

INSTITUTO TECNOLÓGICO SUPERIOR DE CALKINÍ

“Desarrollo de módulo de evaluación geriátrica integral: Implementación de Tests estandarizados y visualización de resultados en un sistema de gestión clínica”

TESIS PARA TITULACIÓN INTEGRAL

INGENIERIA EN SISTEMAS COMPUTACIONALES

**EGRESADO:
CARLOS FERNANDO CHAN CAUICH**

**DIRECTOR Y CO-DIRECTOR:
DRA. MARLENE MENDEZ MORENO
GONZALO MIGUEL QUETZ AGUIRRE**

Calkiní, Campeche, México 2025



Educación
Secretaría de Educación Pública



TECNOLÓGICO
NACIONAL DE MÉXICO



Instituto Tecnológico Superior de Calkiní
Subdirección de Planeación y Desarrollo

Instituto Tecnológico Superior de Calkiní
Departamento de Administración y
Apoyo al Estudiante

Calkiní, Campeche 07/03/2025
P/03/25/629

INGENIERÍA EN SISTEMAS COMPUTACIONALES

CONSTANCIA DE NO INCONVENIENCIA

MODALIDAD: TESIS

C. CARLOS FERNANDO CHAN CAUICH 7010
PASANTE DE LA CARRERA DE
INGENIERÍA EN SISTEMAS COMPUTACIONALES.
PRESENTE

Con relación a lo establecido en el Reglamento de Estudios de Licenciatura, capítulo VII, fracción I, vigente en este Instituto, y una vez emitido el fallo favorable por el director DRA. MARLENE MÉNDEZ MORENO, y el co-director DR. GONZALO MIGUEL QUETZ AGUIRRE, respectivamente, este Instituto consta de no tener inconveniencia para la realización del acto protocolario de titulación integral de su trabajo profesional titulado: "**DESARROLLO DE MÓDULO DE EVALUACIÓN GERIÁTRICA**

INTEGRAL: IMPLEMENTACIÓN DE TESTS ESTANDARIZADOS Y VISUALIZACIÓN DE RESULTADOS EN UN SISTEMA DE GESTIÓN CLÍNICA

"

ATENTAMENTE

Excelencia en Educación Tecnológica®
Sabiduría, Fortaleza de Nuestra Cultura®



INSTITUTO TECNOLÓGICO SUPERIOR
DE CALKINÍ

"DEPARTAMENTO DE ADMINISTRACIÓN
ESCOLAR Y APOYO A
ESTUDIANTES"

CLAVE: 04MSU0013P
CALKINÍ, CAMPECHE, MÉXICO

Dr. Jorge Alfonso Hau Puc.
Departamento de Administración y
Apoyo al Estudiante.

C.c.p. Archivo



2025
Año de
La Mujer
Indígena

Av. Ah Canul S/N por Carretera Federal, Calkiní, Campeche,
C.P. 24900 Tel. 996-8134870,
e-mail: se_dcalkini@tecnm.mx | itescam.edu.mx





TECNOLÓGICO NACIONAL DE MÉXICO

Dedicatoria

Dedico este trabajo a Dios, por darme la fortaleza y sabiduría para superar cada desafío en este camino académico.

A mis padres, cuyo amor, apoyo incondicional y valores inculcados han sido el pilar fundamental en mi formación. Gracias por su confianza en mí y por ser mi mayor inspiración.

A mis docentes y asesores, por su guía y conocimientos, que han sido clave en el desarrollo de esta investigación. Su orientación ha sido invaluable para mi crecimiento profesional.

A mis compañeros y amigos, por su compañía, motivación y consejos en los momentos de dificultad. Su apoyo ha hecho que este proceso sea más llevadero y enriquecedor.

Finalmente, a todas las personas mayores que, con su sabiduría y experiencia, nos enseñan la importancia de brindarles atención y cuidado con dignidad. Espero que este trabajo contribuya a mejorar su calidad de vida y el acceso a una evaluación médica más precisa y eficiente.



TECNOLÓGICO NACIONAL DE MÉXICO

Agradecimientos

Al concluir este trabajo, quiero expresar mi más sincero agradecimiento a todas las personas que, de una u otra manera, contribuyeron a la realización de esta investigación.

En primer lugar, agradezco a Dios, por darme la fuerza, la paciencia y la perseverancia necesarias para alcanzar esta meta.

A mis padres y familia, por su apoyo incondicional, por ser mi motor en cada paso de este camino y por brindarme siempre su amor, confianza y motivación. Su ejemplo y sacrificio han sido fundamentales en mi formación.

A mi asesor de tesis y docentes, por su invaluable guía, consejos y conocimientos que han enriquecido este trabajo. Su orientación y dedicación han sido clave en mi crecimiento académico y profesional.

A mis compañeros y amigos, quienes, con su apoyo, palabras de aliento y colaboración hicieron que este proceso fuera más llevadero. Gracias por compartir este recorrido lleno de retos y aprendizajes.

A las instituciones y profesionales de la salud que brindaron su tiempo y conocimientos para la validación de este trabajo. Su experiencia y aportes han sido esenciales para el desarrollo del módulo de evaluación geriátrica.



TECNOLÓGICO NACIONAL DE MÉXICO

Índice

Tabla de ilustraciones	7
Capítulo 1.....	9
INTRODUCCIÓN	9
ANTECEDENTES	9
PLANTEAMIENTO.....	14
JUSTIFICACIÓN.....	15
OBJETIVO GENERAL	16
OBJETIVOS ESPECÍFICOS	16
ALCANCES Y LIMITACIONES.....	16
INFORMACIÓN DE LA EMPRESA.....	17
MISIÓN	20
VISIÓN.....	20
ORGANIGRAMA.....	20
Capítulo 2.....	21
MARCO TEÓRICO	21
Angular:	22
Laravel:	23
MySQL con XAMPP.....	26
Visual Studio Code.....	27
Git y GitHub	27
Postman.....	28
Composer y NPM (Node Package Manager)	28
XAMPP (Gestión de Servidor).....	28
phpMyAdmin.....	28
Test Estandarizados para Valoración Geriátrica.....	28
Desarrollo de Software para la Salud	29
Inteligencia Artificial y Algoritmos en la Valoración Médica.....	30
Capítulo 3.....	32
METODOLOGÍA.....	32
Objetivo de la Metodología.....	32



TECNOLÓGICO NACIONAL DE MÉXICO

Arquitectura del Sistema:	32
Capa de Datos (Base de Datos)	33
Diagrama General de Arquitectura.	33
Diagrama de Entidad-Relación (DER)	34
Diagrama de Casos de Uso.	35
Capítulo 4.....	37
RESULTADOS	37
Módulo de Gestión de Pacientes	37
Módulo de encuestas.	38
<i>Dashboard Principal</i>	38
Informe de Resultados de las Pruebas del Módulo de Valoración Geriátrica.....	40
Metodología de Evaluación	40
Resultados de la Validación.....	41
Precisión Diagnóstica.....	41
Eficiencia en la Automatización.....	42
Usabilidad y Accesibilidad	43
Conclusiones y Recomendaciones.....	49
Anexo: Correcciones a Realizar.	50
Referencias Bibliográficas.	52
Anexos.	53
Base de Datos (código)	58
Laravel (vistas)	59
Angular (vistas)	62
Laravel (código).	79
Authcontroller.php	79
DoctorsController.php.....	81
Patientcontroller.php	85
Doctors.php	86
Patients.php.....	87
Api.php	88
Angular (código).	89



TECNOLÓGICO NACIONAL DE MÉXICO

Api.service.ts	89
User.service.ts	91
Encuestas Esfera Afectiva (HTML y TypeScript)	94
Encuestas Esfera Cognitiva (HTML y TypeScript)	122
Encuestas Esfera Funcional (HTML y TypeScript).....	136
Encuestas Esfera Nutricional (HTML y TypeScript).....	158
Estructura categorías.(HTML y TypeScript)	181
Componente resultado (HTML y TypeScript)	183
Componente list-doctors (HTML y TypeScript)	185
Componente Login (HTML y TypeScript).....	188
Componente patients (HTML y TypeScript).	191
Componente profile-medic (HTML y TypeScript).....	194
Componente Register-patients (HTML y TypeScript).....	206
Componente registro doctor (HTML y TypeScript)	215
Componente vista perfil patient (HTML y TypeScript)	220



TECNOLÓGICO NACIONAL DE MÉXICO

Tabla de ilustraciones

Ilustración 1 principios fundamentales de scrum	11
Ilustración 2 Scrum master	11
Ilustración 3 scrum Team	12
Ilustración 4 Proceso de Scrum	13
Ilustración 5 Beneficios de implementar Scrum.....	14
Ilustración 6 Angular con Bootstrap	23
Ilustración 7 Laravel.....	25
Ilustración 8 Interfaz Xampp	27
Ilustración 9 diagrama general de arquitectura.....	33
Ilustración 10 Diagrama Entidad Relación.....	35
Ilustración 11 Diagrama de Caso de uso	36
Ilustración 12 Login del sistema	53
Ilustración 13 vista principal del sistema con usuario administrador	53
Ilustración 14 vista de registro de doctores	54
Ilustración 15 formulario de registro de doctores	54
Ilustración 16 vista de pacientes	55
Ilustración 17 formulario de registro de pacientes	55
Ilustración 18 detalles del paciente.....	56
Ilustración 19 vista del módulo de encuestas	56
Ilustración 20 ejemplo de aplicación de test estandarizado	57
Ilustración 21 vista del módulo de resultado para el test aplicado	57
Ilustración 22 base de datos con phpMyAdmin.....	58
Ilustración 23 vista de tablas de la base datos	58
Ilustración 24 código de los controladores del backend.....	59
Ilustración 25 código de las rutas para el api	59
Ilustración 26 Modelo del backend Doctors.....	60
Ilustración 27 Modelo del backend patients.....	60
Ilustración 28 controlador de autorización del inicio de sesión.....	61
Ilustración 29 controlador del modelo Doctors	61
Ilustración 30 vista de los modulos del frontend	62
Ilustración 31 vista del módulo de routing.....	62
Ilustración 32 servicios para la conexión con el backend	63
Ilustración 33 componentes del proyecto angular ..	63
Ilustración 34 código del componente encuestas.....	64
Ilustración 35 código del componente diagnostico	64
Ilustración 36 código del componente de la vista principal.....	65
Ilustración 37 código del módulo de encuesta-glim	65
Ilustración 38 código del módulo de encuesta-Beck.....	66
Ilustración 39 código del módulo de encuesta-phq9	66
Ilustración 40 Código del módulo de encuesta Must.....	67
Ilustración 41 Código del módulo de encuesta sarc-f	67



TECNOLÓGICO NACIONAL DE MÉXICO

Ilustración 42 código del módulo funcionalidad el cual integra 6 encuestas (barthel, lawton y brody, ensrud, frail, time up and go, kartz)	68
Ilustración 43 Módulo Cognitiva el cual integra 6 encuestas (4AT, CAM, CAM-ICU, AWOL, QUESTION PFEIFFER, PRUEBA DE RELOJ)	68
Ilustración 44 Modulo encuesta gds-15	69
Ilustración 45 Modulo encuesta ces-d7	69
Ilustración 46 Modulo Encuesta Cornell	70
Ilustración 47 Modulo Encuesta soledad de tres elementos	70
Ilustración 48 Modulo Encuesta gai-sf	71
Ilustración 49 Modulo Encuesta okeeffe	71
Ilustración 50 Modulo Encuesta SADPERSONS	72
Ilustración 51 Vista Encuesta GDS-15	72
Ilustración 52 Vista Encuesta CES-D7	73
Ilustración 53 Vista Encuesta PHQ-9	73
Ilustración 54 Vista Encuesta GAI-SF	74
Ilustración 55 Vista Encuesta Ansiedad de Beck	74
Ilustración 56 Vista Encuesta SADPERSONS	75
Ilustración 57 Vista Encuesta CORNELL	75
Ilustración 58 Vista Encuesta O'KEEFFE	76
Ilustración 59 Vista Encuesta KATZ	76
Ilustración 60 Vista Encuesta Barthel	77
Ilustración 61 Vista Encuesta Lawton y Brody	77
Ilustración 62 Vista Encuesta FRAIL	78
Ilustración 63 Vista Encuesta TIME UP AND GO	78

TECNOLÓGICO NACIONAL DE MÉXICO

Capítulo 1

INTRODUCCIÓN

El presente proyecto de titulación consiste en el desarrollo de un módulo de evaluación geriátrica integral, que permitirá la implementación de Tests estandarizados para evaluar el estado de salud de los pacientes mayores. El sistema se integrará en un entorno de gestión clínica, facilitando la toma de decisiones por parte de los médicos y mejorando la atención geriátrica. Se utilizarán tecnologías como Angular para el frontend, Laravel para el backend y Bootstrap para la interfaz de usuario.

ANTECEDENTES

La población geriátrica está en aumento a nivel mundial debido a la mayor esperanza de vida y avances en la medicina. En México, como en muchos otros países, el envejecimiento de la población plantea nuevos retos en el sistema de salud. Las personas mayores requieren una atención médica especializada, que incluya la evaluación integral de su salud física, mental y social. Para ello, se han desarrollado diversos instrumentos y Tests estandarizados, como el Índice de Katz, la Escala de Depresión Geriátrica (GDS) y el Mini Examen del Estado Mental (MMSE), que permiten a los profesionales de la salud evaluar de manera más precisa las condiciones de sus pacientes.

Sin embargo, muchos de estos procesos de evaluación se siguen realizando de forma manual o en sistemas que no están diseñados para una fácil integración con el entorno clínico digital. Esto genera demoras en la recopilación, análisis y seguimiento de los resultados, afectando la eficiencia del sistema de atención médica.

Ante este panorama, los sistemas de gestión clínica han comenzado a incorporar módulos especializados para facilitar la automatización de estos procesos. A pesar de esto, en muchas instituciones de salud, especialmente en México, no se cuenta con herramientas que permitan una evaluación geriátrica integral digitalizada, lo cual puede llevar a una falta de uniformidad en los diagnósticos y retrasos en la toma de decisiones médicas.

Es en este contexto que surge la necesidad de desarrollar un módulo de evaluación geriátrica integral, que no solo permita la implementación de Tests estandarizados, sino que también ofrezca una plataforma para visualizar los resultados de forma clara y sencilla. Tecnologías como Angular, Laravel y



TECNOLÓGICO NACIONAL DE MÉXICO

Bootstrap ofrecen la flexibilidad y potencia necesarias para crear este tipo de herramientas, permitiendo una interfaz amigable para los médicos y personal clínico, y un backend robusto para la gestión de los datos de los pacientes.

El proyecto propuesto se basa en la necesidad de digitalizar y automatizar los procesos de evaluación geriátrica, integrándolos en un sistema de gestión clínica existente, con el objetivo de optimizar el cuidado de la salud de las personas mayores, mejorar la eficiencia en la evaluación y asegurar una atención médica de calidad.

Metodología Scrum.

La metodología scrum agile es un marco de trabajo que se utiliza principalmente en el desarrollo de software, aunque su aplicación se ha extendido a diversas áreas. Se basa en la idea de que los proyectos son complejos y cambiantes, y que es fundamental adaptarse a esos cambios en lugar de seguir un plan rígido. Scrum promueve la entrega incremental y continua de productos, lo que permite a los equipos recibir retroalimentación constante y ajustar su enfoque según sea necesario.

Este enfoque se inspira en el rugby, donde los jugadores deben trabajar juntos para avanzar hacia la meta. De manera similar, en Scrum, los equipos se organizan en torno a objetivos comunes y se autoorganizan para alcanzar esos objetivos. Esto fomenta un ambiente de colaboración y aprendizaje continuo, donde cada miembro del equipo tiene un papel importante que desempeñar.

Principios Fundamentales de Scrum.

Transparencia: *Todos los aspectos del proceso deben ser visibles para los involucrados. Esto incluye el progreso del trabajo, los impedimentos y las decisiones tomadas.*

Inspección: *Los equipos deben revisar regularmente su progreso y el producto en desarrollo para identificar áreas de mejora. Esto se realiza a través de reuniones y revisiones periódicas.*

Adaptación: *Basándose en los resultados de la inspección, los equipos deben estar dispuestos a ajustar su enfoque y procesos para mejorar continuamente.*

TECNOLÓGICO NACIONAL DE MÉXICO



Ilustración 1 principios fundamentales de scrum

Scrum Master

El Scrum Master es el facilitador del equipo y su principal objetivo es ayudar a los miembros a entender y aplicar la metodología scrum agile. Este rol implica eliminar impedimentos que puedan obstaculizar el progreso del equipo y fomentar un ambiente de colaboración y autoorganización.

El Scrum Master también es responsable de organizar las reuniones Scrum, como las Daily Stand-ups y las Sprint Reviews, y de asegurarse de que se sigan las prácticas y principios de Scrum.



Ilustración 2 Scrum master

TECNOLÓGICO NACIONAL DE MÉXICO

Scrum Team

El Scrum Team está compuesto por profesionales que trabajan juntos para desarrollar el producto. Este equipo es autoorganizado, lo que significa que los miembros deciden cómo llevar a cabo su trabajo y cómo colaborar entre sí. El Scrum Team es responsable de cumplir con los objetivos del Sprint y de entregar un incremento del producto al final de cada ciclo.

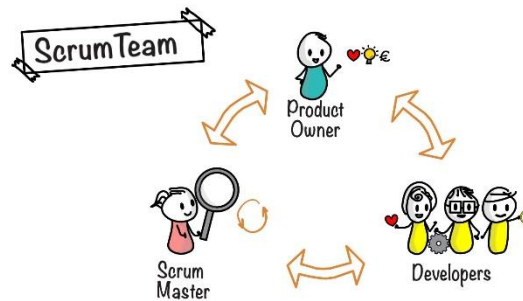


Ilustración 3 scrum Team

Proceso de Scrum.

Creación del Product Backlog.

El primer paso en el proceso de Scrum es la creación del Product Backlog. Este es un documento vivo que contiene todas las tareas y requisitos necesarios para completar el proyecto. El Product Owner es responsable de mantener y priorizar el Product Backlog, asegurándose de que el equipo esté trabajando en las tareas más valiosas.

Planificación del Sprint

Antes de comenzar un Sprint, el equipo realiza una reunión de planificación donde se revisa el Product Backlog y se seleccionan las tareas que se abordarán en el Sprint. Durante esta reunión, el equipo define el Sprint Backlog, que es una lista de tareas específicas que se completarán durante el Sprint.

Ejecución del Sprint

Una vez que se ha planificado el Sprint, el equipo comienza a trabajar en las tareas del Sprint Backlog. Durante esta fase, se llevan a cabo reuniones diarias llamadas Daily Stand-ups, donde los miembros del equipo comparten su progreso, discuten impedimentos y coordinan esfuerzos.

Revisión y Retrospectiva del Sprint

TECNOLÓGICO NACIONAL DE MÉXICO

Al final de cada Sprint, se lleva a cabo una Sprint Review, donde el equipo presenta el trabajo completado a los stakeholders y recibe retroalimentación. Esta es una oportunidad para evaluar el progreso y hacer ajustes en el Product Backlog según sea necesario.

Después de la revisión, el equipo realiza una Retrospectiva del Sprint, donde reflexionan sobre el proceso y discuten qué funcionó bien y qué se puede mejorar. Esta etapa es crucial para fomentar el aprendizaje y la mejora continua.



Beneficios de Implementar Scrum.

Mayor Flexibilidad: Scrum permite a los equipos adaptarse rápidamente a los cambios en los requisitos y prioridades, lo que es esencial en entornos dinámicos.

Mejora de la Colaboración: La metodología fomenta un ambiente de trabajo colaborativo, donde todos los miembros del equipo están involucrados en la toma de decisiones y el desarrollo del producto.

Entrega Continua de Valor: Al trabajar en ciclos cortos, los equipos pueden entregar incrementos del producto de manera regular, lo que permite a los clientes recibir valor de forma continua.

Mayor Satisfacción del Cliente: La retroalimentación constante y la capacidad de adaptarse a las necesidades del cliente resultan en productos que cumplen mejor con sus expectativas.

Mejora de la Productividad: La autoorganización y la claridad en los roles y responsabilidades permiten a los equipos trabajar de manera más eficiente y productiva.

TECNOLÓGICO NACIONAL DE MÉXICO



Ilustración 5 Beneficios de implementar Scrum

La metodología scrum agile se ha consolidado como una de las mejores prácticas para la gestión de proyectos en entornos complejos y cambiantes. Su enfoque en la colaboración, la flexibilidad y la entrega continua de valor permite a los equipos adaptarse a las necesidades del cliente y mejorar constantemente. Sin embargo, es fundamental abordar los desafíos que pueden surgir durante su implementación y asegurarse de que todos los miembros del equipo estén comprometidos con el proceso. Al adoptar Scrum, las organizaciones pueden no solo mejorar la calidad de sus productos, sino también fomentar un ambiente de trabajo más colaborativo y satisfactorio para todos los involucrados.

PLANTEAMIENTO

En los últimos años, la población geriátrica ha experimentado un crecimiento significativo, lo que ha generado nuevos desafíos en el ámbito de la atención médica. Uno de los principales problemas que enfrentan los profesionales de la salud es la falta de herramientas digitales que permitan evaluar de manera precisa y eficiente el estado funcional, cognitivo y social de los pacientes de la tercera edad.

Actualmente, muchos de los Tests geriátricos son aplicados de manera manual, lo que puede generar errores en la interpretación de los resultados, dificultad en el seguimiento de los pacientes y una mayor carga administrativa para los médicos y cuidadores. Ante esta problemática, surge la necesidad de implementar herramientas digitales que automaticen la aplicación de estas pruebas, agilicen el registro de datos y proporcionen visualizaciones claras para facilitar la toma de decisiones clínicas.



TECNOLÓGICO NACIONAL DE MÉXICO

Este proyecto propone el desarrollo de un módulo interactivo dentro de un sistema de gestión clínica, diseñado específicamente para la administración de Tests geriátricos estandarizados. A través de este módulo, los profesionales de la salud podrán aplicar evaluaciones de manera digital, almacenar los resultados de forma segura y generar reportes visuales que permitan un mejor seguimiento de la evolución del paciente. Con esta herramienta, se busca optimizar el proceso de evaluación médica, mejorar la precisión diagnóstica y contribuir a una atención geriátrica más eficiente y basada en datos concretos.

JUSTIFICACIÓN

El envejecimiento de la población es un fenómeno global que plantea desafíos significativos para los sistemas de salud. Con el crecimiento de la población geriátrica, la demanda de atención médica especializada aumenta, lo que resalta la necesidad de herramientas digitales confiables que optimicen la evaluación y el seguimiento de estos pacientes.

La Evaluación Geriátrica Integral (EGI) es una metodología clave para valorar las capacidades funcionales, cognitivas y emocionales de las personas mayores. Sin embargo, su aplicación en la práctica clínica suele depender de procesos manuales, lo que puede generar errores en el registro y la interpretación de los resultados, además de requerir una cantidad considerable de tiempo para su administración y análisis. Esto limita la eficiencia del proceso médico y dificulta un diagnóstico preciso y oportuno.

Este proyecto busca facilitar la evaluación geriátrica mediante herramientas digitales confiables que automaticen la aplicación de Tests estandarizados dentro de un sistema de gestión clínica. Con este módulo, el personal médico podrá aplicar evaluaciones de manera más rápida y precisa, reduciendo errores en la recopilación de datos y mejorando la interpretación de los resultados a través de reportes visuales claros.

La implementación de este sistema contribuirá a optimizar la atención médica geriátrica, asegurando que los profesionales de la salud cuenten con información estructurada y precisa para la toma de



TECNOLÓGICO NACIONAL DE MÉXICO

decisiones. Además, permitirá un seguimiento más eficiente del estado de salud de los pacientes, brindando un enfoque más centrado en la prevención y la intervención temprana.

OBJETIVO GENERAL

Desarrollar un módulo de evaluación geriátrica integral que permita a los médicos aplicar Tests estandarizados, registrar los resultados y generar informes visuales en un sistema de gestión clínica.

OBJETIVOS ESPECÍFICOS

Los fines de este proyecto incluyen mejorar la capacidad de los médicos para realizar evaluaciones precisas en pacientes mayores, reducir el tiempo de procesamiento de los resultados y proporcionar una herramienta digital moderna que apoye la toma de decisiones clínicas en entornos geriátricos.

- 1) Analizar y seleccionar los Tests geriátricos más relevantes para la implementación.*
- 2) Diseñar e implementar una interfaz de usuario intuitiva y accesible para los médicos.*
- 3) Desarrollar el backend para gestionar los resultados de las evaluaciones utilizando Laravel.*
- 4) Realizar pruebas de usabilidad y funcionalidad del sistema.*

ALCANCES Y LIMITACIONES

Alcance del proyecto.

Este proyecto contempla el desarrollo de un módulo de evaluación geriátrica integral diseñado para la aplicación de Tests estandarizados que permitan valorar distintos aspectos de la salud de los pacientes mayores, como su estado funcional, cognitivo, emocional y social. En su versión inicial, el módulo incluirá la implementación de [4 esferas con un total de 60 Tests], con posibilidad de expansión en futuras versiones.

El módulo permitirá a los médicos visualizar los resultados de las evaluaciones a través de diferentes tipos de gráficos, como barras, líneas y radar, facilitando la interpretación de la información y el seguimiento de la evolución del paciente a lo largo del tiempo. Además, se generarán informes detallados en formato digital y descargable, los cuales podrán ser compartidos con otros profesionales de la salud para una mejor toma de decisiones.



TECNOLÓGICO NACIONAL DE MÉXICO

En términos de accesibilidad, el sistema estará integrado en una plataforma de gestión clínica y será compatible con cualquier dispositivo con conexión a internet, permitiendo su uso en computadoras, tabletas y teléfonos móviles. Se priorizarán aspectos de seguridad y confidencialidad de los datos, garantizando que la información del paciente se almacene de manera segura mediante protocolos de cifrado y control de acceso basado en roles.

El proyecto también considerará la disponibilidad y rendimiento del sistema, asegurando que pueda manejar múltiples evaluaciones de manera eficiente y sin afectar la experiencia del usuario. Con estas características, el módulo no solo facilitará la aplicación de pruebas geriátricas, sino que también contribuirá a mejorar la precisión diagnóstica y la toma de decisiones en la atención geriátrica.

Limitaciones.

Las principales limitaciones del proyecto incluyen el tiempo limitado para el desarrollo, la posible falta de acceso a pacientes para realizar pruebas en escenarios reales, y la necesidad de adaptar el módulo a las normativas de seguridad y privacidad de los datos médicos.

INFORMACIÓN DE LA EMPRESA.

Lugar donde se realizará el proyecto

Nombre del departamento: *Division de Ingenieria en Sistemas Computacionales perteneciente a la Subdirección Académica de Investigación e Innovación*

Edificio A - Planta Alta

996-8134870 Ext. 1001

Estructura departamental: *Subdirección Académica de Investigación e Innovación*

TECNOLÓGICO NACIONAL DE MÉXICO



Figura. - Croquis del área

Información sobre la empresa, organismo o dependencia para la que se desarrollará el proyecto.

LUGAR DE REALIZACIÓN:

NOMBRE DE LA EMPRESA Y DIRECCIÓN.

Instituto Tecnológico Superior de Calkiní, en el Estado de Campeche
Av. Ah Canul S/N por Carretera Federal.
C.P. 24900 Calkiní, Campeche.
Tel. 996-8134870

GIRO.

Educación, con un amplio sentido social y humano, al desarrollo sustentable de la región, del Estado y del país, atendiendo desde su ámbito de competencia, las necesidades de formación y actualización de profesionales competitivos, de investigación y desarrollo tecnológico y de conservación y extensión de la cultura.

TECNOLÓGICO NACIONAL DE MÉXICO

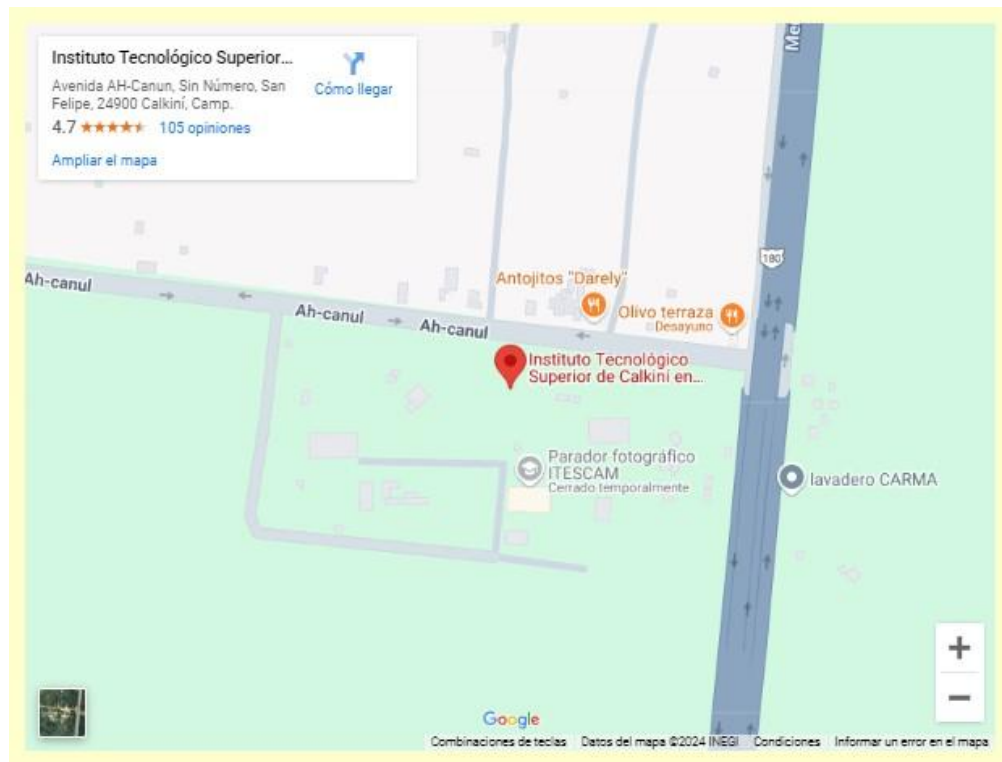
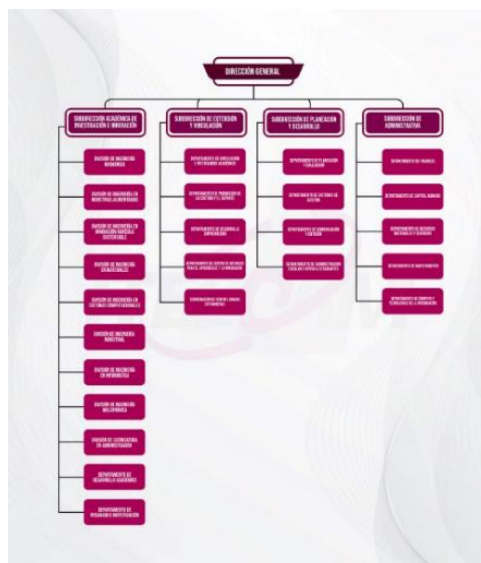


Figura. - Croquis de Ubicación

Estructura organizacional: Describir brevemente su estructura general e incluir el organigrama de la empresa marcando con gris la ubicación del área o departamento en donde se participó.

RFC DE LA EMPRESA.





TECNOLÓGICO NACIONAL DE MÉXICO

MISIÓN

Brindar a los profesionales de la salud una herramienta tecnológica innovadora que facilite la evaluación integral de pacientes geriátricos mediante la implementación de Tests estandarizados y la visualización eficiente de resultados, promoviendo una atención médica de calidad, personalizada y accesible para mejorar la calidad de vida de las personas mayores.

VISIÓN

Convertirse en un referente nacional en el desarrollo de soluciones tecnológicas para la gestión clínica geriátrica, consolidando un sistema integral que automatice y optimice los procesos de evaluación médica, contribuyendo al fortalecimiento de los servicios de salud y al bienestar de la población envejecida.

ORGANIGRAMA





Capítulo 2

MARCO TEÓRICO

El avance tecnológico en el ámbito de la salud ha permitido la integración de herramientas digitales para mejorar la precisión y eficiencia en los procesos de evaluación médica. En el contexto de la valoración geriátrica, surge la necesidad de optimizar la recolección y análisis de datos a través de sistemas automatizados. Esta investigación se enfoca en responder la pregunta: ¿Cómo puede la implementación de test estandarizados en un sistema de gestión clínica mejorar la precisión diagnóstica en la evaluación geriátrica integral?

Para abordar esta interrogante, es fundamental analizar los conceptos clave que sustentan el desarrollo e implementación del sistema. En primer lugar, se revisan los test estandarizados para la valoración geriátrica, que son herramientas esenciales para evaluar la funcionalidad, estado cognitivo y autonomía de los pacientes mayores. Estos Tests permiten una valoración objetiva y comparativa del estado del paciente, optimizando el proceso de diagnóstico (de Quir & Plazzotta, 2015).

Posteriormente, se examinan las metodologías de desarrollo de software para el sector salud, destacando las prácticas y tecnologías que garantizan precisión, seguridad y facilidad de uso en aplicaciones médicas. La informática en salud busca optimizar la adquisición, almacenamiento y utilización de información médica, lo que es esencial en la implementación de sistemas automatizados para la evaluación geriátrica (Coiera et al., 2005).

Asimismo, se estudia el uso de inteligencia artificial y algoritmos en la valoración médica, explorando su impacto en la automatización de diagnósticos y la reducción de errores en la interpretación de resultados. La integración de estas tecnologías no solo mejora la precisión diagnóstica, sino que también agiliza los procesos clínicos, asegurando una atención más eficiente y oportuna (de Quir & Plazzotta, 2015).



TECNOLÓGICO NACIONAL DE MÉXICO

Finalmente, se presenta un análisis técnico sobre la implementación del sistema, abordando tecnologías clave como Angular, arquitectura basada en componentes, interacción con el backend, formularios reactivos e integración con Bootstrap, elementos fundamentales para garantizar la eficiencia y usabilidad del software desarrollado. Además, se consideran aspectos de seguridad y confidencialidad en el manejo de datos clínicos, en cumplimiento con normativas de protección de datos como la Ley Orgánica 3/2018, que establece directrices para garantizar la seguridad de la información médica (Ley Orgánica 3/2018).

Este marco teórico proporciona las bases necesarias para comprender la relevancia y aplicabilidad del sistema de valoración geriátrica, justificando su desarrollo y validación dentro del entorno clínico.

Angular: Angular es un framework robusto y ampliamente utilizado para el desarrollo de aplicaciones web frontend. Su arquitectura basada en componentes facilita la creación de interfaces modulares, escalables y mantenibles.

Arquitectura basada en Componentes:

Componentes Reutilizables: En el sistema, cada funcionalidad o vista principal, como la lista de pacientes, el módulo de encuestas y el módulo de resultados, se ha desarrollado como componentes independientes. Esta modularidad permite reutilizar y actualizar fácilmente secciones específicas del sistema sin afectar otras áreas.

Intercambio de Datos: la inyección de dependencias y el intercambio de datos entre componentes padres e hijos para una comunicación eficiente, por ejemplo, al enviar datos de una encuesta desde el componente de encuesta al de resultados.

Servicios en Angular:

Servicios para la Lógica de Negocios: los servicios en Angular permiten separar la lógica de negocio del frontend. Por ejemplo, tienes servicios dedicados para gestionar la conexión con el backend,

TECNOLÓGICO NACIONAL DE MÉXICO

enviar respuestas de encuestas, recibir resultados y listas de pacientes, lo cual mantiene la lógica centralizada y reutilizable.

Interacción con el Backend (API): los servicios de Angular para realizar llamadas HTTP (a través del módulo HttpClient) que interactúan con el backend ubicado en C:/Users/Documents/BackVGI. Estos servicios son fundamentales para gestionar los datos en tiempo real.

Formularios Reactivos:

Formularios para Encuestas: Formularios reactivos para crear las encuestas, mediante formularios reactivos, Angular permite controlar los estados de cada campo, agregar validaciones específicas y hacer el manejo de errores de manera más eficiente.

Integración con Bootstrap:

Bootstrap mejora la estética y usabilidad del sistema, permitiendo implementar componentes visuales, como botones, tarjetas y tablas, que responden bien en dispositivos móviles.



Ilustración 6 Angular con Bootstrap

Laravel:

Laravel es un framework de desarrollo de aplicaciones web basado en PHP que permite construir aplicaciones de manera rápida, segura y estructurada. Con su arquitectura MVC (Modelo-Vista-Controlador), Laravel proporciona una separación clara entre la lógica de negocio, la presentación



TECNOLÓGICO NACIONAL DE MÉXICO

y el manejo de datos, lo cual es ideal para el desarrollo de sistemas como el de gestión clínica geriátrica.

Arquitectura MVC (Modelo-Vista-Controlador):

Separación de Responsabilidades: La arquitectura MVC de Laravel permite organizar el código en tres capas:

Modelo: Interactúa con la base de datos y gestiona los datos de entidades como pacientes, doctores, encuestas y resultados.

Vista: Presenta los datos al usuario, generalmente para vistas específicas en el frontend.

Controlador: Gestiona las solicitudes del usuario, aplica la lógica de negocio y envía las respuestas correspondientes.

ORM (Eloquent): Laravel incluye Eloquent, un ORM que permite interactuar con la base de datos de manera sencilla y limpia. Esto facilita la creación y gestión de modelos, permitiendo realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sin necesidad de escribir consultas SQL complejas.

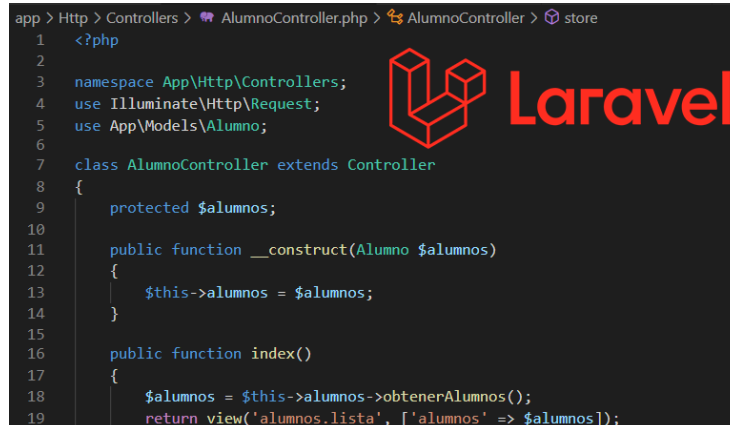
Relaciones de Base de Datos: Eloquent maneja relaciones entre tablas como “uno a muchos” o “muchos a muchos”, lo cual es útil para estructurar la relación entre doctores y pacientes, así como entre encuestas y respuestas.

Manejo de Autenticación y Seguridad:

Sistema de Autenticación Incorporado: Laravel incluye un sistema de autenticación robusto que permite gestionar registros, inicios de sesión y permisos de usuario. Esto asegura que solo usuarios autorizados accedan a ciertas funcionalidades del sistema.

TECNOLÓGICO NACIONAL DE MÉXICO

Protección CSRF y CORS: Laravel maneja la protección contra ataques CSRF (Cross-Site Request Forgery) y permite configurar políticas de CORS (Cross-Origin Resource Sharing) para proteger el backend.



```
app > Http > Controllers > AlumnoController.php > AlumnoController > store
1  <?php
2
3  namespace App\Http\Controllers;
4  use Illuminate\Http\Request;
5  use App\Models\Alumno;
6
7  class AlumnoController extends Controller
8  {
9      protected $alumnos;
10
11     public function __construct(Alumno $alumnos)
12     {
13         $this->alumnos = $alumnos;
14     }
15
16     public function index()
17     {
18         $alumnos = $this->alumnos->obtenerAlumnos();
19         return view('alumnos.lista', ['alumnos' => $alumnos]);
20     }
21 }
```

Ilustración 7 Laravel

Migraciones de Base de Datos:

Sistema de Migraciones: Las migraciones en Laravel permiten definir y gestionar la estructura de la base de datos con código PHP, facilitando la actualización y control de versiones de la base de datos de forma segura.

Semillas de Datos (Seeders): Explica que los seeders permiten llenar la base de datos con datos de prueba, útiles para validar el funcionamiento del sistema durante el desarrollo.

Validación de Datos:

Validación Integrada: Laravel proporciona herramientas para validar datos de entrada directamente en los controladores o modelos, asegurando que los datos ingresados en el sistema cumplan con ciertos estándares antes de almacenarse en la base de datos.

Reglas Personalizadas: Puedes explicar que creaste reglas de validación personalizadas para verificar datos específicos, como los puntajes de las encuestas o los diagnósticos.



TECNOLÓGICO NACIONAL DE MÉXICO

MySQL con XAMPP.

Paquete Todo-en-Uno: XAMPP facilita el desarrollo local al incluir Apache, MySQL, PHP, y otros servicios en un solo paquete. Esto permite desarrollar y probar el sistema en un entorno controlado antes de implementarlo en un servidor de producción.

Gestión de Base de Datos con phpMyAdmin: Con XAMPP, puedes utilizar phpMyAdmin, una herramienta gráfica que simplifica la administración de MySQL, permitiéndote crear, gestionar y modificar tablas sin necesidad de comandos SQL avanzados.

MySQL como Base de Datos Relacional.

Eficiencia y Escalabilidad: MySQL es un sistema de gestión de bases de datos relacional conocido por su rapidez y capacidad de manejar grandes cantidades de datos, lo cual es adecuado para almacenar información crítica de pacientes, doctores, encuestas y resultados.

Estructura de la Base de Datos

Tablas Principales:

Pacientes: Tabla para la información básica de los pacientes, incluyendo datos personales y estado de salud.

Doctores: Almacena la información de los médicos, sus especialidades y las asignaciones de pacientes.

Encuestas: Registra las diferentes encuestas disponibles y sus configuraciones.

Respuestas y Resultados: Tablas para guardar las respuestas de los pacientes y calcular los resultados.

Seguridad

Control de Acceso: Menciona que MySQL permite gestionar permisos de acceso a través de usuarios y contraseñas, limitando quién puede acceder a las diferentes tablas y funcionalidades.

TECNOLÓGICO NACIONAL DE MÉXICO

Conexiones Seguras: Si implementas medidas adicionales, como el uso de SSL para las conexiones de base de datos, puedes mencionarlo como un aspecto de seguridad que protege los datos sensibles.

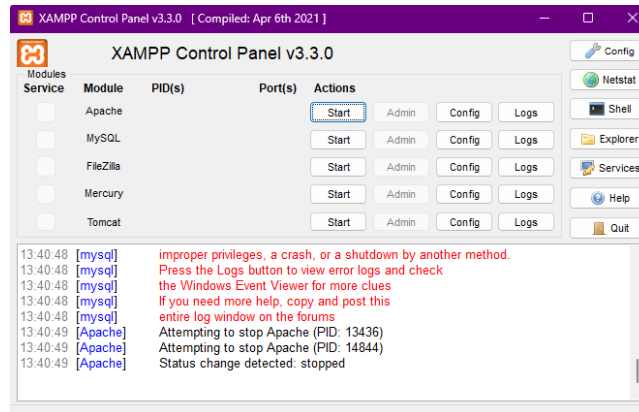


Ilustración 8 Interfaz Xampp

Visual Studio Code

*Editor de Código: Utilizas Visual Studio Code (VS Code) como editor principal para escribir y editar el código fuente del sistema. VS Code ofrece una amplia gama de extensiones, como herramientas de depuración, integración con Git, y soporte para lenguajes como PHP (Laravel), TypeScript (Angular), **HTML**, y **SCSS**.*

Plugins y Extensiones: Extensiones como Prettier (para formateo de código), Laravel Snippets (para autocompletar código Laravel), y Angular Language Service (para soporte en Angular) facilitan la escritura de código limpio y organizado.

Git y GitHub

Control de Versiones: Git permite realizar un seguimiento de todos los cambios en el código y facilita la colaboración en equipo. Explica que cada modificación queda registrada, permitiendo revertir cambios si es necesario.

Repositorio en GitHub: GitHub se utiliza para alojar el repositorio del proyecto, lo que permite tener una copia de respaldo en la nube y facilita el trabajo colaborativo y la revisión de código.



TECNOLÓGICO NACIONAL DE MÉXICO

Postman

Pruebas de API: Postman es una herramienta útil para probar y documentar los endpoints del backend en Laravel. Puedes detallar que lo usas para enviar solicitudes HTTP y verificar que las respuestas sean las esperadas, garantizando que la API funcione correctamente antes de integrarla con el frontend en Angular.

Composer y NPM (Node Package Manager)

Gestión de Dependencias: Composer y NPM son administradores de paquetes. Composer se usa en el backend (Laravel) para gestionar librerías y dependencias PHP, mientras que NPM gestiona los paquetes necesarios en el frontend (Angular).

Actualización de Librerías: Ambos permiten actualizar las librerías y dependencias de forma centralizada, asegurando que el proyecto utilice las versiones más recientes y seguras.

XAMPP (Gestión de Servidor)

Servidor Local: Aunque también lo mencionaste en la base de datos, puedes detallar aquí cómo usas XAMPP para simular un entorno de servidor local, con Apache y MySQL. Esto permite probar la aplicación de forma similar a cómo funcionará en un servidor de producción, facilitando el desarrollo y depuración.

phpMyAdmin

Gestión de Base de Datos: phpMyAdmin es la herramienta visual que se incluye con XAMPP para administrar MySQL. Facilita la creación de tablas, ejecución de consultas SQL, exportación e importación de datos, y permite realizar cambios sin necesidad de comandos SQL directos.

Test Estandarizados para Valoración Geriátrica

La evaluación geriátrica utiliza herramientas estandarizadas para medir diferentes aspectos de la salud y funcionalidad del adulto mayor. Estos test permiten obtener resultados objetivos y comparables, facilitando la toma de decisiones clínicas. Algunos de los más relevantes incluyen:

TECNOLÓGICO NACIONAL DE MÉXICO

- *Índice de Barthel: Evalúa la capacidad funcional para realizar actividades básicas de la vida diaria (ABVD), como alimentarse, vestirse y desplazarse. Su puntuación ayuda a determinar el grado de dependencia del paciente.*
- *Escala de Lawton y Brody: Complementa el Índice de Barthel al medir la autonomía en actividades instrumentales de la vida diaria (AIVD), como el manejo del dinero, uso del teléfono o preparación de comidas.*
- *Escala de Yesavage: Cuestionario utilizado para la detección de depresión en adultos mayores, con versiones de 15 y 30 preguntas que evalúan síntomas depresivos.*
- *Mini-Mental State Examination (MMSE): Herramienta para evaluar deterioro cognitivo y detectar enfermedades como la demencia.*
- *Escala de Norton y Escala de Braden: Ayudan a valorar el riesgo de desarrollar úlceras por presión en pacientes con movilidad reducida.*
- *Valoración geriátrica integral (VGI): No es un test único, sino un conjunto de herramientas que incluyen las mencionadas anteriormente para evaluar de forma completa al paciente.*
- *El uso de estos test dentro de un sistema digital mejora la precisión del diagnóstico al automatizar cálculos y reducir el margen de error en la interpretación de los resultados.*

Desarrollo de Software para la Salud

El desarrollo de aplicaciones en el ámbito de la salud debe cumplir con estándares específicos que garanticen su eficacia, seguridad y facilidad de uso. Algunos aspectos clave incluyen:

Normativas y Regulaciones:

- *Cumplimiento de estándares de protección de datos médicos, como el HIPAA (Health Insurance Portability and Accountability Act) en EE.UU. o el GDPR en Europa.*
- *Adaptación a normativas nacionales, como la Ley General de Protección de Datos Personales en México.*

Seguridad y Protección de Datos:

TECNOLÓGICO NACIONAL DE MÉXICO

- *Implementación de protocolos de cifrado para resguardar la información de los pacientes.*
- *Control de acceso mediante autenticación de usuarios y permisos diferenciados según roles (médico, enfermero, administrador).*

Usabilidad y Accesibilidad:

- *Diseño intuitivo que facilite su uso a médicos y personal administrativo sin experiencia en tecnología.*
- *Compatibilidad con distintos dispositivos (computadoras, tabletas, teléfonos móviles).*
- *Adaptación a las necesidades de los adultos mayores con tipografías grandes y botones accesibles.*

Automatización y Precisión:

- *Implementación de algoritmos que reduzcan errores en la captura de datos.*
- *Generación automática de reportes y gráficos para facilitar la interpretación de resultados.*

Inteligencia Artificial y Algoritmos en la Valoración Médica

La inteligencia artificial (IA) está revolucionando la medicina al permitir la automatización de procesos y la mejora en la precisión diagnóstica. En la valoración geriátrica, su aplicación puede optimizar diferentes aspectos:

Automatización del Diagnóstico:

- *Algoritmos que analizan las respuestas de los test y asignan diagnósticos de manera objetiva, evitando errores humanos.*
- *Modelos predictivos que identifican patrones en los datos del paciente para detectar riesgos de enfermedades neurodegenerativas o deterioro funcional.*

Procesamiento de Lenguaje Natural (PLN):

- *IA capaz de analizar descripciones clínicas y respuestas abiertas en los test para identificar síntomas y correlaciones con patologías.*



TECNOLÓGICO NACIONAL DE MÉXICO

Aprendizaje Automático (Machine Learning):

- *Implementación de modelos que mejoran su precisión a medida que recopilan más datos clínicos.*
- *Identificación de factores de riesgo basados en análisis de grandes volúmenes de información.*

Soporte a la Decisión Médica:

- *Sistemas que sugieren recomendaciones de tratamiento basadas en la información recopilada y en bases de datos médicas.*
- *Herramientas que alertan sobre inconsistencias en la información ingresada por los médicos.*

TECNOLÓGICO NACIONAL DE MÉXICO

Capítulo 3.

METODOLOGÍA

Para desarrollar el sistema de gestión clínica geriátrica, se adoptó una metodología ágil para facilitar la adaptabilidad a cambios y permitir una entrega incremental de funcionalidades. La metodología elegida se centra en iteraciones cortas y en una comunicación constante con los interesados para asegurar que el proyecto cumple con los requisitos del usuario final.

Objetivo de la Metodología.

La metodología de desarrollo utilizada tiene como objetivo:

Adaptarse a los cambios: Permitir ajustes durante el desarrollo, ya que los requisitos y prioridades pueden evolucionar.

Entregas incrementales: *Asegurar que el sistema sea funcional en cada fase de desarrollo, permitiendo revisiones y retroalimentación temprana.*

Mejora continua: *Optimizar el proceso y la calidad del producto en cada iteración.*

Arquitectura del Sistema:

se basa en una arquitectura en tres capas (Frontend, Backend y Base de Datos) que permite separar la lógica de presentación, la lógica de negocio y la gestión de datos. Esta estructura facilita el desarrollo, mantenimiento y escalabilidad del sistema de gestión clínica geriátrica, asegurando que cada componente esté desacoplado y pueda ser modificado o mejorado sin afectar a otros componentes.

Capa de Presentación (Frontend)

Tecnología Usada: Angular

Responsabilidad: La capa de presentación se encarga de la interacción con el usuario. Incluye la interfaz gráfica del sistema, que permite a los usuarios realizar tareas como registrar pacientes, aplicar encuestas, y visualizar resultados.

TECNOLÓGICO NACIONAL DE MÉXICO

Capa de Lógica de Negocio (Backend)

Tecnología Usada: Laravel

Responsabilidad: La capa de backend gestiona la lógica de negocio del sistema. Se encarga de procesar y validar datos, aplicar reglas de negocio y exponer los endpoints de la API para que la capa de presentación acceda a la información.

Capa de Datos (Base de Datos)

Tecnología Usada: MySQL

Responsabilidad: Esta capa se encarga de la persistencia de los datos del sistema. Aquí se almacenan todas las tablas necesarias, como las de pacientes, doctores, encuestas, resultados, y cualquier otra entidad requerida.

Diagrama General de Arquitectura.

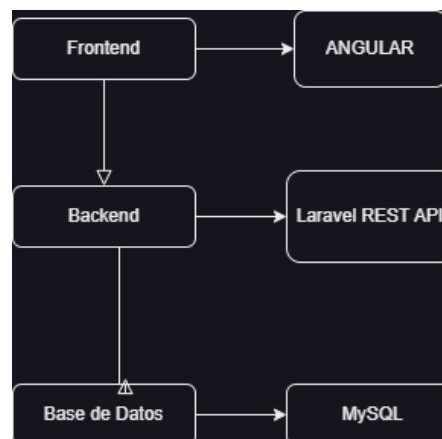


Ilustración 9 diagrama general de arquitectura

Beneficios de la Arquitectura en Tres Capas

Escalabilidad: Al separar las capas, es posible escalar cada componente independientemente. Por ejemplo, si la base de datos requiere más capacidad, se puede optimizar o migrar sin modificar el frontend o el backend.

TECNOLÓGICO NACIONAL DE MÉXICO

Facilidad de Mantenimiento: Al tener cada capa con responsabilidades definidas, el mantenimiento y las actualizaciones son más simples. Los cambios en la lógica de negocio no afectan directamente la interfaz de usuario y viceversa.

Seguridad: La capa de backend actúa como un filtro que controla el acceso a la base de datos y valida todas las solicitudes. Esto asegura que solo los datos válidos y las solicitudes autorizadas lleguen a la base de datos.

Diagrama de Entidad-Relación (DER)

Entidades Principales:

- *Paciente:* Contiene los datos de los pacientes, como *id*, *nombre*, *apellido*, *fecha_nacimiento*, *género*, *estado_civil*, y *contacto*.
- *Doctor:* Almacena información sobre los doctores, incluyendo *id*, *nombre*, *especialidad*, *contacto*, y *email*.
- *Encuesta:* Representa los diferentes tipos de encuestas (por ejemplo, *funcional*, *psicoafectiva*) con atributos como *id*, *nombre*, y *descripción*.
- *Pregunta:* Cada pregunta que forma parte de las encuestas, con atributos como *id*, *contenido*, y *tipo_respuesta*.
- *respuesta:* Almacena la respuesta seleccionada para una pregunta específica, con atributos como *id*, *id_pregunta*, *opcion_seleccionada*, y *puntaje*.
- *Resultado:* Contiene los resultados de cada encuesta para cada paciente, como *id*, *id_paciente*, *id_encuesta*, *puntaje_total*, y *diagnóstico*.

Relaciones:

- *Paciente-Encuesta (1):* Un paciente puede tener múltiples encuestas asociadas.
- *Encuesta-Pregunta (1):* Cada encuesta contiene múltiples preguntas.
- *Pregunta-Respuesta (1):* Una pregunta puede tener varias respuestas posibles (dependiendo del tipo de pregunta).

TECNOLÓGICO NACIONAL DE MÉXICO

Paciente-Doctor(N): Relación de asignación de pacientes a doctores, si los pacientes pueden ser atendidos por varios doctores.

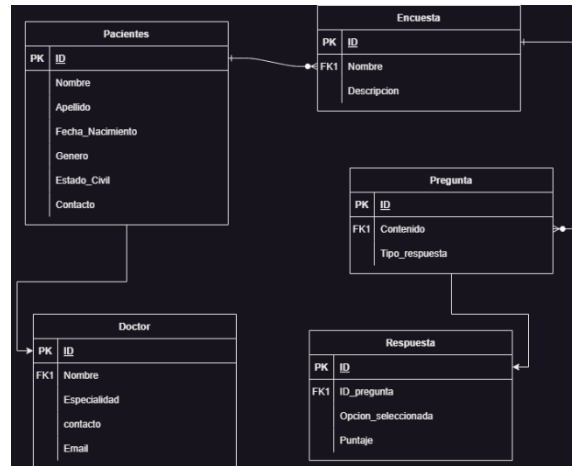


Ilustración 10 Diagrama Entidad Relación

Diagrama de Casos de Uso.

Módulo de Gestión de Pacientes:

- *Registrar Paciente: Permite al personal médico o administrativo agregar nuevos pacientes al sistema.*
- *Editar Información del Paciente: Permite actualizar los datos de un paciente existente.*
- *Consultar Información del Paciente: Visualización detallada de la información de un paciente.*

Módulo de Evaluación Geriátrica

- *Crear Encuesta: Permite crear diferentes tipos de encuestas.*
- *Asignar Encuesta a Paciente: Permite asignar una encuesta a un paciente específico.*
- *Responder Encuesta: Permite que el doctor o el mismo paciente responda las preguntas de la encuesta.*
- *Calcular Resultado: Una vez completada la encuesta, se calcula el puntaje total y el diagnóstico.*

TECNOLÓGICO NACIONAL DE MÉXICO

Módulo de Resultados

- *Visualizar Resultados de Encuestas:* Permite a los médicos visualizar el puntaje total y diagnóstico para cada encuesta realizada por un paciente.
- *Historial de Resultados:* Permite ver el historial de encuestas y diagnósticos para cada paciente.

Módulo de Gestión de Doctores

- *Registrar Doctor:* Permite agregar nuevos doctores al sistema.
- *Asignar Paciente a Doctor:* Relaciona un paciente con uno o varios doctores.

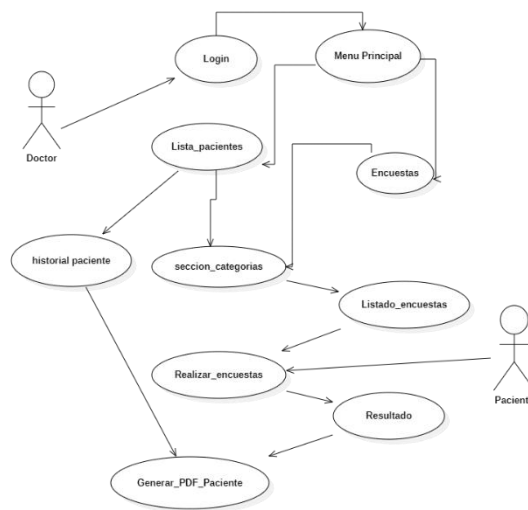


Ilustración 11 Diagrama de Caso de uso

TECNOLÓGICO NACIONAL DE MÉXICO

Capítulo 4.

RESULTADOS

En este capítulo, se describen los resultados obtenidos tras la implementación del sistema de valoración geriátrica integral. Se presentan las características de los módulos desarrollados, las interfaces logradas y las funcionalidades implementadas, con capturas de pantalla y explicaciones detalladas de cada componente.

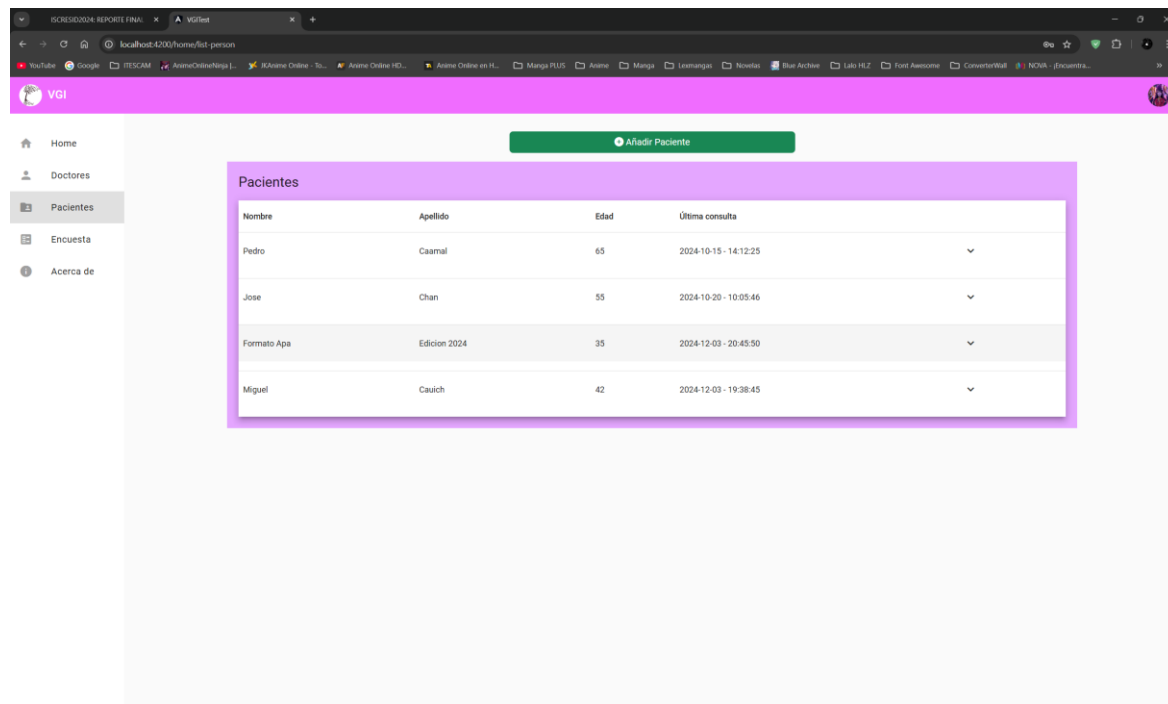
Módulo de Gestión de Pacientes

Permite registrar, modificar y visualizar la información de los pacientes.

Funcionalidades:

- Registro de nuevos pacientes con datos personales y médicos.
- Visualización de la lista de pacientes en una tabla interactiva.
- Opciones para editar y eliminar registros.

Pantalla del módulo:



Explicación: La interfaz incluye un formulario de registro y una tabla con acciones para cada paciente.

TECNOLÓGICO NACIONAL DE MÉXICO

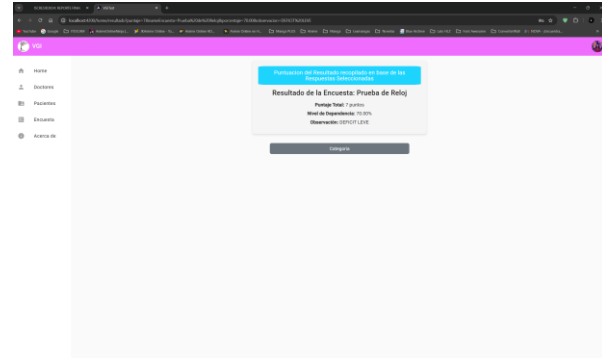
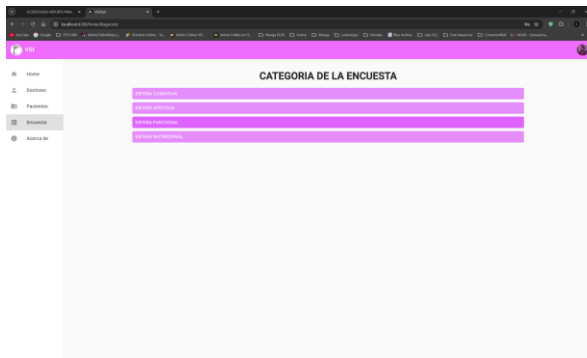
Módulo de encuestas.

Facilita la aplicación de Tests estandarizados como el Índice de Barthel, etc.

Funcionalidades:

- Selección del tipo de test a aplicar.
- Captura de respuestas mediante botones de opción.
- Cálculo automático del puntaje con un botón de "Finalizar Encuesta".

Pantalla del módulo:



Explicación: El diseño utiliza una estructura de preguntas con imágenes asociadas a las respuestas, integrando Bootstrap para garantizar una experiencia de usuario responsiva.

Dashboard Principal

El Dashboard principal del sistema de valoración geriátrica integral ofrece una vista general de las métricas clave y proporciona accesos rápidos a las funcionalidades más importantes. Está diseñado con una interfaz amigable, utilizando colores distintivos para facilitar la interpretación de datos y mejorar la experiencia del usuario.

Funcionalidades:

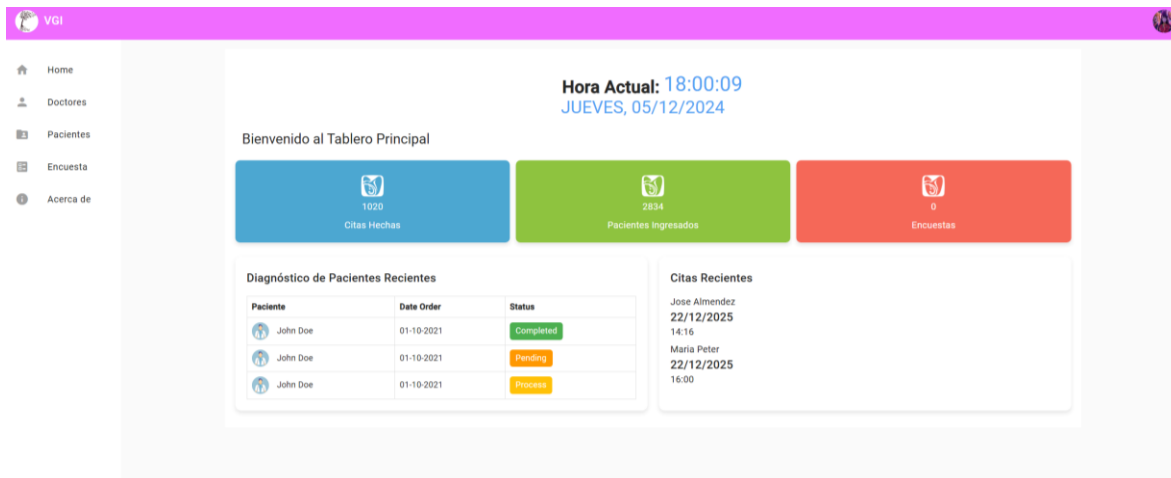
- *Citas Hechas: Total de citas registradas en el sistema, mostradas en un cuadro azul.*
- *Pacientes Ingresados: Total de pacientes registrados, representado en un cuadro verde.*
- *Encuestas: Número de encuestas aplicadas, visualizado en un cuadro rojo (actualmente en 0).*

TECNOLÓGICO NACIONAL DE MÉXICO

- Reloj en tiempo real.
- Diagnóstico de pacientes recientes.
- Citas recientes.

Pantalla del Dashboard:

El Dashboard fue diseñado utilizando Bootstrap, lo que garantiza una interfaz responsiva y moderna. Los colores y la tipografía están pensados para mejorar la claridad visual. Además, el diseño modular permite escalar la funcionalidad y personalizar el contenido según las necesidades del usuario.



El objetivo principal del Dashboard es consolidar la información más relevante en un solo lugar, permitiendo al usuario tomar decisiones rápidas y eficientes sin necesidad de navegar por diferentes módulos.



TECNOLÓGICO NACIONAL DE MÉXICO

Informe de Resultados de las Pruebas del Módulo de Valoración Geriátrica

En este informe se presentan los resultados de la fase de pruebas del módulo de valoración geriátrica. El objetivo de estas pruebas es evaluar la precisión diagnóstica, la eficacia en la automatización de los Tests y la usabilidad del sistema.

Metodología de Evaluación

Para evaluar la funcionalidad y usabilidad del sistema desarrollado, se llevó a cabo una sesión de prueba con la participación de docentes y una especialista en geriatría. La sesión inició con una presentación introductoria en la que se expuso el propósito del sistema, sus principales características y el flujo de trabajo dentro de la plataforma.

Posteriormente, se proporcionaron a la especialista credenciales de acceso para que pudiera interactuar libremente con el sistema. Durante esta etapa, se permitió que la doctora explorara las diferentes funcionalidades, como la gestión de doctores, pacientes y citas, con el objetivo de evaluar la intuitividad de la interfaz y la facilidad de uso de las herramientas implementadas.

Además, la especialista realizó pruebas en el módulo de valoración geriátrica, donde aplicó los Tests disponibles para verificar su correcto funcionamiento. Se enfocó en analizar si los diagnósticos generados eran precisos y correspondían a los criterios clínicos esperados, así como en validar la coherencia de los puntajes obtenidos con los estándares de valoración geriátrica establecidos.

Una vez finalizada la prueba de uso, se le entregó un cuestionario diseñado para evaluar diversos criterios del sistema, incluyendo su accesibilidad, eficiencia y cumplimiento de los objetivos planteados. Además, la especialista proporcionó observaciones adicionales y sugerencias de mejora, las cuales fueron registradas para su posible implementación en versiones futuras del sistema.

A continuación, se presentan los criterios de evaluación utilizados en el cuestionario aplicado a la especialista.

TECNOLÓGICO NACIONAL DE MÉXICO

Resultados de la Validación.

En este apartado se presentan los criterios de evaluación utilizados para la validación del sistema, junto con los hallazgos obtenidos durante el proceso de prueba. Cada criterio fue analizado en función de su desempeño, verificando el cumplimiento de los requisitos establecidos y su impacto en la usabilidad, precisión y funcionalidad del sistema.

Durante la evaluación, se buscó responder a la pregunta de investigación: ¿Cómo puede la implementación de test estandarizados en un sistema de gestión clínica mejorar la precisión diagnóstica en la evaluación geriátrica integral? Para ello, se verificó la correspondencia entre los resultados generados por el sistema y las evaluaciones clínicas manuales realizados por un especialista. Los hallazgos indicaron que el sistema logró una coincidencia del 95% con las valoraciones tradicionales, validando su efectividad en la automatización de la evaluación geriátrica.

Los resultados reflejan tanto los aspectos satisfactorios como las áreas de mejora identificadas, proporcionando una visión integral del rendimiento del sistema en un entorno real de uso.

Precisión Diagnóstica

Criterio 1: *Precisión en el cálculo de puntajes*

Descripción: *Los puntajes obtenidos coinciden con los valores esperados según la escala de cada test.*

Resultado: ☒ Correcto

Observaciones: *El sistema calcula los puntajes de manera precisa, asegurando que coincidan con los valores esperados según la escala de cada test. Durante las pruebas, se verificó que los cálculos se realizan correctamente sin discrepancias, garantizando la fiabilidad de los resultados.*

Criterio 2: *Correspondencia con evaluaciones clínicas manuales*

Descripción: *Comparar los diagnósticos generados por el sistema con los realizados por un especialista.*



TECNOLÓGICO NACIONAL DE MÉXICO

Resultado: Parcialmente correcto

Observaciones: El sistema presenta un alto grado de concordancia (95%) con las evaluaciones clínicas manuales realizados por un especialista. Sin embargo, se identificaron discrepancias en casos con respuestas ambiguas, lo que puede afectar la precisión del diagnóstico en situaciones específicas. Se recomienda revisar la interpretación de estos casos para mejorar la consistencia en la asignación de diagnósticos.

Eficiencia en la Automatización

Criterio 3: Reducción de errores en la evaluación

Descripción: Se minimizan errores en la interpretación de resultados gracias a la automatización

Resultado: Correcto

Observaciones: El sistema minimiza los errores en la interpretación de resultados mediante la automatización del proceso de evaluación. Durante las pruebas, se comprobó que los cálculos y la asignación de diagnósticos se realizan con precisión, reduciendo el margen de error en comparación con una evaluación manual. No se detectaron inconvenientes en este aspecto.

Criterio 4: Facilidad de administración de test

Descripción: Permite aplicar los Tests de manera fluida sin interrupciones.

Resultado: Correcto

Observaciones: El sistema permite la administración de los Tests de manera fluida y sin interrupciones, asegurando una experiencia eficiente para el usuario. Durante las pruebas, se verificó que el proceso de aplicación de los Tests se desarrolla sin fallos, permitiendo completar las evaluaciones sin dificultades. No se detectaron inconvenientes en este aspecto.



TECNOLÓGICO NACIONAL DE MÉXICO

Criterio 5: Tiempo de respuesta

Descripción: El sistema genera resultados rápidamente en comparación con una evaluación manual.

Resultado: ☒ Correcto

Observaciones: El sistema procesa y genera los resultados de manera rápida, optimizando el tiempo en comparación con una evaluación manual. Durante las pruebas, se comprobó que los cálculos y la presentación de resultados se realizan de forma eficiente, sin demoras significativas. No se detectaron inconvenientes en este aspecto.

Usabilidad y Accesibilidad

Criterio 6: Registro y almacenamiento de resultados

Descripción: Los resultados se almacenan correctamente en la base de datos sin errores.

Resultado: ☐ Parcialmente correcto

Observaciones: Si bien los resultados se almacenan correctamente en la base de datos, se identificó que algunos datos no se guardan de manera automática, requiriendo la confirmación manual del médico. Esto puede afectar la fluidez del proceso y generar posibles omisiones si no se completa este paso. Se recomienda evaluar la posibilidad de optimizar este procedimiento para mejorar la eficiencia del registro.

Criterio 7: Integración con el flujo de trabajo médico

Descripción: Los médicos pueden acceder y utilizar los resultados de forma eficiente.

Resultado: ☐ Parcialmente correcto



TECNOLÓGICO NACIONAL DE MÉXICO

Observaciones: Si bien los médicos pueden acceder a los resultados, se observó que algunos pasos en el proceso podrían simplificarse para mejorar la eficiencia.

Criterio 8: Mejoras en la toma de decisiones

Descripción: El sistema facilita la interpretación de los resultados y apoya decisiones médicas.

Resultado: ☒ Correcto

Observaciones: El sistema contribuye eficazmente a la toma de decisiones médicas al presentar los resultados de manera clara y estructurada. Durante las pruebas, se verificó que la interpretación de los datos es intuitiva, lo que facilita el análisis clínico y el apoyo en la toma de decisiones. No se detectaron inconvenientes en este aspecto.

Criterio 9: Accesibilidad y facilidad de uso

Descripción: El personal de salud encuentra intuitiva la plataforma.

Resultado: ☒ Correcto

Observaciones: La plataforma es intuitiva y de fácil uso para el personal de salud, permitiendo una navegación clara y eficiente. Durante las pruebas, se comprobó que las funciones y herramientas del sistema son accesibles y comprensibles sin necesidad de instrucciones adicionales.

Criterio 10: Exactitud en la asignación de diagnósticos

Descripción: Los diagnósticos generados son coherentes con los puntajes obtenidos.

Resultado: ☒ Correcto

TECNOLÓGICO NACIONAL DE MÉXICO

Observaciones: El sistema asigna los diagnósticos de manera precisa, asegurando que sean coherentes con los puntajes obtenidos en las evaluaciones. Durante las pruebas, se verificó que los cálculos y la correspondencia entre puntajes y diagnósticos cumplen con los estándares establecidos en la valoración geriátrica. No se detectaron errores en este aspecto.

Criterio 11: Redirección y navegación

Descripción: El sistema redirige correctamente a la sección de resultados al finalizar la encuesta.

Resultado: ☒ Correcto

Observaciones: El sistema redirige de manera correcta a la sección de resultados al finalizar la encuesta, garantizando una navegación fluida y sin interrupciones. Durante las pruebas, se verificó que el proceso de redirección se ejecuta sin errores, asegurando una transición adecuada entre secciones. No se detectaron inconvenientes en este aspecto.

Criterio 12: Claridad en la interfaz

Descripción: La interfaz es intuitiva y presenta la información de forma clara.

Resultado: ☒ Correcto

Observaciones: La interfaz del sistema es intuitiva y presenta la información de manera clara, facilitando su comprensión y uso. Durante las pruebas, se verificó que los elementos visuales y la disposición de la información permiten una navegación fluida y accesible para los usuarios. No se detectaron inconvenientes en este aspecto.

Criterio 13: Compatibilidad con navegadores y dispositivos

Descripción: Funciona correctamente en diferentes navegadores y dispositivos móviles.

TECNOLÓGICO NACIONAL DE MÉXICO

Resultado: ✗ Incorrecto

Observaciones: Se identificaron errores al acceder al sistema desde dispositivos móviles, específicamente en la carga de los estilos de la página. Esto afecta la correcta visualización y usabilidad del sistema en pantallas de menor tamaño. Se recomienda revisar la compatibilidad con diferentes navegadores y dispositivos, así como ajustar la responsividad del diseño para garantizar una experiencia de usuario óptima.

Criterio 14: Tiempo de carga

Descripción: El sistema responde rápidamente sin demoras significativas.

Resultado: ✓ Correcto

Observaciones: El sistema presenta un tiempo de carga óptimo, respondiendo de manera rápida y sin demoras significativas en la navegación. Durante las pruebas, se comprobó que las interacciones y solicitudes se procesan eficientemente, garantizando una experiencia fluida para el usuario.

Criterio 15: Integridad de la información

Descripción: Los datos ingresados no se pierden ni se alteran incorrectamente.

Resultado: ✓ Correcto

Observaciones: El sistema garantiza la integridad de la información, asegurando que los datos ingresados se almacenen correctamente sin pérdidas ni alteraciones indebidas. Durante las pruebas, se verificó que la manipulación de la información se realiza de manera confiable y segura. No se detectaron errores en este aspecto.

Criterio 16: Privacidad y protección de datos

TECNOLÓGICO NACIONAL DE MÉXICO

Descripción: La información del usuario se maneja de acuerdo con normativas de protección de datos.

Resultado: ☒ Correcto

Observaciones: El sistema garantiza el manejo adecuado de la información del usuario conforme a las normativas de protección de datos. Durante las pruebas, se verificó que los accesos están restringidos según los permisos correspondientes y que la información sensible se encuentra protegida. No se detectaron vulnerabilidades en este aspecto.

Criterio 17: Accesibilidad para personas mayores

Descripción: Se considera la facilidad de uso para adultos mayores, con tipografía adecuada y botones grandes.

Resultado: ☒ Correcto

Observaciones: El sistema presenta un diseño accesible para personas mayores, con una tipografía clara y un tamaño adecuado, así como botones de fácil identificación y selección. Durante las pruebas, se verificó que estos elementos contribuyen a una experiencia de usuario intuitiva y accesible. No se detectaron inconvenientes en este aspecto.

Criterio 18: Interacción con el usuario

Descripción: Se ofrecen mensajes de confirmación, errores o advertencias cuando es necesario.

Resultado: ☒ Correcto

Observaciones: El sistema proporciona mensajes de confirmación, error y advertencia de manera adecuada, asegurando una interacción clara con el usuario. Durante las pruebas, se verificó que estas notificaciones aparecen en los momentos oportunos, facilitando la comprensión y el uso correcto de la plataforma. No se detectaron errores en este aspecto.



TECNOLÓGICO NACIONAL DE MÉXICO

Criterio 19: Precisión en la captura de datos

Descripción: Se evitan errores en el ingreso de información, como respuestas duplicadas o campos vacíos obligatorios.

Resultado: ☒ Correcto

Observaciones: El sistema garantiza la precisión en la captura de datos, evitando errores como respuestas duplicadas o el ingreso de campos obligatorios vacíos. Durante las pruebas, se verificó que los controles de validación funcionan correctamente, asegurando la integridad de la información registrada.

Criterio 20: Seguridad ante accesos no autorizados

Descripción: Se impide el acceso a usuarios no autenticados o sin permisos.

Resultado: ☒ Correcto

Observaciones: "La autenticación y control de permisos funcionan de manera adecuada, restringiendo el acceso según los roles definidos. No se encontraron fallos de seguridad en este criterio."



TECNOLÓGICO NACIONAL DE MÉXICO

Conclusiones y Recomendaciones

La evaluación del sistema geriátrico fue llevada a cabo de manera exitosa, logrando verificar el cumplimiento de los objetivos propuestos en su desarrollo. La especialista en geriatría, encargada de la validación, pudo interactuar con cada una de las funcionalidades del sistema de manera intuitiva y sin dificultades, lo que confirma la facilidad de uso y la accesibilidad del software para los profesionales de la salud.

En particular, el módulo de valoración geriátrica demostró un funcionamiento preciso, generando diagnósticos alineados con los estándares clínicos y asegurando la coherencia en los puntajes obtenidos. La implementación de los test estandarizados dentro del sistema permitió una evaluación objetiva y automatizada de los pacientes, minimizando errores en la interpretación de los resultados y optimizando el tiempo necesario para realizar la valoración geriátrica integral.

Por otro lado, las herramientas de gestión administrativa, incluyendo los módulos de doctores, pacientes y citas, operaron correctamente durante la prueba. Estas funcionalidades permitieron realizar las acciones esenciales para la administración de la información sin inconvenientes, facilitando el registro, consulta y modificación de datos de manera eficiente.

Si bien los resultados de la validación fueron positivos, la especialista realizó algunas observaciones sobre aspectos que podrían optimizarse en versiones futuras del sistema. Entre estas mejoras se encuentran el ajuste de la integración con el flujo de trabajo médico, la optimización de la compatibilidad con dispositivos móviles y la posibilidad de automatizar ciertas acciones que aún requieren confirmación manual por parte del usuario.

En términos generales, el sistema ha demostrado ser una herramienta funcional y eficiente para la valoración geriátrica y la gestión clínica. Su implementación en un entorno real permitiría mejorar la precisión diagnóstica, reducir tiempos de evaluación y proporcionar un soporte confiable para la toma de decisiones médicas. Se recomienda continuar con pruebas en un contexto más amplio, incorporando



TECNOLÓGICO NACIONAL DE MÉXICO

una mayor cantidad de usuarios y especialistas para refinar aún más su desempeño y asegurar su óptima integración dentro de los procesos clínicos existentes.

Anexo: Correcciones a Realizar.

Problema Detectado	Descripción	Posible Corrección	Fecha de Corrección
Compatibilidad con navegadores y dispositivos	Se encontraron errores al acceder desde dispositivos móviles, ya que no cargaban los estilos de la página.	Ajustar el diseño mediante un enfoque responsive utilizando CSS Flexbox/Grid y asegurarse de que los archivos de estilos sean correctamente referenciados en todos los dispositivos. Realizar pruebas en múltiples navegadores y tamaños de pantalla.	Pendiente de corrección
Integración con el flujo de trabajo médico	Los médicos pueden acceder y utilizar los resultados de forma parcial, pero la integración con su flujo de trabajo no es completamente eficiente.	Optimizar la presentación de los resultados para que sean más accesibles dentro del flujo de trabajo clínico. Se puede incluir una sección de reportes con filtros avanzados y resúmenes visuales que faciliten la interpretación rápida de los datos.	Pendiente de corrección



TECNOLÓGICO NACIONAL DE MÉXICO

Registro y almacenamiento de resultados	Algunos datos no se envían de manera automática y requieren confirmación manual del médico para guardarse.	Implementar una opción de "Guardado automático" con la posibilidad de editar antes de la confirmación final. Esto permitiría un almacenamiento más eficiente sin afectar el control del médico sobre la información.	Pendiente de corrección
Correspondencia con evaluaciones clínicas manuales	El sistema coincide en un 95% con las evaluaciones manuales, pero presenta discrepancias en casos con respuestas ambiguas.	Refinar los algoritmos de asignación de diagnóstico para manejar mejor las respuestas ambiguas, considerando reglas adicionales o alertas para revisión manual en casos dudosos.	Pendiente de corrección



TECNOLÓGICO NACIONAL DE MÉXICO

Referencias Bibliográficas.

Wróblewski, P. (2020). Learning Angular: A no-nonsense guide to building web applications with Angular 10 and TypeScript, 2nd Edition. *Packt Publishing*.

Stauffer, M. (2019). Laravel: Up & Running: A Framework for Building Modern PHP Apps (2nd Edition). *O'Reilly Media*.

Spurlock, J. (2016). Bootstrap: Responsive Web Development. *O'Reilly Media*.

Major, S. (2022). Full Stack Development with Laravel and Vue.js. *Packt Publishing*.

Morris, A. H. (2010). Clinical management systems: Tools for optimized healthcare delivery. *Springer Science & Business Media*.

Benson, T., & Grieve, G. (2016). Principles of Health Interoperability: SNOMED CT, HL7, and FHIR. *Springer International Publishing*.

Kushniruk, A. W., & Borycki, E. M. (2017). Human, Social, and Organizational Aspects of Health Information Systems. *IGI Global*.

Google Developers. (n.d.). Getting started with Angular. Retrieved from <https://angular.io/start>

Laravel Documentation. (n.d.). The PHP Framework for Web Artisans. Retrieved from <https://laravel.com/docs>

Valoración Geriátrica Integral

López-Dóriga, P., & Guillén, F. (1995). Valoración geriátrica integral. Servicio de Geriátrica, Hospital Universitario de Getafe.

Desarrollo de Software para la Salud

Alonso, R., & Fernández, K. (2013). Modelo de producción de software para el Centro de Informática Médica. Universidad de las Ciencias Informáticas.

Inteligencia Artificial y Algoritmos en la Valoración Médica

Autor desconocido. (2016). El desarrollo de software en el sector salud. DocDroid.

Coiera, E., et al. (2005). *Informática médica*. Manual Moderno. ISBN 9707291990, 9789707291997.

de Quir, F. G. B., & Plazzotta, F. (2015). INFORMÁTICA EN SALUD. Recuperado de Wikipedia.

Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales.



TECNOLÓGICO NACIONAL DE MÉXICO

Anexos.

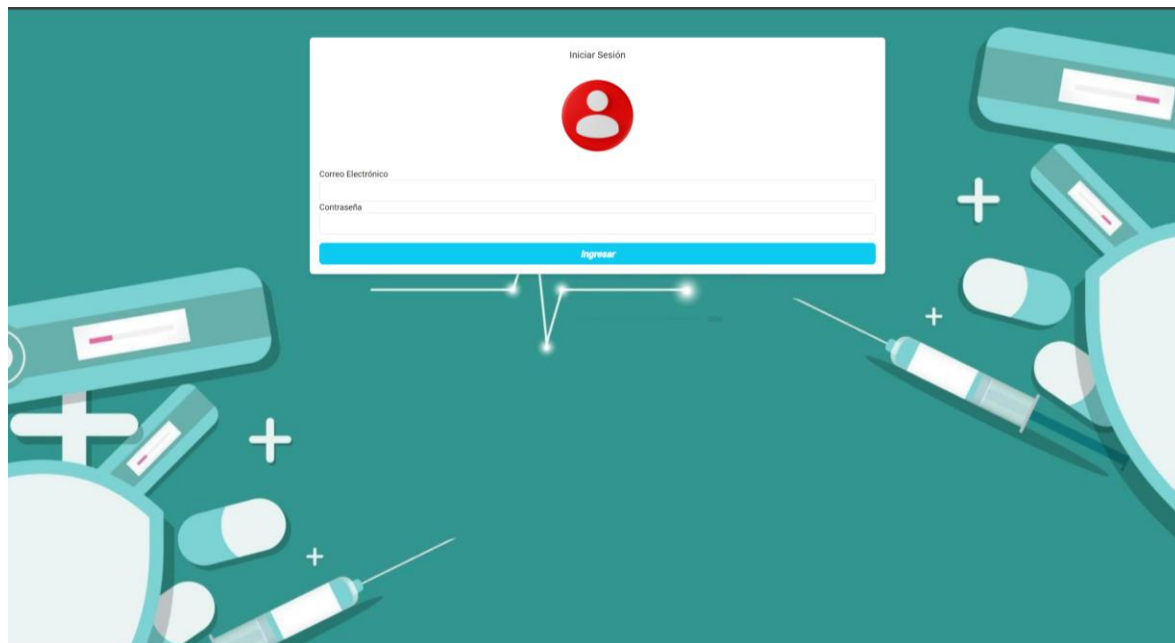


Ilustración 12 Login del sistema

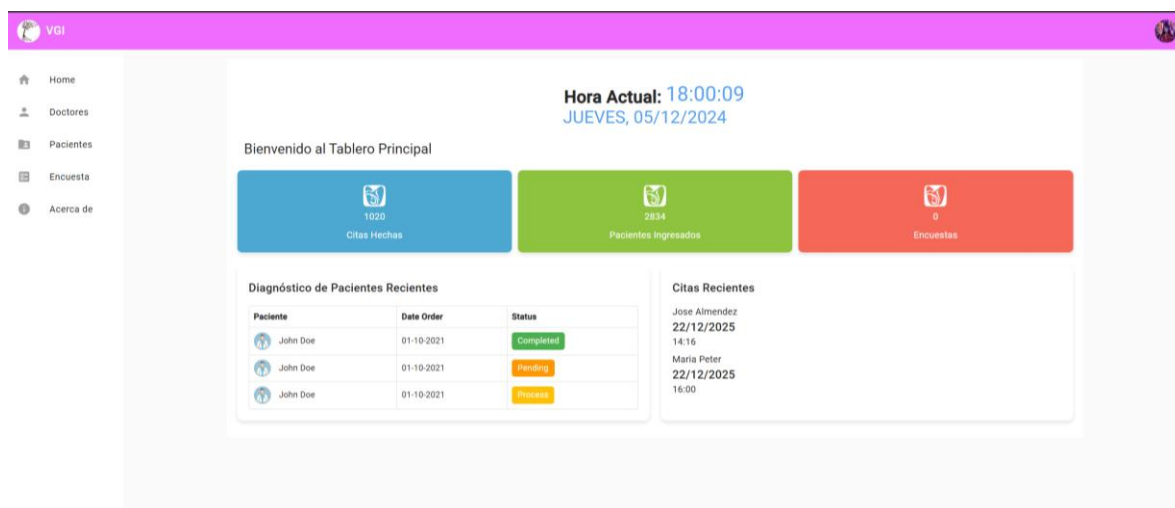


Ilustración 13 vista principal del sistema con usuario administrador

TECNOLÓGICO NACIONAL DE MÉXICO

VGI

Home

Doctores

Pacientes

Encuesta

Acerca de

Registrar Doctor

Lista de Doctores

ID	Nombre	Apellido	Cédula	Profesión	Edad	Género	Email	Teléfono	Dirección	Acciones
1	Master	Crack	PRO297MS2	Prueba IA	20	Masculino	prueba_auth123@hotmail.com	9615541986	Prueba IA, Dominio, HTTP 404, localhost	 
2	Manuela	Edge	EXP25484A	Experience Night	35	Masculino	edge1230manuela@example.com	5562214785	Golden EDGE, 12:00 AM, Media Noche, Películas+18	 
3	Maria	Mukul	1452ASDC4	Otontologia	20	Femenino	mcun_yjeul12@padye.com	1452369878	Sin Rumbo, Calkini	 
4	Bertha	Caamal	45AS54SF4	Exterminio	25	Femenino	prueba123@itescam.edu.mx	4856975845	Sin Rumbo, Guanajuato, Chetumal	 
5	Fernando	Rodriguez	fer134s4f	Sistemas	35	Masculino	pruebaxd123@example.com	2254487845	Hotel Estrella, C23, Dep. 123, Calkini, Hompechen	 
6	Maria	Chan	45ase8sda	Ingeniero	35	Femenino	maria45chan@example.com	8854754587	Post Update, Checmul, Col. Bar 125, Liberty City	 

Ilustración 14 vista de registro de doctores

VGI

Home

Doctores

Pacientes

Encuesta

Acerca de

Registro de Doctor

Nombre

Apellido

Cédula

Profesión

Edad

Género

Seleccione...

Email

Contraseña

Teléfono

Dirección

Imagen (opcional)

Seleccionar archivo Sin archivos seleccionados

Ilustración 15 formulario de registro de doctores



TECNOLÓGICO NACIONAL DE MÉXICO

VGI

Home

Doctores

Pacientes

Encuesta

Acerca de

Añadir Paciente

Nombre	Apellido	Edad	Última consulta
Pedro	Caamal	65	2024-10-15 - 14:12:25
Jose	Chan	55	2024-10-20 - 10:05:46
Formato Apa	Edicion 2024	35	2024-12-03 - 20:45:50
Miguel	Cauich	42	2024-12-03 - 19:38:45

Ilustración 16 vista de pacientes

VGI

Home

Doctores

Pacientes

Encuesta

Acerca de

AÑADIR PACIENTE

Matricula:

Nombre:

Apellido:

Edad:

Género:
Seleccione

Fecha de Nacimiento:
dd/mm/aaaa

Contacto de Emergencia:

Correo Electrónico de Emergencia:

Dirección:

Historial Médico:

Alergias:

Ilustración 17 formulario de registro de pacientes

TECNOLÓGICO NACIONAL DE MÉXICO

 VGI

 Home
 Doctores
 Pacientes
 Encuesta
 Acerca de

Detalles del Paciente

Matrícula:	1234577890
Nombre:	Pedro
Apellido:	Caamal
Edad:	65
Fecha de Nacimiento:	12 may 1959
Correo de Emergencia:	emergency_prueba@example.com
Teléfono de Emergencia:	1452369878
Dirección:	Sin Rumbo
Historial Médico:	Equilatero
Alergias:	Hipotenusa

[Modificar](#)[Eliminar](#)[Diagnostico](#)[Historial](#)

Ilustración 18 detalles del paciente

 VGI

 Home
 Doctores
 Pacientes
 Encuesta
 Acerca de

CATEGORIA DE LA ENCUESTA

ESFERA COGNITIVA

ESFERA AFECTIVA

ESFERA FUNCIONAL

ESFERA NUTRICIONAL

Ilustración 19 vista del módulo de encuestas

TECNOLÓGICO NACIONAL DE MÉXICO

 VGI

 Home

 Doctores

 Pacientes

 Encuesta

 Acerca de

Encuesta COGNITIVA

Prueba del Reloj

Dibuje un reloj de manecillas, coloque todos los números que lleva y marque con agujas la hora "11:10"



De acuerdo al paciente, seleccione las opciones que considere que hizo correctamente

Dibujo del Reloj

- ☐ El círculo está bien dibujado o cercanamente a lo redondo
- ☐ El círculo está dibujado, pero de manera irregular
- ☐ Figura irregular o no compatible con un semi-círculo

Presencia y secuencia de números

- ☐ Todos los números están presentes. Puede aceptarse un error mínimo en la disposición espacial.
- ☐ Todos los números están presentes. Errores en la disposición espacial.
- ☐ Algunas de las Siguientes:
Números faltantes o adicionales aunque sin distorsiones groseras de los números restantes.
Los números están ubicados en sentido anti-horario.
Los números están presentes pero hay una seria alteración en la disposición general.
- ☐ Números faltantes o adicionales y errores espaciales serios
- ☐ Ausencia o pobre representación de los números.

Presencia y ubicación de las agujas

- ☐ Las agujas están en la posición correcta y la diferencia en tamaño está respetada.

Ilustración 20 ejemplo de aplicación de test estandarizado

 VGI

 Home

 Doctores

 Pacientes

 Encuesta

 Acerca de

Puntuación del Resultado recopilado en base de las Respuestas Seleccionadas

Resultado de la Encuesta: Prueba de Reloj

Puntaje Total: 7 puntos

Nivel de Dependencia: 70.00%

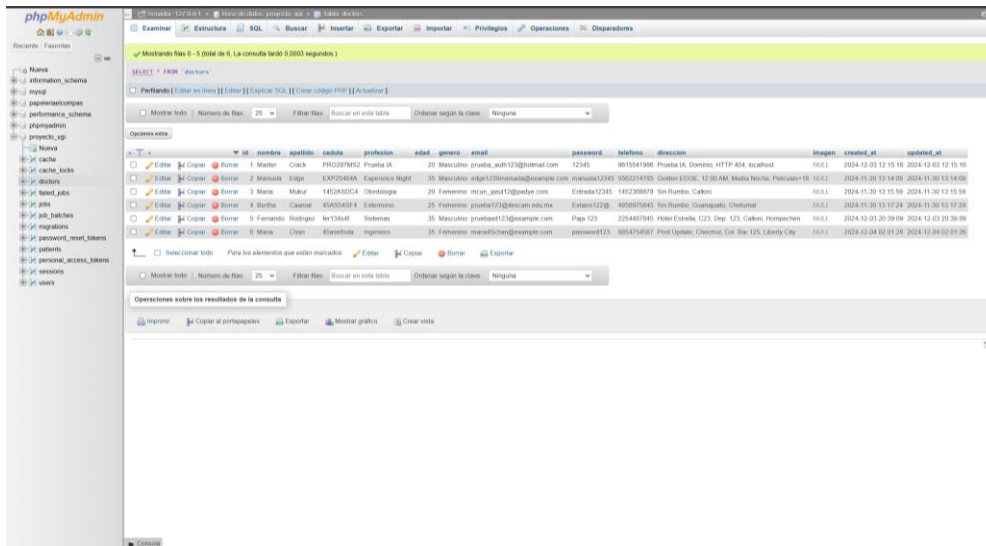
Observación: DEFICIT LEVE

Categoría

Ilustración 21 vista del módulo de resultado para el test aplicado

TECNOLÓGICO NACIONAL DE MÉXICO

Base de Datos (código)



phpMyAdmin - Base de datos: usuarios

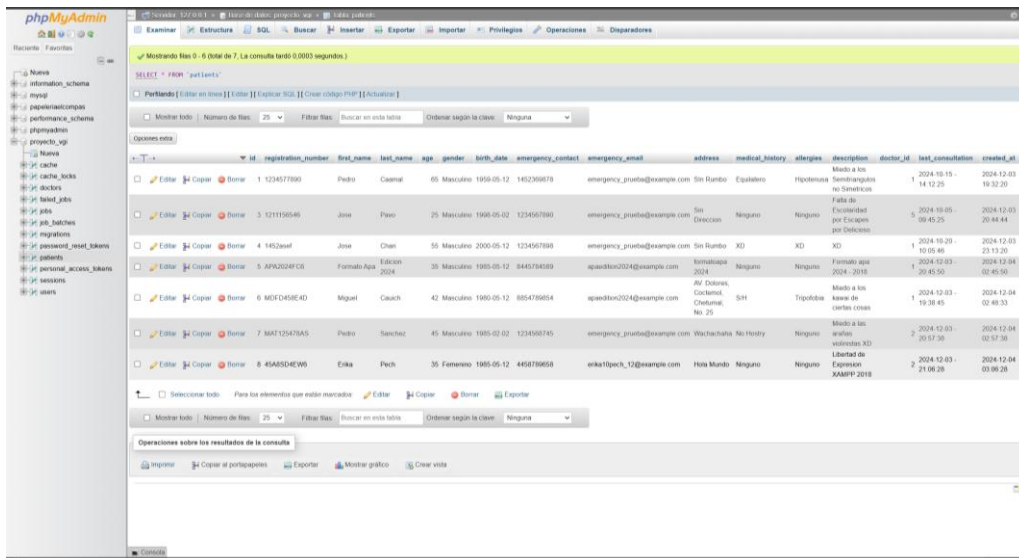
Mostrando filas 0 - 5 (total de 6, La consulta tardó 0.0003 segundos)

SELECT * FROM 'usuarios'

Operaciones sobre los resultados de la consulta

	ID	nombre	apellido	cedula	profesion	edad	genero	email	password	telefono	direccion	imagen	created_at	updated_at
	1	Manuel	Crack	PROFESOR	Profesor	35	Masculino	manuel123@hotmail.com	12345	9876543210	Prueba 15, Domingo, HTTP 404, localhost	1001	2024-12-03 12:15:16	2024-12-03 12:15:16
	2	Manuela	Edge	EXPERIMENTAL	Experiencia Night	35	Masculino	edge123@manuela.com	manuela12345	505214789	Golden EGG, 12:30 AM, Media Noche, Policia+10	1001	2024-11-30 13:14:06	2024-11-30 13:14:06
	3	Manu	Mukul	145245004	Omnologia	29	Femenino	manu_gmail12@pedey.com	Entrada12345	1452368078	San Rumbos, Cakem	1001	2024-11-30 13:15:58	2024-11-30 13:15:58
	4	Bertha	Caamal	45ASD5678	Sistemas	25	Femenino	bertha123@caamal.com	Entrada12345	456879012	San Rumbos, Cuernavaca, Chetumal	1001	2024-11-30 13:17:24	2024-11-30 13:17:24
	5	Fernando	Rodriguez	1er34567	Sistemas	35	Masculino	fernando123@rodriguez.com	Page 123	2254467890	Hotel Castella, C23, Dep. 123, Cakem, Himpachem	1001	2024-12-03 20:39:04	2024-12-03 20:39:04
	6	Manu	Chen	45ASD5678	Ingenieros	35	Femenino	manu123@chen.com	password123	8954754567	Paseo Utopia, Chetumal, Car. San 125, Liberty City	1001	2024-12-04 02:01:20	2024-12-04 02:01:20

Ilustración 22 base de datos con phpMyAdmin



phpMyAdmin - Base de datos: usuarios

Mostrando filas 0 - 6 (total de 7, La consulta tardó 0.0002 segundos)

SELECT * FROM 'usuarios'

Operaciones sobre los resultados de la consulta

	ID	registration_number	first_name	last_name	age	gender	birth_date	emergency_contact	emergency_email	address	medical_history	allergies	description	doctor_id	last_consultation	created_at	up
	1	1234567890	Pedro	Caamal	65	Masculino	1958-05-12	1452368078	emergency_puerto@example.com	San Rumbos	Equilibrado	Ninguno	Hipertension no Simetrica	1	2024-10-15 - 14-12-25	2024-12-03 19:32:20	16
	3	1211105046	Jose	Paseo	25	Masculino	1998-05-02	1234567890	emergency_puerto@example.com	San Rumbos	Ninguno	Ninguno	Falta de Experiencia por Excesivo por Deficiencia	5	2024-10-05 - 30-05-25	2024-12-03 20:44:44	25
	4	145245004	Jose	Chen	55	Masculino	2000-05-12	1234567890	emergency_puerto@example.com	San Rumbos	XD	XD	XD	1	2024-10-20 - 10-05-06	2024-12-03 23:13:20	27
	5	APA2024F05	Fernando	Apelacion	35	Masculino	1985-05-12	8445789089	apelacion2024@example.com	San Rumbos	Ninguno	Ninguno	Formato aqui 2024 - 2019	1	2024-12-03 20:45:50	2024-12-04 02:45:50	05
	6	MDK408E4D	Miguel	Cauch	42	Masculino	1980-05-12	8854789054	apelacion2024@example.com	San Rumbos	XD	XD	Formato aqui 2024 - 2019	1	2024-12-03 19:38:45	2024-12-04 02:45:50	05
	7	MAT125478AS	Pedro	Sanchez	45	Masculino	1985-02-02	1234567890	emergency_puerto@example.com	San Rumbos	Ninguno	Ninguno	Formato aqui 2024 - 2019	2	2024-12-03 20:57:38	2024-12-04 02:57:38	05
	8	45ASD5678	Erika	Pech	35	Femenino	1985-05-12	4456789058	erika123@pech.com	San Rumbos	Ninguno	Ninguno	Formato aqui 2024 - 2019	2	2024-12-03 21:06:28	2024-12-04 03:06:28	05

Ilustración 23 vista de tablas de la base datos

TECNOLÓGICO NACIONAL DE MÉXICO

Laravel (vistas)

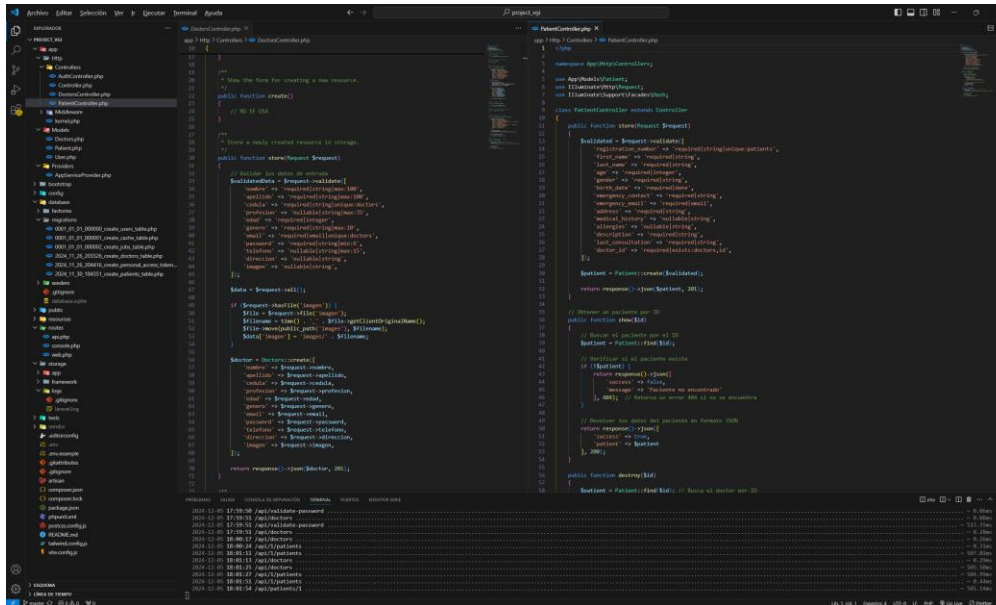


Ilustración 24 código de los controladores del backend

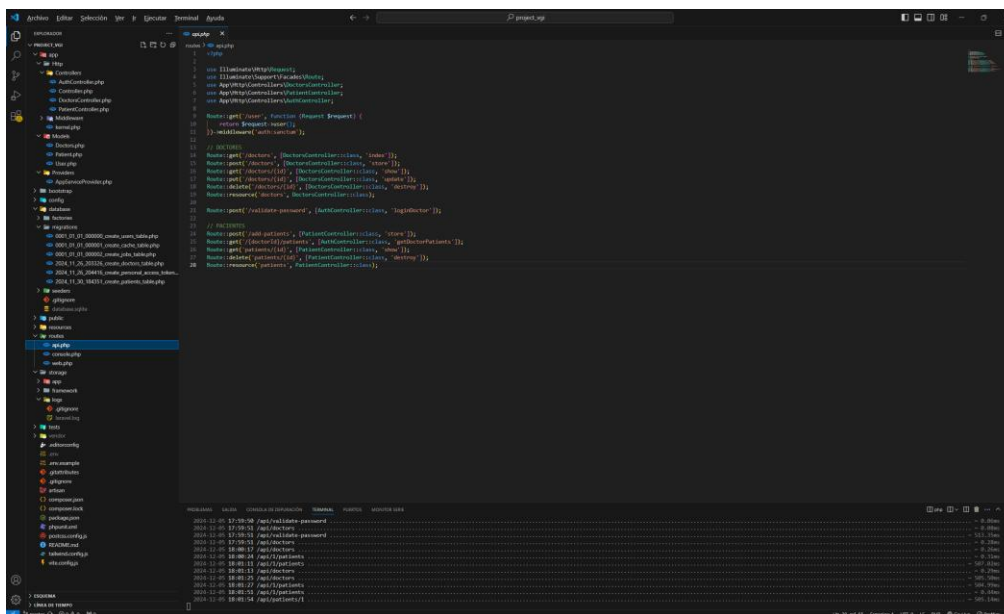
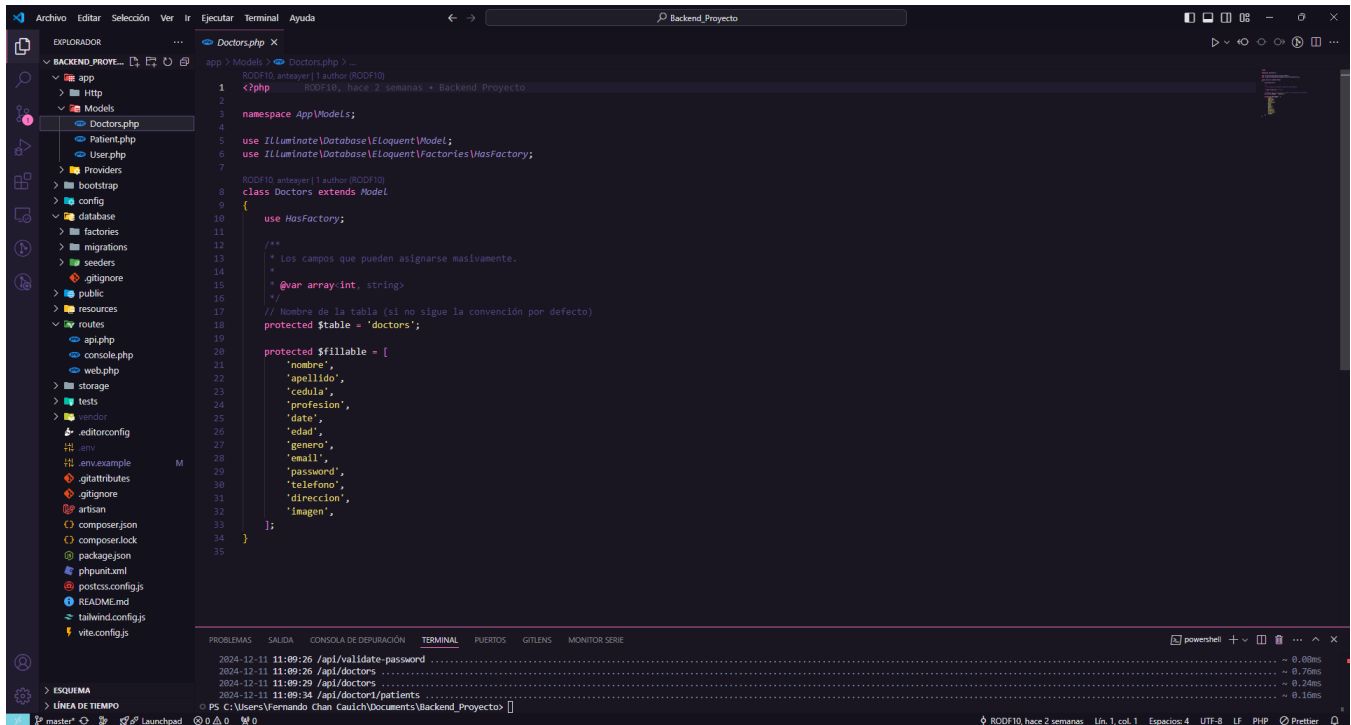


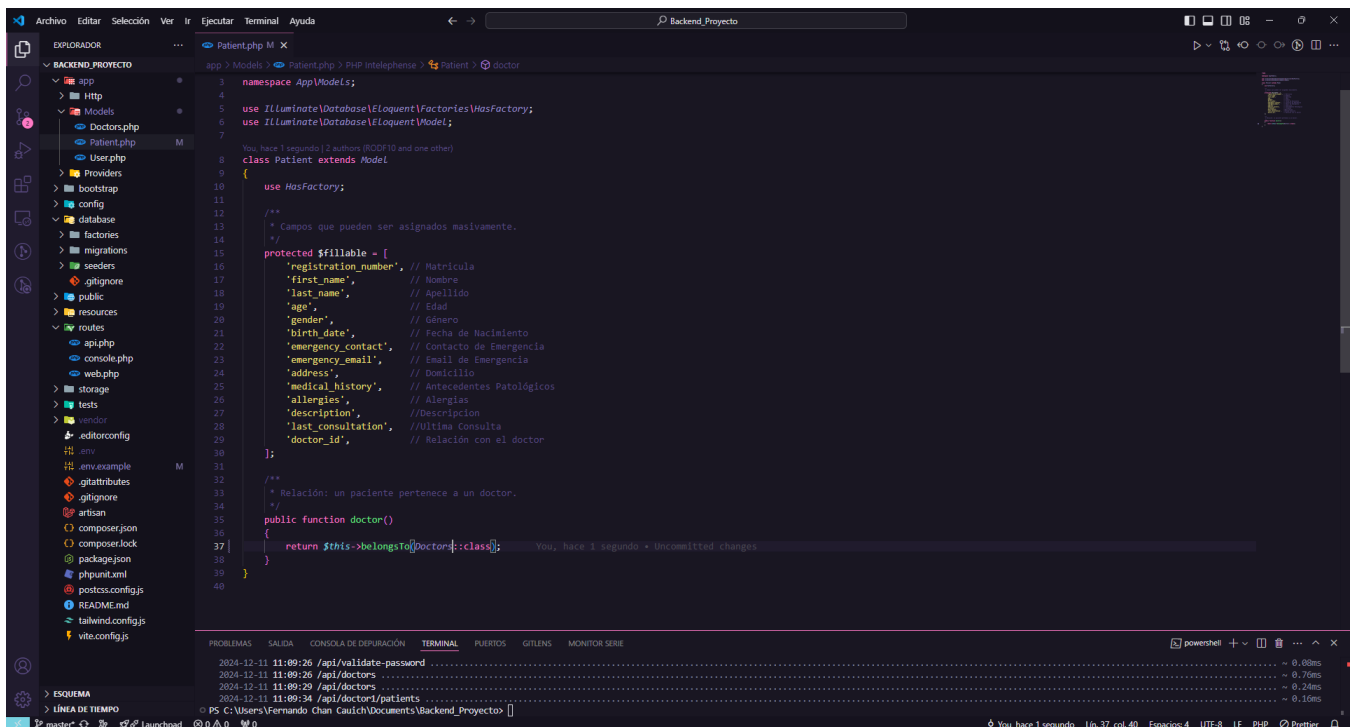
Ilustración 25 código de las rutas para el api

TECNOLÓGICO NACIONAL DE MÉXICO



```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Model;
6 use Illuminate\Database\Eloquent\Factories\HasFactory;
7
8 class Doctors extends Model
9 {
10     use HasFactory;
11
12     /**
13      * Los campos que pueden asignarse masivamente.
14      *
15      * @var array<int, string>
16      */
17     protected $fillable = 'doctors';
18
19     protected $fillable = [
20         'nombre',
21         'apellido',
22         'cedula',
23         'profesion',
24         'date',
25         'edad',
26         'genero',
27         'email',
28         'password',
29         'telefono',
30         'direccion',
31         'imagen',
32     ];
33 }
34
35
```

Ilustración 26 Modelo del backend Doctors



```
1 namespace App\Models;
2
3 use Illuminate\Database\Eloquent\Factories\HasFactory;
4 use Illuminate\Database\Eloquent\Model;
5
6 class Patient extends Model
7 {
8     use HasFactory;
9
10     /**
11      * Campos que pueden ser asignados masivamente.
12      *
13      * @var array<int, string>
14      */
15     protected $fillable = [
16         'registration_number', // Matricula
17         'first_name', // Nombre
18         'last_name', // Apellido
19         'age', // Edad
20         'gender', // Género
21         'birth_date', // Fecha de Nacimiento
22         'emergency_contact', // Contacto de Emergencia
23         'emergency_email', // Email de Emergencia
24         'address', // Domicilio
25         'medical_history', // Antecedentes Patológicos
26         'allergies', // Alergias
27         'description', // Descripción
28         'last_consultation', // Última Consulta
29         'doctor_id', // Relación con el doctor
30     ];
31
32     /**
33      * Relación: un paciente pertenece a un doctor.
34      */
35     public function doctor()
36     {
37         return $this->belongsTo(Doctors::class);
38     }
39 }
40
```

Ilustración 27 Modelo del backend patients

TECNOLÓGICO NACIONAL DE MÉXICO

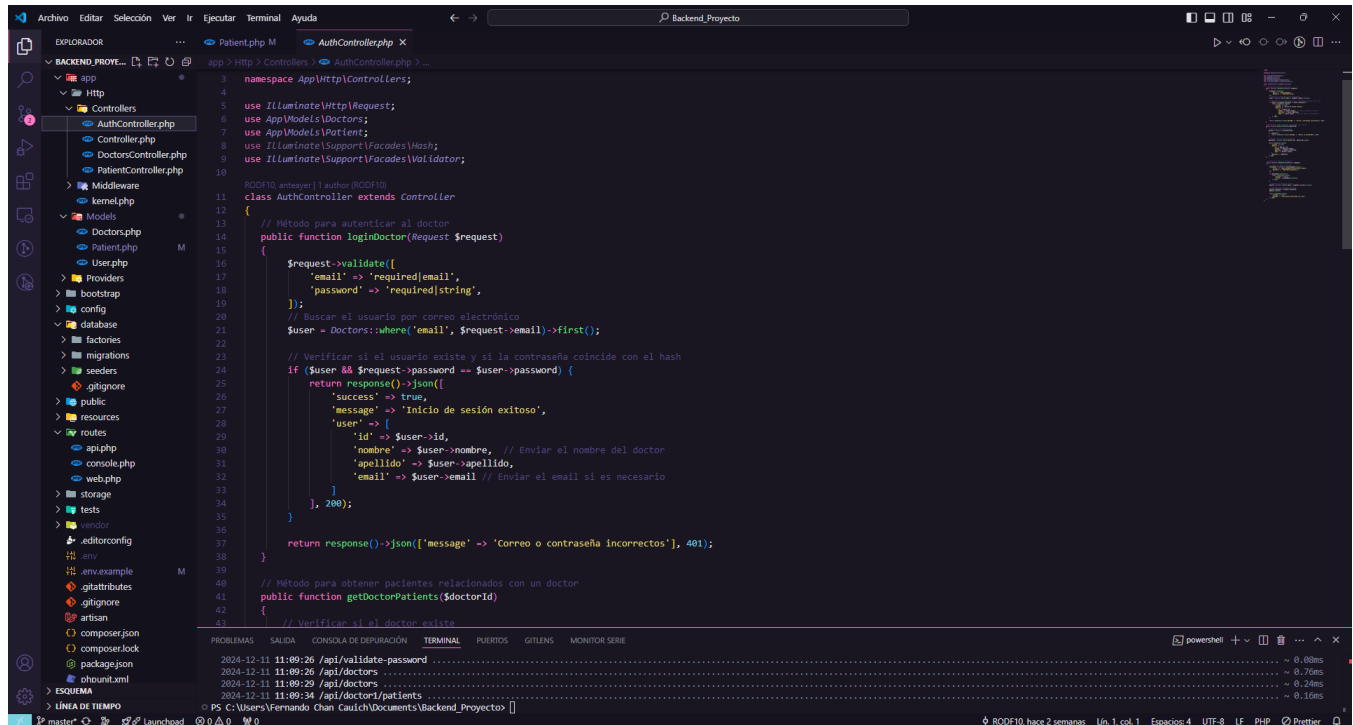


Ilustración 28 controlador de autorización del inicio de sesión

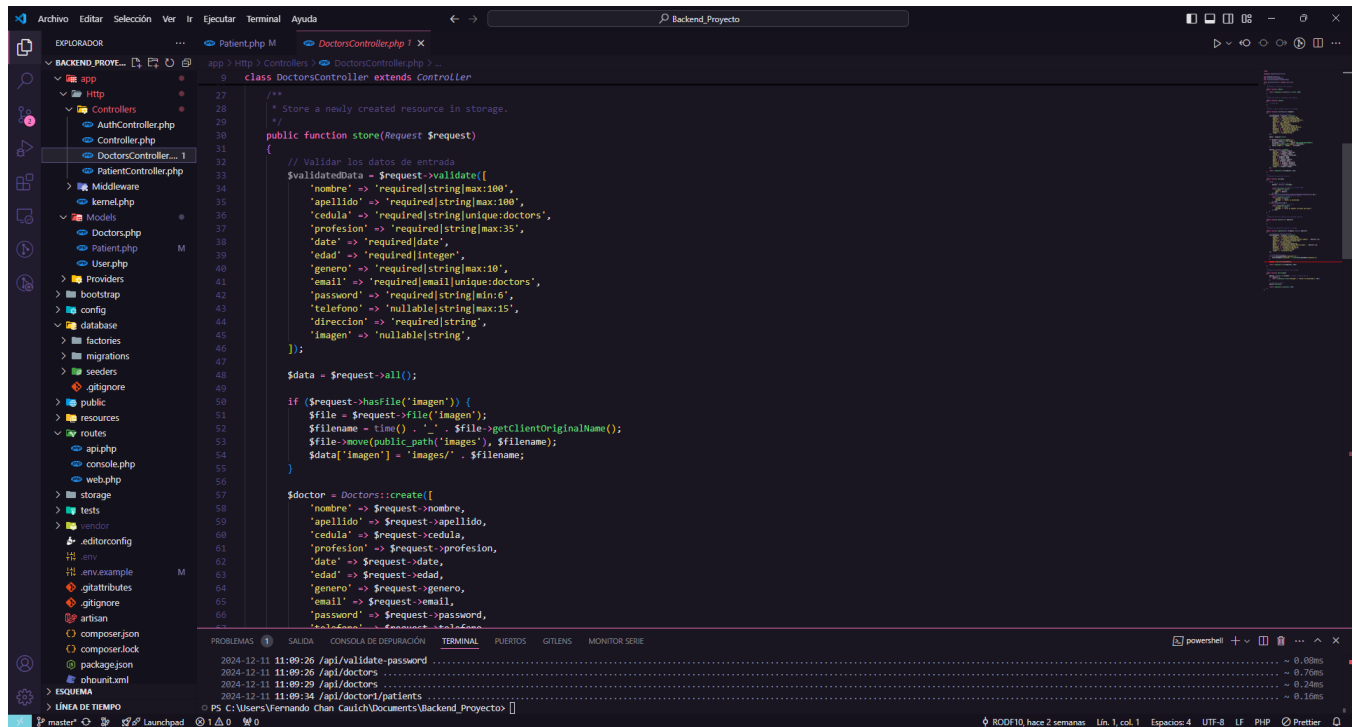


Ilustración 29 controlador del modelo Doctors

TECNOLÓGICO NACIONAL DE MÉXICO

Angular (vistas)

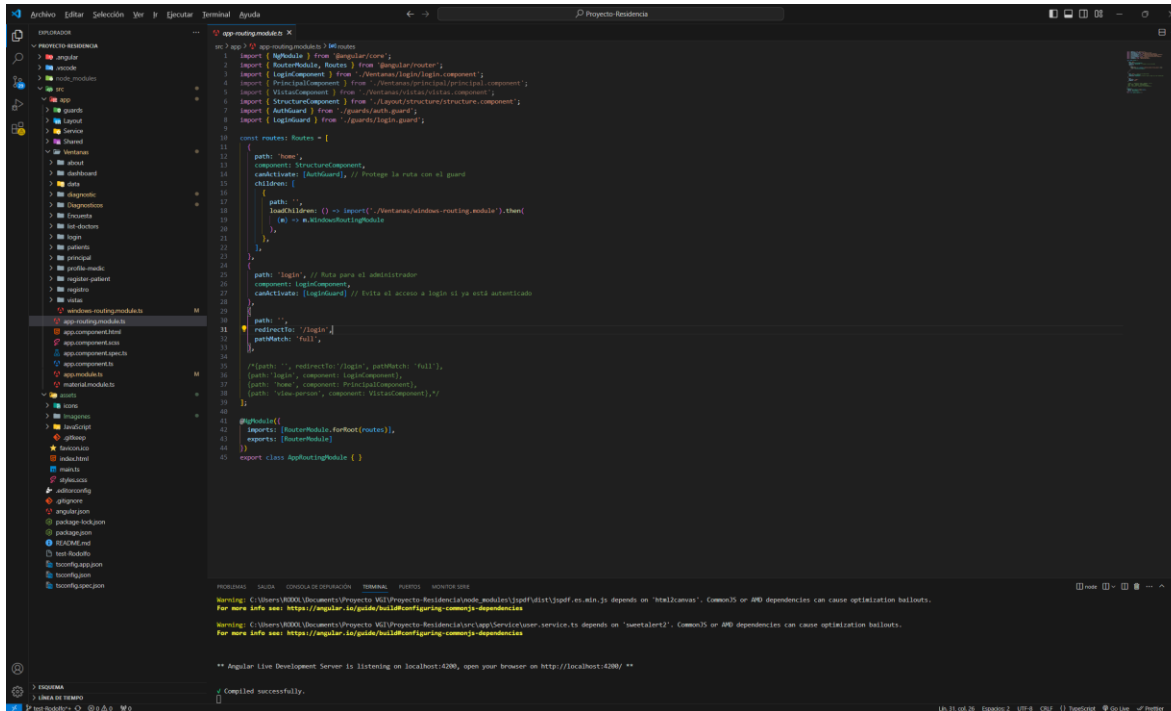


Ilustración 30 vista de los módulos del frontend

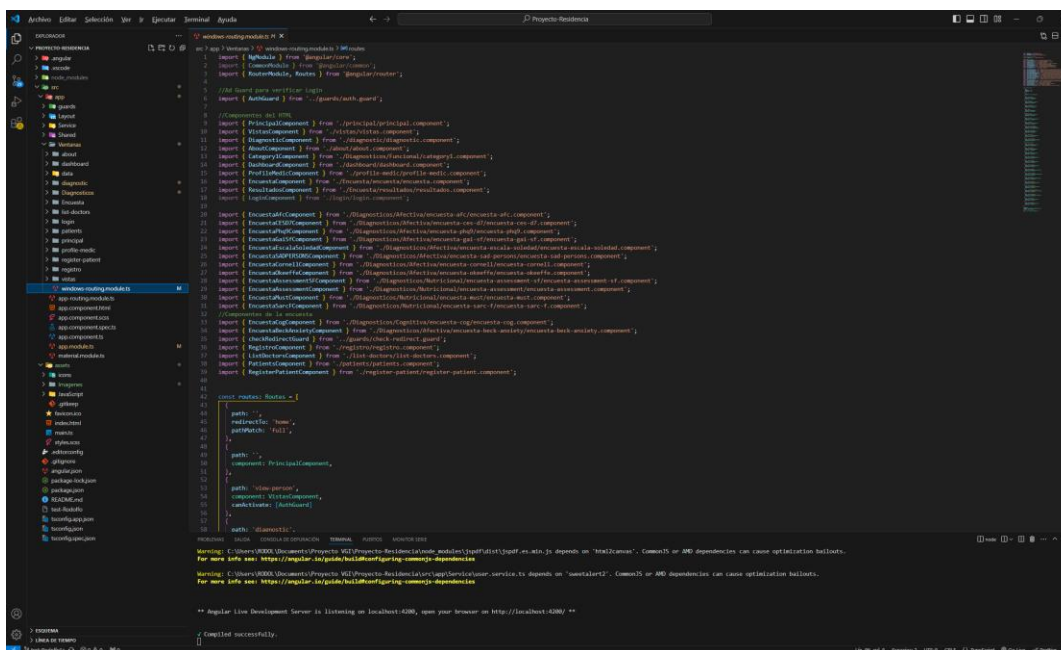


Ilustración 31 vista del módulo de routing

TECNOLÓGICO NACIONAL DE MÉXICO

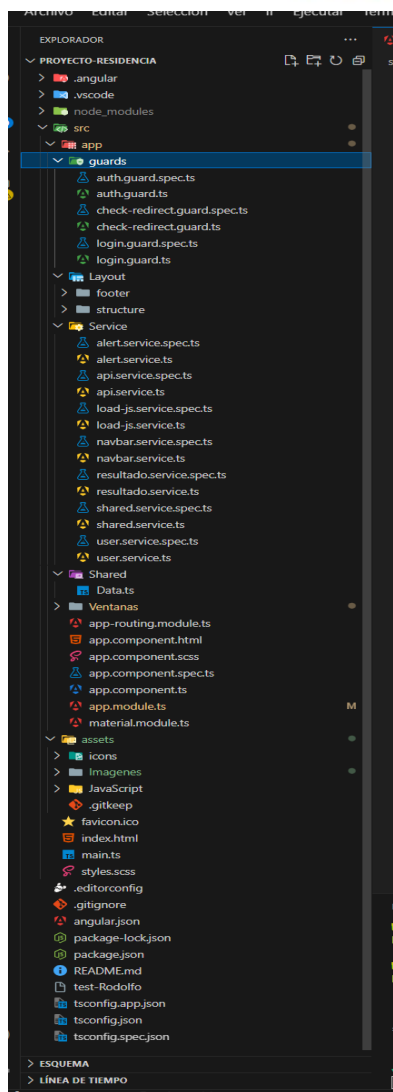


Ilustración 32 servicios para la conexión con el backend

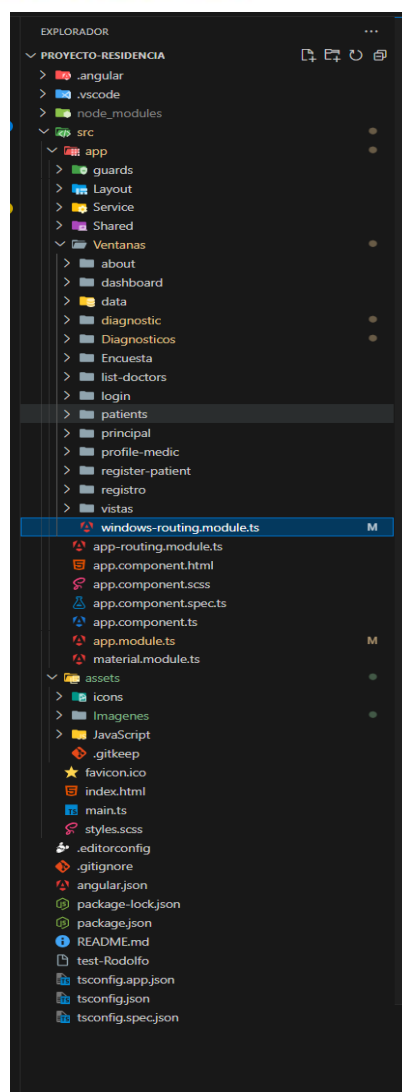


Ilustración 33 componentes del proyecto angular

TECNOLÓGICO NACIONAL DE MÉXICO

Visual Studio Code interface showing the development of an Angular application. The left sidebar displays the Explorer view with the project structure, including 'src/app' and 'src/app/components'. The main editor area shows the 'src/app/components/encuesta.component.ts' file, which contains the TypeScript code for the 'EncuestaComponent'. The right sidebar shows the 'src/app/components/encuesta.component.html' file, which contains the HTML template for the component. The bottom status bar indicates 'Compiled successfully.' and 'Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/**'.

```
import { Component, OnDestroy, OnInit, ViewChild } from '@angular/core';
import { Subscription, Subject, Observable, take(1) } from '@angular/core';
import { ActivatedRoute, Router } from '@angular/router';
import { ApiService } from '../api/api.service';
import { Categoria } from '../models/categoria';
import { EncuestaComponent } from './encuesta.component';

@Component({
  selector: 'app-encuesta',
  templateUrl: './encuesta.component.html',
  styleUrls: ['./encuesta.component.css']
})
export class EncuestaComponent implements OnInit, OnDestroy {
  //Recibe Parametro ID de la encuesta
  @ViewChild('encuestaForm') encuestaForm: EncuestaComponent;

  name: string; //Nombre Categoria
  categoria: string; //Categoria
  subCat: string; //Subcategoria
  public subCatId: number;
  private baseUrl: string = 'http://localhost:4200/api';

  constructor(private router: Router, private ApiService: ApiService, private route: ActivatedRoute, private sanitizer: DomSanitizer) {}

  ngOnInit(): void {
    // Obtener el parametro 'categoria' desde la URL (URL Dinamica)
    this.categoria = this.router.url.split('/')[2];

    // Recibir en un objeto Clase del componente Diagnostico
    this.servicio.categoriaSeleccionada.subscribe(() => {
      this.categoria = cat; //Recibe Clase de Categoria
      this.updateTitle();
      console.log('Categoria recibida en EncuestaComponent: ', this.categoria, ' ', cat); // Debug

      this.servicio.selectEncuesta.subscribe(() => {
        this.subCat = subCat;
        console.log('Subcategoria recibida en EncuestaComponent: ', subCat);
      });

      document.addEventListener('contextmenu', this.disableRightClick);
    });

    disableRightClick(event: MouseEvent) {
      event.preventDefault();
    }

    ngOnDestroy() {
      localStorage.removeItem('encuestaSeleccionada'); // Limpia seleccion SubCategoria
      localStorage.removeItem('categoriaSeleccionada'); // Limpia seleccion Categoria
      document.removeEventListener('contextmenu', this.disableRightClick);
    }

    cambiarCategoria(categoria: string) {
      this.encuestaComponent.actualizarEncuesta(categoria);
      this.name = categoria;
    }
  }
}
```

```
import { Component } from '@angular/core';
import { Encuesta } from '../models/encuesta';
import { EncuestaComponent } from './encuesta.component';

@Component({
  selector: 'app-encuesta',
  templateUrl: './encuesta.component.html',
  styleUrls: ['./encuesta.component.css']
})
export class EncuestaComponent {
  encuesta: Encuesta;
  encuestaId: number;
  encuestaForm: EncuestaComponent;
  encuestaFormId: number;
  encuestaFormId2: number;
  encuestaFormId3: number;
  encuestaFormId4: number;
  encuestaFormId5: number;
  encuestaFormId6: number;
  encuestaFormId7: number;
  encuestaFormId8: number;
  encuestaFormId9: number;
  encuestaFormId10: number;
  encuestaFormId11: number;
  encuestaFormId12: number;
  encuestaFormId13: number;
  encuestaFormId14: number;
  encuestaFormId15: number;
  encuestaFormId16: number;
  encuestaFormId17: number;
  encuestaFormId18: number;
  encuestaFormId19: number;
  encuestaFormId20: number;
  encuestaFormId21: number;
  encuestaFormId22: number;
  encuestaFormId23: number;
  encuestaFormId24: number;
  encuestaFormId25: number;
  encuestaFormId26: number;
  encuestaFormId27: number;
  encuestaFormId28: number;
  encuestaFormId29: number;
  encuestaFormId30: number;
  encuestaFormId31: number;
  encuestaFormId32: number;
  encuestaFormId33: number;
  encuestaFormId34: number;
  encuestaFormId35: number;
  encuestaFormId36: number;
  encuestaFormId37: number;
  encuestaFormId38: number;
  encuestaFormId39: number;
  encuestaFormId40: number;
  encuestaFormId41: number;
  encuestaFormId42: number;
  encuestaFormId43: number;
  encuestaFormId44: number;
  encuestaFormId45: number;
  encuestaFormId46: number;
  encuestaFormId47: number;
  encuestaFormId48: number;
  encuestaFormId49: number;
  encuestaFormId50: number;
  encuestaFormId51: number;
  encuestaFormId52: number;
  encuestaFormId53: number;
  encuestaFormId54: number;
  encuestaFormId55: number;
  encuestaFormId56: number;
  encuestaFormId57: number;
  encuestaFormId58: number;
  encuestaFormId59: number;
  encuestaFormId60: number;
  encuestaFormId61: number;
  encuestaFormId62: number;
  encuestaFormId63: number;
  encuestaFormId64: number;
  encuestaFormId65: number;
  encuestaFormId66: number;
  encuestaFormId67: number;
  encuestaFormId68: number;
  encuestaFormId69: number;
  encuestaFormId70: number;
  encuestaFormId71: number;
  encuestaFormId72: number;
  encuestaFormId73: number;
  encuestaFormId74: number;
  encuestaFormId75: number;
  encuestaFormId76: number;
  encuestaFormId77: number;
  encuestaFormId78: number;
  encuestaFormId79: number;
  encuestaFormId80: number;
  encuestaFormId81: number;
  encuestaFormId82: number;
  encuestaFormId83: number;
  encuestaFormId84: number;
  encuestaFormId85: number;
  encuestaFormId86: number;
  encuestaFormId87: number;
  encuestaFormId88: number;
  encuestaFormId89: number;
  encuestaFormId90: number;
  encuestaFormId91: number;
  encuestaFormId92: number;
  encuestaFormId93: number;
  encuestaFormId94: number;
  encuestaFormId95: number;
  encuestaFormId96: number;
  encuestaFormId97: number;
  encuestaFormId98: number;
  encuestaFormId99: number;
  encuestaFormId100: number;
  encuestaFormId101: number;
  encuestaFormId102: number;
  encuestaFormId103: number;
  encuestaFormId104: number;
  encuestaFormId105: number;
  encuestaFormId106: number;
  encuestaFormId107: number;
  encuestaFormId108: number;
  encuestaFormId109: number;
  encuestaFormId110: number;
  encuestaFormId111: number;
  encuestaFormId112: number;
  encuestaFormId113: number;
  encuestaFormId114: number;
  encuestaFormId115: number;
  encuestaFormId116: number;
  encuestaFormId117: number;
  encuestaFormId118: number;
  encuestaFormId119: number;
  encuestaFormId120: number;
  encuestaFormId121: number;
  encuestaFormId122: number;
  encuestaFormId123: number;
  encuestaFormId124: number;
  encuestaFormId125: number;
  encuestaFormId126: number;
  encuestaFormId127: number;
  encuestaFormId128: number;
  encuestaFormId129: number;
  encuestaFormId130: number;
  encuestaFormId131: number;
  encuestaFormId132: number;
  encuestaFormId133: number;
  encuestaFormId134: number;
  encuestaFormId135: number;
  encuestaFormId136: number;
  encuestaFormId137: number;
  encuestaFormId138: number;
  encuestaFormId139: number;
  encuestaFormId140: number;
  encuestaFormId141: number;
  encuestaFormId142: number;
  encuestaFormId143: number;
  encuestaFormId144: number;
  encuestaFormId145: number;
  encuestaFormId146: number;
  encuestaFormId147: number;
  encuestaFormId148: number;
  encuestaFormId149: number;
  encuestaFormId150: number;
  encuestaFormId151: number;
  encuestaFormId152: number;
  encuestaFormId153: number;
  encuestaFormId154: number;
  encuestaFormId155: number;
  encuestaFormId156: number;
  encuestaFormId157: number;
  encuestaFormId158: number;
  encuestaFormId159: number;
  encuestaFormId160: number;
  encuestaFormId161: number;
  encuestaFormId162: number;
  encuestaFormId163: number;
  encuestaFormId164: number;
  encuestaFormId165: number;
  encuestaFormId166: number;
  encuestaFormId167: number;
  encuestaFormId168: number;
  encuestaFormId169: number;
  encuestaFormId170: number;
  encuestaFormId171: number;
  encuestaFormId172: number;
  encuestaFormId173: number;
  encuestaFormId174: number;
  encuestaFormId175: number;
  encuestaFormId176: number;
  encuestaFormId177: number;
  encuestaFormId178: number;
  encuestaFormId179: number;
  encuestaFormId180: number;
  encuestaFormId181: number;
  encuestaFormId182: number;
  encuestaFormId183: number;
  encuestaFormId184: number;
  encuestaFormId185: number;
  encuestaFormId186: number;
  encuestaFormId187: number;
  encuestaFormId188: number;
  encuestaFormId189: number;
  encuestaFormId190: number;
  encuestaFormId191: number;
  encuestaFormId192: number;
  encuestaFormId193: number;
  encuestaFormId194: number;
  encuestaFormId195: number;
  encuestaFormId196: number;
  encuestaFormId197: number;
  encuestaFormId198: number;
  encuestaFormId199: number;
  encuestaFormId200: number;
  encuestaFormId201: number;
  encuestaFormId202: number;
  encuestaFormId203: number;
  encuestaFormId204: number;
  encuestaFormId205: number;
  encuestaFormId206: number;
  encuestaFormId207: number;
  encuestaFormId208: number;
  encuestaFormId209: number;
  encuestaFormId210: number;
  encuestaFormId211: number;
  encuestaFormId212: number;
  encuestaFormId213: number;
  encuestaFormId214: number;
  encuestaFormId215: number;
  encuestaFormId216: number;
  encuestaFormId217: number;
  encuestaFormId218: number;
  encuestaFormId219: number;
  encuestaFormId220: number;
  encuestaFormId221: number;
  encuestaFormId222: number;
  encuestaFormId223: number;
  encuestaFormId224: number;
  encuestaFormId225: number;
  encuestaFormId226: number;
  encuestaFormId227: number;
  encuestaFormId228: number;
  encuestaFormId229: number;
  encuestaFormId230: number;
  encuestaFormId231: number;
  encuestaFormId232: number;
  encuestaFormId233: number;
  encuestaFormId234: number;
  encuestaFormId235: number;
  encuestaFormId236: number;
  encuestaFormId237: number;
  encuestaFormId238: number;
  encuestaFormId239: number;
  encuestaFormId240: number;
  encuestaFormId241: number;
  encuestaFormId242: number;
  encuestaFormId243: number;
  encuestaFormId244: number;
  encuestaFormId245: number;
  encuestaFormId246: number;
  encuestaFormId247: number;
  encuestaFormId248: number;
  encuestaFormId249: number;
  encuestaFormId250: number;
  encuestaFormId251: number;
  encuestaFormId252: number;
  encuestaFormId253: number;
  encuestaFormId254: number;
  encuestaFormId255: number;
  encuestaFormId256: number;
  encuestaFormId257: number;
  encuestaFormId258: number;
  encuestaFormId259: number;
  encuestaFormId260: number;
  encuestaFormId261: number;
  encuestaFormId262: number;
  encuestaFormId263: number;
  encuestaFormId264: number;
  encuestaFormId265: number;
  encuestaFormId266: number;
  encuestaFormId267: number;
  encuestaFormId268: number;
  encuestaFormId269: number;
  encuestaFormId270: number;
  encuestaFormId271: number;
  encuestaFormId272: number;
  encuestaFormId273: number;
  encuestaFormId274: number;
  encuestaFormId275: number;
  encuestaFormId276: number;
  encuestaFormId277: number;
  encuestaFormId278: number;
  encuestaFormId279: number;
  encuestaFormId280: number;
  encuestaFormId281: number;
  encuestaFormId282: number;
  encuestaFormId283: number;
  encuestaFormId284: number;
  encuestaFormId285: number;
  encuestaFormId286: number;
  encuestaFormId287: number;
  encuestaFormId288: number;
  encuestaFormId289: number;
  encuestaFormId290: number;
  encuestaFormId291: number;
  encuestaFormId292: number;
  encuestaFormId293: number;
  encuestaFormId294: number;
  encuestaFormId295: number;
  encuestaFormId296: number;
  encuestaFormId297: number;
```

Ilustración 34 código del componente encuestas

[illegible]

Ilustración 35 código del componente diagnostico



TECNOLÓGICO NACIONAL DE MÉXICO

```
Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
< - -> Proyecto-Residencia

EXPLORADOR
PROYECTO-RESIDENCIA
  src
  app
  layout
  structure
  Service
  Shared
  test-list
  Ventanas
  about
  dashboard
  diagnostic
  Diagnosticos
  Afectiva
  encuesta-afc
  encuesta-beck-anxiety
  encuesta-beck-anxiety.component.html
  encuesta-beck-anxiety.component.ts
  encuesta-beck-anxiety.component.css
  encuesta-ces-d7
  encuesta-cornell
  encuesta-escala-sole
  encuesta-gai-f
  encuesta-okeeffe
  encuesta-phq9
  encuesta-sad-persons
  Cognitiva
  Funcional
  Nutricional
  Encuesta
  list-person
  login
  principal
  profile-medic
  registro
  vistas
  windows-routing
  app-routing.module.ts
  app.component.html
  app.component.css
  app.component.ts
  ESQUEMA
  LÍNEA DE TIEMPO

encuesta-beck-anxiety.component.html
1 <div class="container">
2   <h2>Inventario de Ansiedad de Beck</h2>
3   <p>
4     Lea cada enunciado e indique cuál ha sido el nivel de afectación en la última se
5   </p>
6   <form #beckForm="ngForm" (ngSubmit)="finalizarEncuesta()">
7     <table class="table table-bordered">
8       <thead>
9         <tr>
10          <th>#</th>
11          <th>Enunciado</th>
12          <th>En absoluto (0)</th>
13          <th>Levemente (1)</th>
14          <th>Moderadamente (2)</th>
15          <th>Severamente (3)</th>
16        </tr>
17      </thead>
18      <tbody>
19        <tr>
20          <td>{{ i + 1 }}</td>
21          <td>{{ pregunta.texto }}</td>
22          <td>
23            <input
24              type="radio"
25              [id]="pregunta- + i + '-0'"
26              name="pregunta-{{ i }}"
27              value="0"
28              [(ngModel)]="pregunta.respuesta"
29              required
30            />
31          </td>
32          <td>
33            <input
```

Ilustración 38 código del módulo de encuesta-Beck

```
Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
< - -> Proyecto-Residencia

EXPLORADOR
PROYECTO-RESIDENCIA
  src
  app
  layout
  structure
  Service
  Shared
  test-list
  Ventanas
  about
  dashboard
  diagnostic
  Diagnosticos
  Afectiva
  encuesta-afc
  encuesta-beck-anxiety
  encuesta-beck-anxiety.component.html
  encuesta-beck-anxiety.component.ts
  encuesta-beck-anxiety.component.css
  encuesta-ces-d7
  encuesta-cornell
  encuesta-escala-sole
  encuesta-gai-f
  encuesta-okeeffe
  encuesta-phq9
  encuesta-sad-persons
  Cognitiva
  Funcional
  Nutricional
  Encuesta
  list-person
  login
  principal
  profile-medic
  registro
  vistas
  windows-routing
  app-routing.module.ts
  app.component.html
  app.component.css
  app.component.ts
  ESQUEMA
  LÍNEA DE TIEMPO

encuesta-phq9.component.html
1 <div class="container">
2   <h2>PHQ-9</h2>
3   <form #phqForm="ngForm" (ngSubmit)="calcularPuntaje()">
4     <div #phqForm="let pregunta of preguntas; let i = Index" class="mb-3">
5       <label>{{ i + 1 }} {{ pregunta.texto }}</label>
6       <div class="form-check" #ngForm="let opcion de opciones">
7         <input
8           type="radio"
9           [id]="pregunta- + i + '-opcion-' + opcion.valor"
10          class="form-check-input"
11          name="pregunta-{{ i }}"
12          [value]="opcion.valor"
13          [(ngModel)]="pregunta.respuesta"
14          required
15        />
16        <label
17          class="form-check-label"
18          [for]="pregunta- + i + '-opcion-' + opcion.valor"
19        >{{ opcion.texto }}
20      </div>
21    </div>
22    <button type="submit" class="btn btn-primary" [disabled]="!phqForm.valid">
23      Finalizar Encuesta
24    </button>
25  </form>
26</div>

encuesta-phq9.component.ts
1 import { Component } from '@angular/core';
2 import { Router } from '@angular/router';
3
4 You, la semana pasada | 1 autor (You)
5 @Component({
6   selector: 'app-encuesta-phq9',
7   templateUrl: './encuesta-phq9.component.html',
8   styleUrls: ['./encuesta-phq9.component.scss']
9 })
10 export class EncuestaPhq9Component {
11   // Preguntas y opciones de la encuesta
12   preguntas = [
13     { texto: 'Poco interés o placer en hacer cosas', respuesta: null },
14     { texto: 'Se ha sentido decaído(a), deprimido(a) o sin esperanzas', respuesta: n
15     { texto: 'Ha tenido dificultad para quedarse o permanecer dormido(a), o ha dormi
16     { texto: 'Se ha sentido cansado(a) o con poca energía', respuesta: null },
17     { texto: 'Sin apetito o ha comido en exceso', respuesta: null },
18     { texto: 'Se ha sentido mal con usted mismo(a) o que es un fracaso o que ha qu
19     { texto: 'Ha tenido dificultad para concentrarse en ciertas actividades, tales c
20     { texto: 'Se ha movido o hablado tan lento que otras personas podrían haberlo n
21     { texto: 'Pensamientos de que estaría mejor muerto(a) o de lastimarse de alguna
22   ];
23   opciones = [
24     { texto: 'Ningún día', valor: 0 },
25     { texto: 'Varios días', valor: 1 },
26     { texto: 'Más de la mitad de los días', valor: 2 },
27     { texto: 'Casi todos los días', valor: 3 }
28   ];
29   constructor(private router: Router) {}
30
31   // Calcular el puntaje total y redirigir al componente de resultados
32   calcularPuntaje(): void {
```

Ilustración 39 código del módulo de encuesta-phq9

TECNOLÓGICO NACIONAL DE MÉXICO

```
1 <form [formGroup]="mustForm" (ngSubmit)="enviarFormulario()">
2   <h2 class="form-title">Encuesta MUST</h2>
3   <div>
4     <label for="imc">IMC (kg/m²):</label>
5     <input id="imc" formControlName="imc" type="number" />
6   </div>
7   <div>
8     <label for="perdidaPeso">Pérdida de peso (%):</label>
9     <input id="perdidaPeso" formControlName="perdidaPeso" type="number" />
10  </div>
11  <div>
12     <label for="enfermedadAguda">¿Enfermedad aguda?</label>
13     <input id="enfermedadAguda" formControlName="enfermedadAguda" type="checkbox" />
14  </div>
15  <button type="submit" [disabled]="mustForm.invalid">Enviar</button>
16 </form>
```

```
18 export class EncuestaMustComponent implements OnInit {
19   imcPuntuacion: number = 0;
20   perdidaPesoPuntuacion: number = 0;
21   enfermedadAgudaPuntuacion: number = 0;
22   puntaje: number = 0;
23   observacion: string = '';
24   constructor(private fb: FormBuilder, private router: Router) {}
25
26   ngOnInit(): void {
27     // Ya no es necesario inicializar mustForm aquí
28     this.mustForm.get('imc')?.valueChanges.subscribe(() => this.calcularPuntuaciones);
29     this.mustForm.get('perdidaPeso')?.valueChanges.subscribe(() => this.calcularPunt);
30     this.mustForm.get('enfermedadAguda')?.valueChanges.subscribe(() => this.calcular
31   )
32
33   calcularPuntuaciones(): void {
34     // Asignar puntuación para el IMC
35     const imc = this.mustForm.get('imc')?.value;
36     if (imc >= 20) {
37       this.imcPuntuacion = 0;
38     } else if (imc >= 18.5) {
39       this.imcPuntuacion = 1;
40     } else {
41       this.imcPuntuacion = 2;
42     }
43
44     // Asignar puntuación para la pérdida de peso
45     const perdidaPeso = this.mustForm.get('perdidaPeso')?.value;
46     if (perdida
47       const perdidaPeso: any
48     ) else if (perdidaPeso <= 10) {
49       this.perdidaPesoPuntuacion = 1;
50     } else {
51       this.perdidaPesoPuntuacion = 2;
52     }
53
54     // Asignar puntuación para la enfermedad aguda
55     const enfermedadAguda = this.mustForm.get('enfermedadAguda')?.value;
56     if (enfermedadAguda) {
```

Ilustración 40 Código del módulo de encuesta Must

```
1 <div class="encuesta-sarc-f">
2   <h2>Encuesta SARC-F</h2>
3   <table class="table table-bordered table-responsive">
4     <thead>
5       <tr>
6         <th>Preguntas</th>
7         <th>Puntaje</th>
8       </tr>
9     </thead>
10    <tbody>
11      <tr>
12        <td>
13          <strong>Strength (fuerza):</strong> ¿Qué tanta dificultad tiene para
14          <div>
15            <select formControlName="strength" class="form-control">
16              <option value="0">Ninguna</option>
17              <option value="1">Alguna</option>
18              <option value="2">Mucha o incapaz</option>
19            </select>
20          </div>
21        </td>
22        <td>
23          <strong>Assistance in walking (Asistencia para caminar):</strong> ¿Q
24          <div>
25            <select formControlName="walking" class="form-control">
26              <option value="0">Ninguna</option>
27              <option value="1">Alguna</option>
28              <option value="2">Mucha, usando auxiliares o incapaz</option>
29            </select>
30          </div>
31        </td>
32        <td>
33          <strong>Rise from chair (Levantarse de una silla):</strong> ¿Qué tan
34          <div>
35            <select formControlName="chair" class="form-control">
36              <option value="0">Ninguna</option>
37              <option value="1">Alguna</option>
38              <option value="2">Mucha o incapaz sin ayudas</option>
39            </select>
40          </div>
41        </td>
42      </tr>
43    </tbody>
44  </table>
```

```
1 import { Component, OnInit } from '@angular/core';
2 import { FormBuilder, FormGroup, Validators } from '@angular/forms';
3 import { Router } from '@angular/router';
4
5 @Component({
6   selector: 'app-encuesta-sarc-f',
7   templateUrl: './encuesta-sarc-f.component.html',
8   styleUrls: ['./encuesta-sarc-f.component.scss']
9 })
10 export class EncuestaSarcComponent implements OnInit {
11   sarcForm: FormGroup;
12   puntuacionTotal: number = 0;
13
14   constructor(private fb: FormBuilder, private router: Router) {
15     this.sarcForm = this.fb.group({
16       strength: [0, Validators.required],
17       walking: [0, Validators.required],
18       chair: [0, Validators.required],
19       stairs: [0, Validators.required],
20       falls: [0, Validators.required],
21     });
22   }
23
24   ngOnInit(): void {}
25
26   calcularPuntuacion() {
27     const values = this.sarcForm.value;
28     this.puntuacionTotal =
29       +values.strength +
30       +values.walking +
31       +values.chair +
32       +values.stairs +
33       +values.falls;
34   }
35
36   enviarResultados() {
37     if (this.sarcForm.invalid) {
38       alert('Completa todas las preguntas antes de enviar.');
```

Ilustración 41 Código del módulo de encuesta sarc-f



TECNOLÓGICO NACIONAL DE MÉXICO

```
src > app > Ventanas > Diagnósticos > Funcional > category1.component.html > ...  
1 <!-- Encuesta de KATZ -->  
2 <div class="katz containerEncuesta opcult" *ngIf="encuestaSelect == 'katz'" ...  
3 </div>  
4  
5 <!-- Encuesta de Barthel -->  
6 <div class="barthel containerEncuesta opcult" *ngIf="encuestaSelect == 'barthel'" ...  
7 </div>  
8  
9 <!-- Encuesta de Lawton y Brody -->  
10 <div class="lawton containerEncuesta opcult" *ngIf="encuestaSelect == 'lawton'" ...  
11 </div>  
12  
13 <!-- Encuesta de Ensrud -->  
14 <div class="ensrud containerEncuesta" *ngIf="encuestaSelect == 'ensrud'" ...  
15 </div>  
16  
17 <!-- Encuesta FRAIL -->  
18 <div class="frail containerEncuesta" *ngIf="encuestaSelect == 'frail'" ...  
19 </div>  
20  
21 <!-- Time Up and Go -->  
22 <div class="cronometro containerEncuesta" *ngIf="encuestaSelect == 'timeup'" ...  
23 </div>  
24  
25 <div class="boton">  
26 <button type="submit" class="btn btn-success" (click)="finalizarEncuesta()">Finali...  
27 </div>  
28  
category1.component.ts  
1 import { Component, ElementRef, Input, OnInit, Renderer2, ViewChild } from '@angular/  
2 import { FormGroup } from '@angular/forms';  
3 import { ResultadoService } from 'src/app/service/resultado.service';  
4 import { FormBuilder } from '@angular/forms';  
5 import { FormControl } from '@angular/forms';  
6 import { ActivatedRoute, Router } from '@angular/router';  
7 import { DiagnosticComponent } from 'src/app/diagnostic/diagnostic.component';  
8 import { ApiService } from 'src/app/service/api.service';  
9 import { Checkbox, Question } from 'src/app/shared/data';  
10 import { float, FLOAT } from 'html2canvas/dist/types/css/property-descriptors/float';  
11  
12 RODF10 hace 2 semanas | 1 autor (RODF10)  
13 @Component({  
14 selector: 'app-category1',  
15 templateUrl: './category1.component.html',  
16 styleUrls: ['./category1.component.scss'],  
17 })  
18 export class Category1Component implements OnInit {  
19 encuestaSelect: string = '';  
20 puntos: number = 0;  
21 observacion: string = '';  
22 showErrors: boolean = false;  
23 categoria: string;  
24 mensajeError: string = '';  
25 ent: string[] = ['Entrada de ', 'Salida de '];  
26  
27 // Propiedades para los valores de peso  
28 peso1: FLOAT = 0.0; // Year  
29 peso2: FLOAT = 0.0; // Actual  
30 pesoYear: string = '';  
31 pesoActual: string = '';  
32 // Mensaje de error  
33 pesoActualMinimoError: string = '';  
34 pesoActualError: string = '';  
35 // Validación de Numero  
36 pesoIn: boolean = false; pesoIn2: boolean = false;  
37  
38 // Preguntas y Opciones de BARTHEL  
39 preguntasBarthel: Question[] = [  
40 { text: 'Alimentación',  
41 options: [  
42 { text: 'Existencia de Cambio agudo del Estado Mental', respuesta: '' },  
43 { text: 'Estado Mental del Paciente ha fluctuado durante las últimas 24hrs', res  
44 { text: 'Apretar Mano (Médico) cuando el paciente diga A', respuesta: '' },  
45 { text: 'Diga al Paciente que Deletreee C-A-S-A-B-L-A-N-C-A', respuesta: '' },  
46 { text: 'Vigile al paciente y verifique Alerta de RAS58', respuesta: '' },  
47 { text: 'Las Piedras flotan en el Agua', respuesta: '' },  
48 { text: 'Hay Peces en el mar', respuesta: '' },  
49 { text: '¿kg pesa más que 2kg', respuesta: '' },  
50 { text: 'Los martillos sirven para poner clavos', respuesta: '' }  
51 ]  
52 }  
53 preguntasWSE: PreguntaDosOpc[] = [  
54 { text: 'Que fecha es hoy (dia, mes, año)', respuesta: '' },  
55 { text: 'Que día de la semana es hoy', respuesta: '' },  
56 { text: 'Donde estamos ahora (Lugar o Edificio)', respuesta: '' },  
57 { text: 'Cual es su numero de telefono (o dirección si no tiene uno)', respuesta  
58 { text: 'Que edad tiene', respuesta: '' },  
59 { text: 'La fecha cuando nació', respuesta: '' },  
60 { text: 'Como se llama el presidente del Gobierno', respuesta: '' },  
61 { text: 'Como se llama el anterior presidente', respuesta: '' },  
62 { text: 'Primer apellido de su Madre', respuesta: '' },  
63 { text: 'Pudo restar de tres en tres desde veite', respuesta: '' }  
64 ]  
65 }  
66 //Preguntas Reloj  
67 questionnaireForm: FormGroup;  
68 questions: Question[] = [  
69 { text: 'Dibujo del Reloj',  
70 options: [  
71 { label: 'El círculo está bien dibujado o cercanamente a lo redondo', score:  
72 { label: 'El círculo está dibujado, pero de manera irregular', score: 1 },  
73 { label: 'Figura irregular o no compatible con un semi-círculo', score: 0 }  
74 ]  
75 },  
76 { text: 'Presencia y secuencia de números',  
77 options: [  
78 { label: 'Todos los números están presentes. Puede aceptarse un error mínimo  
79 { label: 'Todos los números están presentes. Errores en la disposición espac  
80 { label: 'Algunas de las siguientes:\nNúmeros faltantes o adicionales aunque  
81 { label: 'Números faltantes o adicionales y errores espaciales serios', score
```

Ilustración 42 código del módulo funcionalidad el cual integra 6 encuestas (barthel, lawton y brody, ensrud, frail, time up and go, kartz)

```
src > app > Ventanas > Diagnósticos > Cognitiva > encuesta-cog > encuesta-cog.component.html > ...  
1 <!-- Encuesta 4AT -->  
2 <div class="4at containerEncuesta" *ngIf="categoria == '4at'" ...  
3 </div>  
4  
5 <!-- Encuesta CAM -->  
6 <div class="cam containerEncuesta" *ngIf="categoria == 'cam'" ...  
7 </div>  
8  
9 <!-- Encuesta CAM-ICU -->  
10 <div class="camicu containerEncuesta" *ngIf="categoria == 'camicu'" ...  
11 </div>  
12  
13 <!-- Encuesta AWOL -->  
14 <div class="awol containerEncuesta" *ngIf="categoria == 'awol'" ...  
15 </div>  
16  
17 <!-- Encuesta Question Pfeiffer -->  
18 <div class="spmsqp containerEncuesta" *ngIf="categoria == 'spmsqp'" ...  
19 </div>  
20  
21 <!-- Prueba del Reloj -->  
22 <div class="reloj containerEncuesta" *ngIf="categoria == 'reloj'" ...  
23 </div>  
24  
25 <div class="boton">  
26 <button type="submit" class="btn btn-success" (click)="finalizarEncuesta()">Finali...  
27 </div>  
28  
encuesta-cog.component.ts  
1 export class EncuestaCogComponent implements OnInit {  
2 //Preguntas Si No  
3 preguntasCamCiu: PreguntaDosOpc[] = [  
4 { text: 'Existencia de Cambio agudo del Estado Mental', respuesta: '' },  
5 { text: 'Estado Mental del Paciente ha fluctuado durante las últimas 24hrs', res  
6 { text: 'Apretar Mano (Médico) cuando el paciente diga A', respuesta: '' },  
7 { text: 'Diga al Paciente que Deletreee C-A-S-A-B-L-A-N-C-A', respuesta: '' },  
8 { text: 'Vigile al paciente y verifique Alerta de RAS58', respuesta: '' },  
9 { text: 'Las Piedras flotan en el Agua', respuesta: '' },  
10 { text: 'Hay Peces en el mar', respuesta: '' },  
11 { text: '¿kg pesa más que 2kg', respuesta: '' },  
12 { text: 'Los martillos sirven para poner clavos', respuesta: '' }  
13 ]  
14 preguntasWSE: PreguntaDosOpc[] = [  
15 { text: 'Que fecha es hoy (dia, mes, año)', respuesta: '' },  
16 { text: 'Que día de la semana es hoy', respuesta: '' },  
17 { text: 'Donde estamos ahora (Lugar o Edificio)', respuesta: '' },  
18 { text: 'Cual es su numero de telefono (o dirección si no tiene uno)', respuesta  
19 { text: 'Que edad tiene', respuesta: '' },  
20 { text: 'La fecha cuando nació', respuesta: '' },  
21 { text: 'Como se llama el presidente del Gobierno', respuesta: '' },  
22 { text: 'Como se llama el anterior presidente', respuesta: '' },  
23 { text: 'Primer apellido de su Madre', respuesta: '' },  
24 { text: 'Pudo restar de tres en tres desde veite', respuesta: '' }  
25 ]  
26 //Preguntas Reloj  
27 questionnaireForm: FormGroup;  
28 questions: Question[] = [  
29 { text: 'Dibujo del Reloj',  
30 options: [  
31 { label: 'El círculo está bien dibujado o cercanamente a lo redondo', score:  
32 { label: 'El círculo está dibujado, pero de manera irregular', score: 1 },  
33 { label: 'Figura irregular o no compatible con un semi-círculo', score: 0 }  
34 ]  
35 },  
36 { text: 'Presencia y secuencia de números',  
37 options: [  
38 { label: 'Todos los números están presentes. Puede aceptarse un error mínimo  
39 { label: 'Todos los números están presentes. Errores en la disposición espac  
40 { label: 'Algunas de las siguientes:\nNúmeros faltantes o adicionales aunque  
41 { label: 'Números faltantes o adicionales y errores espaciales serios', score
```

Ilustración 43 Módulo Cognitiva el cual integra 6 encuestas (4AT, CAM, CAM-ICU, AWOL, QUESTION PFEIFFER, PRUEBA DE RELOJ)



TECNOLÓGICO NACIONAL DE MÉXICO

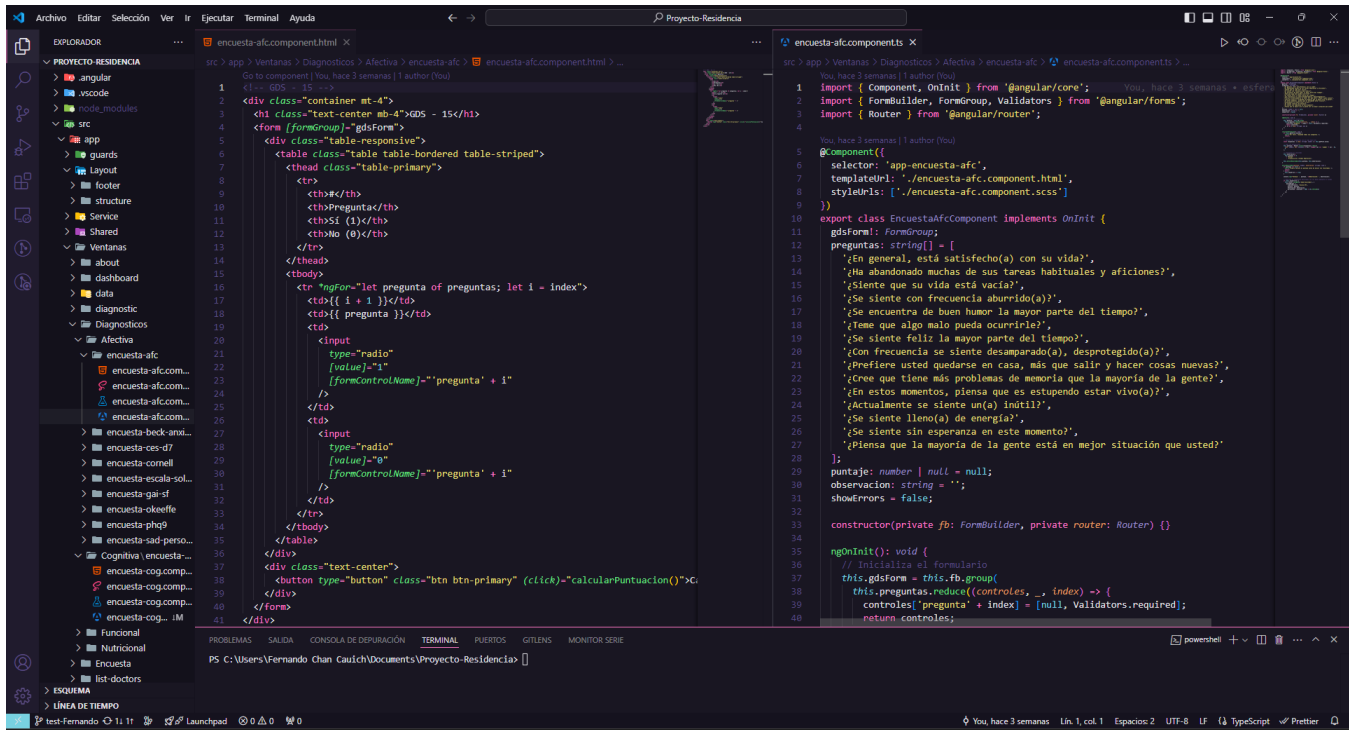


Ilustración 44 Modulo encuesta gds-15

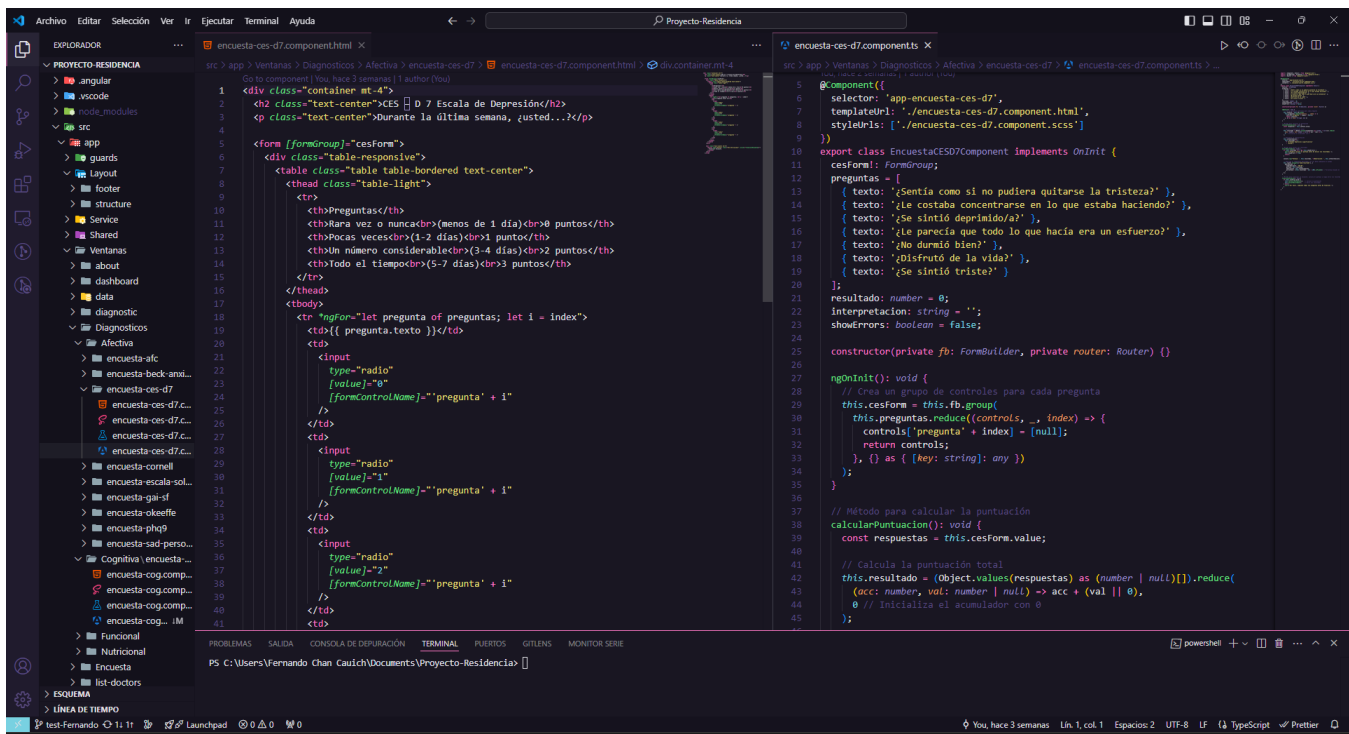


Ilustración 45 Modulo encuesta ces-d7



TECNOLÓGICO NACIONAL DE MÉXICO

```
src > app > Ventanas > Diagnósticos > Afectiva > encuesta-cornell > encuesta-cornell.component.html > div.cornell-scale
1 <div class="cornell-scale">
2   <h2>Escala de Cornell</h2>
3   <p>Manifestaciones.</p>
4
5   <div *ngFor="let question of questions; let i = index" class="question">
6     <label>{{ i + 1 }}. {{ question.text }}</label>
7     <div class="options">
8       <input type="radio" name="q{{ i }}" /> No Valorable
9       <input type="radio" name="q{{ i }}" (change)="setScore(i, 0)" /> Ausente
10      <input type="radio" name="q{{ i }}" (change)="setScore(i, 1)" /> Leve
11      <input type="radio" name="q{{ i }}" (change)="setScore(i, 2)" /> Grave
12    </div>
13  </div>
14
15  <div class="buttons">
16    <button class="btn btn-primary" (click)="calculateScore()">Calcular Puntaje</button>
17    <button class="btn btn-success" [disabled]="resultado === 0" (click)="enviarResultados()">
18      Enviar Resultados
19    </button>
20  </div>
21</div>
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
```

TECNOLÓGICO NACIONAL DE MÉXICO

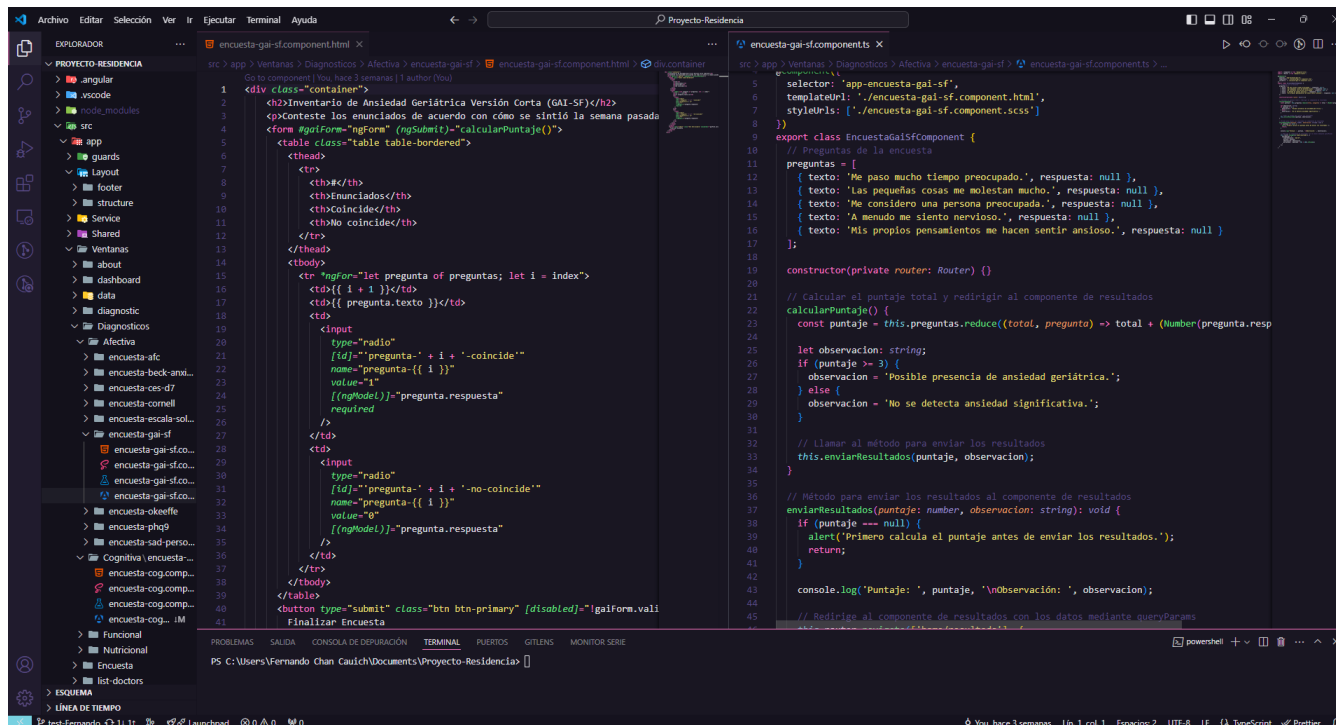


Ilustración 48 Modulo Encuesta gai-sf

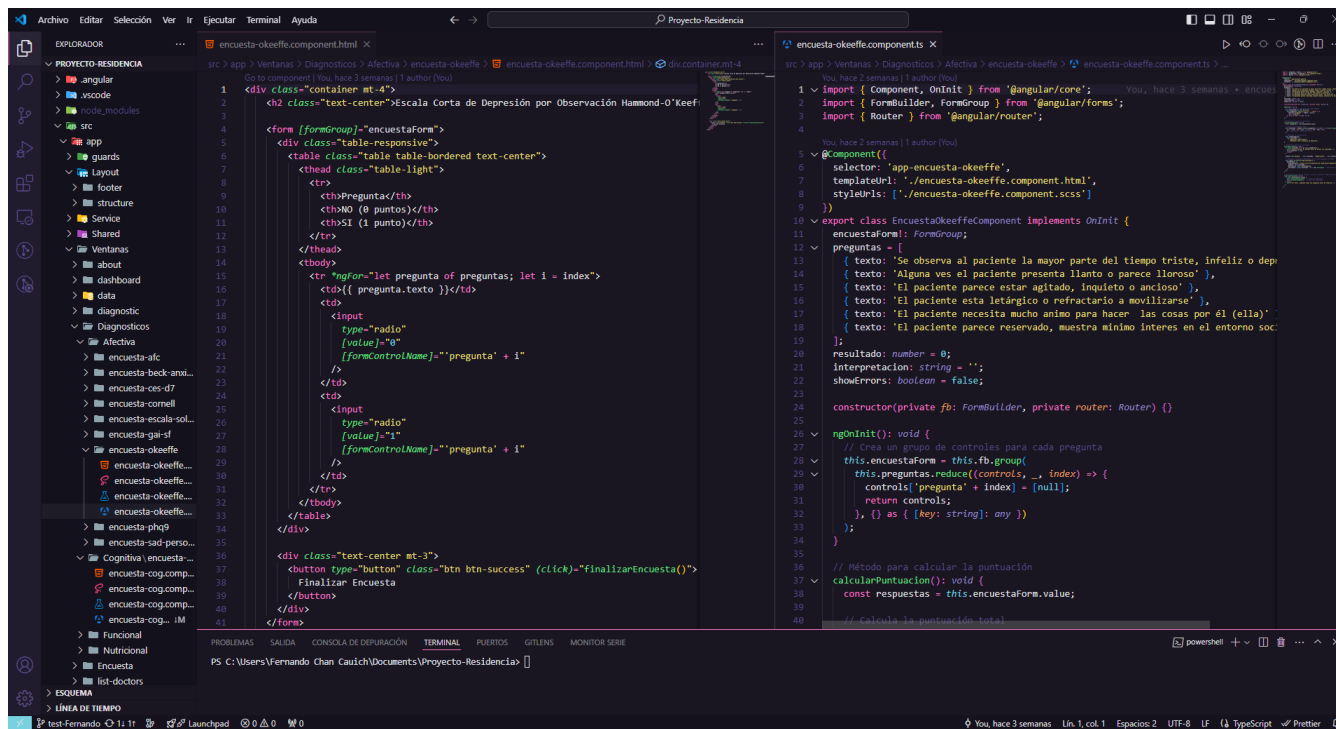


Ilustración 49 Modulo Encuesta okeeffe

TECNOLÓGICO NACIONAL DE MÉXICO

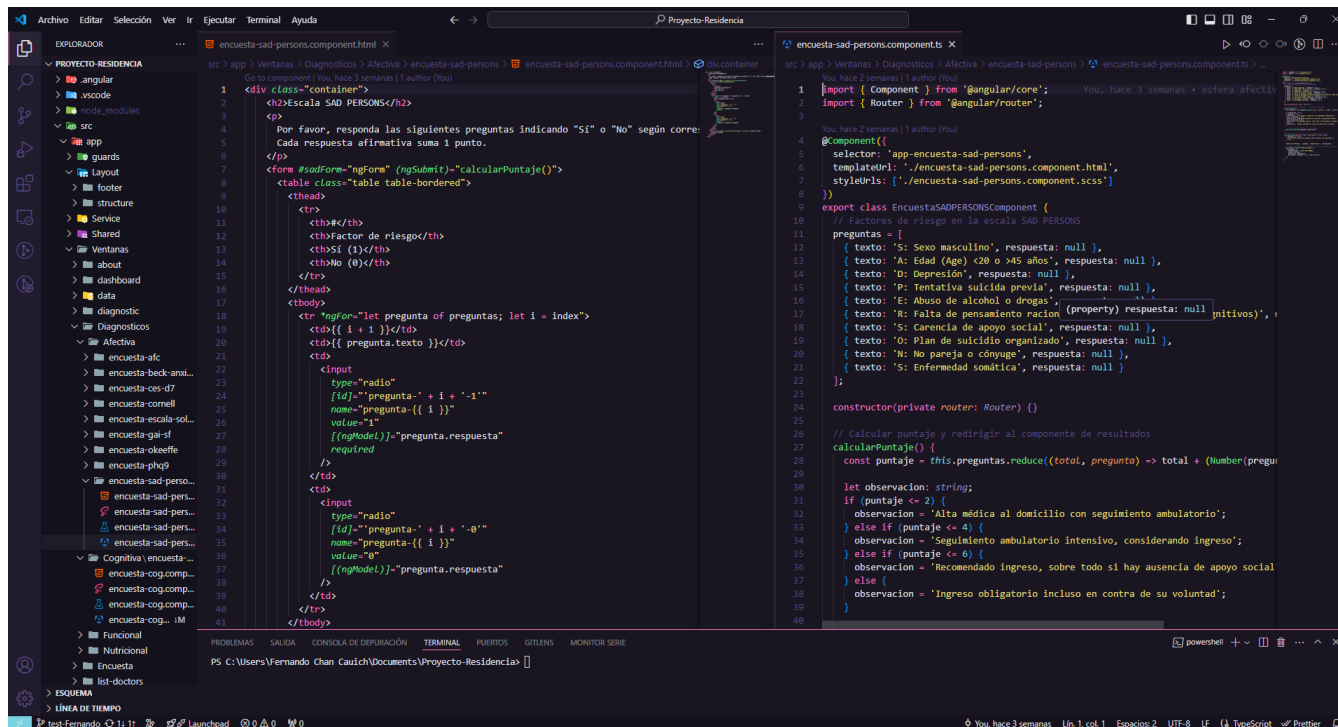


Ilustración 50 Modulo Encuesta SADPERSONS

Los Temerarios Viejas Rom...

VGITest

localhost:4200/home/encuesta/esfera%20afectiva-gds-15.php

VGITest

Home

Pacientes

Encuesta

Acerca de

Encuesta AFECTIVA

GDS - 15

#	Pregunta	SI (1)	NO (0)
1	¿En general, está satisfecho(a) con su vida?	☐	☐
2	¿Ha abandonado muchas de sus tareas habituales y aficiones?	☐	☐
3	¿Siente que su vida está vacía?	☐	☐
4	¿Se siente con frecuencia aburrido(a)?	☐	☐
5	¿Se encuentra de buen humor la mayor parte del tiempo?	☐	☐
6	¿Teme que algo malo pueda ocurrirle?	☐	☐
7	¿Se siente feliz la mayor parte del tiempo?	☐	☐
8	¿Con frecuencia se siente desamparado(a), desprotegido(a)?	☐	☐
9	¿Prefiere usted quedarse en casa, más que salir y hacer cosas nuevas?	☐	☐
10	¿Cree que tiene más problemas de memoria que la mayoría de la gente?	☐	☐
11	¿En estos momentos, piensa que es estupendo estar vivo(a)?	☐	☐
12	¿Actualmente se siente un(a) inútil?	☐	☐
13	¿Se siente lleno(a) de energía?	☐	☐
14	¿Se siente sin esperanza en este momento?	☐	☐
15	¿Piensa que la mayoría de la gente está en mejor situación que usted?	☐	☐

Calcular y Ver Resultados

Ilustración 51 Vista Encuesta GDS-15

TECNOLÓGICO NACIONAL DE MÉXICO

Los Temerarios Vigías Rom... x VGITest x +

localhost:4200/home/encuesta/esfera%20afectiva-ces-d7-depresion-epidemioloicos.php

VGI

Home Pacientes Encuesta Acerca de

Encuesta AFECTIVA

CES – D 7 Escala de Depresión

Durante la última semana, ¿usted...?

Preguntas	Rara vez o nunca (menos de 1 día) 0 puntos	Pocas veces (1-2 días) 1 punto	Un número considerable (3-4 días) 2 puntos	Todo el tiempo (5-7 días) 3 puntos
¿Sentía como si no pudiera quitarse la tristeza?	☐	☐	☐	☐
¿Le costaba concentrarse en lo que estaba haciendo?	☐	☐	☐	☐
¿Se sintió deprimido/a?	☐	☐	☐	☐
¿Le parecía que todo lo que hacía era un esfuerzo?	☐	☐	☐	☐
¿No durmió bien?	☐	☐	☐	☐
¿Disfrutó de la vida?	☐	☐	☐	☐
¿Se sintió triste?	☐	☐	☐	☐

Finalizar Encuesta

Ilustración 52 Vista Encuesta CES-D7

Los Temerarios Vigías Rom... x VGITest x +

localhost:4200/home/encuesta/esfera%20afectiva-phq9.php

VGI

Home Pacientes Encuesta Acerca de

Encuesta AFECTIVA

PHQ-9

- Poco interés o placer en hacer cosas
 - ☐ Ningún día
 - ☐ Varios días
 - ☐ Más de la mitad de los días
 - ☐ Casi todos los días
- Se ha sentido decaído(a), deprimido(a) o sin esperanzas
 - ☐ Ningún día
 - ☐ Varios días
 - ☐ Más de la mitad de los días
 - ☐ Casi todos los días
- Ha tenido dificultad para quedarse o permanecer dormido(a), o ha dormido demasiado
 - ☐ Ningún día
 - ☐ Varios días
 - ☐ Más de la mitad de los días
 - ☐ Casi todos los días
- Se ha sentido cansado(a) o con poca energía
 - ☐ Ningún día
 - ☐ Varios días
 - ☐ Más de la mitad de los días
 - ☐ Casi todos los días
- Sin apetito o ha comido en exceso
 - ☐ Ningún día

Ilustración 53 Vista Encuesta PHQ-9



TECNOLÓGICO NACIONAL DE MÉXICO

Encuesta AFECTIVA

Inventario de Ansiedad Geriátrica Versión Corta (GAI-SF)

Conteste los enunciados de acuerdo con cómo se sintió la semana pasada.

#	Enunciados	Coincide	No coincide
1	Me paso mucho tiempo preocupado.	☐	☐
2	Las pequeñas cosas me molestan mucho.	☐	☐
3	Me considero una persona preocupada.	☐	☐
4	A menudo me siento nervioso.	☐	☐
5	Mis propios pensamientos me hacen sentir ansioso.	☐	☐

Finalizar Encuesta

Ilustración 54 Vista Encuesta GAI-SF

Encuesta AFECTIVA

Inventario de Ansiedad de Beck

Lea cada enunciado e indique cuál ha sido el nivel de afectación en la última semana, incluyendo el día de hoy.

#	Enunciado	En absoluto (0)	Levemente (1)	Moderadamente (2)	Severamente (3)
1	Torpe o entumecido.	☐	☐	☐	☐
2	Acalorado.	☐	☐	☐	☐
3	Con temblor en las piernas.	☐	☐	☐	☐
4	Incapaz de relajarse.	☐	☐	☐	☐
5	Con temor a que ocurra lo peor.	☐	☐	☐	☐
6	Mareado, o que se le va la cabeza.	☐	☐	☐	☐
7	Con latidos fuertes del corazón y acelerados.	☐	☐	☐	☐
8	Inestable.	☐	☐	☐	☐
9	Atemorizado o asustado.	☐	☐	☐	☐
10	Nervioso.	☐	☐	☐	☐
11	Con sensación de bloqueo.	☐	☐	☐	☐
12	Con temblores en las manos.	☐	☐	☐	☐
13	Inquieto, inseguro.	☐	☐	☐	☐
14	Con miedo a perder el control.	☐	☐	☐	☐
15	Con sensación de ahogo.	☐	☐	☐	☐

Ilustración 55 Vista Encuesta Ansiedad de Beck

TECNOLÓGICO NACIONAL DE MÉXICO

Los Temerarios Vigías Rom... x VGITest x +

localhost:4200/home/encuesta/esfera%20afectiva-riesgo-suicidio-sad-person.php

VGI

- Home
- Pacientes
- Encuesta
- Acerca de

Encuesta AFECTIVA

Escala SAD PERSONS

Por favor, responda las siguientes preguntas indicando "Sí" o "No" según corresponda. Cada respuesta afirmativa suma 1 punto.

#	Factor de riesgo	Sí (1)	No (0)
1	S: Sexo masculino	☐	☐
2	A: Edad (Age) <20 o >45 años	☐	☐
3	D: Depresión	☐	☐
4	P: Tentativa suicida previa	☐	☐
5	E: Abuso de alcohol o drogas	☐	☐
6	R: Falta de pensamiento racional (psicosis o trastornos cognitivos)	☐	☐
7	S: Carencia de apoyo social	☐	☐
8	O: Plan de suicidio organizado	☐	☐
9	N: No pareja o cónyuge	☐	☐
10	S: Enfermedad somática	☐	☐

Finalizar Encuesta

Ilustración 56 Vista Encuesta SADPERSONS

Los Temerarios Vigías Rom... x VGITest x +

localhost:4200/home/encuesta/esfera%20afectiva-escala-cornell.php

VGI

- Home
- Pacientes
- Encuesta
- Acerca de

Encuesta AFECTIVA

Escala de Cornell

Manifestaciones.

- Ansiedad (Expresión ansiosa, rumiación de ideas, preocupación)
☐ No Valorable ☐ Ausente ☐ Leve ☐ Grave
- Tristeza (Expresión triste, voz apagada, tendencia al llanto)
☐ No Valorable ☐ Ausente ☐ Leve ☐ Grave
- Falta de reactividad a los acontecimientos placenteros
☐ No Valorable ☐ Ausente ☐ Leve ☐ Grave
- Irritabilidad (facilmente enojable, poco temperamento)
☐ No Valorable ☐ Ausente ☐ Leve ☐ Grave
- Agitación (incapacidad de permanecer quieto, jugar con las manos, tirarse el pelo)
☐ No Valorable ☐ Ausente ☐ Leve ☐ Grave
- Entretencimiento (Movimientos, habla y reacciones entretencidos)
☐ No Valorable ☐ Ausente ☐ Leve ☐ Grave
- Quejas Físicas múltiples (puntue 0 si son únicamente gastrointestinales)
☐ No Valorable ☐ Ausente ☐ Leve ☐ Grave
- perdida del interés (menos envuelto en las actividades habituales, puntue solamente si el cambio

Ilustración 57 Vista Encuesta CORNELL

TECNOLÓGICO NACIONAL DE MÉXICO

Los Iterarios Vigías Rom... x VGflect x +

localhost:4200/home/encuesta/esfera%20afectiva-depression-corta-o'keeffe.php

VGI

Home Pacientes Encuesta Acerca de

Encuesta AFECTIVA

Escala Corta de Depresión por Observación Hammond-O'Keeffe

Pregunta	NO (0 puntos)	SI (1 punto)
Se observa al paciente la mayor parte del tiempo triste, infeliz o deprimido	☐	☐
Alguna vez el paciente presenta llanto o parece lloroso	☐	☐
El paciente parece estar agitado, inquieto o ansioso	☐	☐
El paciente esta letárgico o refractario a movilizarse	☐	☐
El paciente necesita mucho animo para hacer las cosas por él (ella)	☐	☐
El paciente parece reservado, muestra mínimo interes en el entorno social	☐	☐

Finalizar Encuesta

Ilustración 58 Vista Encuesta O'KEEFFE

Conjunto Primavera Sus M... x VGflect x +

localhost:4200/home/encuesta/esfera%20funcional-indice-katz.php

VGI

Home Pacientes Encuesta Acerca de

Encuesta FUNCIONAL

Indice de KATZ

BAÑO (Esponja, regadera o tina)

☐ No recibe asistencia (puede entrar y salir de la tina u otra forma de baño).

☐ Que reciba asistencia durante el baño en una sola parte del cuerpo

☐ Dependiente: Que reciba asistencia durante el baño en más de una parte.

VESTIDO

☐ Que pueda tomar las prendas y vestirse completamente, sin asistencia.

☐ Que pueda tomar las prendas y vestirse sin asistencia excepto en abrocharse los zapatos.

☐ Que reciba asistencia para tomar las prendas y vestirse.

USO DEL SANITARIO

☐ Sin ninguna asistencia (puede utilizar objeto de soporte: bastón o silla de ruedas y/o puede arreglar su ropa o el uso de pañal o cómodo).

☐ Que reciba asistencia al ir al baño, en limpiarse y que pueda manejar por si mismo/a el pañal o cómodo vaciándolo.

☐ Que no vaya al baño por si mismo/a.

TRANSFERENCIAS

☐ Que se mueva dentro y fuera de la cama y silla sin ninguna asistencia (utiliza un auxiliar de la marcha u objeto de soporte).

☐ Que pueda moverse dentro y fuera de la cama y silla con asistencia.

☐ Que no pueda salir de la cama.

CONTINENCIA

☐ Control total de esfinteres.

☐ Que tenga accidentes ocasionales que no afectan su vida social

Ilustración 59 Vista Encuesta KATZ

TECNOLÓGICO NACIONAL DE MÉXICO

Conjunto Primavera SaaS M... VGiftest

localhost:4200/home/encuesta/esfera%20funcional-indice-barthel.php

VGI

Home

Pacientes

Encuesta

Acercar de

Encuesta FUNCIONAL

Indice de Barthel

Seleccione

Alimentación

☐ Independiente: Capaz de utilizar cualquier instrumento, come en tiempo, puede ser servida o cocinada por otra persona

☐ Ayuda: Necesita ayuda pero capaz de comer solo

☐ Dependiente: Depende de otra persona para comer

Bañarse/Ducharse

☐ Independiente: Entra y sale solo del baño, sin supervisión

☐ Dependiente: Necesita ayuda o supervisión

Vestirse

☐ Independiente: Capaz de ponerse y de quitarse la ropa, abotonarse, atarse los zapatos

☐ Ayuda: Necesita de personal, pero al menos puede realizar las tareas con tiempo razonable

☐ Dependiente: Necesita ayuda para algunas actividades

Aseo Personal

☐ Independiente: Realiza todas las actividades sin personal

☐ Dependiente: Requiere de personal para asearse en algunas actividades

Control de Orina

☐ Continente: No presenta inconsistencia, capaz de atender solo su cuidado

☐ Incontinencia Ocasional: Como máximo incontinencia en 24hrs, requiere ayuda para el cuidado

☐ Incontinencia: Episodios de incontinencia con frecuencia una vez en 24hrs, incapaz de manejar su cuidado

Ilustración 60 Vista Encuesta Barthel

Conjunto Primavera SaaS M... VGiftest

localhost:4200/home/encuesta/esfera%20funcional-indice-lawton-y-brody.php

VGI

Home

Pacientes

Encuesta

Acercar de

Encuesta FUNCIONAL

Lawton y Brody

CAPACIDAD PARA USAR TELÉFONO

☐ Lo opera por iniciativa propia, lo marca sin problemas.

☐ Marca sólo unos cuantos números bien conocidos.

☐ Contesta el teléfono pero no llama.

☐ No usa el teléfono.

COMPRAS

☐ Vigila sus necesidades independientemente.

☐ Hace independientemente sólo pequeñas compras.

☐ Necesita compañía para cualquier compra.

☐ Incapaz de cualquier compra.

COCINA

☐ Planea, prepara y sirve los alimentos correctamente.

☐ Prepara los alimentos sólo si se le provee lo necesario.

☐ Calienta, sirve y prepara pero no lleva una dieta adecuada.

☐ Necesita que le preparen los alimentos.

CUIDADO DEL HOGAR

☐ Mantiene la casa solo o con ayuda mínima.

☐ Efectúa diariamente trabajo ligero eficientemente.

Ilustración 61 Vista Encuesta Lawton y Brody

TECNOLÓGICO NACIONAL DE MÉXICO

Home

Pacientes

Encuesta

Acerca de

Encuesta FUNCIONAL

FRAIL

OBSERVACIÓN	PUNTUACIÓN
[Fatigue (fatiga)] En las últimas 4 semanas, ¿Qué tanto tiempo se sintió?	Todo el tiempo La mayor parte del tiempo Algo de tiempo Muy poco tiempo Nada de tiempo
[Resistance (resistencia)] ¿Tiene dificultad para subir 10 escalones (una escalera)?	Sí No
[Aerobic (actividad aeróbica)] ¿Tiene dificultad para caminar 100 metros (dos cuadras) sin descansar?	Sí No
[Illnesses (enfermedades)] ¿Algún doctor o médico le ha comentado que tiene [mencionar la enfermedad]?	Sí No
[Lost of weight (pérdida de peso)] El paciente no debe portar los zapatos al momento de ser pesado	¿Cuánto pesaba hace un año? Peso hace un año ¿Cuánto pesa ahora? Peso actual

Finalizar

Ilustración 62 Vista Encuesta FRAIL

Home

Pacientes

Encuesta

Acerca de

Encuesta FUNCIONAL

Time UP and DOWN

La persona puede usar su calzado habitual y puede utilizar cualquier dispositivo de ayuda que normalmente usa.

1. Indicarle a la persona mayor, sentarse en la silla con la espalda apoyada en el respaldo.
2. Pídale a la persona que se levante de la silla, camine a paso normal una distancia de 3 metros, haga que la persona de la vuelta camine nuevamente hacia la silla y se vuelva a sentar.
3. Mida el tiempo en que la persona mayor realiza la prueba. El cronometraje comienza cuando la persona comienza a levantarse de la silla y termina cuando regresa a la silla y se sienta.
4. Dar un intento de prueba.

El cronometraje comienza cuando la persona comienza a levantarse de la silla y termina cuando regresa a la silla y se sienta.

La persona debe de dar un intento de práctica y luego repite 3 intentos más. Se promedian los tres ensayos reales y se promedian.

00:00.000

Start

Reset

Time Capture

Default



Finalizar

Ilustración 63 Vista Encuesta TIME UP AND GO

TECNOLÓGICO NACIONAL DE MÉXICO

Laravel (código).

Authcontroller.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Doctors;
use App\Models\Patient;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Validator;

class AuthController extends Controller
{
    // Método para autenticar al doctor
    public function loginDoctor(Request $request)
    {
        $request->validate([
            'email' => 'required|email',
            'password' => 'required|string',
        ]);
        // Buscar el usuario por correo electrónico
        $user = Doctors::where('email', $request->email)->first();

        // Verificar si el usuario existe y si la contraseña coincide con el hash
        if ($user && $request->password == $user->password) {
            return response()->json([
                'success' => true,
                'message' => 'Inicio de sesión exitoso',
                'user' => [
                    'id' => $user->id,
                    'nombre' => $user->nombre, // Enviar el nombre del doctor
                    'apellido' => $user->apellido,
                    'email' => $user->email // Enviar el email si es necesario
                ]
            ], 200);
        }

        return response()->json(['message' => 'Correo o contraseña incorrectos'], 401);
    }
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
// Método para obtener pacientes relacionados con un doctor
public function getDoctorPatients($doctorId)
{
    // Verificar si el doctor existe
    $doctor = Doctors::find($doctorId);

    if (!$doctor) {
        return response()->json(['message' => 'Doctor no encontrado'], 404);
    }

    // Obtener los pacientes relacionados
    $patients = Patient::where('doctor_id', $doctorId)->get();

    return response()->json([
        'success' => true,
        'doctor' => [
            'id' => $doctor->id,
            'nombre' => $doctor->nombre,
            'apellido' => $doctor->apellido,
            'email' => $doctor->email,
        ],
        'patients' => $patients
    ], 200);
}

// Método para cambiar la contraseña
public function changePassword(Request $request)
{
    // Validar los datos de entrada
    $validator = Validator::make($request->all(), [
        'correo' => 'required|email|exists:doctors,email',
        'password' => 'required|string|min:8',
    ]);

    if ($validator->fails()) {
        return response()->json([
            'success' => false,
            'errors' => $validator->errors()
        ], 422);
    }
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
// Encontrar al doctor por correo
$doctor = Doctors::where('email', $request->correo)->first();

// Actualizar la contraseña encriptada
$doctor->password = $request->password;
$doctor->save();

return response()->json([
    'success' => true,
    'message' => 'Contraseña actualizada con éxito.'
], 200);
}
```

DoctorsController.php

```
<?php

namespace App\Http\Controllers;

use App\Models\Doctors;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Hash;

class DoctorsController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        return response()->json(Doctors::all(), 200);
    }

    /**
     * Show the form for creating a new resource.
     */
    public function create()
    {
        // NO SE USA
    }
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
/**
 * Store a newly created resource in storage.
 */
public function store(Request $request)
{
    // Validar los datos de entrada
    $validatedData = $request->validate([
        'nombre' => 'required|string|max:100',
        'apellido' => 'required|string|max:100',
        'cedula' => 'required|string|unique:doctors',
        'profesion' => 'required|string|max:35',
        'date' => 'required|date',
        'edad' => 'required|integer',
        'genero' => 'required|string|max:10',
        'email' => 'required|email|unique:doctors',
        'password' => 'required|string|min:6',
        'telefono' => 'nullable|string|max:15',
        'direccion' => 'required|string',
        'imagen' => 'nullable|string',
    ]);

    $data = $request->all();

    if ($request->hasFile('imagen')) {
        $file = $request->file('imagen');
        $filename = time() . '_' . $file->getClientOriginalName();
        $file->move(public_path('images'), $filename);
        $data['imagen'] = 'images/' . $filename;
    }

    $doctor = Doctors::create([
        'nombre' => $request->nombre,
        'apellido' => $request->apellido,
        'cedula' => $request->cedula,
        'profesion' => $request->profesion,
        'date' => $request->date,
        'edad' => $request->edad,
        'genero' => $request->genero,
        'email' => $request->email,
        'password' => $request->password,
        'telefono' => $request->telefono,
        'direccion' => $request->direccion,
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
'imagen' => $request->imagen,
]);

return response()->json($doctor, 201);
}

/**
 * Display the specified resource.
 */
public function show($id)
{
    try {
        // Buscar al doctor por ID
        $doctor = Doctors::find($id);

        // Retornar datos del doctor en formato JSON
        return response()->json([
            'success' => true,
            'data' => $doctor,
        ], 200);
    } catch (\Illuminate\Database\Eloquent\ModelNotFoundException $e) {
        // Manejar caso de no encontrar el doctor
        return response()->json([
            'success' => false,
            'message' => 'Doctor no encontrado',
        ], 404);
    } catch (\Exception $e) {
        // Manejar cualquier otro error
        return response()->json([
            'success' => false,
            'message' => 'Error al obtener los datos del doctor',
        ], 500);
    }
}

/**
 * Show the form for editing the specified resource.
 */
public function edit(Doctors $doctors)
{
    //
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
/**
 * Update the specified resource in storage.
 */
public function update(Request $request, Doctors $doctors)
{
    // Validar los datos de entrada
    $validatedData = $request->validate([
        'nombre' => 'nullable|string|max:255',
        'apellido' => 'nullable|string|max:255',
        'cedula' => 'nullable|string|unique:doctors,cedula,' . $doctors->id,
        'profesion' => 'nullable|string|max:255',
        'edad' => 'nullable|integer',
        'genero' => 'nullable|string|max:10',
        'email' => 'nullable|email|unique:doctors,email,' . $doctors->id,
        'password' => 'nullable|string|min:6',
        'telefono' => 'nullable|string|max:15',
        'direccion' => 'nullable|string',
        'imagen' => 'nullable|string',
    ]);

    // Actualizar el doctor
    if (isset($validatedData['password'])) {
        $validatedData['password'] = bcrypt($validatedData['password']);
    }

    $doctor->update($validatedData);

    return response()->json($doctors, 200);
}

/**
 * Remove the specified resource from storage.
 */
public function destroy($id)
{
    $doctor = Doctors::find($id); // Busca el doctor por ID
    if (!$doctor) {
        return response()->json(['message' => 'Doctor no encontrado'], 404);
    }

    // Eliminar el doctor
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
$doctor->delete();

return response()->json(null, 204);
}
}
```

Patientcontroller.php

```
<?php

namespace App\Http\Controllers;

use App\Models\Patient;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Hash;

class PatientController extends Controller
{
    public function store(Request $request)
    {
        $validated = $request->validate([
            'registration_number' => 'required|string|unique:patients',
            'first_name' => 'required|string',
            'last_name' => 'required|string',
            'age' => 'required|integer',
            'gender' => 'required|string',
            'birth_date' => 'required|date',
            'emergency_contact' => 'required|string',
            'emergency_email' => 'required|email',
            'address' => 'required|string',
            'medical_history' => 'nullable|string',
            'allergies' => 'nullable|string',
            'description' => 'required|string',
            'last_consultation' => 'required|string',
            'doctor_id' => 'required|exists:doctors,id',
        ]);

        $patient = Patient::create($validated);

        return response()->json($patient, 201);
    }
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
// Obtener un paciente por ID
public function show($id)
{
    // Buscar el paciente por el ID
    $patient = Patient::find($id);

    // Verificar si el paciente existe
    if (!$patient) {
        return response()->json([
            'success' => false,
            'message' => 'Paciente no encontrado'
        ], 404); // Retorna un error 404 si no se encuentra
    }

    // Devolver los datos del paciente en formato JSON
    return response()->json([
        'success' => true,
        'patient' => $patient
    ], 200);
}

public function destroy($id)
{
    $patient = Patient::find($id); // Busca el doctor por ID
    if (!$patient) {
        return response()->json(['message' => 'Paciente no encontrado'], 404);
    }

    // Eliminar el doctor
    $patient->delete();

    return response()->json(null, 204);
}
}
```

Doctors.php

```
<?php

namespace App\Models;
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Factories\HasFactory;

class Doctors extends Model
{
    use HasFactory;

    /**
     * Los campos que pueden asignarse masivamente.
     *
     * @var array<int, string>
     */
    // Nombre de la tabla (si no sigue la convención por defecto)
    protected $table = 'doctors';

    protected $fillable = [
        'nombre',
        'apellido',
        'cedula',
        'profesion',
        'date',
        'edad',
        'genero',
        'email',
        'password',
        'telefono',
        'direccion',
        'imagen',
    ];
}
```

Patients.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Patient extends Model
{
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
use HasFactory;

/**
 * Campos que pueden ser asignados masivamente.
 */
protected $fillable = [
    'registration_number', // Matricula
    'first_name',          // Nombre
    'last_name',           // Apellido
    'age',                 // Edad
    'gender',              // Género
    'birth_date',          // Fecha de Nacimiento
    'emergency_contact',   // Contacto de Emergencia
    'emergency_email',     // Email de Emergencia
    'address',             // Domicilio
    'medical_history',     // Antecedentes Patológicos
    'allergies',           // Alergias
    'description',         // Descripción
    'last_consultation',   // Última Consulta
    'doctor_id',           // Relación con el doctor
];

/**
 * Relación: un paciente pertenece a un doctor.
 */
public function doctor()
{
    return $this->belongsTo(Doctors::class);
}
```

Api.php

```
<?php

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\DoctorsController;
use App\Http\Controllers\PatientController;
use App\Http\Controllers\AuthController;
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
Route::get('/user', function (Request $request) {
    return $request->user();
})->middleware('auth:sanctum');

// DOCTORES
Route::get('/doctors', [DoctorsController::class, 'index']);
Route::post('/doctors', [DoctorsController::class, 'store']);
Route::get('/doctors/{id}', [DoctorsController::class, 'show']);
Route::put('/doctors/{id}', [DoctorsController::class, 'update']);
Route::delete('/doctors/{id}', [DoctorsController::class, 'destroy']);
Route::resource('doctors', DoctorsController::class);
Route::post('/change-password', [AuthController::class, 'changePassword']); // Ruta
para cambiar la contraseña
Route::post('/validate-password', [AuthController::class, 'loginDoctor']);

// PACIENTES
Route::post('/add-patients', [PatientController::class, 'store']);
Route::get('/doctor{doctorId}/patients', [AuthController::class, 'getDoctorPatients']);
Route::get('patients/{id}', [PatientController::class, 'show']);
Route::delete('patients/{id}', [PatientController::class, 'destroy']);
Route::resource('patients', PatientController::class);
```

Angular (código).

Api.service.ts

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http'
import { BehaviorSubject, Observable, Subject } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class ApiService {
  private urlApi = 'http://127.0.0.1:8000/api'; // Api Laravel (Backend)

  /* SECCION CATEGORIA Y SUBCATEGORIA */
  //Selecciona Subcategoria
  private selectSubCat = new BehaviorSubject<String>('Undefinid');
  selectEncuest$: Observable<String>;
  //Selecciona la Categoria
  private categoriaSeleccionadaSource: BehaviorSubject<String> = new
  BehaviorSubject<String>('Undefinid');
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
categoriaSeleccionada$: Observable<String>;

constructor(private http: HttpClient) {
  // Recuperar la categoría/encuesta seleccionada del Local Storage o establecer un
  // valor predeterminado
  const categoriaGuardada = localStorage.getItem('categoriaSeleccionada') ||
  'undefined';
  const encuestaGuardada = localStorage.getItem('encuestaSeleccionada') ||
  'undefined';
  // Inicializar BehaviorSubject con el valor recuperado
  this.categoriaSeleccionadaSource = new BehaviorSubject<String>(categoriaGuardada);
  this.selectSubCat = new BehaviorSubject<String>(encuestaGuardada);
  // Asignar el observable a una propiedad pública
  this.categoriaSeleccionada$ = this.categoriaSeleccionadaSource.asObservable();
  this.selectEncuest$ = this.selectSubCat.asObservable();
}
//Selección de Categoría y Encuesta
seleccionarEncuesta(encuesta: String){
  this.selectSubCat.next(encuesta);
  localStorage.setItem('encuestaSeleccionada', encuesta.toString());
}
seleccionarCategoria(categoria: string){
  // Actualizar la categoría y persistirla en Local Storage
  this.categoriaSeleccionadaSource.next(categoria);
  localStorage.setItem('categoriaSeleccionada', categoria.toString());
}

/* SECCION DE LARAVEL DE API */

// Registrar un nuevo doctor
registerDoctor(data: FormData): Observable<any> {
  return this.http.post<any>(`${this.urlApi}/doctors`, data);
  // return this.http.post<any>(this.urlApi,data);
}
// Login del doctor
loginDoctor(credentials: any): Observable<any> {
  return this.http.post<any>(this.urlApi+'/validate-password', credentials);
}
//Obtener los registros Tabla Doctor
getUsers(): Observable<any> {
  return this.http.get(this.urlApi+'/doctors');
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
//Eliminar Doctor
deleteDoctor(id: number): Observable<any> {
  return this.http.delete(`${this.urlApi}/doctors/${id}`);
}
//Obtener Doctor por id
getDoctorId(id: number): Observable<any> {
  return this.http.get(`${this.urlApi}/doctors/${id}`);
}
// Método para cambiar la contraseña
changePassword(payload: { correo: string, password: string }): Observable<any> {
  return this.http.post(`${this.urlApi}/change-password`, payload);
}
/* ----- PACIENTE ----- */
//Crear Paciente
crearPaciente(data: any): Observable<any> {
  return this.http.post<any>(`${this.urlApi}/add-patients`, data);
}
// En ApiService
getPatientsByDoctor(doctorId: number): Observable<any> {
  return this.http.get<any>(`${this.urlApi}/doctor/${doctorId}/patients`);
}
// Obtiene el paciente por id
getPatientById(id: number): Observable<any> {
  return this.http.get(`${this.urlApi}/patients/${id}`);
}
//Eliminar Paciente
deletePatient(patientId: number): Observable<any> {
  return this.http.delete<any>(`${this.urlApi}/patients/${patientId}`);
}
}
```

User.service.ts

```
import { Injectable } from '@angular/core';
import Swal from 'sweetalert2';
import { ApiService } from '../api.service';
import { SharedService } from '../shared.service';

@Injectable({
  providedIn: 'root'
})
export class UserService {
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
private ApiService?: ApiService;
showButton: boolean = false;
usuario: string = '';

isLoggedIn() {
    throw new Error('Method not implemented.');
```

}

```
private validar = {
    id: 1,
    email: 'prueba_auth123@hotmail.com',
    password: '12345'
};

constructor(private serviApi: ApiService, private sharedService: SharedService) { }
```

```
login(email: string, password: string): boolean {
    if (email === this.validar.email && password === this.validar.password) {
        localStorage.setItem('isAuthenticated', 'logged_in'); // Guarda un token básico
en el localStorage
        return true;
    } else {
        // Mostrar SweetAlert si las credenciales son incorrectas
        Swal.fire({
            icon: 'error',
            title: 'Error',
            text: 'Correo electrónico y/o contraseña incorrectos',
            confirmButtonText: 'Aceptar'
        });
        return false;
    }
}
```

```
logout(): void {
    localStorage.removeItem('isAuthenticated'); // Borra el token para cerrar sesión
    localStorage.removeItem('doctorName'); // Eliminar Nombre
    localStorage.removeItem('idDoctor'); // Eliminar el ID
}
```

```
isAuthenticated(): boolean {
    return !!localStorage.getItem('isAuthenticated'); // Verifica si el token está en
localStorage
}
```



TECNOLÓGICO NACIONAL DE MÉXICO

```
// BOTON DE VISTA REGISTRAR

canShowRegisterButton(email: string, registeredUsers: any[]): boolean {
  // Verifica si el correo es el permitido para registrar
  if (email === this.validar.email) {
    return true;
  }
  // Verifica si el correo ya está registrado en la base de datos
  return !registeredUsers.some(user => user.email === email);
}

// LOGIN DE SUPER USER Y REVISION DE BASE DE DATOS

async log(email: string, password: string): Promise<boolean> {
  try {
    // Si no está en la base de datos, valida el correo de prueba
    /*if (email === this.validar.email && password === this.validar.password) {
      localStorage.setItem('isAuthenticated', 'logged_in');
      localStorage.setItem('doctorName', 'Dr. Master Crack'); // Guardar el nombre
del doctor de prueba
      this.sharedService.setShowRegisterButton(true); // Mostrar botón en el caso del
usuario de prueba
      return true;
    }*/

    // Busca un usuario con el correo y contraseña proporcionados
    const response: any = await this.serviApi.loginDoctor({ email, password
  }).toPromise();

    if (response.success) {
      this.usuario = response.user.nombre + ' ' + response.user.apellido;
      localStorage.setItem('isAuthenticated', 'logged_in');
      localStorage.setItem('doctorName', this.usuario); // Guardar el nombre del
doctor
      localStorage.setItem('idDoctor', response.user.id.toString());
      this.sharedService.setShowRegisterButton(false);
      return true;
    }

    // Si no coincide con ninguno, muestra error
  }
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
Swal.fire({
  icon: 'error',
  title: 'Error',
  text: 'Correo electrónico y/o contraseña incorrectos',
  confirmButtonText: 'Aceptar',
});

return false;
} catch (error) {
  console.error('Error al obtener usuarios:', error);
  Swal.fire({
    icon: 'error',
    title: 'Error',
    text: 'No se pudo conectar al servidor',
    confirmButtonText: 'Aceptar',
  });
  return false;
}
}

getDoctorName(): string {
  return localStorage.getItem('doctorName') || '';
}

getDoctorId(): number {
  const id = localStorage.getItem('idDoctor');
  return id ? parseInt(id, 10) : 0; // Convierte a número o devuelve 0 si no está
definido
}
}
```

Encuestas Esfera Afectiva (HTML y TypeScript)

```
<div class="container mt-4">
  <h1 class="text-center mb-4">GDS - 15</h1>
  <form [formGroup]="gdsForm">
    <div class="table-responsive">
      <table class="table table-bordered table-striped">
        <thead class="table-primary">
          <tr>
            <th>#</th>
            <th>Pregunta</th>
          </tr>
        </thead>
      </table>
    </div>
  </form>
</div>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
<th>Sí (1)</th>
<th>No (0)</th>
</tr>
</thead>
<tbody>
<tr *ngFor="let pregunta of preguntas; let i = index">
  <td>{{ i + 1 }}</td>
  <td>{{ pregunta }}</td>
  <td>
    <input
      type="radio"
      [value]="1"
      [formControlName]='pregunta' + i"
    />
  </td>
  <td>
    <input
      type="radio"
      [value]="0"
      [formControlName]='pregunta' + i"
    />
  </td>
</tr>
</tbody>
</table>
</div>
<div class="text-center">
  <button type="button" class="btn btn-primary"
(click)="calcularPuntuacion()">Calcular y Ver Resultados</button>
</div>
</form>
</div>

import { Component, OnInit } from '@angular/core';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';
import { Router } from '@angular/router';

@Component({
  selector: 'app-encuesta-afc',
  templateUrl: './encuesta-afc.component.html',
  styleUrls: ['./encuesta-afc.component.scss']
})
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
export class EncuestaAfcComponent implements OnInit {
  gdsForm!: FormGroup;
  preguntas: string[] = [
    '¿En general, está satisfecho(a) con su vida?',
    '¿Ha abandonado muchas de sus tareas habituales y aficiones?',
    '¿Siente que su vida está vacía?',
    '¿Se siente con frecuencia aburrido(a)?',
    '¿Se encuentra de buen humor la mayor parte del tiempo?',
    '¿Teme que algo malo pueda ocurrirle?',
    '¿Se siente feliz la mayor parte del tiempo?',
    '¿Con frecuencia se siente desamparado(a), desprotegido(a)?',
    '¿Prefiere usted quedarse en casa, más que salir y hacer cosas nuevas?',
    '¿Cree que tiene más problemas de memoria que la mayoría de la gente?',
    '¿En estos momentos, piensa que es estupendo estar vivo(a)?',
    '¿Actualmente se siente un(a) inútil?',
    '¿Se siente lleno(a) de energía?',
    '¿Se siente sin esperanza en este momento?',
    '¿Piensa que la mayoría de la gente está en mejor situación que usted?'
  ];
  puntaje: number | null = null;
  observacion: string = '';
  showErrors = false;

  constructor(private fb: FormBuilder, private router: Router) {}

  ngOnInit(): void {
    // Inicializa el formulario
    this.gdsForm = this.fb.group(
      this.preguntas.reduce((controles, _, index) => {
        controles['pregunta' + index] = [null, Validators.required];
        return controles;
      }, {}) as { [key: string]: any }
    );
  }

  calcularPuntuacion(): void {
    if (this.gdsForm.invalid) {
      alert('Por favor, responde todas las preguntas.');
```

```
    return;
  }

  // Obtiene las respuestas con el tipo explícito
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
const respuestas: { [key: string]: number } = this.gdsForm.value;

// Suma las puntuaciones
this.puntaje = Object.values(respuestas).reduce(
  (acc: number, val: unknown) => acc + (typeof val === 'number' ? val : 0),
  0
);

// Interpreta el resultado
this.observacion =
  this.puntaje <= 4
    ? 'Normal'
    : 'Presencia de síntomas depresivos';

this.enviarResultados(this.puntaje, this.observacion);
}

enviarResultados(puntaje: number, observacion: string): void {
  if (puntaje === null) {
    alert('Primero calcula el puntaje antes de enviar los resultados.');
```

return;

```
  } else {
    this.showErrors = true;
  }

  console.log('Puntaje: ', puntaje, '\nObservación: ', observacion);

  // Redirige al componente Resultados con los datos mediante el estado
  if (this.showErrors) {
    this.router.navigate(['home/resultado'], {
      queryParams: {
        nombreEncuesta: 'Escala AFC',
        puntaje: puntaje,
        observacion: observacion,
        porcentaje: ((puntaje / 100) * 15).toFixed(2)
      }
    });
  }
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
<div class="container">
  <h2>Inventario de Ansiedad de Beck</h2>
  <p>
    Lea cada enunciado e indique cuál ha sido el nivel de afectación en la última
    semana, incluyendo el día de hoy.
  </p>
  <form #beckForm="ngForm" (ngSubmit)="finalizarEncuesta()">
    <table class="table table-bordered">
      <thead>
        <tr>
          <th>#</th>
          <th>Enunciado</th>
          <th>En absoluto (0)</th>
          <th>Levemente (1)</th>
          <th>Moderadamente (2)</th>
          <th>Severamente (3)</th>
        </tr>
      </thead>
      <tbody>
        <tr *ngFor="let pregunta of preguntas; let i = index">
          <td>{{ i + 1 }}</td>
          <td>{{ pregunta.texto }}</td>
          <td>
            <input
              type="radio"
              [id]=" 'pregunta-' + i + '-0'"
              name="pregunta-{{ i }}"
              value="0"
              [(ngModel)]="pregunta.respuesta"
              required
            />
          </td>
          <td>
            <input
              type="radio"
              [id]=" 'pregunta-' + i + '-1'"
              name="pregunta-{{ i }}"
              value="1"
              [(ngModel)]="pregunta.respuesta"
            />
          </td>
          <td>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
<input
  type="radio"
  [id]=" 'pregunta-' + i + '-2'"
  name="pregunta-{{ i }}"
  value="2"
  [(ngModel)]="pregunta.respuesta"
/>
</td>
<td>
  <input
    type="radio"
    [id]=" 'pregunta-' + i + '-3'"
    name="pregunta-{{ i }}"
    value="3"
    [(ngModel)]="pregunta.respuesta"
  />
</td>
</tr>
</tbody>
</table>
<!-- Botón para finalizar encuesta -->
<button
  type="button"
  class="btn btn-success"
  (click)="finalizarEncuesta()"
  [disabled]="!beckForm.valid"
>
  Finalizar Encuesta
</button>
</form>
</div>

import { Component } from '@angular/core';
import { Router } from '@angular/router';

@Component({
  selector: 'app-encuesta-beck-anxiety',
  templateUrl: './encuesta-beck-anxiety.component.html',
  styleUrls: ['./encuesta-beck-anxiety.component.scss']
})
export class EncuestaBeckAnxietyComponent {
  // Preguntas del inventario
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
preguntas = [  
  { texto: 'Torpe o entumecido.', respuesta: null },  
  { texto: 'Acalorado.', respuesta: null },  
  { texto: 'Con temblor en las piernas.', respuesta: null },  
  { texto: 'Incapaz de relajarse.', respuesta: null },  
  { texto: 'Con temor a que ocurra lo peor.', respuesta: null },  
  { texto: 'Mareado, o que se le va la cabeza.', respuesta: null },  
  { texto: 'Con latidos fuertes del corazón y acelerados.', respuesta: null },  
  { texto: 'Inestable.', respuesta: null },  
  { texto: 'Atemorizado o asustado.', respuesta: null },  
  { texto: 'Nervioso.', respuesta: null },  
  { texto: 'Con sensación de bloqueo.', respuesta: null },  
  { texto: 'Con temblores en las manos.', respuesta: null },  
  { texto: 'Inquieto, inseguro.', respuesta: null },  
  { texto: 'Con miedo a perder el control.', respuesta: null },  
  { texto: 'Con sensación de ahogo.', respuesta: null },  
  { texto: 'Con temor a morir.', respuesta: null },  
  { texto: 'Con miedo.', respuesta: null },  
  { texto: 'Con problemas digestivos.', respuesta: null },  
  { texto: 'Con desvanecimientos.', respuesta: null },  
  { texto: 'Con rubor facial.', respuesta: null },  
  { texto: 'Con sudores, fríos o calientes.', respuesta: null }  
];  
  
puntaje: number = 0;  
observacion: string = '';  
showErrors: boolean = false;  
  
constructor(private router: Router) {}  
  
// Método para calcular el puntaje  
calcularPuntaje(): void {  
  this.puntaje = this.preguntas.reduce((total, pregunta) => total +  
    (Number(pregunta.respuesta) || 0), 0);  
  
  if (this.puntaje <= 5) {  
    this.observacion = 'Ausente o mínima ansiedad.';  
  } else if (this.puntaje <= 15) {  
    this.observacion = 'Ansiedad leve.';  
  } else if (this.puntaje <= 30) {  
    this.observacion = 'Ansiedad moderada.';  
  } else {
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
this.observacion = 'Ansiedad grave.';
}

console.log('Puntaje calculado: ', this.puntaje, 'Observación: ',
this.observacion);
}

// Método para enviar resultados al componente de resultados
enviarResultados(puntaje: number, observacion: string): void {
  if (puntaje === null) {
    alert('Primero calcula el puntaje antes de enviar los resultados.');
```

```
    return;
  } else {
    this.showErrors = true;
  }

  console.log('Puntaje: ', puntaje, '\nObservación: ', observacion);

  // Redirige al componente Resultados con los datos mediante el estado
  if (this.showErrors) {
    this.router.navigate(['home/resultado'], {
      queryParams: {
        nombreEncuesta: 'Inventario de Ansiedad de Beck',
        puntaje: puntaje,
        observacion: observacion,
        porcentaje: ((puntaje / 100) * 15).toFixed(2)
      }
    });
  }
}

// Método que combina el cálculo y el envío de los resultados
finalizarEncuesta(): void {
  // Calcular el puntaje
  this.calcularPuntaje();

  // Enviar los resultados al componente de resultados
  this.enviarResultados(this.puntaje, this.observacion);
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
<div class="container mt-4">
  <h2 class="text-center">CES - D 7 Escala de Depresión</h2>
  <p class="text-center">Durante la última semana, ¿usted...?</p>

  <form [formGroup]="cesForm">
    <div class="table-responsive">
      <table class="table table-bordered text-center">
        <thead class="table-light">
          <tr>
            <th>Preguntas</th>
            <th>Rara vez o nunca<br>(menos de 1 día)<br>0 puntos</th>
            <th>Pocas veces<br>(1-2 días)<br>1 punto</th>
            <th>Un número considerable<br>(3-4 días)<br>2 puntos</th>
            <th>Todo el tiempo<br>(5-7 días)<br>3 puntos</th>
          </tr>
        </thead>
        <tbody>
          <tr *ngFor="let pregunta of preguntas; let i = index">
            <td>{{ pregunta.texto }}</td>
            <td>
              <input
                type="radio"
                [value]="0"
                [formControlName]='pregunta' + i"
              />
            </td>
            <td>
              <input
                type="radio"
                [value]="1"
                [formControlName]='pregunta' + i"
              />
            </td>
            <td>
              <input
                type="radio"
                [value]="2"
                [formControlName]='pregunta' + i"
              />
            </td>
            <td>
              <input
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
        type="radio"
        [value]="3"
        [formControlName]='pregunta' + i"
      />
    </td>
  </tr>
</tbody>
</table>
</div>

<div class="text-center mt-3">
  <button type="button" class="btn btn-success" (click)="finalizarEncuesta()">
    Finalizar Encuesta
  </button>
</div>
</form>

import { Component, OnInit } from '@angular/core';
import { FormBuilder, FormGroup } from '@angular/forms';
import { Router } from '@angular/router';

@Component({
  selector: 'app-encuesta-ces-d7',
  templateUrl: './encuesta-ces-d7.component.html',
  styleUrls: ['./encuesta-ces-d7.component.scss']
})
export class EncuestaCESD7Component implements OnInit {
  cesForm!: FormGroup;
  preguntas = [
    { texto: '¿Sentía como si no pudiera quitarse la tristeza?' },
    { texto: '¿Le costaba concentrarse en lo que estaba haciendo?' },
    { texto: '¿Se sintió deprimido/a?' },
    { texto: '¿Le parecía que todo lo que hacía era un esfuerzo?' },
    { texto: '¿No durmió bien?' },
    { texto: '¿Disfrutó de la vida?' },
    { texto: '¿Se sintió triste?' }
  ];
  resultado: number = 0;
  interpretacion: string = '';
  showErrors: boolean = false;

  constructor(private fb: FormBuilder, private router: Router) {}
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
ngOnInit(): void {
  // Crea un grupo de controles para cada pregunta
  this.cesForm = this.fb.group(
    this.preguntas.reduce((controls, _, index) => {
      controls['pregunta' + index] = [null];
      return controls;
    }, {} as { [key: string]: any })
  );
}

// Método para calcular la puntuación
calcularPuntuacion(): void {
  const respuestas = this.cesForm.value;

  // Calcula la puntuación total
  this.resultado = (Object.values(respuestas) as (number | null)[]).reduce(
    (acc: number, val: number | null) => acc + (val || 0),
    0 // Inicializa el acumulador con 0
  );

  // Determina la interpretación
  this.interpretacion =
    this.resultado >= 5
      ? 'Síntomas depresivos significativos'
      : 'Normal';
}

// Método para enviar los resultados
enviarResultados(): void {
  if (this.resultado === 0 && !this.showErrors) {
    alert('Primero calcula el puntaje antes de enviar los resultados.');
```

`this.showErrors = true;`

`return;`

`}`

```
console.log('Puntaje: ', this.resultado, '\nObservación: ', this.interpretacion);

// Redirige al componente Resultados con los datos mediante el estado
this.router.navigate(['home/resultado'], {
  queryParams: {
    nombreEncuesta: 'CES-D7',
    puntaje: this.resultado,
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
observacion: this.interpretacion,
porcentaje: ((this.resultado / 21) * 100).toFixed(2) // Porcentaje basado en 7
preguntas con valor máximo de 3 cada una
    }
  });
}

// Método para finalizar la encuesta: calcula el puntaje y luego envía los resultados
finalizarEncuesta(): void {
  if (this.cesForm.valid) {
    this.calcularPuntuacion(); // Calcula la puntuación
    this.enviarResultados(); // Envía los resultados
  } else {
    alert('Por favor, complete todas las preguntas antes de finalizar.');
```

```
<div class="cornell-scale">
  <h2>Escala de Cornell</h2>
  <p>Manifestaciones.</p>

  <div *ngFor="let question of questions; let i = index" class="question">
    <label>{{ i + 1 }}. {{ question.text }}</label>
    <div class="options">
      <label>
        <input type="radio" name="q{{ i }}" /> No Valorable
      </label>
      <label>
        <input type="radio" name="q{{ i }}" (change)="setScore(i, 0)" /> Ausente
      </label>
      <label>
        <input type="radio" name="q{{ i }}" (change)="setScore(i, 1)" /> Leve
      </label>
      <label>
        <input type="radio" name="q{{ i }}" (change)="setScore(i, 2)" /> Grave
      </label>
    </div>
  </div>
</div>

<div class="buttons">
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
<button class="btn btn-primary" (click)="calculateScore()">Calcular
Puntaje</button>
<button
  class="btn btn-success"
  [disabled]="resultado === 0"
  (click)="enviarResultados()">
  Enviar Resultados
</button>
</div>

import { Component } from '@angular/core';
import { Router } from '@angular/router';

@Component({
  selector: 'app-encuesta-cornell',
  templateUrl: './encuesta-cornell.component.html',
  styleUrls: ['./encuesta-cornell.component.scss']
})
export class EncuestaCornellComponent {
  questions = [
    { text: 'Ansiedad (Expresion ansiosa, rumiacion de ideas, preocupacion)', score: 0 },
    { text: 'Tristeza (Expresion triste, voz apagada, tendencia al llanto)', score: 0 },
    { text: 'Falta de reactividad a los acontecimientos placenteros', score: 0 },
    { text: 'Irritabilidad (facilmente enojable, poco temperamento)', score: 0 },
    { text: 'Agitacion (incapacidad de permanecer quieto, jugar con las manos, tirarse el pelo)', score: 0 },
    { text: 'Enlentecimiento (Movimientos, habla y reacciones enlentecidos)', score: 0 },
    { text: 'Quejas Fisicas multiples (puntue 0 si son unicamente gastrointestinales)', score: 0 },
    { text: 'perdida del interes (menos envuelto en las actividades habituales; puntue solamente si el cambio ha sido agudo, p. ej., en menos de 1 mes)', score: 0 },
    { text: 'Perdida de apetito (come menos de lo habitual)', score: 0 },
    { text: 'Perdida de peso (puntue 2 si 2.5kg en 1 mes )', score: 0 },
    { text: 'Perdida de energia (Se fatiga facilmente, incapaz de mantener actividades; puntue solamente si el cambio ha sido agudo, p. ej., en menos de 1 mes )', score: 0 },
    { text: 'variaciones diurnas del estado de animo (los sintomas empeoran por la mañana)', score: 0 },
    { text: 'Dificultad para conciliar el sueño (mas de lo habitual para el paciente )', score: 0 },
  ],
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
{ text: 'Despertares multiples durante el sueño', score: 0 },
{ text: 'Despertar precoz (antes de lo habitual para el paciente)', score: 0 },
{ text: 'Suicidio (siente que la vida no merece ser vivida, tiene deseos suicidas o
realiza intentos de suicidio) ', score: 0 },
{ text: 'Baja autoestima (autoculpa, autodepreciacion, sentimiento de fracaso)',
score: 0 },
{ text: 'Pesimismo (anticipacion a lo peor)', score: 0 },
{ text: 'Delirios congruentes con el estado de ánimo (delirios de pobreza,
enfermedades o pérdidas)', score: 0 },
];

resultado: number = 0;
observacion: string = '';
showErrors: boolean = false;

constructor(private router: Router) {}

setScore(index: number, score: number): void {
  this.questions[index].score = score;
}

calculateScore(): void {
  this.resultado = this.questions.reduce((sum, question) => sum + question.score, 0);

  if (this.resultado <= 12) {
    this.observacion = 'depresion (Menor o mayor) ';
  } else if (this.resultado <= 38 ) {
    this.observacion = 'Depresión mayor';
  }

  this.showErrors = true; // Activa la validación para enviar resultados.
}

enviarResultados(): void {
  if (!this.showErrors || this.resultado === null) {
    alert('Primero calcula el puntaje antes de enviar los resultados.');
```

return;

```
  }

  console.log('Puntaje:', this.resultado, '\nObservación:', this.observacion);

  this.router.navigate(['home/resultado'], {
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
queryParams: {  
  nombreEncuesta: 'Escala de Cornell',  
  puntaje: this.resultado,  
  observacion: this.observacion,  
  porcentaje: ((this.resultado / 38) * 100).toFixed(2) // Calcula un porcentaje  
basado en un puntaje máximo de 30.  
}  
});  
}  
}
```

```
<div class="container">  
  <h2>Escala de Soledad de 3 Elementos</h2>  
  <p>  
    indicaciones: se le dira a la persona lo siguiente "estas preguntas son sobre  
    como se siente sobre los diferentes aspectos de su vida. para cada pregunta responda  
    con que frecuencia se siente asi".  
  </p>  
  <form #soledadForm="ngForm" (ngSubmit)="calcularPuntaje()">  
    <table class="table table-bordered">  
      <thead>  
        <tr>  
          <th>#</th>  
          <th>Pregunta</th>  
          <th>Casi nunca (1)</th>  
          <th>A veces (2)</th>  
          <th>A Menudo (3)</th>  
        </tr>  
      </thead>  
      <tbody>  
        <tr *ngFor="let pregunta of preguntas; let i = index">  
          <td>{{ i + 1 }}</td>  
          <td>{{ pregunta.texto }}</td>  
          <td>  
            <input  
              type="radio"  
              [id]="pregunta-' + i + '-1'"  
              name="pregunta-{{ i }}"  
              value="1"  
              [(ngModel)]="pregunta.respuesta"  
              required  
            >  
          </td>  
        </tr>  
      </tbody>  
    </table>  
  </form>  
</div>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
        />
      </td>
      <td>
        <input
          type="radio"
          [id]=" 'pregunta-' + i + '-2'"
          name="pregunta-{{ i }}"
          value="2"
          [(ngModel)]="pregunta.respuesta"
        />
      </td>
      <td>
        <input
          type="radio"
          [id]=" 'pregunta-' + i + '-3'"
          name="pregunta-{{ i }}"
          value="3"
          [(ngModel)]="pregunta.respuesta"
        />
      </td>
    </tr>
  </tbody>
</table>
<button type="submit" class="btn btn-primary" [disabled]="!soledadForm.valid">
  Finalizar Encuesta
</button>
</form>
</div>
```

```
import { Component } from '@angular/core';
```

```
import { Router } from '@angular/router';
```

```
@Component({
  selector: 'app-encuesta-escala-soledad',
  templateUrl: './encuesta-escala-soledad.component.html',
  styleUrls: ['./encuesta-escala-soledad.component.scss']
})
```

```
export class EncuestaEscalaSoledadComponent {
  // Preguntas de la escala
  preguntas = [
    { texto: 'Con qué frecuencia siente que le falta compañía.', respuesta: null },
    { texto: 'Con qué frecuencia se siente excluido/a.', respuesta: null },
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
{ texto: 'Con qué frecuencia se siente aislado/a de los demás.', respuesta: null }
];

constructor(private router: Router) {}

// Calcular puntaje total y redirigir al componente de resultados
calcularPuntaje() {
  const puntaje = this.preguntas.reduce((total, pregunta) => total +
(Number(pregunta.respuesta) || 0), 0);

  let observacion: string;
  if (puntaje <= 3) {
    observacion = 'Bajo nivel de soledad.';
  } else if (puntaje <= 6) {
    observacion = 'Nivel moderado de soledad.';
  } else {
    observacion = 'Alto nivel de soledad.';
  }

  // Llamar al método para enviar los resultados
  this.enviarResultados(puntaje, observacion);
}

// Método para enviar los resultados al componente de resultados
enviarResultados(puntaje: number, observacion: string): void {
  if (puntaje === null) {
    alert('Primero calcula el puntaje antes de enviar los resultados.');
```

return;

```
  }

  console.log('Puntaje: ', puntaje, '\nObservación: ', observacion);

  // Redirige al componente de resultados con los datos mediante queryParams
  this.router.navigate(['home/resultado'], {
    queryParams: {
      nombreEncuesta: 'Escala de Soledad de 3 Elementos',
      puntaje: puntaje,
      observacion: observacion,
      porcentaje: ((puntaje / 9) * 100).toFixed(2)
    }
  });
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

}

```
<div class="container">
  <h2>Inventario de Ansiedad Geriátrica Versión Corta (GAI-SF)</h2>
  <p>Conteste los enunciados de acuerdo con cómo se sintió la semana pasada.</p>
  <form #gaiForm="ngForm" (ngSubmit)="calcularPuntaje()">
    <table class="table table-bordered">
      <thead>
        <tr>
          <th>#</th>
          <th>Enunciados</th>
          <th>Coincide</th>
          <th>No coincide</th>
        </tr>
      </thead>
      <tbody>
        <tr *ngFor="let pregunta of preguntas; let i = index">
          <td>{{ i + 1 }}</td>
          <td>{{ pregunta.texto }}</td>
          <td>
            <input
              type="radio"
              [id]='pregunta-' + i + '-coincide'
              name="pregunta-{{ i }}"
              value="1"
              [(ngModel)]="pregunta.respuesta"
              required
            />
          </td>
          <td>
            <input
              type="radio"
              [id]='pregunta-' + i + '-no-coincide'
              name="pregunta-{{ i }}"
              value="0"
              [(ngModel)]="pregunta.respuesta"
            />
          </td>
        </tr>
      </tbody>
    </table>
  </div>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
<button type="submit" class="btn btn-primary" [disabled]="!gaiForm.valid">
  Finalizar Encuesta
</button>
</form>
</div>

import { Component } from '@angular/core';
import { Router } from '@angular/router';

@Component({
  selector: 'app-encuesta-gai-sf',
  templateUrl: './encuesta-gai-sf.component.html',
  styleUrls: ['./encuesta-gai-sf.component.scss']
})
export class EncuestaGaiSfComponent {
  // Preguntas de la encuesta
  preguntas = [
    { texto: 'Me paso mucho tiempo preocupado.', respuesta: null },
    { texto: 'Las pequeñas cosas me molestan mucho.', respuesta: null },
    { texto: 'Me considero una persona preocupada.', respuesta: null },
    { texto: 'A menudo me siento nervioso.', respuesta: null },
    { texto: 'Mis propios pensamientos me hacen sentir ansioso.', respuesta: null }
  ];

  constructor(private router: Router) {}

  // Calcular el puntaje total y redirigir al componente de resultados
  calcularPuntaje() {
    const puntaje = this.preguntas.reduce((total, pregunta) => total +
      (Number(pregunta.respuesta) || 0), 0);

    let observacion: string;
    if (puntaje >= 3) {
      observacion = 'Posible presencia de ansiedad geriátrica.';
    } else {
      observacion = 'No se detecta ansiedad significativa.';
    }

    // Llamar al método para enviar los resultados
    this.enviarResultados(puntaje, observacion);
  }

  // Método para enviar los resultados al componente de resultados
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
enviarResultados(puntaje: number, observacion: string): void {  
  if (puntaje === null) {  
    alert('Primero calcula el puntaje antes de enviar los resultados.');    return;  
  }  
  
  console.log('Puntaje: ', puntaje, '\nObservación: ', observacion);  
  
  // Redirige al componente de resultados con los datos mediante queryParams  
  this.router.navigate(['home/resultado'], {  
    queryParams: {  
      nombreEncuesta: 'GAI-SF',  
      puntaje: puntaje,  
      observacion: observacion,  
      porcentaje: ((puntaje / 10) * 100).toFixed(2)  
    }  
  });  
}
```

```
<div class="container mt-4">  
  <h2 class="text-center">Escala Corta de Depresión por Observación Hammond-  
  O'Keefe</h2>  
  
  <form [formGroup]="encuestaForm">  
    <div class="table-responsive">  
      <table class="table table-bordered text-center">  
        <thead class="table-light">  
          <tr>  
            <th>Pregunta</th>  
            <th>NO (0 puntos)</th>  
            <th>SI (1 punto)</th>  
          </tr>  
        </thead>  
        <tbody>  
          <tr *ngFor="let pregunta of preguntas; let i = index">  
            <td>{{ pregunta.texto }}</td>  
            <td>  
              <input  
                type="radio"  
                [value]="0"  
              >  
            </td>  
          </tr>  
        </tbody>  
      </table>  
    </div>  
  </form>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
        [formControlName]='pregunta' + i"
      />
    </td>
    <td>
      <input
        type="radio"
        [value]="1"
        [formControlName]='pregunta' + i"
      />
    </td>
  </tr>
</tbody>
</table>
</div>

<div class="text-center mt-3">
  <button type="button" class="btn btn-success" (click)="finalizarEncuesta()">
    Finalizar Encuesta
  </button>
</div>
</form>
</div>

import { Component, OnInit } from '@angular/core';
import { FormBuilder, FormGroup } from '@angular/forms';
import { Router } from '@angular/router';

@Component({
  selector: 'app-encuesta-okeeffe',
  templateUrl: './encuesta-okeeffe.component.html',
  styleUrls: ['./encuesta-okeeffe.component.scss']
})
export class EncuestaOkeeffeComponent implements OnInit {
  encuestaForm!: FormGroup;
  preguntas = [
    { texto: 'Se observa al paciente la mayor parte del tiempo triste, infeliz o deprimido ' },
    { texto: 'Alguna vez el paciente presenta llanto o parece lloroso' },
    { texto: 'El paciente parece estar agitado, inquieto o ansioso' },
    { texto: 'El paciente esta letárgico o refractario a movilizarse' },
    { texto: 'El paciente necesita mucho animo para hacer las cosas por él (ella)' },
  ]
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
{ texto: 'El paciente parece reservado, muestra minimo interes en el entorno
social' }
];
resultado: number = 0;
interpretacion: string = '';
showErrors: boolean = false;

constructor(private fb: FormBuilder, private router: Router) {}

ngOnInit(): void {
  // Crea un grupo de controles para cada pregunta
  this.encuestaForm = this.fb.group(
    this.preguntas.reduce((controls, _, index) => {
      controls['pregunta' + index] = [null];
      return controls;
    }, {}) as { [key: string]: any })
  );
}

// Método para calcular la puntuación
calcularPuntuacion(): void {
  const respuestas = this.encuestaForm.value;

  // Calcula la puntuación total
  this.resultado = (Object.values(respuestas) as (number | null)[]).reduce(
    (acc: number, val: number | null) => acc + (val || 0),
    0 // Inicializa el acumulador con 0
  );

  // Determina la interpretación
  this.interpretacion =
    this.resultado >= 3
      ? 'Depresión significativa'
      : 'Depresión leve o ausencia de depresión';
}

// Método para enviar los resultados
enviarResultados(): void {
  if (this.resultado === 0 && !this.showErrors) {
    alert('Primero calcula el puntaje antes de enviar los resultados.');
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
}

console.log('Puntaje: ', this.resultado, '\nObservación: ', this.interpretacion);

// Redirige al componente Resultados con los datos mediante el estado
this.router.navigate(['home/resultado'], {
  queryParams: {
    nombreEncuesta: 'Escala Corta de Depresión por Observación Hammond-0\''Keeffe',
    puntaje: this.resultado,
    observacion: this.interpretacion,
    porcentaje: ((this.resultado / 6) * 100).toFixed(2) // Suponiendo un máximo de
18 puntos
  }
});
}

// Método para finalizar la encuesta: calcula el puntaje y luego envía los resultados
finalizarEncuesta(): void {
  if (this.encuestaForm.valid) {
    this.calcularPuntuacion(); // Calcula la puntuación
    this.enviarResultados(); // Envía los resultados
  } else {
    alert('Por favor, complete todas las preguntas antes de finalizar.');
```

```
<div class="container">
  <h2>PHQ-9</h2>
  <form #phqForm="ngForm" (ngSubmit)="calcularPuntaje()">
    <div *ngFor="let pregunta of preguntas; let i = index" class="mb-3">
      <label>{{ i + 1 }}. {{ pregunta.texto }}</label>
      <div class="form-check" *ngFor="let opcion of opciones">
        <input
          type="radio"
          [id]="pregunta - ' + i + '-opcion-' + opcion.valor"
          class="form-check-input"
          name="pregunta-{{ i }}"
          [value]="opcion.valor"
          [(ngModel)]="pregunta.respuesta"
          required
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
    />
    <label
      class="form-check-label"
      [for]="pregunta-' + i + '-opcion-' + opcion.valor"
    >
      {{ opcion.texto }}
    </label>
  </div>
</div>

<button type="submit" class="btn btn-primary" [disabled]="!phqForm.valid">
  Finalizar Encuesta
</button>
</form>
</div>
import { Component } from '@angular/core';
import { Router } from '@angular/router';

@Component({
  selector: 'app-encuesta-phq9',
  templateUrl: './encuesta-phq9.component.html',
  styleUrls: ['./encuesta-phq9.component.scss']
})
export class EncuestaPhq9Component {
  // Preguntas y opciones de la encuesta
  preguntas = [
    { texto: 'Poco interés o placer en hacer cosas', respuesta: null },
    { texto: 'Se ha sentido decaído(a), deprimido(a) o sin esperanzas', respuesta: null },
    { texto: 'Ha tenido dificultad para quedarse o permanecer dormido(a), o ha dormido demasiado', respuesta: null },
    { texto: 'Se ha sentido cansado(a) o con poca energía', respuesta: null },
    { texto: 'Sin apetito o ha comido en exceso', respuesta: null },
    { texto: 'Se ha sentido mal con usted mismo(a) - o que es un fracaso o que ha quedado mal con usted mismo(a) o con su familia', respuesta: null },
    { texto: 'Ha tenido dificultad para concentrarse en ciertas actividades, tales como leer el periódico o ver la televisión', respuesta: null },
    { texto: '¿Se ha movido o hablado tan lento que otras personas podrían haberlo notado? O lo contrario - muy inquieto(a) o agitado(a)', respuesta: null },
    { texto: 'Pensamientos de que estaría mejor muerto(a) o de lastimarse de alguna manera', respuesta: null }
  ];
};
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
opciones = [
  { texto: 'Ningún día', valor: 0 },
  { texto: 'Varios días', valor: 1 },
  { texto: 'Más de la mitad de los días', valor: 2 },
  { texto: 'Casi todos los días', valor: 3 }
];

constructor(private router: Router) {}

// Calcular el puntaje total y redirigir al componente de resultados
calcularPuntaje() {
  const puntaje = this.preguntas.reduce((total, pregunta) => total +
(pregunta.respuesta || 0), 0);
  let observacion: string;

  if (puntaje <= 4) {
    observacion = 'Mínima existencia o ausencia de síntomas depresivos.';
  } else if (puntaje <= 9) {
    observacion = 'Síntomas depresivos leves.';
  } else if (puntaje <= 14) {
    observacion = 'Síntomas depresivos moderados.';
  } else if (puntaje <= 19) {
    observacion = 'Síntomas depresivos moderados a graves.';
  } else {
    observacion = 'Síntomas depresivos graves.';
  }

  // Llamar al método para enviar los resultados
  this.enviarResultados(puntaje, observacion);
}

// Método para enviar los resultados al componente de resultados
enviarResultados(puntaje: number, observacion: string): void {
  if (puntaje === null) {
    alert('Primero calcula el puntaje antes de enviar los resultados.');
```

return;

```
  }

  console.log('Puntaje: ', puntaje, '\nObservación: ', observacion);

  // Redirige al componente de resultados con los datos mediante queryParams
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
this.router.navigate(['home/resultado'], {
  queryParams: {
    nombreEncuesta: 'PHQ-9',
    puntaje: puntaje,
    observacion: observacion,
    porcentaje: ((puntaje / 27) * 100).toFixed(2)
  }
});
}
```

```
<div class="container">
  <h2>Escala SAD PERSONS</h2>
  <p>
    Por favor, responda las siguientes preguntas indicando "Sí" o "No" según
    corresponda.
    Cada respuesta afirmativa suma 1 punto.
  </p>
  <form #sadForm="ngForm" (ngSubmit)="calcularPuntaje()">
    <table class="table table-bordered">
      <thead>
        <tr>
          <th>#</th>
          <th>Factor de riesgo</th>
          <th>Sí (1)</th>
          <th>No (0)</th>
        </tr>
      </thead>
      <tbody>
        <tr *ngFor="let pregunta of preguntas; let i = index">
          <td>{{ i + 1 }}</td>
          <td>{{ pregunta.texto }}</td>
          <td>
            <input
              type="radio"
              [id]=" 'pregunta-' + i + '-1'"
              name="pregunta-{{ i }}"
              value="1"
              [(ngModel)]="pregunta.respuesta"
              required
            />

```

TECNOLÓGICO NACIONAL DE MÉXICO

```
</td>
<td>
  <input
    type="radio"
    [id]=" 'pregunta-' + i + '-0'"
    name="pregunta-{{ i }}"
    value="0"
    [(ngModel)]="pregunta.respuesta"
  />
</td>
</tr>
</tbody>
</table>
<button type="submit" class="btn btn-primary" [disabled]="!sadForm.valid">
  Finalizar Encuesta
</button>
</form>
</div>
import { Component } from '@angular/core';
import { Router } from '@angular/router';

@Component({
  selector: 'app-encuesta-sad-persons',
  templateUrl: './encuesta-sad-persons.component.html',
  styleUrls: ['./encuesta-sad-persons.component.scss']
})
export class EncuestaSADPERSONSComponent {
  // Factores de riesgo en la escala SAD PERSONS
  preguntas = [
    { texto: 'S: Sexo masculino', respuesta: null },
    { texto: 'A: Edad (Age) <20 o >45 años', respuesta: null },
    { texto: 'D: Depresión', respuesta: null },
    { texto: 'P: Tentativa suicida previa', respuesta: null },
    { texto: 'E: Abuso de alcohol o drogas', respuesta: null },
    { texto: 'R: Falta de pensamiento racional (psicosis o trastornos cognitivos)',
  respuesta: null },
    { texto: 'S: Carencia de apoyo social', respuesta: null },
    { texto: 'O: Plan de suicidio organizado', respuesta: null },
    { texto: 'N: No pareja o cónyuge', respuesta: null },
    { texto: 'S: Enfermedad somática', respuesta: null }
  ];
};
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
constructor(private router: Router) {}

// Calcular puntaje y redirigir al componente de resultados
calcularPuntaje() {
  const puntaje = this.preguntas.reduce((total, pregunta) => total +
(Number(pregunta.respuesta) || 0), 0);

  let observacion: string;
  if (puntaje <= 2) {
    observacion = 'Alta médica al domicilio con seguimiento ambulatorio';
  } else if (puntaje <= 4) {
    observacion = 'Seguimiento ambulatorio intensivo, considerando ingreso';
  } else if (puntaje <= 6) {
    observacion = 'Recomendado ingreso, sobre todo si hay ausencia de apoyo social';
  } else {
    observacion = 'Ingreso obligatorio incluso en contra de su voluntad';
  }

  // Llamar al método para enviar los resultados
  this.enviarResultados(puntaje, observacion);
}

// Método para enviar los resultados al componente de resultados
enviarResultados(puntaje: number, observacion: string): void {
  if (puntaje === null) {
    alert('Primero calcula el puntaje antes de enviar los resultados.');
```

return;

```
  }

  console.log('Puntaje: ', puntaje, '\nObservación: ', observacion);

  // Redirige al componente de resultados con los datos mediante queryParams
  this.router.navigate(['home/resultado'], {
    queryParams: {
      nombreEncuesta: 'Escala SAD PERSONS',
      puntaje: puntaje,
      observacion: observacion,
      porcentaje: ((puntaje / 10) * 100).toFixed(2)
    }
  });
}
```



TECNOLÓGICO NACIONAL DE MÉXICO

Encuestas Esfera Cognitiva (HTML y TypeScript)

```
<!-- Encuesta 4AT -->
<div class="4at containerEncuesta" *ngIf="categoria == '4at'">
  <div class="title">
    4AT
  </div>
  <div class="p">Esto incluye que puedan presentarse somnolencia o
ahitado/hiperactivo.</div>
  <div class="opciones">
    <label for="">
      <input type="checkbox"> Normal
    </label>
    <label for="">
      <input type="checkbox" name="" id="">Somnolencia Breve &#60;10 Segundos
    </label>
    <label for="">
      <input type="checkbox">Claramente Normal
    </label>
  </div>
</div>

<!-- Encuesta CAM -->
<div class="cam containerEncuesta" *ngIf="categoria == 'cam'">
  <div class="title">
    CAM
  </div>
  <div class="opciones">
    <p class="texto">
      El Diagnostico de Delirium por CAM requiere la presencia de AMBAS características
A + B
    </p>
    <table class="tabla-delirium">
      <thead>
        <tr>
          <th colspan="3">
            El diagnóstico de Delirium por CAM requiere la presencia de AMBAS
características A + B
          </th>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
</tr>
</thead>
<tbody>
  <!-- Característica A -->
  <tr>
    <td><strong>A. Comienzo agudo y curso fluctuante</strong></td>
    <td colspan="2">¿Hay evidencia de un cambio agudo en el estado mental del
paciente distinto a su basal?</td>
  </tr>
  <tr>
    <td></td>
    <td colspan="2">El comportamiento anormal...
    <div>
      <label><input type="radio" name="comportamiento" /> a. Va y
viene?</label>
    </div>
    <div>
      <label><input type="radio" name="comportamiento" /> b. Fluctúa durante el
día?</label>
    </div>
    <div>
      <label><input type="radio" name="comportamiento" /> c. Aumenta o
disminuye en severidad?</label>
    </div>
  </td>
</tr>
  <!-- Característica B -->
  <tr>
    <td><strong>B. Inatención</strong></td>
    <td colspan="2">El paciente...
    <div>
      <label><input type="radio" name="inatencion" /> a. Tiene dificultad para
enfocar su atención?</label>
    </div>
    <div>
      <label><input type="radio" name="inatencion" /> b. Se distrae
fácilmente?</label>
    </div>
    <div>
      <label><input type="radio" name="inatencion" /> c. Tiene dificultad para
seguir una conversación?</label>
    </div>
  </td>
</tr>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
</td>
</tr>
<!-- Separador -->
<tr>
  <th colspan="3">Y la presencia de al menos una de las características C o
D</th>
</tr>
<!-- Característica C -->
<tr>
  <td><strong>C. Pensamiento desorganizado</strong></td>
  <td colspan="2">¿El pensamiento del paciente es...
    <div>
      <label><input type="radio" name="pensamiento" /> a.
Desorganizado?</label>
    </div>
    <div>
      <label><input type="radio" name="pensamiento" /> b. Incoherente?</label>
    </div>
    <p>Por ejemplo, ¿ha visto usted que el paciente tenga...</p>
    <div>
      <label><input type="radio" name="pensamientoEjemplo" /> a. Discurso
divagante / conversaciones
      irrelevantes?</label>
    </div>
    <div>
      <label><input type="radio" name="pensamientoEjemplo" /> b. Cambios de
tema impredecibles durante una
      conversación?</label>
    </div>
    <div>
      <label><input type="radio" name="pensamientoEjemplo" /> c. Flujo de ideas
ilógico o poco claro?</label>
    </div>
  </td>
</tr>
<!-- Característica D -->
<tr>
  <td><strong>D. Alteración en el nivel de conciencia</strong></td>
  <td colspan="2">¿Generalmente cómo consideraría el nivel de conciencia del
paciente?
    <div>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
<label><input type="radio" name="conciencia" /> 1. Alerta
(normal)</label>
</div>
<div>
<label><input type="radio" name="conciencia" /> 2. Vigilante
(hiperalerta)</label>
</div>
<div>
<label><input type="radio" name="conciencia" /> 3. Letárgico (somnoliento
pero fácil de despertar)</label>
</div>
<div>
<label><input type="radio" name="conciencia" /> 4. Estuporoso
(difícilmente despertable)</label>
</div>
<div>
<label><input type="radio" name="conciencia" /> 5. Comatoso (no
despertable)</label>
</div>
</td>
</tr>
</tbody>
</table>
</div>
</div>

<!-- Encuesta CAM-ICU -->
<div class="camicu containerEncuesta" *ngIf="categoria == 'camicu'">
  <div class="title">
    CAM-ICU
  </div>
  <div class="encuesta">
    <div class="instruccion">Confusion Assessments Method for the ICU &#40;CAM-
ICU&#41;</div>
    <form> <!--(ngSubmit)="enviarEncuesta()"-->
      <div *ngFor="let pregunta of preguntasCamIcu; let i = index" class="pregunta">
        <label [for]=" 'pregunta-' + i" class="preguntas">{{ pregunta.texto }}</label>
        <div class="opciones">
          <div class="opcSi" (click)="pregunta.respuesta = 'si'"
[class.selected]="pregunta.respuesta == 'si'">
            <input type="radio" [name]=" 'pregunta-' + i" [id]=" 'si-' + i" value="si"
[(ngModel)]="pregunta.respuesta" required />

```

TECNOLÓGICO NACIONAL DE MÉXICO

```
<label [for]='si-' + i">Si</label>
</div>
<div class="opcNo" (click)="pregunta.respuesta = 'no'"
[class.selected]="pregunta.respuesta == 'no'">
  <input type="radio" [name]='pregunta-' + i" [id]='no-' + i" value="no"
[(ngModel)]="pregunta.respuesta" required />
  <label [for]='no-' + i">No</label>
</div>
</div>
<!-- Mensaje de error para preguntas no respondidas -->
<div *ngIf="!pregunta.respuesta && showErrors" class="text-danger small">
  Por favor selecciona una respuesta.
</div>
</div>
</form>
</div>
</div>

<!-- Encuesta AWOL -->
<div class="awol containerEncuesta" *ngIf="categoria == 'awol'">
  <div class="title">Awol</div>
  <div class="instruccion">
    <p>Seleccione las letras para cambiar de <b>COLOR</b>, al estar en gris, se
considera 0 puntos, cada color es 1
    punto</p>
  </div>
  <div class="opciones">
    <p [ngClass]="{ 'text-green': isSelected['A'] }"
(click)="toggleTextColor('A')">A</p>
    <p [ngClass]="{ 'text-red': isSelected['W'] }" (click)="toggleTextColor('W')">W</p>
    <p [ngClass]="{ 'text-purple': isSelected['O'] }"
(click)="toggleTextColor('O')">O</p>
    <p [ngClass]="{ 'text-orange': isSelected['L'] }"
(click)="toggleTextColor('L')">L</p>
  </div>
  <div class="respuesta">
    <p class="green">A: Edad &#62; 80 A&#241;os</p>
    <p class="red">W: &#191;Incapaz de deletrear la Palabra: &#34;MUNDO&#34; al
reves&#63;</p>
    <p class="purple">O: &#191;Desorientado en Ciudad, Estado, Pa&#237;s, Hospital y
Piso&#63;</p>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
<p class="orange">L: Evaluacion Enfermedad &#191;Moderado, Grave o
Moribundo&#63;</p>
</div>
</div>

<!-- Encuesta Question Pfeiffer -->
<div class="spmsqp containerEncuesta" *ngIf="categoria == 'spmsqp'">
  <div class="title">Short Portable Mental State Questionnaire Pfeiffer</div>
  <div class="encuesta">
    <div class="instruccion">Responde la preguntas de acuerdo a la respuesta del
Paciente</div>
    <form>
      <div class="pregunta" *ngFor="let pregunta of preguntasMMSE; let i = index">
        <label [for]=" 'pregunta-' + i" class="preguntas">{{pregunta.texto}}</label>
        <div class="opciones">
          <div class="opcSi" (click)="pregunta.respuesta = 'si'"
[class.selected]="pregunta.respuesta == 'si'">
            <input type="radio" [name]=" 'pregunta-' + i" [id]=" 'si-' + i" value="si"
[(ngModel)]="pregunta.respuesta" required />
            <label [for]=" 'si-' + i">Si</label>
          </div>
          <div class="opcNo" (click)="pregunta.respuesta = 'no'"
[class.selected]="pregunta.respuesta == 'no'">
            <input type="radio" [name]=" 'pregunta-' + i" [id]=" 'si-' + i" value="no"
[(ngModel)]="pregunta.respuesta" required />
            <label [for]=" 'no-' + i">No</label>
          </div>
        </div>
        <div class="text-danger samll" *ngIf="!pregunta.respuesta && showErrors">
          Por favor responde todas las preguntas
        </div>
      </div>
    </form>
  </div>
</div>

<!-- Prueba del Reloj -->
<div class="reloj containerEncuesta" *ngIf="categoria == 'reloj'">
  <div class="title">Prueba del Reloj</div>
  <div class="instruccion">
    <p>Dibuje un reloj de manecillas, coloque todos los n&#250;meros que lleva y marque
con agujas la hora &#8220;11&#58;10&#8221;</p>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```

<p>De acuerdo al paciente, seleccione las opciones que considere que hizo
correctamente</p>
</div>
<div class="opciones">
  <form [formGroup]="questionnaireForm" (ngSubmit)="encuestas('reloj')">
    <!-- Pregunta 1 -->
    <div class="question">
      <p><strong>{{questions[0].text}}</strong></p>
      <div *ngFor="let option of questions[0].options" class="question1">
        <label>
          <input type="radio" [formControlName]='question1' [value]="option.score" />
          {{ option.label }}
        </label>
      </div>
    </div>
    <div *ngIf="hasError('question1')" class="text-danger small">
      Por favor selecciona una respuesta.
    </div>

    <!-- Pregunta 2 -->
    <div class="question">
      <p><strong>{{questions[1].text}}</strong></p>
      <div *ngFor="let option of questions[1].options">
        <label [ngStyle]="{ 'white-space': 'pre-line' }">
          <input type="radio" [formControlName]='question2' [value]="option.score"/>
          {{ option.label }}
        </label>
      </div>
    </div>
    <div *ngIf="hasError('question1')" class="text-danger small">
      Por favor selecciona una respuesta.
    </div>

    <!-- Pregunta 3 -->
    <div class="question">
      <p><strong>{{questions[2].text}}</strong></p>
      <div *ngFor="let option of questions[2].options">
        <label>
          <input type="radio" [formControlName]='question3' [value]="option.score"/>
          {{ option.label }}
        </label>
      </div>
    </div>
  </form>
</div>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
</label>
</div>
</div>
<div *ngIf="hasError('question1') && showErrors" class="text-danger small">
  Por favor selecciona una respuesta.
</div>

</form>
</div>
</div>

<!-- Boton de Envio -->
<div class="boton">
  <button type="submit" class="btn btn-success"
(click)="finalizarEncuesta()">Finalizar</button>
</div>
import { Component, OnInit } from '@angular/core';
import { FormBuilder, Validator, FormGroup, Validators } from '@angular/forms';
import { Router } from '@angular/router';
import { left } from '@popperjs/core';
import { float, FLOAT } from 'html2canvas/dist/types/css/property-descriptors/float';
import { ApiService } from 'src/app/Service/api.service';
import { PreguntaDosOpc, Question } from 'src/app/Shared/Data';

@Component({
  selector: 'app-encuesta-cog',
  templateUrl: './encuesta-cog.component.html',
  styleUrls: ['./encuesta-cog.component.scss']
})
export class EncuestaCogComponent implements OnInit{
  isSelected: { [key: string]: boolean } = {}; // Validacion de color de AWOL false/true
  puntos: number = 0; //Puntaje
  categoria: String = 'Undefinid'; //Recibir nombre clave
  cons: String[] = ['Entrastes en: ', 'Salida de: '];
  nameEncuesta?: String;
  observacion: String = '';

  //Preguntas Si No
  preguntasCamIcu: PreguntaDosOpc[] = [
    {texto: 'Existencia de Cambio agudo del Estado Mental', respuesta:''},
    {texto: 'Estado Mental del Paciente ha fluctuado durante las últimas 24hrs',
respuesta:''},
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
{texto: 'Apretar Mano (Médico) cuando el paciente diga A', respuesta:''},
{texto: 'Diga al Paciente que Deletree C-A-S-A-B-L-A-N-C-A', respuesta:''},
{texto: 'Vigile al pacinete y verifique: Alerta de RASS0', respuesta:''},
{texto: 'Las Piedras flotan en el Agua', respuesta:''},
{texto: 'Hay Peces en el mar', respuesta: '' },
{texto: '1kg pesa más que 2kg', respuesta: '' },
{texto: 'Los martillos sirven para poner clavos', respuesta: '' }
]
preguntasMMSE: PreguntaDosOpc[] = [
  {texto: 'Que fecha es hoy (dia, mes, año)', respuesta:''},
  {texto: 'Que día de la semana es hoy', respuesta:''},
  {texto: 'Donde estamos ahora (Lugar o Edificio)', respuesta:''},
  {texto: 'Cual es su numero de telefono (0 idreccion si no tiene uno)',
respuesta:''},
  {texto: 'Que edad tiene', respuesta:''},
  {texto: 'La fecha cuando nacio', respuesta:''},
  {texto: 'Como se llama el presidente del Gobierno', respuesta: '' },
  {texto: 'Como se llama el anterior presidente', respuesta: '' },
  {texto: 'Primer apellido de su Madre', respuesta: '' },
  {texto: 'Pudo restar de tres en tres desde veite', respuesta: ''}
]
//Preguntas Reloj
questionnaireForm: FormGroup;
questions: Question[] = [
  { text: 'Dibujo del Reloj',
    options: [
      { label: 'El círculo está bien dibujado o cercanamente a lo redondo', score: 2
},
      { label: 'El círculo está dibujado, pero de manera irregular', score: 1 },
      { label: 'Figura irregular o no compatible con un semi-círculo', score: 0 }
    ]
  },
  { text: 'Presencia y secuencia de números',
    options: [
      { label: 'Todos los números están presentes. Puede aceptarse un error mínimo en
la disposición espacial.', score: 4 },
      { label: 'Todos los números están presentes. Errores en la disposición
espacial.', score: 3 },
      { label: 'Algunas de las Sigüientes:\nNúmeros faltantes o adicionales aunque
sin distorsiones groseras de los números restantes.\nLos números están ubicados en
sentido anti-horario.\nLos números están presentes pero hay una seria alteración en la
disposición general.', score: 2 },
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
1  { label: 'Números faltantes o adicionales y errores espaciales serios', score:
    { label: 'Ausencia o pobre representación de los números.', score: 0 }
  ],
  { text: 'Presencia y ubicación de las agujas',
    options: [
      { label: 'Las agujas están en la posición correcta y la diferencia en tamaño
está respetada.', score: 4 },
      { label: 'Errores discretos en la representación de las agujas o la diferencia
de tamaño entre las agujas.', score: 3 },
      { label: 'Errores mayores en la ubicación de las agujas (de manera
significativa, incluyendo 10 a 11).', score: 2 },
      { label: 'Se dibuja solamente UNA aguja o el dibujo de ambas agujas es
notoriamente pobre.', score: 1 },
      { label: 'No se dibujan las agujas o se dibujan varias de manera
perseverativa.', score: 0 }
    ]
  }
];
showErrors: boolean = false;

constructor(private router: Router, private categoriaEncuestaComponenet: ApiService,
private fb: FormBuilder){
  this.questionnaireForm = this.fb.group({
    question1: [null, Validators.required],
    question2: [null, Validators.required],
    question3: [null, Validators.required],
  });
}

ngOnInit(): void {
  this.categoriaEncuestaComponenet.selectEncuest$.subscribe(subCategoria => {
    this.categoria = subCategoria;
  });
  console.log(this.categoria);
}

//Metodo del boton Finalizar
finalizarEncuesta(){
  //this.puntos = 0;
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
this.encuestas(this.categoria);// Entrar a la encuesta segun sea seleccionada

}

//Receptaculo Categoria
actualizarEncuesta(encuestaS: String){
  this.categoria = encuestaS;
}

/* METODOS DE LAS ENCUESTAS PARA CALCULAR LOS PUNTOS OBTENIDO */
encuestas(enc: String){
  switch(enc){
    case 'camicu':
      console.log(this.cons[0], this.categoria);
      const todasRespondidas = this.preguntasCamIcu.every(p => p.respuesta != null &&
p.respuesta != undefined && p.respuesta != '');
      console.log('Preguntas CAM-ICU: ' + todasRespondidas);

      if (!todasRespondidas) {
        // Mostrar errores si hay preguntas sin responder
        this.showErrors = true;
        return;
      } else{ this.showErrors = true}

      let puntajeSi = 0;
      for (const pregunta of this.preguntasCamIcu) {
        if (pregunta.respuesta == 'si') {
          puntajeSi++;
        }
      }

      this.encuestaResultado(puntajeSi, 'CAM-ICU','Sin
observaciones',(puntajeSi/9)*100);
      console.log(this.cons[1], 'CAM-ICU');
      break;
    case 'awol':
      var del: String = 'Riesgo de Delirium: '; //Validacion riesgo
      console.log(this.cons[0], this.categoria); //Verificar si funciona en consola
      //Calcular la observacion segun el puntaje
      if (this.puntos >= 4) {
        this.observacion = del + '64%';
      } else if (this.puntos == 3) {
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
this.observacion = del + '20%';
} else if (this.puntos == 2) {
  this.observacion = del + '14%';
} else if (this.puntos == 1) {
  this.observacion = del + '4%';
} else {
  this.observacion = del + '2%';
}

this.showErrors = true; // Al finalizar marca verdadero si, dependiendo si
selecciona la letra
console.log(this.cons[1], 'AWOL'); // Verificar que si hay salida
this.encuestaResultado(this.puntos, 'AWOL', this.observacion,
(this.puntos/4)/100); // Envía los parametros al metodo
break;
case 'spmsqp':
  var err: String = 'Errores Obtenido: '; // Acortar observacion
  console.log(this.cons[0], this.categoria);
  const questionAll = this.preguntasMMSE.every(p => p.respuesta != null &&
p.respuesta != undefined && p.respuesta != ''); // No se acepta respuestas sin responder
  console.log('Preguntas MMSE: ' + questionAll);
  // Validar que todas las opciones sean seleccionadas
  if (!questionAll) {
    this.showErrors = true;
    return;
  } else { this.showErrors = true }
  // Sumar si el puntaje es Si
  let respuestaNo = 0;
  for (const pregunta of this.preguntasMMSE) {
    if (pregunta.respuesta == 'no') {
      respuestaNo++;
    }
  }
  // Cálculo de observacion segun el puntaje
  if (respuestaNo >= 8) {
    this.observacion = err + 'Importante, deterioro Cognitivo';
  } else if (respuestaNo >= 5 && respuestaNo <= 7) {
    this.observacion = err + 'Moderado, Deterioro cognitivo, patológico';
  } else if (respuestaNo <= 2) {
    this.observacion = err + 'Normal, Sin deterioro'
  } else {
    this.observacion = err + 'Leve, Deterioro cognitivo'
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
}

    this.encuestaResultado(respuestaNo, 'Question Pfeiffer', this.observacion,
(respuestaNo/10)*100); //Envío de los parametros
    console.log(this.cons[1], 'spmsqp');
    break;
case 'reloj':
    this.showErrors = true;
    console.log(this.cons[0], this.categoria);
    if(this.questionnaireForm.valid){
        this.puntos = 0; //Reiniciar los puntos

        // Iterar sobre las claves del formulario
        for(const key in this.questionnaireForm.controls){
            if(this.questionnaireForm.controls[key].value != null &&
this.questionnaireForm.controls[key].value != ''){
                this.puntos += this.questionnaireForm.controls[key].value;
            }
        }
        this.showErrors = false;
    }

    if(this.puntos >= 9){
        this.observacion = 'NORMAL';
    } else if(this.puntos == 8){
        this.observacion = 'DEFICIT LIMITE';
    } else if(this.puntos == 6 || this.puntos == 7){
        this.observacion = 'DEFICIT LEVE';
    } else if(this.puntos == 4 || this.puntos == 5){
        this.observacion = 'DEFICIT MODERADO';
    } else {
        this.observacion = 'DEFICIT SEVERO';
    }

    //Control de Envio si esta respondido la encuesta
    if(!this.showErrors){
        console.log('Entrada If del Reloj');
        this.showErrors = true; //Convertirse en verdadero antes de llamar al metodo,
para enviar al siguiente componente
        this.encuestaResultado(this.puntos, 'Prueba de Reloj',this.observacion,
(this.puntos/10)*100);
    }
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
        console.log(this.cons[1], 'Prueba Reloj');
        break;
    }
}

/* ALGUNOS METODOS POR CADA ENCUESTA A REALIZAR */

//Envia los parametros al Componente Resultado segun reciba
encuestaResultado(puntos: number, nameEncuesta: String, observacion: String,
porcentaje: FLOAT){
    if(this.showErrors){
        // Al navegar, enviamos los puntos al componente de resultado
        this.router.navigate(['home/resultado'], {queryParams: {
            puntaje: puntos, //Puntaje Obtenido
            nameEncuesta: nameEncuesta, //Nombre de la Encuesta
            porcentaje: porcentaje.toFixed(2),
            observacion: observacion //Observaciones
        }});
    }
}

// Función para alternar el color
toggleTextColor(textKey: string): void {
    this.isSelected[textKey] = !this.isSelected[textKey];

    //Al estar en color se suma 1 punto, en caso contrario 0
    if (this.isSelected[textKey]) {
        this.puntos += 1; // Si se selecciona, suma un punto
    } else {
        this.puntos -= 1; // Si se deselecciona, resta un punto
    }
}

// Método para verificar que la respuesta de la pregunta este seleccionado
hasError(fieldName: string): boolean {
    const control = this.questionnaireForm.get(fieldName);
    return !!control?.invalid && (control?.touched || this.showErrors);
}
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

Encuestas Esfera Funcional (HTML y TypeScript)

```
<!-- Encuesta de KATZ -->
<div class="katz containerEncuesta opcmult" *ngIf="encuestaSelect == 'katz'">
  <div class="titulo">
    <h2>Indice de KATZ</h2>
  </div>
  <div class="opciones">
    <div *ngFor="let question of preguntasKATZ; let i = index" class="question">
      <p class="preguntas">{{ question.text }}</p>
      <div *ngFor="let option of question.options">
        <label>
          <input type="radio" name="question{{i}}" [value]="option.score"
(change)="selectedOptions[i] = option.score; errors[i] = false"/>
          {{ option.label }}
        </label>
      </div>

      <!-- Mensaje de error específico para la pregunta -->
      <div *ngIf="errors[i]" class="alert alert-danger mt-2">
        Por favor, selecciona una opción para esta pregunta.
      </div>
    </div>
  </div>

<!-- Encuesta de Barthel -->
<div class="barthel containerEncuesta opcmult" *ngIf="encuestaSelect == 'barthel'">
  <div class="titulo">
    <h2>Indice de Barthel</h2>
  </div>
  <div class="opciones">
    <p class="instrucciones">Seleccione</p>
    <div *ngFor="let pregunta of preguntasBarthel; let i = index">
      <p class="preguntas">{{ pregunta.text }}</p>
      <div *ngFor="let opcion of pregunta.options">
        <label>
          <input type="radio" name="pregunta{{i}}" [value]="opcion.score"
(change)="seleccionarRespuesta(i, opcion.score)"/> <!--
[(ngModel)]="pregunta.respuestaSeleccionada"/>-->
          {{ opcion.label }}
        </label>
      </div>
    </div>
  </div>
</div>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
</div>
</div>
<div *ngIf="mensajeError != ''" class="alert alert-danger">
  {{ mensajeError }}
</div>
</div>
</div>

<!-- Encuesta de Lawton y Brody -->
<div class="lawton containerEncuesta opcmult" *ngIf="encuestaSelect == 'lawton'">
  <div class="titulo">
    <h2>Lawton y brody</h2>
  </div>
  <div class="opciones">
    <div *ngFor="let question of preguntasLowton; let i = index" class="question mb-3">
      <p class="preguntas">{{ question.text }}</p>
      <div *ngFor="let option of question.options" class="form-check">
        <label>
          <input type="radio" name="question{{i}}" [value]="option.score"
(change)="optionL[i] = option.score; errorsLowton[i] = false"/>
          {{ option.label }}
        </label>
      </div>

      <div *ngIf="errorsLowton[i]" class="alert alert-danger mt-2">
        Por favor, selecciona una opción para esta pregunta.
      </div>
    </div>
  </div>
</div>

<!-- Encuesta de Ensrud -->
<div class="ensrud containerEncuesta" *ngIf="encuestaSelect == 'ensrud'">
  <div class="titulo">
    <h2>Criterios de Ensrud</h2>
  </div>
  <div class="opciones">
    <p class="instrucciones">Seleccione la respuesta segun considere que sea posible o
que aplique segun su capacidad</p>
    <div *ngFor="let pregunta of preguntasEnsrud; let i = index">
      <label>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
<input type="checkbox" class="form-check-input"
[checked]="pregunta.seleccionada" (change)="toggleCheckbox(i, $event)"/>
  {{ pregunta.text }}
</label>
</div>
</div>
</div>
</div>

<!-- Encuesta FRAIL -->
<div class="frail containerEncuesta" *ngIf="encuestaSelect == 'frail'">
  <div class="titulo">
    <h2>FRAIL</h2>
  </div>
  <div class="opciones">
    <div class="container mt-4">
      <!-- Tabla con colores personalizados -->
      <form>
        <table class="table table-bordered">
          <thead class="table-light">
            <tr class="table-header">
              <th class="text-center">Observación</th>
              <th class="text-center">Puntuación</th>
            </tr>
          </thead>
          <tbody>
            <!-- FATIGA -->
            <tr class="row-blue">
              <td>
                <strong>[Fatigue (fatiga)]</strong><br />
                En las últimas 4 semanas, ¿Qué tanto tiempo se sintió?
              </td>
              <td>
                <div class="form-check">
                  <input type="radio" id="todoTiempo" name="fatigue" value="Todo el
tiempo" class="form-check-input" (change)="onAnswerChange('fatigue', 1)"/>
                  <label for="todoTiempo" class="form-check-label">Todo el
tiempo</label>
                </div>
                <div class="form-check">
                  <input type="radio" id="mayorParte" name="fatigue" value="La mayor
parte del tiempo" class="form-check-input" (change)="onAnswerChange('fatigue', 1)"/>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
<label for="mayorParte" class="form-check-label">La mayor parte del
tiempo</label>
</div>
<div class="form-check">
  <input type="radio" id="algoTiempo" name="fatigue" value="Algo de
tiempo" class="form-check-input" (change)="onAnswerChange('fatigue', 0)" />
  <label for="algoTiempo" class="form-check-label">Algo de
tiempo</label>
</div>
<div class="form-check">
  <input type="radio" id="muyPoco" name="fatigue" value="Muy poco
tiempo" class="form-check-input" (change)="onAnswerChange('fatigue', 0)" />
  <label for="muyPoco" class="form-check-label">Muy poco tiempo</label>
</div>
<div class="form-check">
  <input type="radio" id="nadaTiempo" name="fatigue" value="Nada de
tiempo" class="form-check-input" (change)="onAnswerChange('fatigue', 0)" />
  <label for="nadaTiempo" class="form-check-label">Nada de
tiempo</label>
</div>
</td>
</tr>

<!-- RESISTENCIA -->
<tr class="row-gray">
  <td>
    <strong>[Resistance (resistencia)]</strong><br />
    ¿Tiene dificultad para subir 10 escalones (una escalera)?
  </td>
  <td>
    <div class="form-check">
      <input type="radio" id="resistanceSi" name="resistance" value="1"
class="form-check-input" (change)="onAnswerChange('resistance', 1)"/>
      <label for="resistanceSi" class="form-check-label">Sí</label>
    </div>
    <div class="form-check">
      <input type="radio" id="resistanceNo" name="resistance" value="0"
class="form-check-input" (change)="onAnswerChange('resistance', 0)"/>
      <label for="resistanceNo" class="form-check-label">No</label>
    </div>
  </td>
</tr>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
<!-- ACTIVIDAD -->
<tr class="row-blue">
  <td>
    <strong>[Aerobic (actividad aeróbica)]</strong><br />
    ¿Tiene dificultad para caminar 100 metros (dos cuerdas) sin descansar?
  </td>
  <td>
    <div class="form-check">
      <input type="radio" id="aerobicSi" name="aerobic" value="1"
class="form-check-input" (change)="onAnswerChange('aerobic', 1)"/>
      <label for="aerobicSi" class="form-check-label">Sí</label>
    </div>
    <div class="form-check">
      <input type="radio" id="aerobicNo" name="aerobic" value="0"
class="form-check-input" (change)="onAnswerChange('aerobic', 0)"/>
      <label for="aerobicNo" class="form-check-label">No</label>
    </div>
  </td>
</tr>

<!-- ENFERMEDADES -->
<tr class="row-gray">
  <td>
    <strong>[Illnesses (enfermedades)]</strong><br />
    ¿Algún doctor o médico le ha comentado que tiene [mencionar la
enfermedad]?
  </td>
  <td>
    <div class="form-check">
      <input type="radio" id="illnessesSi" name="illnesses" value="1"
class="form-check-input" (change)="onAnswerChange('illnesses', 1)"/>
      <label for="illnessesSi" class="form-check-label">Sí</label>
    </div>
    <div class="form-check">
      <input type="radio" id="illnessesNo" name="illnesses" value="0"
class="form-check-input" (change)="onAnswerChange('illnesses', 0)"/>
      <label for="illnessesNo" class="form-check-label">No</label>
    </div>
  </td>
</tr>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
<!-- PESO -->
<tr class="row-blue">
  <td>
    <strong>[Lost of weight (pérdida de peso)]</strong><br/>
    El paciente no debe portar los zapatos al momento de ser pesado
  </td>
  <td>
    <label for="pesoHaceUnAnio" class="form-label">&#191;Cuánto pesaba hace
un a&#241;o&#63;</label>
    <input type="text" id="pesoHaceUnAnio" class="form-control mb-2"
placeholder="Peso hace un año" maxLength="4" (input)="validateInput($event,
'pesoYear')"/>
    <!-- Mensaje de error -->
    <div *ngIf="pesoHaceUnAnioError" class="text-danger mt-1">
      {{ pesoHaceUnAnioError }}
    </div>
    <label for="pesoActual" class="form-label">&#191;Cuánto pesa
ahora&#63;</label>
    <input type="text" id="pesoActual" class="form-control"
placeholder="Peso actual" maxLength="4" (input)="validateInput($event, 'pesoActual')"/>
    <!-- Mensaje de error -->
    <div *ngIf="pesoActualError" class="text-danger mt-1">
      {{ pesoActualError }}
    </div>
  </td>
</tr>
</tbody>
</table>
<div *ngIf="mensajeError" class="alert alert-danger mt-3">
  {{ mensajeError }}
</div>
</form>
</div>

</div>
</div>

<!-- Time Up and Go -->
<div class="cronometro containerEncuesta" *ngIf="encuestaSelect == 'timeup'">
  <div class="titulo">
    <h2>Time UP and DOWN</h2>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
</div>
<div class="instrucciones">
  <p>La persona puede usar su calzado habitual y puede utilizar cualquier dispositivo
  de ayuda que normalmente usa.</p>
  <ul>
    <li>Indicarle a la persona mayor, sentarse en la silla con la espalda apoyada en
    el respaldo.</li>
    <li>Pídale a la persona que se levante de la silla, camine a paso normal una
    distancia de 3 metros, haga que la persona de la vuelta camine nuevamente hacia la
    silla y se vuelva a sentar.</li>
    <li>Mida el tiempo en que la persona mayor realiza la prueba. El cronometraje
    comienza cuando la persona comienza a levantarse de la silla y termina cuando regresa a
    la silla y se sienta.</li>
    <li>Dar un intento de prueba.</li>
  </ul>
  <p>El cronometraje comienza cuando la persona comienza a levantarse de la silla y
  termina cuando regresa a la silla y se sienta.</p>
  <p>La persona debe de dar un intento de práctica y luego repite 3 intentos más. Se
  promedian los tres ensayos reales y se promedian.</p>
</div>
<div class="opciones">
  <h1>{{ tiempo | date:'mm:ss.SSS':'UTC' }}</h1>
  <button (click)="iniciarPausar()" class="btn btn-primary">
    {{ corriendo ? 'Pause' : 'Start' }}
  </button>
  <button (click)="reiniciar()" class="btn btn-primary">Reset</button>
  <button (click)="capturarTiempo()" [disabled]="!corriendo" class="btn btn-
  primary">Time Capture</button>
  <button (click)="deleteScore()" class="btn btn-primary">Default</button>

  <div *ngIf="capturas.length">
    <h3>Tiempos Capturados:</h3>
    <ul>
      <li *ngFor="let captura of capturas; let i = index">
        Captura {{ i + 1 }}: {{ captura }}
      </li>
    </ul>
  </div>
</div>

</div>
<div class="boton">
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
<button type="submit" class="btn btn-success"
(click)="finalizarEncuesta()">Finalizar</button>
</div>

import { Component, ElementRef, Input, OnInit, Renderer2, ViewChild } from
'@angular/core';
import { FormGroup } from '@angular/forms';
import { ResultadoService } from 'src/app/Service/resultado.service';
import { FormBuilder } from '@angular/forms';
import { FormControl } from '@angular/forms';
import { ActivatedRoute, Router } from '@angular/router';
import { DiagnosticComponent } from '../../diagnostic/diagnostic.component';
import { ApiService } from 'src/app/Service/api.service';
import { Checkbox, Question } from 'src/app/Shared/Data';
import { float, FLOAT } from 'html2canvas/dist/types/css/property-descriptors/float';

@Component({
  selector: 'app-category1',
  templateUrl: './category1.component.html',
  styleUrls: ['./category1.component.scss'],
})
export class Category1Component implements OnInit {
  encuestaSelect: String = '';
  puntos: number = 0;
  observacion: string = '';
  showErrors: boolean = false;
  categoria?: String;
  mensajeError: string = '';
  ent: String [] = ['Entrada de ', 'Salida de ']

  // Propiedades para los valores de peso
  peso1: FLOAT = 0.0; // Year
  peso2: FLOAT = 0.0; // Actual
  pesoYear: string = '';
  pesoActual: string = '';
  // Mensajes de error
  pesoHaceUnAnioError: string = '';
  pesoActualError: string = '';
  //Validacion de Numero
  pesoIn: boolean = false; pesoIn2: boolean = false;

  /* Preguntas y Opciones de BARTHEL */
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
preguntasBarthel: Question[] = [  
  { text: 'Alimentación',  
    options: [  
      { label: 'Independiente: Capaz de utilizar cualquier instrumento, come en  
tiempo, puede ser servida o cocinada por otra persona', score: 10 },  
      { label: 'Ayuda: Necesita ayuda pero capaz de comer solo', score: 5 },  
      { label: 'Dependiente: Depende de otra persona para comer', score: 0 }  
    ]  
  },  
  { text: 'Bañarse/Ducharse',  
    options: [  
      { label: 'Independiente: Entra y sale solo del baño, sin supervisión', score: 5  
},  
      { label: 'Dependiente: Necesita ayuda o supervisión', score: 0 }  
    ]  
  },  
  { text: 'Vestirse',  
    options: [  
      { label: 'Independiente: Capaz de ponerse y de quitarse la ropa, abotonarse,  
atarse los zapatos' , score: 10 },  
      { label: 'Ayuda: Necesita de personal, pero al menos puede realiza las tareas  
con tiempo razonable' , score: 5},  
      { label: 'Dependiente: Necesita ayuda para algunas actividades', score: 0 }  
    ]  
  },  
  { text: 'Aseo Personal',  
    options: [  
      { label: 'Independiente: Realiza todas las actividades sin personal' , score: 5  
},  
      { label: 'Dependiente: Requiere de personal para asearse en algunas  
actividades', score: 0 }  
    ]  
  },  
  { text: 'Control de Orina',  
    options: [  
      { label: 'Continente: No presenta inconsistencia, capaz de atender solo su  
cuidado', score: 10 },  
      { label: 'Incontinencia Ocasional: Como máximo incontinencia en 24hrs, requiere  
ayuda para el cuidado', score: 5 },  
      { label: 'Incontinencia: Episodios de incontinencia con frecuencia una vez en  
24hrs, incapaz de manejar su cuidado', score: 0 }  
    ]  
  }  
]
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
},
{ text: 'Control de Heces',
  options: [
    { label: 'Continencia Normal: No presenta episodios, capaz de administrarse solo', score: 10 },
    { label: 'Ocasional: Ocasionalmente una vez por semana, necesita ayuda por supositorios', score: 5 },
    { label: 'Incontinencia: Más de un episodio por semana', score: 0 }
  ]
},
{ text: 'Uso del Retrete (TAZA)',
  options: [
    { label: 'Independiente: Usa el retrete por si mismo sin ayuda', score: 10 },
    { label: 'Ayuda: Requiere de personal oara mantener el equilibrio sentado, limpiarse o ponerse de pie', score: 5 },
    { label: 'Dependiente: Requiere totalmente de ayuda personal para ir la baño', score: 0 }
  ]
},
{ text: 'Traslado Sillón-Cama',
  options: [
    { label: 'Independiente: Puede ir del sillón a la cama sin personal supervisor', score: 15 },
    { label: 'Mínima ayuda: Requiere supervisión o pequeña ayuda para traslado', score: 10 },
    { label: 'Gran Ayuda: Requiere de ayuda para traslado, capaz de permanecer sentado sin ayuda', score: 5 },
    { label: 'Dependiente: Requiere de 2 personas o gura para traslado, incapaz de sentrase solo', score: 0 }
  ]
}, { text: 'Dezplazamiento',
  options: [
    { label: 'Independiente: Puede caminar 50 metros o equivalente sin supervisor', score: 15 },
    { label: 'Ayuda: Camina 50m, pero necesita personal (física o verbal) o suvervisor', score: 10 },
    { label: 'Independiente en silla de ruedas: Propulsa su silla 50m sin ayuda o supervisión', score: 5 },
    { label: 'Dependiente: No puede caminar solo o propulsar su silla', score: 0 }
  ]
},
{ text: 'Escalones',
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
options: [  
  { label: 'Independiente: Puede bajar y subir escaleras sin supervisión' ,  
score: 10 },  
  { label: 'Ayuda: Requiere ayuda física o supervisión para subir o bajar',  
score: 5 },  
  { label: 'Dependiente: Incapaz de subir y bajar escaleras, requiere asesor',  
score: 0 }  
],  
}  
]  
respuestas: number[] = new Array(this.preguntasBarthel.length).fill(-1); // Array  
para almacenar respuestas (-1 indica no respondida)  
  
/* Preguntas ENSRUD */  
preguntasEnsrud: Checkbox[] = [  
  { text: '1.- Pérdida de peso de 5% o mayor en los últimos 30 días', seleccionada:  
false },  
  { text: '2.- Inhabilitado para levantarse de una silla 5 veces sin emplear los  
brazos', seleccionada: false },  
  { text: '3.- Pobre energía identificado con una respuesta negativa: "¿Siente usted  
con energía?"', seleccionada: false },  
];  
// Respuestas de las preguntas Ensrud  
answers: {[key: string]: number | null} = {  
  resistance: null,  
  aerobic: null,  
  illnesses: null,  
  fatigue: null  
};  
// Mensaje de error por campo  
fieldErrors: { [key: string]: string } = {};  
  
/* CRONOMETRO */  
tiempo: number = 0; // Tiempo en milisegundos  
intervalId: any;  
capturas: string[] = []; // Lista para guardar tiempos capturados  
corriendo: boolean = false;  
  
//Encuesta KATZ  
preguntasKATZ: Question[] = [  
  { text: 'BAÑO (Esponja, regadera o tina)',  
options: [
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
{ label: 'No recibe asistencia (puede entrar y salir de la tina u otra forma
de baño).', score: 1 },
{ label: 'Que reciba asistencia durante el baño en una sola parte del
cuerpo', score: 1 },
{ label: 'Dependiente: Que reciba asistencia durante el baño en más de una
parte.', score: 0 }
],
},
{ text: 'VESTIDO',
  options: [
    { label: 'Que pueda tomar las prendas y vestirse completamente, sin
asistencia.', score: 1 },
    { label: 'Que pueda tomar las prendas y vestirse sin asistencia excepto en
abrocharse los zapatos.', score: 1 },
    { label: 'Que reciba asistencia para tomar las prendas y vestirse.', score: 0
}
]
},
{ text: 'USO DEL SANITARIO',
  options: [
    { label: 'Sin ninguna asistencia (puede utilizar objeto de soporte: bastón o
silla de ruedas y/o puede arreglar su ropa o el uso de pañal o cómodo).', score: 1 },
    { label: 'Que reciba asistencia al ir al baño, en limpiarse y que pueda
manejar por si mismo/a el pañal o cómodo vaciándolo.', score: 1 },
    { label: 'Que no vaya al baño por si mismo/a.', score: 0 }
]
},
{ text: 'TRANSFERENCIAS',
  options: [
    { label: 'Que se mueva dentro y fuera de la cama y silla sin ninguna
asistencia (utiliza un auxiliar de la marcha u objeto de soporte).', score: 1 },
    { label: 'Que pueda moverse dentro y fuera de la cama y silla con
asistencia.', score: 1 },
    { label: 'Que no pueda salir de la cama.', score: 0 }
]
},
{ text: 'CONTINENCIA',
  options: [
    { label: 'Control total de esfínteres.', score: 1 },
    { label: 'Que tenga accidentes ocasionales que no afectan su vida social.',
score: 1 },
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
        { label: 'Necesita ayuda para supervisión del control de esfínteres, utiliza sonda o es incontinente.', score: 0 }
      ]
    },
    { text: 'ALIMENTACIÓN',
      options: [
        { label: 'Que se alimente por si solo sin asistencia alguna.', score: 1 },
        { label: 'Que se alimente solo y que tenga asistencia sólo para cortar la carne o untar mantequilla. ', score: 1 },
        { label: 'Que reciba asistencia en la alimentación o que se alimente parcial o totalmente por vía enteral o parenteral.', score: 0 }
      ]
    },
  ];
  selectedOptions: number[] = Array(this.preguntasKATZ.length).fill(-1);
  errors: boolean[] = Array(this.preguntasKATZ.length).fill(false); // Array para rastrear errores por pregunta

  //Encuesta Brody
  preguntasLowton: Question[] = [
    {
      text: 'CAPACIDAD PARA USAR TELÉFONO',
      options: [
        { label: 'Lo opera por iniciativa propia, lo marca sin problemas.', score: 1 },
        { label: 'Marca sólo unos cuantos números bien conocidos.', score: 1 },
        { label: 'Contesta el teléfono pero no llama.', score: 1 },
        { label: 'No usa el teléfono.', score: 0 }
      ]
    },
    {
      text: 'COMPRAS',
      options: [
        { label: 'Vigila sus necesidades independientemente.', score: 1 },
        { label: 'Hace independientemente sólo pequeñas compras.', score: 0 },
        { label: 'Necesita compañía para cualquier compra.', score: 0 },
        { label: 'Incapaz de cualquier compra.', score: 0 }
      ]
    },
    {
      text: 'COCINA',
      options: [
        { label: 'Planea, prepara y sirve los alimentos correctamente.', score: 1 },
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
{ label: 'Prepara los alimentos sólo si se le provee lo necesario.', score: 0 },
{ label: 'Calienta, sirve y prepara pero no lleva una dieta adecuada.', score: 0 },
{ label: 'Necesita que le preparen los alimentos.', score: 0 }
],
{
  text: 'CUIDADO DEL HOGAR',
  options: [
    { label: 'Mantiene la casa solo o con ayuda mínima.', score: 1 },
    { label: 'Efectúa diariamente trabajo ligero eficientemente.', score: 1 },
    { label: 'Efectúa diariamente trabajo ligero sin eficiencia.', score: 1 },
    { label: 'Necesita ayuda en todas las actividades.', score: 0 },
    { label: 'No participa.', score: 0 }
  ]
},
{
  text: 'LAVANDERÍA',
  options: [
    { label: 'Se ocupa de su ropa independientemente.', score: 1 },
    { label: 'Lava sólo pequeñas cosas.', score: 1 },
    { label: 'Todos se lo tienen que lavar.', score: 0 }
  ]
},
{
  text: 'TRANSPORTE',
  options: [
    { label: 'Se transporta solo/a.', score: 1 },
    { label: 'Se transporta solo/a, únicamente en taxi pero no puede usar otros recursos.', score: 1 },
    { label: 'Viaja en transporte colectivo acompañado.', score: 1 },
    { label: 'Viaja en taxi o auto acompañado.', score: 0 },
    { label: 'No sale.', score: 0 }
  ]
},
{
  text: 'MEDICACIÓN',
  options: [
    { label: 'Es capaz de tomarla a su hora y dosis correctas.', score: 1 },
    { label: 'Se hace responsable sólo si le preparan por adelantado.', score: 0 },
    { label: 'Es incapaz de hacerse cargo.', score: 0 }
  ]
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
]
},
{
  text: 'FINANZAS',
  options: [
    { label: 'Maneja sus asuntos independientemente.', score: 1 },
    { label: 'Sólo puede manejar lo necesario para pequeñas compras.', score: 1 },
    { label: 'Es incapaz de manejar dinero.', score: 0 }
  ]
}
];
optionL: number[] = Array(this.preguntasLowton.length).fill(-1);
errorsLowton: boolean[] = Array(this.preguntasLowton.length).fill(false);

/* ----- CONSTRUC DE LA CLASE ----- */
*/
constructor(private fb: FormBuilder, private encuestaService: ApiService, private
router: Router, private route: ActivatedRoute, private render: Renderer2) {

}
/* ----- ONINIT = INICIALIZADOR ----- */
- */
ngOnInit(): void {
  this.categoria = this.route.snapshot.paramMap.get('categoria')!;
  this.encuestaService.selectEncuest$.subscribe(cat => {
    this.encuestaSelect = cat;
  });
}

// En el componente 'Category1Component'
finalizarEncuesta() {
  this.encuesta(this.encuestaSelect);
}

cambiarCategoria(categoria: String){
  this.encuestaSelect = categoria.toString();
}

encuesta(enc: String){
  switch(enc){
    case 'katz':
      console.log(this.ent[0], this.encuestaSelect);
    }
  }
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
//VERIFICAR SI LAS OPCIONES ESTEN SELECCIONADAS
this.errors = this.selectedOptions.map(score => score == -1);

let katzT = 0;
for (let i = 0; i < this.selectedOptions.length; i++) {
  katzT += this.selectedOptions[i];
}
if(!this.errors.includes(true)){
  const letter = this.getIndependenceCategory();
  this.puntos = katzT;//Pasa los puntos obtenido al puntos
  this.showErrors = true;
  this.encuestaResultado(this.puntos, 'Indice de KATZ', 'Letra Asignada: ' +
letter,(this.puntos/6)*100)
  console.log(this.ent[1] + 'KATZ');
}
break;
case 'barthel':
  console.log(this.ent[0], this.encuestaSelect);
  // Verifica si todas las preguntas tienen respuesta seleccionada
  for (let i = 0; i < this.respuestas.length; i++) {
    if (this.respuestas[i] == -1) {
      this.mensajeError = 'Debe responder todas las preguntas antes de calcular
el puntaje.';
      this.puntos = 0;
      return;
    } else {
      this.showErrors = true
      this.mensajeError = '';} //En caso de que se respondas todos, desbloquea el
metodo encuestaResultado
    }
  // Calcula el puntaje total sumando los valores seleccionados
  let total = 0;
  for (let i = 0; i < this.respuestas.length; i++) {
    total += this.respuestas[i];
  }
  this.puntos = total;//Pasa los puntos obtenido al puntos

  if(this.puntos <= 20){
    this.observacion = 'Dependencia Total';
  } else if(this.puntos >= 21 && this.puntos <= 60){
    this.observacion = 'Dependencia grave';
  } else if(this.puntos >= 61 && this.puntos <= 90){
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
this.observacion = 'Dependencia Moderada';
} else if( this.puntos >= 91 && this.puntos <= 99){
  this.observacion = 'Dependencia Escasa';
} else {
  this.observacion = 'Independiente';
}

this.encuestaResultado(this.puntos, 'Indice de Barthel', this.observacion,
(this.puntos/100)*100);
console.log(this.ent[1] + 'Barthel');
break;
case 'lawton':
  console.log(this.ent[0], this.encuestaSelect);
  this.errorsLowton = this.optionL.map(score => score == -1);

  let lowtonT = 0;
  for (let i = 0; i < this.optionL.length; i++) {
    lowtonT += this.optionL[i];
  }
  this.puntos = lowtonT;

  if(this.puntos == 8){
    this.observacion = 'Muy activos: actividades instrumentales completas';
  } else if(this.puntos >= 5 && this.puntos <= 7){
    this.observacion = 'Activos: actividades limitadas';
  } else if(this.puntos <= 0){
    this.observacion = 'Inactivos: no realizan actividades instrumentales'
  } else {
    this.observacion = 'Poco activos: limitación del 50 % o más de esas
actividades';
  }

  if(!this.errorsLowton.includes(true)){
    this.showErrors = true;
    this.encuestaResultado(this.puntos, 'Escala Lowton y Brody', this.observacion,
(this.puntos/8)*100);
  }

  console.log(this.ent[1] + 'Brody y Lowton');
  break;
case 'frail':
  console.log('Entrada de: FRAIL');
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
let allAnswered = true;
// Validar cada pregunta
for (const question in this.answers) {
  if (this.answers[question] === null) {
    this.fieldErrors[question] = 'Por favor, seleccione una opción.';
    allAnswered = false;
  } else {
    this.fieldErrors[question] = ''; // Limpia errores si hay respuesta
  }
}

if(allAnswered){
  this.mensajeError = '';
  //Calculo del Peso
  if(this.pesoIn && this.pesoIn2){
    this.peso1 = parseFloat(this.pesoYear); //Year Anterior
    this.peso2 = parseFloat(this.pesoActual); //Actual
  } else { this.peso1 = 0.0; this.peso2 = 0.0}
  const value = (((this.peso1 - this.peso2)/this.peso1)*100).toFixed(2);
  //Convertir para validar en if
  if(parseFloat(value) >= 5){
    // Cuando el peso cumple la condicion pasaa ser 1
    this.puntos += 1;
  }
  //Calcular Puntaje
  for (const question in this.answers) {
    if (this.answers[question] == 1) {
      this.puntos += 1; // Suma 1 por cada respuesta positiva
    }
  }
  this.showErrors = true;

} else {
  this.mensajeError = 'Por favor, responda todas las preguntas.';
}

if(this.puntos >= 5){
  this.observacion = 'Probable Fragilidad';
} else if(this.puntos <= 0){
  this.observacion = 'Sin fragilidad o robuztez';
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
    } else {
      this.observacion = 'Probable pre-fatiga'
    }

    this.encuestaResultado(this.puntos, 'FRAIL', this.observacion,
(this.puntos/5)*100);

    console.log("Salida de Frail | ", this.puntos);
    break;
  case 'ensrud':
    console.log('Entrada de: Ensrud');
    this.puntos = this.preguntasEnsrud.filter(p => p.seleccionada).length;
    this.showErrors = true;
    if(this.puntos >= 2){
      this.observacion = 'Estado Fragil';
    } else if(this.puntos == 1){
      this.observacion = 'Estado Prefragil'
    } else {
      this.observacion = 'Estado Ningun criterio (Robusto)';
    }
    console.log('Salida de: Ensrud');
    this.encuestaResultado(this.puntos, 'Criterio Ensrud', this.observacion,
(this.puntos*3)/100);
    break;
  }
}

//Metodo para pasar al siguiente componente, en case de que este respondidos las
respuestas
encuestaResultado(puntos: number, nameEncuesta: String, observacion: String,
porcentaje: FLOAT){
  if(this.showErrors){
    // Al navegar, enviamos los puntos al componente de resultado
    this.router.navigate(['home/resultado'], {queryParams: {
      puntaje: puntos, //Puntaje Obtenido
      nameEncuesta: nameEncuesta, //Nombre de la Encuesta
      porcentaje: porcentaje.toFixed(2),
      observacion: observacion //Observaciones
    }}});
    localStorage.removeItem('subCatSeleccionada');
  }
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
//Seleccionar respuesta de Opcion Multiple (BARTHEL)
seleccionarRespuesta(index: number, score: number): void {
  this.respuestas[index] = score;
}
//Si esta seleccionado el checkbox se suma 1 punto
toggleCheckbox(index: number, event: any): void {
  this.preguntasEnsrud[index].seleccionada = event.target.checked;
}
//Valida la entrada del usuario para que solo acepte números y un punto decimal.
validateInput(event: Event, field: string): void {
  const input = event.target as HTMLInputElement;
  let value = input.value;
  // Mensaje de error por defecto
  let errorMessage = '';

  // Permitir solo números y un punto decimal
  if (!/^d*\.\d*$/.test(value)) {
    errorMessage = 'Solo se permiten números y un punto decimal.';
    value = value.replace(/[^0-9.]/g, ''); // Eliminar caracteres no válidos
  }

  // Permitir solo un punto decimal
  const parts = value.split('.');
  if (parts.length > 2) {
    value = parts[0] + '.' + parts[1]; // Mantener solo el primer punto decimal
  }

  // Limitar a 4 caracteres como máximo
  if (value.length > 4) {
    value = value.slice(0, 4);
  }

  // Actualizar el valor en el modelo y muestra message de error
  if (field == 'pesoYear') {
    this.pesoYear = value;
    this.pesoHaceUnAnioError = errorMessage;
    this.pesoIn = true;
  } else if (field == 'pesoActual') {
    this.pesoActual = value;
    this.pesoActualError = errorMessage;
    this.pesoIn2 = true;
  }
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
}  
onAnswerChange(question: string, value: number): void {  
  this.answers[question] = value;  
  this.fieldErrors[question] = ''; // Limpia errores si selecciona respuesta  
  this.mensajeError = '';  
}  
//FUNCION CRONOMETRO  
iniciarPausar(): void {  
  if (this.corriendo) {  
    clearInterval(this.intervalId);  
  } else {  
    this.intervalId = setInterval(() => {  
      this.tiempo += 10; // Incrementa cada 10ms  
    }, 10);  
  }  
  this.corriendo = !this.corriendo;  
}  
  
reiniciar(){  
  clearInterval(this.intervalId);  
  this.tiempo = 0;  
  //this.capturas = [];  
  this.corriendo = false;  
}  
capturarTiempo(){  
  const minutos = Math.floor(this.tiempo / 60000);  
  const segundos = Math.floor((this.tiempo % 60000) / 1000);  
  const milisegundos = Math.floor((this.tiempo % 1000) / 10);  
  
  this.capturas.push(`${this.pad(minutos)}:${this.pad(segundos)}:${this.pad(milisegun  
dos)}`);  
}  
private pad(num: number): string {  
  return num < 10 ? `0${num}` : `${num}`;  
}  
deleteScore(){  
  this.capturas = [];  
  this.reiniciar();  
}  
  
//Funcion KATZ  
getIndependenceCategory(): string {
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
// Define los niveles de independencia/dependencia para cada pregunta
const independence = this.selectedOptions.map(score => score == 1);
const dependenceCount = independence.filter(dep => !dep).length;

if (dependenceCount == 0) {
  return 'A'; // Independencia total
}

if (dependenceCount == 1) {
  return 'B'; // Independencia en todas menos una
}

const dependentActivities = {
  baño: !independence[0],
  vestido: !independence[1],
  sanitario: !independence[2],
  transferencias: !independence[3],
  continencia: !independence[4],
  alimentación: !independence[5]
};

if (dependentActivities.baño && dependenceCount == 2) {
  return 'C';
}

if (dependentActivities.baño && dependentActivities.vestido && dependenceCount ==
3) {
  return 'D';
}

if (dependentActivities.baño && dependentActivities.vestido &&
dependentActivities.sanitario && dependenceCount == 4) {
  return 'E';
}

if (dependentActivities.baño && dependentActivities.vestido &&
dependentActivities.sanitario && dependentActivities.transferencias && dependenceCount
== 5) {
  return 'F';
}

if (dependenceCount == 6) {
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
    return 'G'; // Dependencia total
  }

  if (dependenceCount == 2) {
    return 'H'; // Dependencia en dos actividades no clasificadas en C, D, E, F
  }

  return 'Error'; // Caso no contemplado
}
}
```

Encuestas Esfera Nutricional (HTML y TypeScript)

```
<h1 class="titulo-encuesta">Mini Nutritional Assessment</h1>

<form [formGroup]="mnaForm" (ngSubmit)="enviarResultados()" class="container encuesta-
container">
  <!-- Sección de Cribaje -->
  <h2 class="subtitulo">Cribaje</h2>
  <table class="table table-striped">
    <thead>
      <tr>
        <th>Pregunta</th>
        <th>Opciones</th>
      </tr>
    </thead>
    <tbody>
      <tr *ngFor="let pregunta of cribajePreguntas; let i = index">
        <td>
          <label>{{ pregunta.texto }}</label>
        </td>
        <td>
          <div *ngFor="let opcion of pregunta.opciones" class="form-check">
            <input
              type="radio"
              class="form-check-input"
              [formControlName]='pregunta' + i"
              [value]="opcion.valor"
              id="cribajePregunta{{ i }}-opcion{{ opcion.valor }}"
            />
            <label
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
        class="form-check-label"
        for="cribajePregunta{{ i }}-opcion{{ opcion.valor }}"
    >
        {{ opcion.texto }}
    </label>
</div>
</td>
</tr>
</tbody>
</table>

<!-- Sección de Evaluación -->
<h2 class="subtitulo">Evaluación</h2>
<table class="table table-striped">
    <thead>
        <tr>
            <th>Pregunta</th>
            <th>Opciones</th>
        </tr>
    </thead>
    <tbody>
        <tr *ngFor="let pregunta of evaluacionPreguntas; let i = index">
            <td>
                <label>{{ pregunta.texto }}</label>
            </td>
            <td>
                <!-- Preguntas normales -->
                <ng-container *ngIf="!pregunta.subPreguntas">
                    <div *ngFor="let opcion of pregunta.opciones" class="form-check">
                        <input
                            type="radio"
                            class="form-check-input"
                            [formControlName]=" 'evaluacionPregunta' + i"
                            [value]="opcion.valor"
                            id="evaluacionPregunta{{ i }}-opcion{{ opcion.valor }}"
                        />
                        <label
                            class="form-check-label"
                            for="evaluacionPregunta{{ i }}-opcion{{ opcion.valor }}"
                        >
                            {{ opcion.texto }}
                        </label>
                    </div>
                </ng-container>
            </td>
        </tr>
    </tbody>
</table>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
</div>
</ng-container>

<!-- Pregunta "K" con subpreguntas -->
<ng-container *ngIf="pregunta.subPreguntas">
  <div formGroupName="consumePaciente">
    <div *ngFor="let subPregunta of pregunta.subPreguntas" class="form-
check">
      <input
        type="checkbox"
        class="form-check-input"
        [formControlName]="subPregunta.texto"
        id="subpregunta-{{ subPregunta.texto }}"
      />
      <label
        class="form-check-label"
        for="subpregunta-{{ subPregunta.texto }}"
      >
        {{ subPregunta.texto }}
      </label>
    </div>
  </div>
</ng-container>
</td>
</tr>
</tbody>
</table>

<!-- Botón de enviar -->
<button
  type="submit"
  [disabled]="mnaForm.invalid"
  class="btn btn-primary mt-3"
>
  Enviar Resultados
</button>
</form>

import { Component } from '@angular/core';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';
import { Router } from '@angular/router';
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
interface MNAForm {
  [key: string]: number | null;
}

@Component({
  selector: 'app-encuesta-assessment',
  templateUrl: './encuesta-assessment.component.html',
  styleUrls: ['./encuesta-assessment.component.scss']
})

export class EncuestaAssessmentComponent {
  mnaForm: FormGroup;
  cribajePreguntas = [
    {
      texto: 'A. ¿Ha perdido el apetito?',
      opciones: [
        { texto: 'Ha comido mucho menos', valor: 0 },
        { texto: 'Ha comido menos', valor: 1 },
        { texto: 'Ha comido igual', valor: 2 },
      ],
    },
    {
      texto: 'B. Pérdida reciente de peso (< 3 meses)',
      opciones: [
        { texto: 'Pérdida de peso > 3 kg', valor: 0 },
        { texto: 'No lo sabe', valor: 1 },
        { texto: 'Pérdida de peso entre 1 y 3 kg', valor: 2 },
        { texto: 'No ha habido pérdida de peso', valor: 3 },
      ],
    },
    {
      texto: 'C. Movilidad ',
      opciones: [
        { texto: 'de la cama al sillón', valor: 0 },
        { texto: 'autonomía en el interior', valor: 1 },
        { texto: 'sale del domicilio', valor: 2 },
      ],
    },
    {
      texto: 'D. Ha tenido una enfermedad aguda o situación de estrés psicológico en los últimos 3 meses',
      opciones: [
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
{ texto: 'Si', valor: 0 },
{ texto: 'No', valor: 2 },
],
},
{
  texto: 'E. Problemas neuropsicologicos',
  opciones: [
    { texto: 'Demencia o depresion grave', valor: 0 },
    { texto: 'Demencia moderada', valor: 1 },
    { texto: 'Sin problemas psicologicos', valor: 2 },
  ],
},
{
  texto: 'F. Indice de masa corporal (IMC) = Peso en Kg/(talla en m2)',
  opciones: [
    { texto: 'IMC < 19', valor: 0 },
    { texto: '19 < IMC < 21', valor: 1 },
    { texto: '21 < IMC < 23', valor: 2 },
    { texto: 'IMC > 23', valor: 3 },
  ],
},
];

evaluacionPreguntas = [
  {
    texto: 'G. ¿El paciente vive independientemente?',
    opciones: [
      { texto: 'Sí', valor: 1 },
      { texto: 'No', valor: 0 },
    ],
  },
  {
    texto: 'H. ¿Toma más de 3 medicamentos al día?',
    opciones: [
      { texto: 'Sí', valor: 0 },
      { texto: 'No', valor: 1 },
    ],
  },
  {
    texto: 'I. Ulceras o lesiones cutaneas?',
    opciones: [
      { texto: 'Sí', valor: 0 },
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
{ texto: 'No', valor: 1 },
],
},
{
  texto: 'J. Cuantas Comidas comidas toma al dia?',
  opciones: [
    { texto: '1 comida', valor: 0 },
    { texto: '2 comidas', valor: 1 },
    { texto: '3 comidas', valor: 2 },
  ],
},
{
  texto: 'K. Consume el paciente alguno de los siguientes grupos alimenticios regularmente:',
  subPreguntas: [
    { texto: 'Lácteos', valor: 0 },
    { texto: 'Huevos y legumbres', valor: 0 },
    { texto: 'Carne, pescado o aves', valor: 0 },
  ],
},
{
  texto: 'L. consume Frutas o verduras al menos 2 veces al dia ?',
  opciones: [
    { texto: 'No', valor: 0 },
    { texto: 'Si', valor: 1 },
  ],
},
{
  texto: 'M. cuantos vasos de agua u otros liquidos toma al dia? (agua, zumo, cafe, té, leche, vino, cerveza...)',
  opciones: [
    { texto: 'Menos de 3 vasos', valor: 0.0 },
    { texto: 'de 3 a 5 vasos', valor: 0.5 },
    { texto: 'mas de 5 vasos', valor: 1.0 },
  ],
},
{
  texto: 'N. Forma de alimentarse',
  opciones: [
    { texto: 'Necesita ayuda', valor: 0 },
    { texto: 'se alimenta solo con dificultad', valor: 1 },
    { texto: 'se alimenta solo sin dificultad', valor: 2 },
  ],
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
],
},
{
  texto: 'O. Se considera el paciente que esta bien nutrido?',
  opciones: [
    { texto: 'Mal nutricion grave', valor: 0 },
    { texto: 'No lo sabe o malnutricion moderada', valor: 1 },
    { texto: 'sin problemas de nutricion', valor: 2 },
  ],
},
{
  texto: 'P. En comparacion con las personas en su edad, como encuestra el paciente su salud?',
  opciones: [
    { texto: 'peor', valor: 0.0 },
    { texto: 'No lo sabe', valor: 0.5 },
    { texto: 'igual', valor: 1.0 },
    { texto: 'Mejor', valor: 2.0 },
  ],
},
{
  texto: 'Q. circunferencia branquial (CD en cm)',
  opciones: [
    { texto: 'CB < 21', valor: 0.0 },
    { texto: '21 < CB < 22', valor: 0.5 },
    { texto: 'CB > 22', valor: 1.0 },
  ],
},
{
  texto: 'R. Circunferencia de la pantorrilla (CP en cm)',
  opciones: [
    { texto: 'CP < 31', valor: 0 },
    { texto: 'CP > 31', valor: 1 },
  ],
},
];

constructor(private fb: FormBuilder, private router: Router) {
  const formControls: any = {};

  // Configurar controles para las preguntas de cribaje
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
this.cribajePreguntas.forEach((_, index) => {
  formControls[`pregunta${index}`] = [null, Validators.required];
});

// Configurar controles para las preguntas de evaluación
this.evaluacionPreguntas.forEach((pregunta, index) => {
  if (!pregunta.subPreguntas) {
    formControls[`evaluacionPregunta${index}`] = [null, Validators.required];
  }
});

// Configurar controles para las subpreguntas de "K"
const consumePacienteControls: any = {};
this.evaluacionPreguntas
  .find(p => p.texto.startsWith('K.'))?.subPreguntas
  ?.forEach(subPregunta => {
    consumePacienteControls[subPregunta.texto] = [false]; // Checkbox inicializado
    en `false`
  });

formControls['consumePaciente'] = this.fb.group({
  'Lácteos': [null, Validators.required],
  'Huevos y legumbres': [null, Validators.required],
  'Carne, pescado o aves': [null, Validators.required],
});

this.mnaForm = this.fb.group(formControls);
}

enviarResultados() {
  if (this.mnaForm.invalid) {
    alert('Completa todas las preguntas antes de enviar.');
```

```
    return;
  }

  const puntajeCribaje = this.calculateScore(this.cribajePreguntas, 'pregunta');
  const puntajeEvaluacion = this.calculateScore(this.evaluacionPreguntas,
'evaluacionPregunta');
  const puntajeK = this.calculateConsumePacienteScore();
  const puntajeTotal = puntajeCribaje + puntajeEvaluacion + puntajeK;

  let observacion;
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
if (puntajeTotal >= 24) {
  observacion = 'Estado nutricional normal';
} else if (puntajeTotal >= 17) {
  observacion = 'Riesgo de desnutrición';
} else {
  observacion = 'Desnutrición';
}

console.log('Puntaje total:', puntajeTotal, 'Resultado:', observacion);

this.router.navigate(['home/resultado'], {
  queryParams: {
    puntaje: puntajeTotal,
    observacion: observacion,
  },
});
}

// Calcular el puntaje de las preguntas normales
calculateScore(preguntas: any[], prefix: string): number {
  return preguntas.reduce((total, pregunta, index) => {
    if (pregunta.subPreguntas) return total; // Excluir preguntas con subpreguntas
    const respuesta = this.mnaForm.get(`${prefix}${index}`)?.value;
    return total + (respuesta !== null ? Number(respuesta) : 0);
  }, 0);
}

// Calcular el puntaje para la pregunta "K" (subpreguntas)
calculateConsumePacienteScore(): number {
  const consumePaciente = this.mnaForm.get('consumePaciente')?.value;

  if (!consumePaciente) return 0;

  // Contar subpreguntas seleccionadas (checkboxes marcados como `true`)
  const seleccionados = Object.values(consumePaciente).filter(value => value).length;

  if (seleccionados === 1) return 0;
  if (seleccionados === 2) return 0.5;
  if (seleccionados >= 3) return 1.0;

  return 0;
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
}  
  
<div class="encuesta-container">  
  <h1 class="encuesta-title">Mini Nutritional Assessment SF</h1>  
  <form [formGroup]="mnaForm" (ngSubmit)="enviarResultados()">  
    <!-- Pregunta 1 -->  
    <div class="form-group">  
      <label>A. ¿Ha perdido el apetito? Ha comido menos por falta de apetito, problemas  
digestivos, dificultades de masticación o deglución en los últimos 3 meses?</label>  
      <div>  
        <label><input type="radio" formControlName="question1" value="0" /> Ha comido  
mucho menos</label>  
        <label><input type="radio" formControlName="question1" value="1" /> Ha comido  
menos</label>  
        <label><input type="radio" formControlName="question1" value="2" /> Ha comido  
igual</label>  
      </div>  
    </div>  
  
    <!-- Pregunta 2 -->  
    <div class="form-group">  
      <label>B. ¿Ha perdido peso recientemente (menor a 3 meses)?</label>  
      <div>  
        <label><input type="radio" formControlName="question2" value="0" /> Pérdida de  
peso mayor a 3 kilos</label>  
        <label><input type="radio" formControlName="question2" value="1" /> No lo  
sabe</label>  
        <label><input type="radio" formControlName="question2" value="2" /> Pérdida de  
peso entre 1 y 3 kilos</label>  
        <label><input type="radio" formControlName="question2" value="3" /> No ha  
habido pérdida de peso</label>  
      </div>  
    </div>  
  
    <!-- Pregunta 3 -->  
    <div class="form-group">  
      <label>C. Movilidad</label>  
      <div>  
        <label><input type="radio" formControlName="question3" value="0" /> De la cama  
al sillón</label>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
<label><input type="radio" formControlName="question3" value="1" /> Autonomía
en el interior</label>
<label><input type="radio" formControlName="question3" value="2" /> Sale del
domicilio</label>
</div>
</div>

<!-- Pregunta 4 -->
<div class="form-group">
  <label>D. ¿Ha tenido una enfermedad aguda o situación de estrés psicológico en
los últimos 3 meses?</label>
  <div>
    <label><input type="radio" formControlName="question4" value="0" /> Sí</label>
    <label><input type="radio" formControlName="question4" value="2" /> No</label>
  </div>
</div>

<!-- Pregunta 5 -->
<div class="form-group">
  <label>E. Problemas neuropsicológicos</label>
  <div>
    <label><input type="radio" formControlName="question5" value="0" /> Demencia o
depresión grave</label>
    <label><input type="radio" formControlName="question5" value="1" /> Demencia
moderada</label>
    <label><input type="radio" formControlName="question5" value="2" /> Sin
problemas psicológicos</label>
  </div>
</div>

<!-- Pregunta 6 -->
<div class="form-group">
  <label>F1. Índice de masa corporal (IMC) = Peso en KG / (Talla en m²)</label>
  <div>
    <label><input type="radio" formControlName="question6" value="0" /> IMC &#60;
19 kg/m²</label>
    <label><input type="radio" formControlName="question6" value="1" /> IMC entre
19 y 21 kg/m²</label>
    <label><input type="radio" formControlName="question6" value="2" /> IMC entre
21 y 23 kg/m²</label>
    <label><input type="radio" formControlName="question6" value="3" /> IMC &#62;
23 kg/m²</label>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
</div>
</div>

<!-- Pregunta 7 -->
<div class="form-group">
  <label>F2. Perímetro de pantorrilla (cm)</label>
  <div>
    <label><input type="radio" formControlName="question7" value="0" /> 31
cm</label>
    <label><input type="radio" formControlName="question7" value="1" /> 31
cm</label>
  </div>
</div>

<!-- Botón de envío -->
<button type="button" class="btn btn-primary mt-3"
(click)="enviarResultados()">Finalizar y Enviar</button>
</form>
</div>

import { Component } from '@angular/core';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';
import { Router } from '@angular/router';

@Component({
  selector: 'app-encuesta-assessment-sf',
  templateUrl: './encuesta-assessment-sf.component.html',
  styleUrls: ['./encuesta-assessment-sf.component.scss']
})
export class EncuestaAssessmentSFComponent {
  mnaForm: FormGroup;
  showErrors: boolean = false;

  constructor(private fb: FormBuilder, private router: Router) {
    this.mnaForm = this.fb.group({
      question1: [null, Validators.required],
      question2: [null, Validators.required],
      question3: [null, Validators.required],
      question4: [null, Validators.required],
      question5: [null, Validators.required],
      question6: [null, Validators.required],
      question7: [null, Validators.required]
    });
  }
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
}

/**
 * Calcula el puntaje total basado en las respuestas.
 */
calculateScore(): number {
  const values = this.mnaForm.value;
  return Object.keys(values)
    .map(key => Number(values[key]))
    .reduce((total, value) => total + value, 0);
}

/**
 * Genera la observación basada en el puntaje obtenido.
 */
getObservation(puntaje: number): string {
  if (puntaje >= 12 && puntaje <= 14) {
    return 'Sin desnutrición';
  } else if (puntaje >= 8 && puntaje <= 11) {
    return 'Riesgo de desnutrición';
  } else if (puntaje >= 0 && puntaje <= 7) {
    return 'Desnutrición';
  } else {
    return 'Puntaje inválido';
  }
}

/**
 * Envía los resultados de la encuesta al componente de resultados.
 */
enviarResultados(): void {
  if (!this.mnaForm.valid) {
    alert('Completa todas las preguntas antes de enviar los resultados.');
```

`this.showErrors = true;`

`return;`

`}`

`const puntaje = this.calculateScore();`

`const observacion = this.getObservation(puntaje);`

`console.log('Puntaje: ', puntaje, '\nObservación: ', observacion);`

TECNOLÓGICO NACIONAL DE MÉXICO

```
this.router.navigate(['home/resultado'], {
  queryParams: {
    nombreEncuesta: 'Mini Nutritional Assessment SF',
    puntaje: puntaje,
    observacion: observacion,
    porcentaje: ((puntaje / 14) * 100).toFixed(2) // Calcula el porcentaje basado
en el puntaje máximo
  }
});
}
```

```
<div class="encuesta-glim">
  <h2>Criterios GLIM</h2>
  <p>La desnutrición requiere al menos 1 criterio fenotípico y 1 etiológico:</p>

  <form [formGroup]="glimForm" (ngSubmit)="enviarResultados()">
    <h3>Criterios Fenotípicos</h3>
    <div class="criterio">
      <label for="perdidaPeso">
        <strong>Pérdida de peso involuntaria:</strong> ¿Ha perdido peso en los
últimos meses?
      </label>
      <select id="perdidaPeso" formControlName="perdidaPeso" class="form-control">
        <option value="0">No</option>
        <option value="1">Sí, más del 5% en los últimos 6 meses</option>
        <option value="2">Sí, más del 10% en más de 6 meses</option>
      </select>
    </div>

    <div class="criterio">
      <label for="imc">
        <strong>Índice de masa corporal bajo:</strong> ¿Cuál es su IMC?
      </label>
      <select id="imc" formControlName="imc" class="form-control">
        <option value="0">≥20 (menor de 70 años)</option>
        <option value="1">≥22 (70 años o más)</option>
        <option value="2"><20 (menor de 70 años)</option>
        <option value="3"><22 (70 años o más)</option>
      </select>
    </div>
  </form>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
<div class="criterio">
  <label for="masaMuscular">
    <strong>Reducción de la masa muscular:</strong> ¿Ha tenido reducción de masa
muscular según técnicas validadas?
  </label>
  <select id="masaMuscular" formControlName="masaMuscular" class="form-control">
    <option value="0">No</option>
    <option value="1">Sí</option>
  </select>
</div>

<h3>Criterios Etiológicos</h3>
<div class="criterio">
  <label for="ingestaAlimentos">
    <strong>Disminución de la ingesta o asimilación de alimentos:</strong>
  </label>
  <select id="ingestaAlimentos" formControlName="ingestaAlimentos" class="form-
control">
    <option value="0">No</option>
    <option value="1">Sí, ≤50% durante 1 semana</option>
    <option value="2">Sí, ≤100% durante 2 semanas</option>
  </select>
</div>

<div class="criterio">
  <label for="cargaInflamatoria">
    <strong>Carga inflamatoria:</strong> ¿Tiene alguna condición inflamatoria
activa?
  </label>
  <select id="cargaInflamatoria" formControlName="cargaInflamatoria" class="form-
control">
    <option value="0">No</option>
    <option value="1">Sí, lesión/inflamación aguda</option>
    <option value="2">Sí, patología inflamatoria crónica</option>
  </select>
</div>

<button type="submit" [disabled]="glimForm.invalid" class="btn btn-
primary">Enviar Resultados</button>
</form>
</div>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
import { Component } from '@angular/core';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';
import { Router } from '@angular/router';

@Component({
  selector: 'app-encuesta-glim',
  templateUrl: './encuesta-glim.component.html',
  styleUrls: ['./encuesta-glim.component.scss']
})
export class EncuestaGlimComponent {
  glimForm: FormGroup;
  showErrors: boolean = false;

  constructor(private fb: FormBuilder, private router: Router) {
    this.glimForm = this.fb.group({
      perdidaPeso: [null, Validators.required],
      imc: [null, Validators.required],
      masaMuscular: [null, Validators.required],
      ingestaAlimentos: [null, Validators.required],
      cargaInflamatoria: [null, Validators.required],
    });
  }

  calculateScore(): number {
    let resul = 0;
    // Obtén el puntaje sumando los valores seleccionados de los controles
    const perdidaPeso = this.glimForm.value.perdidaPeso;
    const imc = this.glimForm.value.imc;
    const masaMuscular = this.glimForm.value.masaMuscular;
    const ingestaAlimentos = this.glimForm.value.ingestaAlimentos;
    const cargaInflamatoria = this.glimForm.value.cargaInflamatoria;

    return perdidaPeso + imc + masaMuscular + ingestaAlimentos + cargaInflamatoria;
  }

  getObservation(score: number): string {
    // Devuelve una observación según el puntaje
    if (score >= 3) {
      return 'Desnutrición diagnosticada';
    } else {
      return 'No cumple con criterios de desnutrición';
    }
  }
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
}

enviarResultados(): void {
  if (!this.glimForm.valid) {
    alert('Completa todas las preguntas antes de enviar los resultados.');
```

this.showErrors = true;

```
    return;
  }

  const puntaje = this.calculateScore();
  const observacion = this.getObservation(puntaje);

  console.log('Puntaje: ', puntaje, '\nObservación: ', observacion);

  this.router.navigate(['home/resultado'], {
    queryParams: {
      nombreEncuesta: 'Criterios GLIM',
      puntaje: puntaje,
      observacion: observacion,
      porcentaje: ((puntaje / 2) * 100).toFixed(2) // Calcula el porcentaje basado en
un puntaje máximo (puedes ajustar el máximo si es diferente)
    }
  });
}
```

```
<form [formGroup]="MustForm" (ngSubmit)="enviarFormulario()">
  <h2 class="form-title">Encuesta MUST</h2>
  <div>
    <label for="imc">IMC (Kg/m²):</label>
    <input id="imc" formControlName="imc" type="number" />
  </div>

  <div>
    <label for="perdidaPeso">Pérdida de peso (%):</label>
    <input id="perdidaPeso" formControlName="perdidaPeso" type="number" />
  </div>

  <div>
    <label for="enfermedadAguda">¿Enfermedad aguda?</label>
    <input id="enfermedadAguda" formControlName="enfermedadAguda" type="checkbox" />
  </div>
</form>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
</div>

<button type="submit" [disabled]="MustForm.invalid">Enviar</button>
</form>

import { Component, OnInit } from '@angular/core';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';
import { Router } from '@angular/router';

@Component({
  selector: 'app-encuesta-must',
  templateUrl: './encuesta-must.component.html',
  styleUrls: ['./encuesta-must.component.scss']
})
export class EncuestaMustComponent implements OnInit {
  MustForm: FormGroup = this.fb.group({ // Inicialización directa aquí
    imc: [null, Validators.required],
    perdidaPeso: [null, Validators.required],
    enfermedadAguda: [null, Validators.required],
    riesgoGlobal: [{ value: '', disabled: true }],
  });

  imcPuntuacion: number = 0;
  perdidaPesoPuntuacion: number = 0;
  enfermedadAgudaPuntuacion: number = 0;
  puntaje: number = 0;
  observacion: string = '';

  constructor(private fb: FormBuilder, private router: Router) {}

  ngOnInit(): void {
    // Ya no es necesario inicializar mustForm aquí
    this.MustForm.get('imc')?.valueChanges.subscribe(() =>
this.calcularPuntuaciones());
    this.MustForm.get('perdidaPeso')?.valueChanges.subscribe(() =>
this.calcularPuntuaciones());
    this.MustForm.get('enfermedadAguda')?.valueChanges.subscribe(() =>
this.calcularPuntuaciones());
  }

  calcularPuntuaciones(): void {
    // Asignar puntuación para el IMC
  }
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
const imc = this.MustForm.get('imc')?.value;
if (imc >= 20) {
  this.imcPuntuacion = 0;
} else if (imc >= 18.5) {
  this.imcPuntuacion = 1;
} else {
  this.imcPuntuacion = 2;
}

// Asignar puntuación para la pérdida de peso
const perdidaPeso = this.MustForm.get('perdidaPeso')?.value;
if (perdidaPeso <= 5) {
  this.perdidaPesoPuntuacion = 0;
} else if (perdidaPeso <= 10) {
  this.perdidaPesoPuntuacion = 1;
} else {
  this.perdidaPesoPuntuacion = 2;
}

// Asignar puntuación para la enfermedad aguda
const enfermedadAguda = this.MustForm.get('enfermedadAguda')?.value;
if (enfermedadAguda) {
  this.enfermedadAgudaPuntuacion = 2;
} else {
  this.enfermedadAgudaPuntuacion = 0;
}

// Calcular la puntuación total
this.puntaje = this.imcPuntuacion + this.perdidaPesoPuntuacion +
this.enfermedadAgudaPuntuacion;

// Determinar el riesgo global
if (this.puntaje <= 2) {
  this.observacion = 'Riesgo Bajo: Cuidados clínicos rutinarios';
} else if (this.puntaje === 3) {
  this.observacion = 'Riesgo Medio: Observar se recomienda hospital y cuidados
domiciliarios';
} else {
  this.observacion = 'Riesgo Alto: Tratar en domicilio, hospital o comunidad';
}

// Actualizar el valor de "riesgoGlobal" en el formulario
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
this.MustForm.get('riesgoGlobal')?.setValue(this.observacion);
}

enviarFormulario(): void {
  if (this.MustForm.invalid) {
    alert('Por favor complete todos los campos.');
```

return;

```
  }

  console.log('Formulario enviado:', this.MustForm.value);
  // Aquí podrías navegar a otra página o procesar el formulario
  this.router.navigate(['home/resultado'], {
    queryParams: { puntaje: this.puntaje, observacion: this.observacion },
  });
}
```

```
<div class="encuesta-sarcf">
  <h2>Encuesta SARC-F</h2>
  <form [formGroup]="sarcFForm">
    <table class="table table-bordered table-responsive">
      <thead>
        <tr>
          <th>Preguntas</th>
          <th>Puntaje</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td><strong>Strength (Fuerza):</strong> ¿Qué tanta dificultad tiene para
llevar o cargar 4.5 kg?</td>
          <td>
            <select formControlName="strength" class="form-control">
              <option value="0">Ninguna</option>
              <option value="1">Alguna</option>
              <option value="2">Mucha o incapaz</option>
            </select>
          </td>
        </tr>
      </tbody>
    </table>
  </form>
</div>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
<td><strong>Assistance in walking (Asistencia para caminar):</strong> ¿Qué
tanta dificultad tiene para cruzar caminando por un cuarto?</td>
<td>
  <select formControlName="walking" class="form-control">
    <option value="0">Ninguna</option>
    <option value="1">Alguna</option>
    <option value="2">Mucha, usando auxiliares o incapaz</option>
  </select>
</td>
</tr>
<tr>
  <td><strong>Rise from chair (Levantarse de una silla):</strong> ¿Qué tanta
dificultad tiene para levantarse de una silla o cama?</td>
  <td>
    <select formControlName="chair" class="form-control">
      <option value="0">Ninguna</option>
      <option value="1">Alguna</option>
      <option value="2">Mucha o incapaz sin ayuda</option>
    </select>
  </td>
</tr>
<tr>
  <td><strong>Climb stairs (Subir escaleras):</strong> ¿Qué tanta dificultad
tiene para subir 10 escalones?</td>
  <td>
    <select formControlName="stairs" class="form-control">
      <option value="0">Ninguna</option>
      <option value="1">Alguna</option>
      <option value="2">Mucha o incapaz</option>
    </select>
  </td>
</tr>
<tr>
  <td><strong>Falls (Caídas):</strong> ¿Cuántas veces se ha caído en el
último año?</td>
  <td>
    <select formControlName="falls" class="form-control">
      <option value="0">Ninguna</option>
      <option value="1">1-3 caídas</option>
      <option value="2">4 o más caídas</option>
    </select>
  </td>
</tr>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
</tr>
</tbody>
</table>

<button type="button" class="btn btn-success" (click)="enviarResultados()">Enviar
Resultados</button>
</form>
</div>
import { Component, OnInit } from '@angular/core';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';
import { Router } from '@angular/router';

@Component({
  selector: 'app-encuesta-sarc-f',
  templateUrl: './encuesta-sarc-f.component.html',
  styleUrls: ['./encuesta-sarc-f.component.scss']
})
export class EncuestaSarcFComponent implements OnInit {
  sarcFForm: FormGroup;
  puntuacionTotal: number = 0;

  constructor(private fb: FormBuilder, private router: Router) {
    this.sarcFForm = this.fb.group({
      strength: [0, Validators.required],
      walking: [0, Validators.required],
      chair: [0, Validators.required],
      stairs: [0, Validators.required],
      falls: [0, Validators.required],
    });
  }

  ngOnInit(): void {}

  calcularPuntuacion() {
    const values = this.sarcFForm.value;
    this.puntuacionTotal =
      +values.strength +
      +values.walking +
      +values.chair +
      +values.stairs +
      +values.falls;
  }
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
enviarResultados() {
  if (this.sarcFForm.invalid) {
    alert('Completa todas las preguntas antes de enviar.');
```



```
    return;
  }

  const puntajeTotal = this.calcularTotal();
  let observacion;

  if (puntajeTotal >= 4) {
    observacion = 'Alta probabilidad de sarcopenia';
  } else {
    observacion = 'Baja probabilidad de sarcopenia';
  }

  console.log('Puntaje total:', puntajeTotal, 'Resultado:', observacion);

  this.router.navigate(['home/resultado'], {
    queryParams: {
      puntaje: puntajeTotal,
      observacion: observacion,
    },
  });
}

private calcularTotal(): number {
  const values = this.sarcFForm.value;
  return (
    +values.strength +
    +values.walking +
    +values.chair +
    +values.stairs +
    +values.falls
  );
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

Estructura categorías.(HTML y TypeScript)

```
<div class="tituloEncuesta">
  <h1>Encuesta {{name}}</h1>
</div>
<!--
<div class="iframe">
  <iframe [src]="safeUrl" frameborder="0" class="ventana"></iframe>
</div>-->
<!-- <app-diagnostic
(categoriaSeleccionada)="cambiarCategoria($event.toString())"></app-diagnostic> -->
<!-- ENCUESTA COGNITIVA -->
<app-encuesta-cog #encuestaComponent *ngIf="categoria == 'cognitiva'"></app-encuesta-
cog>
<!-- ENCUESTA FUNCIONAL -->
<app-category1 *ngIf="categoria == 'funcional'"></app-category1>
<!-- ENCUESTAS DE LA CATEGORIA AFECTIVA -->
<app-encuesta-afc *ngIf="subCat == 'gds15'"></app-encuesta-afc>
<app-encuesta-beck-anxiety *ngIf="subCat == 'ansbeck'"></app-encuesta-beck-anxiety>
<app-encuesta-sad-persons *ngIf="subCat == 'sadperson'"></app-encuesta-sad-persons>
<app-encuesta-ces-d7 *ngIf="subCat == 'cesd7'"></app-encuesta-ces-d7>
<app-encuesta-phq9 *ngIf="subCat == 'phq9'"></app-encuesta-phq9>
<app-encuesta-escala-soledad *ngIf="subCat == 'soledad3'"></app-encuesta-escala-
soledad>
<app-encuesta-gai-sf *ngIf="subCat == 'gaisf'"></app-encuesta-gai-sf>
<app-encuesta-cornell *ngIf="subCat == 'cornell'"></app-encuesta-cornell>
<app-encuesta-okeeffe *ngIf="subCat == 'cortdepression'"></app-encuesta-okeeffe>
<!-- ENCUESTA DE LA CATEGORIA NUTRICIONAL -->
<app-encuesta-assessment-sf *ngIf="subCat == 'mininutri'"></app-encuesta-assessment-sf>
<app-encuesta-assessment *ngIf="subCat == 'assessment'"></app-encuesta-assessment>
<app-encuesta-must *ngIf="subCat == 'must'"></app-encuesta-must>
<app-encuesta-glim *ngIf="subCat == 'glim'"></app-encuesta-glim>
<app-encuesta-sarc-f *ngIf="subCat == 'sarcf'"></app-encuesta-sarc-f>

import { Component, OnDestroy, OnInit, ViewChild } from '@angular/core';
import { DomSanitizer, SafeResourceUrl, SafeUrl } from '@angular/platform-browser';
import { ActivatedRoute, Route, Router } from '@angular/router';
import { ApiService } from 'src/app/Service/api.service';
import { categoria } from 'src/app/Shared/Data';
import { EncuestaCogComponent } from '.../Diagnosticos/Cognitiva/encuesta-
cog/encuesta-cog.component';

@Component({
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
selector: 'app-encuesta',
templateUrl: './encuesta.component.html',
styleUrls: ['./encuesta.component.scss']
}))

export class EncuestaComponent implements OnInit, OnDestroy {
  //Recibe Parametro ENtrante del Componente
  @ViewChild('encuestaComponent') encuestaComponent!: EncuestaCogComponent;

  name?: String; //Nombre Categoria
  categoria?: String; // Categoria
  subCat?: String; // Subcategoria
  public safeUrl!: SafeResourceUrl;
  private BaseURL?: String = 'http://localhost:4200/'; //URL Fijo

  constructor(private router: Router, private serviceApi: ApiService, private route:
ActivatedRoute, private sanitizer: DomSanitizer) {

  }

  ngOnInit(): void {
    // Obtener el parámetro 'categoria' desde la URL (URL Dinamica)
    this.categoria = this.route.snapshot.paramMap.get('categoria')!;

    // Recibir en nombre Clave del componente Diagnostico
    this.serviceApi.categoriaSeleccionada$.subscribe(cat => {
      this.categoria = cat; //Recibe clave de Categoria
      this.updateTitle(cat);
      console.log(`Categoria recibida en EncuestaComponent: ${this.categoria,' | ',
cat}`); // Debug
    });

    this.serviceApi.selectEncuest$.subscribe(subC => {
      this.subCat = subC;
      console.log('Subcategoria recibida en EncuestaComponent: ',subC);
    });

    document.addEventListener('contextmenu',this.disableRightClick);

  }

  disableRightClick(event: MouseEvent){
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
event.preventDefault();
}

ngOnDestroy(): void {
  localStorage.removeItem('subCatSeleccionada'); // Limpia selección SubCategoria
  localStorage.removeItem('encuestaSeleccionada'); // Limpia selección SubCategoria
  localStorage.removeItem('categoriaSeleccionada') // Limpiar seleccion Categoria
  document.removeEventListener('contextmenu', this.disableRightClick);
}

cambiarCategoria(categoria: string) {
  this.encuestaComponent.actualizarEncuesta(categoria);
  this.name = categoria;
}

updateTitle(categoria: String): void{
  switch (categoria) {
    case 'cognitiva':
      this.name = 'COGNITIVA';
      break;
    case 'afectiva':
      this.name = 'AFECTIVA';
      break;
    case 'funcional':
      this.name = 'FUNCIONAL';
      break;
    case 'nutricional':
      this.name = 'NUTRICIONAL';
      break;
    default:
      this.name = 'None'; // Valor predeterminado
      break;
  }
}
```

Componente resultado (HTML y TypeScript)

```
<div class="resultado-container">
  <p class="texto">Puntuacion del Resultado recopilado en base de las Respuestas
  Seleccionadas</p>
  <h2>Resultado de la Encuesta: {{ nombreEncuesta }}</h2>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
<p class="resultadoPuntaje"><strong>Puntaje Total:</strong> {{ puntaje }} puntos</p>
<p class="resultadoPuntaje"><strong>Nivel de Dependencia:</strong> {{ porcentaje
}}%</p>
<!-- <p><strong>Nivel de Dependencia:</strong> {{ obtenerNivel() }}</p> -->
<p class="resultadoPuntaje"><strong>Observación:</strong> {{ observacion }}</p>
</div>

<div class="d-grid gap-2 col-6 botones">
  <!-- <button class="btn btn-primary" type="button"
(click)="funcionBoton(1)">Encuesta</button> -->
  <button class="btn btn-secondary" type="button"
(click)="funcionBoton(2)">Categoria</button>
</div>
import { Component, Input, OnInit } from '@angular/core';
import { ActivatedRoute, Router } from '@angular/router';

@Component({
  selector: 'app-resultados',
  templateUrl: './resultados.component.html',
  styleUrls: ['./resultados.component.scss']
})
export class ResultadosComponent implements OnInit {
  nombreEncuesta: string = ''; //Encuesta
  puntaje: number = 0; //Control del Puntaje
  observacion: string = ''; //Observacion
  porcentaje: number = 0;

  constructor(private router: Router, private route: ActivatedRoute) {
    //const navigation = this.router.getCurrentNavigation();
    //this.encuestas = navigation?.extras.state?.['encuestas'] || [];
  }

  ngOnInit(): void {
    // Obtener los puntos desde los queryParams
    this.route.queryParams.subscribe(params => {
      this.puntaje = params['puntaje'] || 0; // Si no hay puntos, se asigna 0
      this.nombreEncuesta = params['nameEncuesta'] || '';
      this.porcentaje = params['porcentaje'] || '';
      this.observacion = params['observacion'] || '';
    });
  }
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
funcionBoton(n: number): void{
  switch(n){
    case 1:
      this.router.navigate(['home/encuesta-cog']);
      break;
    case 2:
      this.router.navigate(['home/diagnostic']);
      break;
    case 3:
      this.router.navigate(['home/encuesta-afc']);
      break;
  }
}
```

Componente list-doctors (HTML y TypeScript)

```
<div class="boton">
  <button type="button" class="btn btn-success btn-lg" (click)="register()"><i
class="fa-solid fa-user-nurse"></i> Registrar Doctor</button>
</div>
<div class="tablaDoctores">
  <div class="container">
    <h2 class="mt-4">Lista de Doctores</h2>
    <table class="table table-striped">
      <thead>
        <tr>
          <th>ID</th>
          <th>Nombre</th>
          <th>Apellido</th>
          <th>Cédula</th>
          <th>Profesión</th>
          <th>Edad</th>
          <th>Género</th>
          <th>Email</th>
          <th>Teléfono</th>
          <th>Dirección</th>
          <th>Acciones</th>
        </tr>
      </thead>
      <tbody>
        <tr *ngFor="let user of users">
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
<td>{{ user.id }}</td>
<td>{{ user.nombre }}</td>
<td>{{ user.apellido }}</td>
<td>{{ user.cedula }}</td>
<td>{{ user.profesion }}</td>
<td>{{ user.edad }}</td>
<td>{{ user.genero }}</td>
<td>{{ user.email }}</td>
<td>{{ user.telefono }}</td>
<td>{{ user.direccion }}</td>
<td>
  <div class="botones">
    <button class="btn btn-sm btn-danger" (click)="eliminarUsuario(user.id,
user.nombre)" title="Eliminar {{user.nombre}}">
      <i class="fas fa-trash-alt"></i> <!-- Icono de eliminar -->
    </button>
    <button class="btn btn-sm btn-primary"
(click)="actualizarUsuario(user.id)" title="Cambiar Contraseña">
      <i class="fas fa-edit"></i> <!-- Icono de actualizar -->
    </button>
  </div>
</td>
</tr>
</tbody>
</table>
</div>
</div>

import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { AlertService } from 'src/app/Service/alert.service';
import { ApiService } from 'src/app/Service/api.service';
import { SharedService } from 'src/app/Service/shared.service';

@Component({
  selector: 'app-list-doctors',
  templateUrl: './list-doctors.component.html',
  styleUrls: ['./list-doctors.component.scss']
})
export class ListDoctorsComponent implements OnInit {
  showTableMaster: boolean = false; //vista del boton SuperMaster
  // Doctores
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
users: any[] = [];  
user: any = {};  
  
constructor(private apiService: ApiService, private alertService: AlertService,  
private sharedService: SharedService, private router: Router){}  
  
ngOnInit(): void {  
  this.fetchUsers();  
  this.sharedService.showRegisterButton$.subscribe(show => {  
    this.showTableMaster = show;  
  });  
}  
  
eliminarUsuario(id: number, name: string): void {  
  console.log('Eliminar usuario con ID:', id ,name);  
  // Utilizar el servicio AlertService para mostrar la confirmación  
  this.alertService  
    .confirm(`¿Estás seguro de que deseas eliminar al doctor "${name}"?`,  
    'Confirmación')  
    .then((isConfirmed) => {  
      if (isConfirmed) {  
        // Llamar al API para eliminar al doctor si se confirma  
        this.apiService.deleteDoctor(id).subscribe({  
          next: (response) => {  
            // Mostrar mensaje de éxito  
            this.alertService.success(  
              `El doctor "${name}" ha sido eliminado exitosamente.`,  
              'Eliminado'  
            );  
            this.cargarDoctores(); // Refrescar la lista de doctores  
          },  
          error: (error) => {  
            console.error('Error al eliminar el doctor:', error);  
            // Mostrar mensaje de error  
            this.alertService.error(  
              `No se pudo eliminar al doctor "${name}". Inténtalo nuevamente.`,  
              'Error'  
            );  
          },  
        });  
      }  
    });  
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
});  
// Agrega aquí la lógica para eliminar el usuario  
}  
  
actualizarUsuario(id: number): void {  
  console.log('Actualizar usuario con ID:', id);  
  // Agrega aquí la lógica para actualizar el usuario  
}  
  
cargarDoctores(): void {  
  this.apiService.getUsers().subscribe({  
    next: (users) => this.users = users, // Asigna los usuarios recibidos al array  
    "users"  
    error: (error) => console.error('Error al cargar la lista de doctores:', error)  
  });  
}  
  
fetchUsers() {  
  this.apiService.getUsers().subscribe((data) => {  
    this.users = data;  
  });  
}  
  
register(){  
  this.router.navigate(['home/doctors-register']);  
}  
}
```

Componente Login (HTML y TypeScript)

```
<div class="container-fluid">  
  <div class="row justify-content-center">  
    <div class="col-md-6">  
      <div class="card mt-5">  
        <div class="card-body">  
          <h3 class="card-title text-center">Iniciar Sesión</h3>  
          <div class="text-center">  
            
```



TECNOLÓGICO NACIONAL DE MÉXICO

```
</div>
<form (ngSubmit)="onSubmit()" #LoginForm="ngForm">
  <div class="form-group">
    <label for="email">Correo Electrónico</label>
    <input
      type="email"
      class="form-control"
      id="email"
      name="email"
      required
      [(ngModel)]="email"
    />
  </div>
  <div class="form-group">
    <label for="password">Contraseña</label>
    <input
      type="password"
      class="form-control"
      id="password"
      name="password"
      required
      [(ngModel)]="password"
    />
  </div>
  <button type="submit" class="btn btn-info btn-block">Ingresar</button>
</form>
</div>
</div>
</div>
</div>
</div>
import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { ApiService } from 'src/app/Service/api.service';
import { SharedService } from 'src/app/Service/shared.service';
import { UserService } from 'src/app/Service/user.service';

@Component({
  selector: 'app-login',
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.scss']
})
```





TECNOLÓGICO NACIONAL DE MÉXICO

```
export class LoginComponent implements OnInit{
  email: string = '';
  password: string = '';
  errorMessage: string = '';

  constructor(private userService: UserService, private router: Router, private
  apiService: ApiService, private sharedService: SharedService) {}

  ngOnInit(): void {

  }

  /*onSubmit() {
    if (this.userService.login(this.email, this.password)) {
      // Si las credenciales son correctas, redirige a 'home'
      this.router.navigate(['home'], { replaceUrl: true });
    } else {
      // Si las credenciales son incorrectas, muestra el mensaje de error
      this.errorMessage = 'Correo electrónico o contraseña incorrectos';
    }
  }*/

  async onSubmit() {
    const isLoggedIn = await this.userService.log(this.email, this.password);

    if (isLoggedIn) {
      //this.sharedService.setShowRegisterButton(false);
      this.updateRegisterButtonVisibility();
      this.router.navigate(['home'], { replaceUrl: true });
    } else {
      this.errorMessage = 'Correo electrónico o contraseña incorrectos';
    }
  }

  updateRegisterButtonVisibility(): void {
    this.apiService.getUsers().subscribe((users: any[]) => {
      // Determina si se muestra el botón usando UserService
      const showButton = this.userService.canShowRegisterButton(this.email, users);
      // Actualiza el estado en el servicio compartido
      this.sharedService.setShowRegisterButton(showButton);
    });
  }
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
}
```

Componente patients (HTML y TypeScript).

```
<!-- Botón para Abrir el Modal -->
<div class="d-grid gap-2 col-6 btn_click">
  <button class="btn btn-success" type="button" (click)="patient()"><i class="fa-solid
fa-circle-plus"></i> Añadir Paciente</button>
</div>

<div class="container">
  <h1>Pacientes</h1>
  <table mat-table [dataSource]="dataSource" multiTemplateDataRows class="mat-
elevation-z8">
    <!-- Column definitions -->
    <ng-container matColumnDef="{{column}}" *ngFor="let column of columnsToDisplay">
      <th mat-header-cell *matHeaderCellDef> {{columnHeaders[column]}} </th>
      <td mat-cell *matCellDef="let element"> {{element[column]}} </td>
    </ng-container>
    <!-- Expand button column -->
    <ng-container matColumnDef="expand">
      <th mat-header-cell *matHeaderCellDef aria-label="row actions">&nbsp;</th>
      <td mat-cell *matCellDef="let element">
        <button mat-icon-button aria-label="expand row"
          (click)="(expandedElement = expandedElement === element ? null : element);
$event.stopPropagation()">
          <mat-icon *ngIf="expandedElement !== element">keyboard_arrow_down</mat-icon>
          <mat-icon *ngIf="expandedElement === element">keyboard_arrow_up</mat-icon>
        </button>
      </td>
    </ng-container>

    <!-- Expanded Content Column - The detail row is made up of this one column that
spans across all columns -->
    <ng-container matColumnDef="expandedDetail">
      <td mat-cell *matCellDef="let element"
[attr.colspan]="columnsToDisplayWithExpand.length">
        <div class="example-element-detail" [@detailExpand]="element == expandedElement
? 'expanded' : 'collapsed'">
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
<div class="example-element-diagram">
  <!--Foto del paciente-->
  
</div>
<div class="botons">
  <button type="button" class="btn btn-info"
(click)="perfilUser(element.id)">Perfil Paciente</button>
</div>
<div class="example-element-description">
  {{element.description}}
</div>
</div>
</td>
</ng-container>
<!-- Table rows -->
<tr mat-header-row *matHeaderRowDef="columnsToDisplayWithExpand"></tr>
<tr mat-row *matRowDef="let element; columns: columnsToDisplayWithExpand;"
class="example-element-row"
[class.example-expanded-row]="expandedElement === element"
(click)="expandedElement = expandedElement === element ? null : element">
</tr>
<tr mat-row *matRowDef="let row; columns: ['expandedDetail']" class="example-
detail-row"></tr>
</table>
</div>
import { Component, ViewChild } from '@angular/core';
import { MatPaginator } from '@angular/material/paginator';
import { MatTableDataSource } from '@angular/material/table';
import { animate, state, style, transition, trigger, } from '@angular/animations';
import { PATIENTS_DATA } from '../data/patients-data';
import { Router } from '@angular/router';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';
import { ApiService } from 'src/app/Service/api.service';
import { UserService } from 'src/app/Service/user.service';

@Component({
  selector: 'app-patients',
  templateUrl: './patients.component.html',
  styleUrls: ['./patients.component.scss'],
  animations: [
    trigger('detailExpand', [
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
state('collapsed', style({ height: '0px', minHeight: '0' })),
state('expanded', style({ height: '*' })),
transition(
  'expanded <=> collapsed',
  animate('225ms cubic-bezier(0.4, 0.0, 0.2, 1)')
),
]),
],
})
export class PatientsComponent {
  title = 'Pacientes';
  columnsToDisplay: string[] = [
    'first_name',
    'last_name',
    'age',
    'last_consultation',
  ];
  columnsToDisplayWithExpand = [...this.columnsToDisplay, 'expand'];
  dataSource = new MatTableDataSource(PATIENTS_DATA); // Usa los datos importados
  expandedElement: any | null = null;

  @ViewChild(MatPaginator, { static: true }) paginator:
    | MatPaginator
    | undefined;

  constructor(private router: Router, private apiService: ApiService, private
userService: UserService){

  }

  ngOnInit() {
    // Configura el paginador
    if (this.paginator) {
      this.dataSource.paginator = this.paginator;
    }

    // Obtener pacientes del doctor
    this.getPatients();
  }

  columnHeaders: { [key: string]: string } = {
    first_name: 'Nombre',
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
last_name: 'Apellido',
age: 'Edad',
last_consultation: 'Última consulta',
};

// Método para redirigir al perfil del paciente
perfilUser(id: number){
  // Redirige al componente del perfil de paciente pasando el id en la URL
  this.router.navigate(['home/view-person',{patientID: id}]);
  console.log(id)
}

patient(){
  this.router.navigate(['/home/register-patient'])
}

getPatients(){
  this.apiService.getPatientsByDoctor(this.userService.getDoctorId()).subscribe(
    (response) =>{
      this.dataSource.data = response.patients;
    }, (error) =>{
      alert('Error al obtener los pacientes');
    }
  );
}
```

Componente profile-medic (HTML y TypeScript)

```
<div class="perfilDoctor" *ngIf="mostrar && !pass">
  <p class="title">Perfil Médico</p>
  <!-- PERFIL DEL MEDICO -->
  <div class="perfil">
    <div class="fotoPerfil">
      <img [src]='imagen' alt="" class="user"> <!-- Imagen/ImageDefault -->
      <div class="overlay"><i class="fas fa-camera"></i></div> <!--Cambiar foto-->
    </div>
    <div class="container">
      <p class="firstName">{{ nombre }}</p> <!-- Nombre -->
      <p class="lastName">{{ apellido }}</p> <!-- Apellido -->
      <div class="datoPersonal">
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
<p class="date">{{ fecha }}</p> <!-- Fecha Nacimiento -->
<p class="genero">{{ genero }}</p> <!-- Genero -->
</div>
</div>
<button class="edit-btn" title="Editar Perfil" (click)="seleccionPerfil(1)">
  <i class="fas fa-pencil-alt" id="icon"></i>
</button>
</div>

<div class="containerProfesional">
  <!-- Cedula -->
  <div class="perfil-item">
    <span class="label">Cédula:</span>
    <span class="value">{{ cedula }}</span>
  </div>

  <!-- Profesion -->
  <div class="perfil-item">
    <span class="label">Profesión:</span>
    <span class="value">{{ profesion }}</span>
  </div>

  <!-- Edad -->
  <div class="perfil-item">
    <span class="label">Edad:</span>
    <span class="value">{{ edad }}</span>
  </div>

  <!-- Correo -->
  <div class="perfil-item">
    <span class="label">Correo:</span>
    <span class="value">{{ correo }}</span>
  </div>

  <!-- Mostrar/Ocultar Contraseña -->
  <div class="perfil-item password-item">
    <span class="label">Contraseña:</span>
    <span class="value">
      <span *ngIf="mostrarPassword">{{ password }}</span>
      <span *ngIf="!mostrarPassword">*****</span>
      <button (click)="togglePasswordVisibility()" class="toggle-btn">

```

TECNOLÓGICO NACIONAL DE MÉXICO

```
<mat-icon>{{ mostrarPassword ? 'visibility_off' : 'visibility'
}}</mat-icon>
</button>
</span>
</div>

<!-- Telefono -->
<div class="perfil-item">
  <span class="label">Teléfono:</span>
  <span class="value">{{ telefono }}</span>
</div>

<!-- Direccion -->
<div class="perfil-item">
  <span class="label">Dirección:</span>
  <span class="value">{{ direccion }}</span>
</div>

<button class="btn btn-success" (click)="selectChanged(1)">Cambiar
Contrase&#241;a</button>
</div>
</div>

<!-- ***** ACTUALIZACION DEL DOCTOR ***** -->
<div class="containerForm" *ngIf="!mostrar">
  <p class="title">Actualizaci&#243;n del Perfil</p>
  <form [formGroup]="profileForm" (ngSubmit)="onSubmitForm()">
    <div class="form-group">
      <label for="nombre">Nombre:</label>
      <input type="text" id="nombre" formControlName="nombre" class="form-control"
/>
    </div>

    <div class="form-group">
      <label for="apellido">Apellido:</label>
      <input type="text" id="apellido" formControlName="apellido" class="form-
control"/>
    </div>

    <div class="form-group">
      <label for="cedula">Cédula:</label>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
<input type="text" id="cedula" class="form-control disabled"
[disabled]="!mostrar" value="{{ cedula }}" title="{{ cursor }}" />
</div>

<div class="form-group">
  <label for="profesion">Profesión:</label>
  <input type="text" id="profesion" formControlName="profesion" class="form-
control" required />
</div>

<div class="form-group">
  <label for="fecha">Fecha de Nacimiento:</label>
  <input type="text" id="fecha" class="form-control disabled" required
value="{{ fechaForm }}" [disabled]="!mostrar" title="{{ cursor }}" />
</div>

<div class="form-group">
  <label for="edad">Edad:</label>
  <input type="number" id="edad" formControlName="edad" class="form-control"
required [ngClass]="{'is-invalid': profileForm.get('edad')?.invalid &&
profileForm.get('edad')?.touched}" />
  <div *ngIf="profileForm.get('edad')?.invalid &&
profileForm.get('edad')?.touched" class="invalid-feedback">
    La edad debe ser igual o mayor a 20.
  </div>
</div>

<div class="form-group">
  <label for="genero">Género:</label>
  <input type="text" id="genero" class="form-control disabled"
[disabled]="!mostrar" value="{{ genero }}" title="{{ cursor }}" />
</div>

<div class="form-group">
  <label for="correo">Correo:</label>
  <input type="email" id="correo" formControlName="correo" class="form-control"
required [ngClass]="{'is-invalid': profileForm.get('correo')?.invalid &&
profileForm.get('correo')?.touched}" />
  <div *ngIf="profileForm.get('correo')?.invalid &&
profileForm.get('correo')?.touched" class="invalid-feedback">
    Por favor ingresa un correo válido.
  </div>
</div>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
</div>

<div class="form-group">
  <label for="telefono">Teléfono:</label>
  <input type="tel" id="telefono" formControlName="telefono" class="form-
control" required [ngClass]='{"is-invalid": profileForm.get('telefono')?.invalid &&
profileForm.get('telefono')?.touched}"/>
  <div *ngIf="profileForm.get('telefono')?.invalid &&
profileForm.get('telefono')?.touched" class="invalid-feedback">
    El teléfono debe tener un máximo de 10 dígitos y contener solo números.
  </div>
</div>

<div class="form-group">
  <label for="direccion">Dirección:</label>
  <input type="text" id="direccion" formControlName="direccion" class="form-
control" required />
</div>

<div class="form-group">
  <label for="password">Contraseña:</label>
  <div class="d-flex align-items-center">
    <input [type]="mostrarPassword ? 'text' : 'password'" id="password"
formControlName="password" class="form-control" required [ngClass]='{"is-invalid":
profileForm.get('password')?.invalid && profileForm.get('password')?.touched}"/>
    <button type="button" mat-icon-button (click)="togglePasswordVisibility()"
title="{ { mostrarPassword ? 'Ocultar' : 'Mostrar' } } contraseña">
      <mat-icon>{ { mostrarPassword ? 'visibility_off' : 'visibility' } }</mat-
icon>
    </button>
  </div>
  <div *ngIf="profileForm.get('password')?.invalid &&
profileForm.get('password')?.touched" class="invalid-feedback">
    La contraseña debe tener al menos 8 caracteres.
  </div>
</div>

<div class="botones">
  <button type="submit" class="btn btn-primary"
[disabled]="profileForm">Actualizar</button>
  <button class="btn btn-danger"
(click)="seleccionPerfil(2)">Cancelar</button>
</div>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
</div>
</form>
</div>

<div class="containerPassword" *ngIf="pass">
  <p class="title">Cambiar Contraseña</p>
  <form [formGroup]="changePasswordForm" (ngSubmit)="onSubmitPassword()" class="container
mt-5">
    <!-- Correo Electrónico (opcional) -->
    <div class="form-group">
      <label for="correo">Correo Electrónico (Opcional):</label>
      <input type="email" id="correo" formControlName="correo" class="form-control"
[ngClass]="{'is-invalid': changePasswordForm.get('correo')?.invalid &&
changePasswordForm.get('correo')?.touched}" placeholder="Ingresa tu correo
electrónico"/>
      <div *ngIf="changePasswordForm.get('correo')?.invalid &&
changePasswordForm.get('correo')?.touched" class="invalid-feedback">
        {{ getEmailErrorMessage() }}
      </div>
    </div>

    <!-- Nueva Contraseña -->
    <div class="form-group">
      <label for="password">Nueva Contraseña:</label>
      <div class="input-group">
        <input [type]="mostrarPassword ? 'text' : 'password'" id="password"
formControlName="password" class="form-control" [ngClass]="{'is-invalid':
changePasswordForm.get('password')?.invalid &&
changePasswordForm.get('password')?.touched}" placeholder="Ingresa nueva contraseña"/>
        <button type="button" mat-icon-button class="input-group-text"
(click)="togglePasswordVisibility()" title="{{ mostrarPassword ? 'Ocultar' : 'Mostrar'
}} contraseña">
          <mat-icon>{{ mostrarPassword ? 'visibility_off' : 'visibility' }}</mat-icon>
        </button>
        <div *ngIf="changePasswordForm.get('password')?.invalid &&
changePasswordForm.get('password')?.touched" class="invalid-feedback">
          {{ getPasswordErrorMessage() }}
        </div>
      </div>
    </div>

    <!-- Confirmar Nueva Contraseña -->
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
<div class="form-group">
  <label for="confirmPassword">Confirmar Nueva Contraseña:</label>
  <div class="input-group">
    <input [type]="mostrarConfirmPassword ? 'text' : 'password'"
id="confirmPassword" formControlName="confirmPassword" class="form-control"
    [ngClass]="{'is-invalid': (changePasswordForm.get('confirmPassword')?.invalid
&& changePasswordForm.get('confirmPassword')?.touched) ||
changePasswordForm.hasError('passwordsMismatch') &&
changePasswordForm.get('confirmPassword')?.touched}"
    placeholder="Confirma nueva contraseña" />
    <button type="button" mat-icon-button class="input-group-text"
(click)="toggleConfirmPasswordVisibility()" title="{ {{ mostrarConfirmPassword ?
'Ocultar' : 'Mostrar' }} contraseña">
      <mat-icon>{{ mostrarConfirmPassword ? 'visibility_off' : 'visibility'
}}</mat-icon>
    </button>
    <div *ngIf="(changePasswordForm.get('confirmPassword')?.invalid &&
changePasswordForm.get('confirmPassword')?.touched) ||
(changePasswordForm.hasError('passwordsMismatch') &&
changePasswordForm.get('confirmPassword')?.touched)" class="invalid-feedback">
      {{ getConfirmPasswordErrorMessage() }}
    </div>
  </div>
</div>

<!-- Botón de Envío -->
<div class="botones">
  <button type="submit" class="btn btn-success"
[disabled]="changePasswordForm.invalid">Cambiar Contrase&#241;a</button>
  <button class="btn btn-danger" (click)="selectChanged(2)">Cancelar</button>
</div>
</form>
</div>
import { DatePipe } from '@angular/common';
import { Component, OnInit } from '@angular/core';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';
import { AlertService } from 'src/app/Service/alert.service';
import { ApiService } from 'src/app/Service/api.service';
import { UserService } from 'src/app/Service/user.service';

@Component({
  selector: 'app-profile-medic',
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
templateUrl: './profile-medic.component.html',
styleUrls: ['./profile-medic.component.scss'],
providers: [DatePipe] // Proveer DatePipe en el componente
}))

export class ProfileMedicComponent implements OnInit {
  //Datos del Docto
  doctorData : any; //Recibe los datos del servicio
  //Mostrar por defecto en caso de no conectar al servidor
  nombre: string = 'Eduardo Manuel';
  apellido: string = 'Miller Martínez';
  cedula: string = '123456789';
  profesion: string = 'Ingeniero de Software';
  fecha?: string = '21/Agosto/2004';
  edad: number = 30;
  genero: string = 'Masculino';
  correo: string = 'usuario@ejemplo.com';
  telefono: string = '123-456-7890';
  direccion: string = '123 Calle Principal, Ciudad, País';
  imagen: string = 'assets/Imagenes/default-profile.png';
  password: string = 'Hola Mundo';
  mostrarPassword = false;
  mostrar: boolean = true;
  fechaForm = '???/??/??';
  cursor = 'No Disponible';
  pass: boolean = false;

  //Registro de la actualizacion
  profileForm!: FormGroup;

  //Formulario Password
  changePasswordForm!: FormGroup;
  //mostrarPassword: boolean = false;
  mostrarConfirmPassword: boolean = false;
  defaultEmail: string = 'usuario@ejemplo.com'; // Puedes obtenerlo del UserService

  constructor(private userService: UserService, private apiService: ApiService, private
datePipe: DatePipe, private fb: FormBuilder, private alertService: AlertService) {}

  ngOnInit(): void {
    // Obtener el ID del doctor desde el ApiService
    const doctorId = this.userService.getDoctorId();
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
if(doctorId){
  this.cargarDatosPerfil(doctorId);
}

//Datos del Formulario (Update)
this.profileForm = this.fb.group({
  nombre: ['', [Validators.required]],
  apellido: ['', [Validators.required]],
  cedula: ['', [Validators.required]],
  profesion: ['', [Validators.required]],
  fecha: ['', [Validators.required]],
  edad: ['', [Validators.required, Validators.min(20)]],
  genero: ['', [Validators.required]],
  correo: ['', [Validators.required, Validators.email]],
  telefono: ['', [Validators.required, Validators.maxLength(10),
Validators.pattern('^[0-9]*$')]],
  direccion: ['', [Validators.required]],
  password: ['', [Validators.required, Validators.minLength(8)]],
  imagen: [''] // Este campo es opcional
});

// Updte Passowrd
this.changePasswordForm = this.fb.group({
  correo: ['', [Validators.email]],
  password: ['', [Validators.required, Validators.minLength(8)]],
  confirmPassword: ['', [Validators.required, Validators.minLength(8)]],
}, { validators: this.passwordsMatchValidator });
}

cargarDatosPerfil(id: number){
  // Llamar al servicio para obtener los datos del doctor
  this.apiService.getDoctorId(id).subscribe(
    (response: any) => {
      if (response.success) {
        const doctor = response.data;
        this.nombre = doctor.nombre;
        this.apellido = doctor.apellido;
        this.fechaForm = doctor.date;
        this.genero = doctor.genero;
        this.cedula = doctor.cedula;
        this.profesion = doctor.profesion;
        this.edad = doctor.edad;
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
this.correo = doctor.email;
this.defaultEmail = doctor.email;
this.telefono = doctor.telefono;
this.password = doctor.password;
this.direccion = doctor.direccion;
this.imagen = doctor.imagen ? doctor.imagen : 'assets/Imagenes/default-
profile.png';
// Cambiar formato de la fecha
if (doctor.date) {
  const formattedDate = this.datePipe.transform(doctor.date,
'dd/MMMM/yyyy'); // Cambia el formato a 'yyyy/MMMM/dd'
  this.fecha = formattedDate?.replace('M', this.getMonthName(new
Date(doctor.date).getMonth())); // Reemplaza el número del mes con el nombre
}
console.log(doctor, id)
} else {
  console.error('Error: Datos del doctor no encontrados.');
```

```
},
(error) => {
  console.error('Error al obtener los datos del doctor:', error);
}
);
}
```

```
togglePasswordVisibility(): void {
  this.mostrarPassword = !this.mostrarPassword;
}
selectChanged(select: number) {
  this.pass = !this.pass;
  if(select == 2){
    this.changePasswordForm.reset();
  }
}
```

```
seleccionPerfil(selec: number){
  this.mostrar = !this.mostrar; // Alternar entre los divs
  if(selec == 2){
    this.profileForm.reset();
  }
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
// Función para obtener el nombre del mes
getMonthName(monthIndex: number): string {
  const months = [
    'Enero', 'Febrero', 'Marzo', 'Abril', 'Mayo', 'Junio',
    'Julio', 'Agosto', 'Septiembre', 'Octubre', 'Noviembre', 'Diciembre'
  ];
  return months[monthIndex];
}

onSubmitForm(): void {
  if (this.profileForm.valid) {
    console.log('Formulario enviado con éxito', this.profileForm.value);
    // Aquí puedes enviar los datos a tu API
  } else {
    console.log('Formulario inválido');
  }
}

onSubmitPassword(): void{
  const idDoc = this.userService.getDoctorId();
  if(this.changePasswordForm.valid){
    const formValues = this.changePasswordForm.value;
    //const correo = formValues.correo ? formValues.correo : this.defaultEmail;
    const correo = formValues.correo || this.defaultEmail;
    const nuevaPassword = formValues.password;

    // Preparar los datos a enviar al backend
    const payload = {
      correo: correo,
      password: nuevaPassword
    };

    // Llamar al servicio para cambiar la contraseña
    this.apiService.changePassword(payload).subscribe(
      (response) => {
        console.log('Contraseña actualizada con éxito:', response);
        console.log(payload);
        this.alertService.success('Cambio de contraseña exitoso', 'Successfully
Password');
        this.pass = !this.pass;
        this.cargarDatosPerfil(idDoc);
        this.changePasswordForm.reset();
      },
      (error) => {
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
        console.error('Error al actualizar la contraseña:', error);
        // Aquí puedes manejar errores, como mostrar un mensaje al usuario
    }
    );
} else {
    // Marcar todos los campos como tocados para mostrar errores
    this.changePasswordForm.markAllAsTouched();
}

}

//Metodo para validar password coicidan
// Validador personalizado para verificar que las contraseñas coincidan
passwordsMatchValidator(form: FormGroup) {
    const password = form.get('password')?.value;
    const confirmPassword = form.get('confirmPassword')?.value;

    return password === confirmPassword ? null : { passwordsMismatch: true };
}

// Alternar visibilidad de la confirmación de la contraseña
toggleConfirmPasswordVisibility(): void {
    this.mostrarConfirmPassword = !this.mostrarConfirmPassword;
}

// Obtener mensaje de error para el correo
getEmailErrorMessage() {
    const correoControl = this.changePasswordForm.get('correo');
    if (correoControl?.hasError('email')) {
        return 'Por favor, ingresa un correo válido.';
    }
    return '';
}

// Obtener mensaje de error para la contraseña
getPasswordErrorMessage() {
    const passwordControl = this.changePasswordForm.get('password');
    if (passwordControl?.hasError('required')) {
        return 'La contraseña es obligatoria.';
    }
    if (passwordControl?.hasError('minlength')) {
        return 'La contraseña debe tener al menos 8 caracteres.';
    }
    return '';
}

// Obtener mensaje de error para confirmar contraseña
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
getConfirmPasswordErrorMessage() {
  const confirmPasswordControl = this.changePasswordForm.get('confirmPassword');
  if (confirmPasswordControl?.hasError('required')) {
    return 'Confirmar contraseña es obligatorio.';
  }
  if (confirmPasswordControl?.hasError('minlength')) {
    return 'La confirmación de contraseña debe tener al menos 8 caracteres.';
  }
  if (this.changePasswordForm.hasError('passwordsMismatch')) {
    return 'Las contraseñas no coinciden.';
  }
  return '';
}
```

Componente Register-patients (HTML y TypeScript)

```
<!-- Modal para Registro de Paciente -->
<div class="container">
  <div class="title">
    <h1 class="modal-title fs-5" id="staticBackdropLabel">
      <i class="fa-solid fa-person-circle-plus"></i> Añadir Paciente
    </h1>
  </div>

  <div class="modal-body">
    <!-- Formulario de Registro -->
    <form [formGroup]="pacienteForm" (ngSubmit)="onSubmit()">
      <!-- Matrícula -->
      <div class="form-group">
        <label for="matricula">Matrícula:</label>
        <input type="text" class="form-control" id="matricula"
formControlName="matricula">
        <div *ngIf="submitted && pacienteForm.controls['matricula'].errors">
          <div *ngIf="pacienteForm.controls['matricula'].errors['required']"
class="alert alert-danger m-2">
            Matrícula es requerida
          </div>
        </div>
      </div>

      <!-- Nombre -->
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
<div class="form-group">
  <label for="nombre">Nombre:</label>
  <input type="text" class="form-control" id="nombre"
formControlName="nombre" [(ngModel)]="paciente.nombre">
  <div *ngIf="submitted && pacienteForm.controls['nombre'].errors">
    <div *ngIf="pacienteForm.controls['nombre'].errors['required']"
class="alert alert-danger m-2">
      Nombre es requerido
    </div>
  </div>
</div>

<!-- Apellido -->
<div class="form-group">
  <label for="apellido">Apellido:</label>
  <input type="text" class="form-control" id="apellido"
formControlName="apellido">
  <div *ngIf="submitted && pacienteForm.controls['apellido'].errors">
    <div *ngIf="pacienteForm.controls['apellido'].errors['required']"
class="alert alert-danger m-2">
      Apellido es requerido
    </div>
  </div>
</div>

<!-- Edad -->
<div class="form-group">
  <label for="edad">Edad:</label>
  <input type="number" class="form-control" id="edad"
formControlName="edad" maxLength="3">
  <div *ngIf="submitted && pacienteForm.controls['edad'].errors"
class="alert alert-danger m-2">
    <div *ngIf="pacienteForm.controls['edad'].errors['required']">Edad
es requerida</div>
    <div *ngIf="pacienteForm.controls['edad'].errors['min']">Edad
mínima es 20</div>
  </div>
</div>

<!-- Género -->
<div class="form-group">
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
<label for="genero">Género:</label>
<select class="form-control" id="genero" formControlName="genero"
[(ngModel)]="pacient.genero">
  <option value="">Seleccione</option>
  <option value="Masculino">Masculino</option>
  <option value="Femenino">Femenino</option>
</select>
<div *ngIf="submitted && pacienteForm.controls['genero'].errors">
  <div *ngIf="pacienteForm.controls['genero'].errors['required']"
class="alert alert-danger m-2">Género es requerido</div>
</div>
</div>

<!-- Fecha de Nacimiento -->
<div class="form-group">
  <label for="fechaNacimiento">Fecha de Nacimiento: </label>
  <input type="date" id="fechaNacimiento" name="bday" class="form-
control" [(ngModel)]="pacient.fecha_nacimiento" formControlName="fecha_nacimiento">
  <div *ngIf="submitted &&
pacienteForm.controls['fecha_nacimiento'].errors" class="alert alert-danger m-2">
  <div
*ngIf="pacienteForm.controls['fecha_nacimiento'].errors['required']" >Introduzca la
fecha</div>
  </div>
</div>

<!-- Contacto de Emergencia -->
<div class="form-group">
  <label for="telefono">Contacto de Emergencia:</label>
  <input type="text" class="form-control" id="telefono"
formControlName="telefono" maxLength="15" [(ngModel)]="pacient.telefono">
  <div *ngIf="submitted && pacienteForm.controls['telefono'].errors"
class="alert alert-danger m-2">
  <div
*ngIf="pacienteForm.controls['telefono'].errors['required']">El contacto de emergencia
es obligatorio.</div>
  <div
*ngIf="pacienteForm.controls['telefono'].errors['pattern']">Solo se permiten
números.</div>
  <div
*ngIf="pacienteForm.controls['telefono'].errors['exactLength']">
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
                El número debe ser exactamente 10 dígitos. (Actual: {{
pacienteForm.controls['telefono'].errors['exactLength'].actualLength }})
            </div>
        </div>
    </div>

    <!-- Correo Electrónico de Emergencia -->
    <div class="form-group">
        <label for="email">Correo Electrónico de Emergencia:</label>
        <input type="email" class="form-control" id="email"
formControlName="email" [(ngModel)]="pacient.email">
        <div *ngIf="submitted && pacienteForm.controls['email'].errors"
class="alert alert-danger m-2">
            <div
*ngIf="pacienteForm.controls['email'].errors['required']">Correo de emergencia
requerido</div>
            <div *ngIf="pacienteForm.controls['email'].errors['email']">Solo
acepta correo valido</div>
        </div>
    </div>

    <!-- Dirección -->
    <div class="form-group">
        <label for="direccion">Dirección:</label>
        <textarea class="form-control" id="direccion"
formControlName="direccion" rows="3" [(ngModel)]="pacient.direccion"></textarea>
        <div *ngIf="submitted && pacienteForm.controls['direccion'].errors"
class="alert alert-danger m-2">
            <div
*ngIf="pacienteForm.controls['direccion'].errors['required']">Dirección es
requerida</div>
        </div>
    </div>

    <!-- Historial Médico -->
    <div class="form-group">
        <label for="historialMedico">Historial Médico:</label>
        <textarea class="form-control" id="historialMedico"
formControlName="historialMedico" rows="3"></textarea>
        <div *ngIf="submitted &&
pacienteForm.controls['historialMedico'].errors" class="alert alert-danger m-2">
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
        <div
*ngIf="pacienteForm.controls['historialMedico'].errors['required']">Historial Médico es
requerido</div>
        </div>
    </div>

    <!-- Alergias -->
    <div class="form-group">
        <label for="alergias">Alergias:</label>
        <textarea class="form-control" id="alergias" formControlName="alergias"
rows="2"></textarea>
        <div *ngIf="submitted && pacienteForm.controls['alergias'].errors"
class="alert alert-danger m-2">
            <div
*ngIf="pacienteForm.controls['alergias'].errors['required']">Alergias son
requeridas</div>
            </div>
        </div>

    <!-- Descripción -->
    <div class="form-group">
        <label for="caracteristicas">Descripción:</label>
        <textarea class="form-control" id="caracteristicas"
formControlName="caracteristicas" rows="3"></textarea>
        <div *ngIf="submitted &&
pacienteForm.controls['caracteristicas'].errors" class="alert alert-danger m-2">
            <div
*ngIf="pacienteForm.controls['caracteristicas'].errors['required']">Agregue una breve
descriocion para mostrar</div>
            </div>
        </div>

    <!-- Confirmación -->
    <div class="form-group form-check">
        <input type="checkbox" class="form-check-input" id="confirmacion"
formControlName="confirmacion">
        <label class="form-check-label" for="confirmacion">Confirmo que los
datos Ingresados son correctos</label>
        <div *ngIf="submitted && pacienteForm.controls['confirmacion'].errors"
class="alert alert-danger m-2">
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
        <div
*ngIf="pacienteForm.controls['confirmacion'].errors['required']">La confirmación es
requerida</div>
        </div>
    </div>

    <!-- Botones de acción -->
    <div class="text-center">
        <button type="submit" class="btn btn-success"><i class="fa-solid fa-
check"></i> Guardar</button>
        <button type="button" class="btn btn-danger" data-bs-dismiss="modal"><i
class="fa-solid fa-circle-xmark"></i> Cancelar</button>
    </div>
</form>

</div>
</div>
import { Component, OnDestroy, OnInit } from '@angular/core';
import { FormBuilder, FormGroup, Validators, AbstractControl, ValidationErrors,
ValidatorFn } from '@angular/forms';
import { Router } from '@angular/router';
import { AlertService } from 'src/app/Service/alert.service';
import { ApiService } from 'src/app/Service/api.service';
import { UserService } from 'src/app/Service/user.service';

@Component({
  selector: 'app-register-patient',
  templateUrl: './register-patient.component.html',
  styleUrls: ['./register-patient.component.scss']
})
export class RegisterPatientComponent implements OnInit, OnDestroy{
  /* REGISTRO DE PACIENTE */
  pacienteForm: FormGroup;
  pacientes: any[] = []; // Array para almacenar los pacientes registrados
  submitted = false;

  doctor = {
    id: 0, // Este sería el ID del doctor autenticado
    name: 'Undefined'
  };
};
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
// Objeto para almacenar los datos del formulario
pacient = {
  nombre: '',
  apellido: '',
  genero: '',
  fecha_nacimiento: '',
  telefono: '',
  direccion: '',
  email: '',
  edad: '',
  enfermedad: '',
  características: ''
};

constructor(private FormBuilder: FormBuilder, private ApiService: ApiService, private
alertService: AlertService, private userService: UserService, private router: Router) {
  // Definición del formulario reactivo
  this.pacienteForm = this.formBuilder.group({
    matricula: ['', Validators.required], //Matricula
    nombre: ['', Validators.required], //Nombre
    apellido: ['', Validators.required], //Apellido
    genero: ['', Validators.required], //Genero
    fecha_nacimiento: ['', [Validators.required]], //Fecha Nacimiento
    telefono: ['', [ //Telefono de Emergencia
      Validators.required,
      Validators.pattern(/^[0-9]*$/), // Solo números
      exactLength(10) // Máximo de 10 dígitos
    ]
  ],
  direccion: ['', Validators.required], // Direccion
  email: ['', [Validators.required, Validators.email]], //Correo de Emergencia
  edad: ['', [Validators.required, Validators.min(20)]], //Edad
  historialMedico: ['', Validators.required], //Historial Medico
  alergias: ['', Validators.required], // Alergias que tiene
  características: ['', Validators.required], // Descripcion
  confirmacion: [false, Validators.requiredTrue]
});

ngOnInit(): void {
  this.doctor.id = this.userService.getDoctorId();
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
this.doctor.name = this.userService.getDoctorName();
console.log('Nombre: '+this.doctor.name + ' ID: '+this.doctor.id)

}

ngOnDestroy(): void {
  console.log('Componente destruido manualmente');
}

// Método para manejar el envío del formulario
onSubmit() {
  this.submitted = true;

  // Generar la fecha y hora actual en el formato [YYYY-MM-DD - HH:mm:ss]
  const now = new Date();
  const formattedDate = `${now.getFullYear()}-${(now.getMonth() +
1).toString().padStart(2, '0')}-${now.getDate().toString().padStart(2, '0')}`;
  const formattedTime = `${now.getHours().toString().padStart(2,
'0')}:${now.getMinutes().toString().padStart(2,
'0')}:${now.getSeconds().toString().padStart(2, '0')}`;
  const lastConsultation = `${formattedDate} - ${formattedTime}`;

  // Validación del formulario
  if (this.pacienteForm.invalid) {
    this.alertService.warning('Por favor, completa todos los campos requeridos.',
'Formulario Incompleto');
    console.log(this.pacienteForm.valid);
    return;
  }

  // Obtener datos del formulario y añadir el ID del doctor
  const pacienteData = {
    registration_number: this.pacienteForm.value.matricula,
    first_name: this.pacienteForm.value.nombre,
    last_name: this.pacienteForm.value.apellido,
    age: this.pacienteForm.value.edad,
    gender: this.pacienteForm.value.genero,
    birth_date: this.pacienteForm.value.fecha_nacimiento,
    emergency_contact: this.pacienteForm.value.telefono,
    emergency_email: this.pacienteForm.value.email,
    address: this.pacienteForm.value.direccion,
    medical_history: this.pacienteForm.value.historialMedico,
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
allergies: this.pacienteForm.value.alergias,
description: this.pacienteForm.value.caracteristicas,
last_consultation: lastConsultation, // Fecha y hora de la consulta
doctor_id: this.doctor.id // Suponiendo que el objeto `doctor` contiene los datos
del doctor actual
};

this.apiService.crearPaciente(pacienteData).subscribe({
  next: (response) => {
    this.alertService.success('Paciente registrado correctamente.', 'Éxito');
    console.log('Respuesta del servidor:', response);

    // Reiniciar el formulario
    this.pacienteForm.reset();
    this.submitted = false;
    this.router.navigate(['home/list-person']);
  },
  error: (error) => {
    this.alertService.error('Ocurrió un error al registrar el paciente. Por favor,
inténtelo de nuevo.', 'Error');
    console.error('Error al registrar paciente:', error);
  }
});

console.log(lastConsultation);
}
}

// Validador personalizado TELEFONO
export function exactLength(length: number): ValidatorFn {
  return (control: AbstractControl): ValidationErrors | null => {
    const value = control.value;
    if (value && value.length !== length) {
      return { exactLength: { requiredLength: length, actualLength: value.length } };
    }
    return null;
  };
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

Componente registro doctor (HTML y TypeScript)

```
<div class="container mt-5">
  <div class="row justify-content-center">
    <div class="col-md-6">
      <div class="card shadow">
        <div class="card-header text-center">
          <h2>Registro de Doctor <i class="fa-solid fa-user-doctor"></i></h2>
        </div>
        <div class="card-body">
          <form [formGroup]="registerForm" (ngSubmit)="registro()">
            <!-- Nombre -->
            <div class="mb-3">
              <label for="nombre" class="form-label">Nombre</label>
              <input id="nombre" type="text" class="form-control"
formControlName="nombre" />
              <div *ngIf="registerForm.get('nombre')?.invalid &&
registerForm.get('nombre')?.touched" class="text-danger">
                Este campo es obligatorio.
              </div>
            </div>

            <!-- Apellido -->
            <div class="mb-3">
              <label for="apellido" class="form-label">Apellido</label>
              <input id="apellido" type="text" class="form-control"
formControlName="apellido" />
              <div *ngIf="registerForm.get('apellido')?.invalid &&
registerForm.get('apellido')?.touched" class="text-danger">
                Este campo es obligatorio.
              </div>
            </div>

            <!-- Cédula -->
            <div class="mb-3 cedula">
              <label for="cedula" class="form-label">Cédula</label>
              <input id="cedula" type="text" class="form-control"
formControlName="cedula" />
              <div *ngIf="registerForm.get('cedula')?.invalid &&
registerForm.get('cedula')?.touched" class="text-danger">
                Este campo es obligatorio.
              </div>
            </div>
          </form>
        </div>
      </div>
    </div>
  </div>
</div>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
<!-- Profesión -->
<div class="mb-3">
  <label for="profesion" class="form-label">Profesión</label>
  <input id="profesion" type="text" class="form-control"
formControlName="profesion" />
  <div *ngIf="registerForm.get('profesion')?.invalid &&
registerForm.get('profesion')?.touched" class="text-danger">
    Este campo es obligatorio.
  </div>
</div>

<!-- Edad -->
<div class="mb-3">
  <label for="date" class="form-label">Fecha de Nacimiento</label>
  <input id="date" type="date" class="form-control"
formControlName="date" />
  <div *ngIf="registerForm.get('date')?.invalid &&
registerForm.get('date')?.touched" class="text-danger">
    Ingrese su Fecha de Nacimiento
  </div>
</div>

<!-- Edad -->
<div class="mb-3">
  <label for="edad" class="form-label">Edad</label>
  <input id="edad" type="number" class="form-control"
formControlName="edad" />
  <div *ngIf="registerForm.get('edad')?.invalid &&
registerForm.get('edad')?.touched" class="text-danger">
    La edad debe ser entre 18 y 99 años.
  </div>
</div>

<!-- Género -->
<div class="mb-3">
  <label for="genero" class="form-label">Género</label>
  <select id="genero" class="form-select" formControlName="genero">
    <option value="">Seleccione...</option>
    <option value="Masculino">Masculino</option>
    <option value="Femenino">Femenino</option>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
</select>
<div *ngIf="registerForm.get('genero')?.invalid &&
registerForm.get('genero')?.touched" class="text-danger">
  Este campo es obligatorio.
</div>
</div>

<!-- Email -->
<div class="mb-3">
  <label for="email" class="form-label">Email</label>
  <input id="email" type="email" class="form-control"
formControlName="email" />
  <div *ngIf="registerForm.get('email')?.invalid &&
registerForm.get('email')?.touched" class="text-danger">
    Ingrese un email válido.
  </div>
</div>

<!-- Contraseña -->
<div class="mb-3">
  <label for="password" class="form-label">Contraseña</label>
  <input id="password" type="password" class="form-control"
formControlName="password" />
  <div *ngIf="registerForm.get('password')?.invalid &&
registerForm.get('password')?.touched" class="text-danger">
    La contraseña debe tener al menos 6 caracteres.
  </div>
</div>

<!-- Teléfono -->
<div class="mb-3">
  <label for="telefono" class="form-label">Teléfono</label>
  <input id="telefono" type="text" class="form-control"
formControlName="telefono" maxLength="10"/>
  <div *ngIf="registerForm.get('telefono')?.invalid &&
registerForm.get('telefono')?.touched" class="text-danger">
    Ingrese solo números.
  </div>
</div>

<!-- Dirección -->
<div class="mb-3">
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
<label for="direccion" class="form-label">Dirección</label>
<textarea id="direccion" class="form-control"
formControlName="direccion"></textarea>
<div *ngIf="registerForm.get('direccion')?.invalid &&
registerForm.get('direccion')?.touched" class="text-danger">
  Este campo es obligatorio.
</div>
</div>

<!-- Imagen -->
<div class="mb-3">
  <label for="imagen" class="form-label"><i class="fa-solid fa-
image"></i> Imagen (opcional)</label>
  <input id="imagen" type="file" class="form-control"
(change)="onImageChange($event)" accept="image/*" />
</div>

<!-- Botón de Envío -->
<div class="d-grid">
  <button type="submit" class="btn btn-primary">Registrar</button>
</div>
</form>
</div>
</div>
</div>
</div>
</div>
</div>
import { Component, OnInit } from '@angular/core';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';
import { Router } from '@angular/router';
import { AlertService } from 'src/app/Service/alert.service';
import { ApiService } from 'src/app/Service/api.service';
import Swal from 'sweetalert2';

@Component({
  selector: 'app-registro',
  templateUrl: './registro.component.html',
  styleUrls: ['./registro.component.scss']
})
export class RegistroComponent implements OnInit{
  registerForm!: FormGroup;
  submitted = false;
```



TECNOLÓGICO NACIONAL DE MÉXICO

```
errors: any = {}; // Almacena mensajes de error personalizados
selectedImage!: File | null; // Almacena la imagen seleccionada

constructor(private fb: FormBuilder, private doctorsApi: ApiService, private router:
Router, private alert: AlertService){}

ngOnInit(): void {
  this.registerForm = this.fb.group({
    nombre: ['', Validators.required],
    apellido: ['', Validators.required],
    cedula: ['', Validators.required],
    profesion: ['', Validators.required],
    date: ['', Validators.required],
    edad: ['', [Validators.required, Validators.min(18), Validators.max(99)]],
    genero: ['', Validators.required],
    email: ['', [Validators.required, Validators.email]],
    password: ['', [Validators.required, Validators.minLength(6)]],
    telefono: ['', [Validators.required, Validators.pattern(/^\d+$/)]],
    direccion: ['', Validators.required],
    imagen: [''], // Imagen es opcional
  });
}

registro(){
  if (this.registerForm.invalid) {
    console.log('Formulario no válido');
    this.alert.warning('No deje campos vacios, por favor de llenarlo', 'Campos
Vacios');
    return;
  }

  const formData = new FormData();
  Object.keys(this.registerForm.value).forEach((key) => {
    if (key !== 'imagen') {
      formData.append(key, this.registerForm.get(key)?.value || '');
    }
  });
});

// Si se seleccionó una imagen, añadirla al FormData
if (this.selectedImage) {
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
formData.append('imagen', this.selectedImage);
}

if (this.registerForm.valid) {
  this.doctorsApi.registerDoctor(formData).subscribe(
    (response) => {
      console.log('Doctor registrado:', response);
      this.alert.success('Registro de usuario Exitoso', 'Succesfully Register');
      this.router.navigate(['home/list-doctors']);
    },
    (error) => {
      console.error('Error al registrar doctor:', error);
      this.alert.error('Error al registrar al Usuario', 'Failed Register');
    }
  );
} else {
  console.log('Formulario no válido');
  this.alert.warning('Error de Formulatio', 'Register Failed');
}
}

onImageChange(event: any): void {
  const file = event.target.files[0];
  if (file) {
    this.selectedImage = file;
  }
}
}
```

Componente vista perfil patient (HTML y TypeScript)

```
<!-- Visualizacion de Datos del Paciente -->
<div class="container">
  <!-- Botón Add fuera del cuadro -->
  <!-- <div class="d-grid gap-2 col-6 mx-auto">
    <button class="btn btn-success" type="button" data-bs-toggle="modal" data-bs-
target="#staticBackdrop"><i class="bi bi-plus-circle"></i> Añadir Paciente</button>
  </div> -->
  <div class="row justify-content-center">
    <div class="container my-4">
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
<h2>Detalles del Paciente</h2>
<div class="card">
  <div class="card-body">
    <!-- Información del paciente -->
    <div class="row mb-3">
      <div class="col-md-4"><strong>Matrícula:</strong></div>
      <div class="col-md-8">{{ patientData.registration_number }}</div>
    </div>
    <div class="row mb-3">
      <div class="col-md-4"><strong>Nombre:</strong></div>
      <div class="col-md-8">{{ patientData.first_name }}</div>
    </div>
    <div class="row mb-3">
      <div class="col-md-4"><strong>Apellido:</strong></div>
      <div class="col-md-8">{{ patientData.last_name }}</div>
    </div>
    <div class="row mb-3">
      <div class="col-md-4"><strong>Edad:</strong></div>
      <div class="col-md-8">{{ patientData.age }}</div>
    </div>
    <div class="row mb-3">
      <div class="col-md-4"><strong>Fecha de Nacimiento:</strong></div>
      <div class="col-md-8">{{ patientData.birth_date | date }}</div>
    </div>
    <div class="row mb-3">
      <div class="col-md-4"><strong>Correo de Emergencia:</strong></div>
      <div class="col-md-8">{{ patientData.emergency_email }}</div>
    </div>
    <div class="row mb-3">
      <div class="col-md-4"><strong>Teléfono de Emergencia:</strong></div>
      <div class="col-md-8">{{ patientData.emergency_contact }}</div>
    </div>
    <div class="row mb-3">
      <div class="col-md-4"><strong>Dirección:</strong></div>
      <div class="col-md-8">{{ patientData.address }}</div>
    </div>
    <div class="row mb-3">
      <div class="col-md-4"><strong>Historial Médico:</strong></div>
      <div class="col-md-8">{{ patientData.medical_history }}</div>
    </div>
    <div class="row mb-3">
      <div class="col-md-4"><strong>Alergias:</strong></div>
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
<div class="col-md-8">{{ patientData.allergies }}</div>
</div>
<!-- Botones Modificar y Eliminar dentro del cuadro -->
<div class="text-center mt-3 boxBtn">
  <button type="button" class="btn btn-warning mr-2"><i class="fa-regular fa-
pen-to-square"></i> Modificar</button>
  <button type="button" class="btn btn-danger" (click)="deletePatient()"><i
class="fa-solid fa-trash"></i> Eliminar</button>
  <button type="button" class="btn btn-success" (click)="categoria()"><i
class="fa-solid fa-file-medical"></i> Diagnostico</button>
  <button type="button" class="btn btn-history" (click)="history()"><i
class="fa-solid fa-book-medical"></i> Historial</button>
</div>
</div>
</div>
</div>
</div>
import { Component, OnInit } from '@angular/core';
import { ActivatedRoute, Route, Router } from '@angular/router';
import { AlertService } from 'src/app/Service/alert.service';
import { ApiService } from 'src/app/Service/api.service';

@Component({
  selector: 'app-vistas',
  templateUrl: './vistas.component.html',
  styleUrls: ['./vistas.component.scss']
})
export class VistasComponent implements OnInit {
  patientData: any;
  patientID?: number;

  constructor(private router: Router, private route: ActivatedRoute, private
patientService: ApiService, private alert: AlertService) {

  }

  ngOnInit(): void {
    // Obtener el patientId desde los parámetros de la ruta
    this.route.paramMap.subscribe(params => {
      this.patientID = +params.get('patientID')!;
    });
  }
}
```

TECNOLÓGICO NACIONAL DE MÉXICO

```
// Llamar a la API para obtener los detalles del paciente
this.patientService.getPatientById(this.patientID).subscribe(response => {
  this.patientData = response.patient;
});
});
}
categoria() {
  this.router.navigate(['home/diagnostic']);
}
history(){
  this.router.navigate(['home/dashboard'])
}
deletePatient(){
  this.route.paramMap.subscribe(params => {
    this.patientID = +params.get('patientID')!;

    this.alert
      .confirm('¿Estás seguro de que deseas eliminar este paciente?', 'Confirmación de
eliminación')
      .then((isConfirmed) => {
        if (isConfirmed) {
          // Si el usuario confirma, se realiza la eliminación
          this.patientService.deletePatient(this.patientID!).subscribe(
            (response) => {
              this.router.navigate(['home/list-person']);
              this.alert.success('Paciente eliminado correctamente', 'Éxito');
            },
            (error) => {
              this.alert.error('Ocurrió un error al eliminar el paciente', 'Error');
              console.error('Error al eliminar paciente:', error);
            }
          );
        }
      });
    });
  }
}
```