



TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE APIZACO

NOMBRE DEL PROYECTO

I'M TALENTY (FASE 2)

OPCIÓN QUE CORRESPONDA

TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL

PARA OBTENER EL TÍTULO PROFESIONAL DE:

INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES

PRESENTA:

GEOVANNI PATLANI AGUILAR

NOMBRE DEL ASESOR:

KATHY LAURA VARGAS MATAMOROS



APIZACO, TLAX., JULIO DEL 2025



Instituto Tecnológico de Apizaco
Sistemas y Computación

Apizaco, Tlaxcala, 30/junio/2025

Oficio No. S.C./OPV/0267/25

ASUNTO: Liberación de proyecto
para titulación integral

ISRAEL JORGE SÁNCHEZ SÁNCHEZ
JEFE DEL DEPTO. DE DIVISIÓN DE ESTUDIOS PROFESIONALES
P R E S E N T E.

POR ESTE MEDIO LE INFORMO QUE HA SIDO LIBERADO EL SIGUIENTE PROYECTO PARA LA TITULACIÓN INTEGRAL.

NOMBRE DEL EGRESADO:	PATLANI AGUILAR GEOVANNI
CARRERA:	INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES
NO. DE CONTROL:	19371048
NOMBRE DEL PROYECTO:	I'm talenty (fase 2)
PRODUCTO U OPCIÓN:	TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL

AGRADEZCO DE ANTEMANO SU VALIOSO APOYO EN ESTA IMPORTANTE ACTIVIDAD PARA LA FORMACIÓN PROFESIONAL DE NUESTROS EGRESADOS.

ATENTAMENTE

*Excelencia en Educación Tecnológica-
Pensar para Servir, Servir para Triunfar®*

MIQUELINA SÁNCHEZ PULIDO
JEFE DEL DEPTO. DE SISTEMAS Y COMPUTACIÓN

ccp. Coordinación de carrera
Expediente
MSP/mrol*



SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL
DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO
DEPARTAMENTO DE
SISTEMAS Y COMPUTACIÓN

KATHY LAURA VARGAS MATAMOROS
ASESORA INTERNA



2025
Año de
**La Mujer
Indígena**

Av. Instituto Tecnológico No. 418, San Andrés Ahuashuatepec,
Municipio de Tzompantepec, Tlaxcala, Mex. C.P. 90491
Tel. (241) 4172010 Ext. 115, e-mail: sistemas@apizaco.tecnm.mx
tecnm.mx | apizaco.tecnm.mx





Educación
Secretaría de Educación Pública



TECNOLÓGICO
NACIONAL DE MÉXICO



Instituto Tecnológico de Apizaco
División de Estudios Profesionales

Tzompantepec, Tlaxcala, **07/octubre/2025**

GEOVANNI PATLANI AGUILAR
EGRESADO(A) DE LA CARRERA DE INGENIERÍA
EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES
CON NÚMERO DE CONTROL 19371048.
P R E S E N T E.

Por este medio, me permito informar a usted que por aprobación de la comisión revisora asignada para valorar el trabajo que mediante la opción **TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL** y cuyo tema es: **"T'M TALENTY (FASE 2) "**, conforme a lo establecido en el procedimiento para la obtención del Título Profesional en el Instituto Tecnológico, la División de Estudios Profesionales a mi cargo le otorga la:

AUTORIZACIÓN DE IMPRESIÓN

Debiendo entregar **2 USB** del mismo, apropiadamente empastados, a la brevedad, para presentar su Acto de Recepción Profesional.

Sin otro particular por el momento me despido.

ATENTAMENTE

Excelencia en Educación Tecnológica®
Pensar para Servir, Servir para Triunfar®



CECILIA YARELLY ARREGUÍN LARA
JEFA DE LA DIVISIÓN DE ESTUDIOS PROFESIONALES

ccp. Archivo
CYAL/Gsc*



2025
Año de
La Mujer
Indígena

Av. Instituto Tecnológico No. 418, San Andrés Ahuashuatepec, Municipio de
Tzompantepec, Tlaxcala, Mex. C.P. 90491 Tel. (241) 4172010 Ext. 304. e-mail:
dep_titulacion@apizaco.tecnm.mx | apizaco.tecnm.mx



Dedicatoria

Dedico este proyecto a mi familia, que ha sido mi pilar más fuerte, mi refugio y mi mayor fuente de inspiración. A mis padres, por su sacrificio, su amor y su fe inquebrantable en mí. Cada logro que alcanzo es también suyo.

A quienes ya no están físicamente, pero siguen presentes en mi corazón y en cada paso que doy. Su recuerdo me impulsa a seguir adelante con determinación y gratitud.

Y a mí mismo, por no rendirme, por confiar en mis capacidades y por haber llegado hasta aquí con esfuerzo, dedicación y pasión por lo que hago.

Agradecimientos

Agradezco profundamente a todas las personas que formaron parte de este camino y que, de una u otra forma, contribuyeron a la realización de este proyecto.

A mi familia, especialmente a mis padres, **Erica Patlani y Rodolfo Martinez**, por su amor incondicional, su apoyo constante y por enseñarme con su ejemplo el valor del esfuerzo, la perseverancia y la humildad. Gracias por creer en mí incluso en los momentos en los que yo dudaba.

A mis asesores, tanto interno como externo, por su guía, paciencia y compromiso durante el desarrollo de este proyecto. Sus observaciones y acompañamiento fueron fundamentales para alcanzar los objetivos planteados.

A la empresa **SmartSoft America Business Applications**, por brindarme la oportunidad de integrarme a un entorno profesional real, donde pude aplicar mis conocimientos, aprender nuevas herramientas y crecer como futuro ingeniero.

A mis compañeros y amigos, por su apoyo, por compartir experiencias, por las palabras de aliento y por estar presentes en cada etapa de esta formación.

Finalmente, agradezco al **Instituto Tecnológico de Apizaco** y a todos los docentes que me acompañaron a lo largo de esta carrera. Gracias por sembrar en mí la pasión por la tecnología y por prepararme para los retos del mundo profesional.

INDICE

Dedicatoria	0
Agradecimientos.....	0
Resumen	4
Contenido	5
Introducción.....	5
Generalidades	6
Ubicación.....	6
Smartsoft America Business Applications	7
Misión	7
Visión	7
Valores	7
Reconocimientos y Certificaciones	8
Problemas a Resolver	8
Objetivo General.....	9
Objetivos Específicos	9
Justificación	9
Marco Teórico	10
Frameworks y Plataformas de Desarrollo Web	11
Generación de Informes y Análisis de Datos	11
Herramientas Tecnológicas.....	12
Metodología	12
Estándares de nombrado y convenciones.....	13
Nombrado de clases, variables y métodos	13
Restricciones para considerar en clases.....	14
Arquitectura de sistema Java	16
Diagrama de proceso	17
Diagrama E-R.....	18
Codificación	19
Comentarios generales	20

Estructura y Relación de Componentes en una Aplicación Java EE utilizando JPQL.....	22
Entity (Entidades)	22
DAO (Data Access Objects)	22
Business (lógica de negocio)	23
Controller (Controladores)	24
View (Vistas).....	24
Base de datos	25
Dar de alta un proyecto.....	26
Notas adicionales.....	34
NetBeans 8.2 (con su JDK y GlassFish).....	35
Navicat y MySQL 8.....	36
TortoiseSVN	37
Desarrollo de las interfaces del sistema	37
Composición del logotipo	38
Paleta de colores para tipografía	39
Paleta de colores para iconos	40
Paleta de colores para menús y secciones	40
Generalidades	41
Contenido General	42
Diseño de pantallas.....	54
Implementación	62
Pantalla de registro	62
Cuerpo de XHTML para el login.	63
Codigo para las validaciones de los campos.	63
Funcionamiento de las validaciones.	64
Recuperación de contraseña mediante correo	64
Funcionamiento del módulo de recuperar contraseña.....	64
Login.....	66
Funcionamiento de modulo.....	66
Página principal	66

Vacantes recomendadas	68
Mis postulaciones.....	69
Vacantes ofertadas / publicar vacantes	70
Procesos de reclutamiento	70
Modalidades de contratación.	71
Cuestionarios	72
Organizaciones registradas	73
Estudiantes	74
Empleados	74
Perfiles	75
Mis Eventos	75
Eventos	76
Mis pases.....	76
Áreas de empresa.....	77
Vinculaciones	77
Conclusiones.....	78
Recomendaciones	78
Experiencia Obtenida	79

Resumen

El presente proyecto tiene como objetivo el desarrollo de un sistema de recomendación de empleo orientado a mejorar la vinculación académica-laboral de estudiantes y egresados universitarios. A través del análisis de perfiles profesionales, habilidades y preferencias, la plataforma sugiere vacantes laborales alineadas con las competencias y aspiraciones de los usuarios.

El sistema fue desarrollado utilizando **Java 8**, con tecnologías como **HTML, CSS, SQL y JPQL**, y herramientas como **NetBeans, Navicat, TORTOISESVN 1.9.5** para el control de versiones y la gestión de base de datos. La interfaz fue diseñada para ser intuitiva y funcional, facilitando la interacción del usuario con las recomendaciones generadas.

Para garantizar un desarrollo ágil, iterativo y centrado en el usuario, se adoptó la **metodología Scrum**, permitiendo una planificación flexible, entregas incrementales y retroalimentación continua. Esta metodología facilitó la adaptación a los cambios y la mejora constante del sistema durante su construcción.

La lógica del sistema se basa en la comparación entre los perfiles de los candidatos y los requisitos de las vacantes, considerando factores como habilidades técnicas, experiencia y afinidad con el puesto. Las consultas estructuradas mediante **JPQL** permiten una gestión eficiente de la información y una respuesta rápida a las necesidades del usuario.

Los resultados obtenidos muestran que el sistema es capaz de generar recomendaciones personalizadas, mejorando la eficiencia en el proceso de búsqueda de empleo y facilitando la toma de decisiones tanto para los candidatos como para las organizaciones. Este proyecto representa una solución práctica y escalable para fortalecer la vinculación entre el ámbito académico y el mercado laboral.

Contenido

Introducción

En la actualidad, la búsqueda de empleo y la vinculación laboral representan un desafío constante para los recién egresados universitarios, quienes enfrentan un mercado laboral competitivo, cambiante y cada vez más digitalizado. A pesar de contar con formación académica sólida, muchos profesionales encuentran dificultades para insertarse laboralmente de manera efectiva, ya sea por la falta de experiencia, la desconexión entre su perfil y las vacantes disponibles, o la ausencia de herramientas tecnológicas que faciliten este proceso.

En respuesta a esta problemática, surge el desarrollo de una plataforma tecnológica orientada a mejorar la vinculación académica-laboral mediante un sistema de recomendación de empleo. Esta solución busca conectar de forma inteligente a estudiantes y egresados con oportunidades laborales acordes a sus habilidades, intereses y formación profesional, optimizando así el proceso de búsqueda y selección de empleo.

El sistema fue desarrollado utilizando **Java 8**, con tecnologías complementarias como **HTML, CSS, SQL y JPQL**, y herramientas como **NetBeans, Navicat y TORTOISESVN 1.9.5**. La plataforma permite a los usuarios crear un perfil profesional detallado, visualizar vacantes, postularse directamente y recibir recomendaciones personalizadas. Además, se implementó la **metodología ágil Scrum** para gestionar el desarrollo del proyecto, permitiendo una planificación iterativa, entregas incrementales y una mejora continua basada en la retroalimentación.

Este documento presenta el proceso de diseño, desarrollo e implementación del sistema, así como los fundamentos teóricos y técnicos que lo sustentan. Se abordan también los resultados obtenidos y las proyecciones de mejora, con el objetivo de contribuir a la transformación digital del proceso de inserción laboral en el entorno universitario.

Generalidades

Ubicación

Dirección física/fiscal: Calle Adolfo López Mateos 131 A, Texcacoac Chiautempan Tlaxcala, 90806.

Correo electrónico: contacto@smartsoftamerica.com.mx

Número de teléfono: (246) 117 5228

Sitio web: <https://smartsoftamerica.com.mx/>

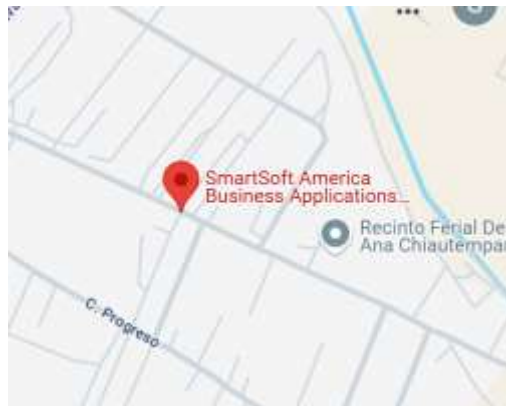


Ilustración 1.Ubicación

Logotipo de la empresa



Ilustración 2.Logotipo de la empresa

Logotipo de proyecto



Ilustración 3.Logotipo de proyecto

Smartsoft America Business Applications

En SmartSoft America BA, aplicamos el motor del cambio desde la fundación de la empresa en mayo de 2009, creamos productos con altos estándares tecnológicos, actualmente en el mercado contamos con productos que dan solución a las necesidades de nuestros clientes con sistemas enfocados a la Industria 4.0, de Internet de las Cosas, para PyMES, para el sector gubernamental, también para logística y diferentes Apps disponibles en Android y en IOS. Destacando el DESARROLLO DE SOFTWARE a la medida, como un servicio con calidad internacional.

Misión

Somos una empresa de Investigación, Innovación, y Desarrollo Tecnológico enfocado a crear y mejorar productos y servicios de Software e Industria 4.0 de calidad que permitan incrementar el desempeño, competitividad y satisfacción de nuestros clientes.

Visión

Ser una organización de clase mundial que brinde productos y servicios de Software e Industria 4.0 cumpliendo con los más altos estándares de calidad, compromiso y eficiencia en beneficio de nuestros clientes, tomando como base la investigación, innovación y el desarrollo tecnológico.

Valores

- Honestidad
- Responsabilidad
- Tolerancia
- Servicio
- Eficiencia
- Calidad

Reconocimientos y Certificaciones

- Evaluados CMMI DEV/5 y CMMI SVC/5
- Norma de Calidad para Desarrollo de Software MOPROSOFT (NMX-I-059/01-NYCE-2005).
- Contando con el Registro Nacional de Instituciones y Empresas Científicas y Tecnológicas (RENIECYT) por parte de CONACyT.
- Empresa afiliada al Clúster de Tecnologías de la Información del estado de Tlaxcala

Problemas a Resolver

En el contexto actual, los procesos de reclutamiento enfrentan múltiples desafíos que afectan tanto a las organizaciones como a los candidatos. Entre los principales problemas identificados se encuentran:

- **Desconexión entre perfiles y vacantes:** Muchos candidatos no logran encontrar ofertas laborales acordes a sus habilidades, formación o intereses, lo que genera frustración y baja efectividad en los procesos de vinculación.
- **Falta de automatización:** Las plataformas tradicionales no cuentan con mecanismos inteligentes que permitan filtrar, analizar y recomendar vacantes de forma personalizada.
- **Procesos lentos y poco eficientes:** La búsqueda manual de empleo y la revisión de perfiles por parte de las empresas consume tiempo y recursos, reduciendo la agilidad del proceso.
- **Ausencia de análisis de datos:** No se aprovechan herramientas tecnológicas para generar estadísticas, informes o patrones que permitan mejorar la toma de decisiones.
- **Limitada comunicación entre actores:** La interacción entre candidatos, instituciones educativas y empresas suele ser fragmentada, dificultando la colaboración efectiva.

Objetivo General

Desarrollar una plataforma web inteligente que permita la gestión y recomendación de ofertas laborales para estudiantes y egresados, optimizando el proceso de vinculación académica-laboral mediante el análisis de perfiles, habilidades y preferencias profesionales.

Objetivos Específicos

- Diseñar un sistema que automatice la generación y gestión de alertas sobre vacantes laborales relevantes para cada usuario.
- Implementar un módulo de análisis de habilidades y competencias que permita evaluar la afinidad entre candidatos y ofertas.
- Facilitar la comunicación entre empresas, instituciones educativas y candidatos mediante una interfaz intuitiva y funcional.
- Incorporar herramientas de análisis estadístico para generar reportes e indicadores sobre el comportamiento del sistema.
- Reducir los tiempos de búsqueda y selección de empleo mediante recomendaciones personalizadas y filtrado inteligente.

Justificación

La implementación de un sistema de recomendación de empleo representa una solución estratégica ante la creciente necesidad de mejorar la vinculación entre egresados universitarios y el mercado laboral. En la actualidad, muchos profesionales enfrentan dificultades para encontrar oportunidades laborales acordes a su perfil, debido a la falta de herramientas tecnológicas que faciliten este proceso de manera personalizada y eficiente.

Este psroyecto surge como respuesta a dicha problemática, con el objetivo de automatizar y optimizar el proceso de búsqueda de empleo mediante una plataforma inteligente que analice habilidades, experiencia y preferencias de los usuarios. La solución propuesta no solo beneficia a los candidatos, al ofrecerles vacantes alineadas con su perfil, sino también a las empresas, al permitirles identificar talento de forma más rápida y precisa.

Además, la plataforma contribuye a reducir los tiempos de búsqueda, mejorar la calidad de las contrataciones y fortalecer la comunicación entre instituciones educativas, egresados y organizaciones. Su desarrollo bajo la metodología ágil Scrum garantiza una evolución continua del sistema, adaptándose a las necesidades reales del entorno laboral.

Marco Teórico

El desarrollo de sistemas de recomendación se fundamenta en diversas disciplinas como la inteligencia artificial, el aprendizaje automático y la ingeniería de software. Estas tecnologías permiten analizar grandes volúmenes de datos para generar sugerencias personalizadas, mejorando la experiencia del usuario y la eficiencia de los procesos.

En el ámbito de la ingeniería de software, se han desarrollado múltiples metodologías y modelos para garantizar la calidad, escalabilidad y mantenibilidad de los sistemas. Entre ellos, destaca la metodología ágil Scrum, que permite una gestión iterativa e incremental del proyecto, facilitando la adaptación a cambios y la entrega continua de valor.

Los sistemas de recomendación aplicados al empleo utilizan técnicas como el análisis de contenido, el filtrado colaborativo y el procesamiento de lenguaje natural para evaluar la afinidad entre perfiles profesionales y vacantes. Estas herramientas permiten identificar patrones, habilidades clave y coincidencias entre candidatos y ofertas, mejorando la precisión en la selección.

Asimismo, el uso de tecnologías como Java, HTML, CSS, SQL y JPQL, junto con herramientas como NetBeans, Navicat y TORTOISESVN 1.9.5, proporciona una base sólida para el desarrollo de aplicaciones web robustas, seguras y escalables, capaces de integrarse con bases de datos y ofrecer una experiencia de usuario fluida.

Frameworks y Plataformas de Desarrollo Web

Para el desarrollo de la plataforma de recomendación de empleo, se utilizaron diversas tecnologías y frameworks que permitieron construir una solución robusta, escalable y eficiente. El sistema fue desarrollado en **Java SE 8**, aprovechando su estabilidad y compatibilidad con aplicaciones empresariales. La interfaz web se implementó utilizando **Java Server Faces (JSF)**, un framework que facilita la creación de interfaces de usuario basadas en componentes reutilizables.

Además, se emplearon tecnologías complementarias como **HTML** y **CSS** para el diseño visual de la plataforma, asegurando una experiencia de usuario intuitiva y adaptable. La gestión de la base de datos se realizó mediante **SQL** y consultas estructuradas con **JPQL**, utilizando **Navicat** como herramienta de administración. El entorno de desarrollo principal fue **NetBeans**, y se utilizó **TORTOISESVN 1.9.5** para el control de versiones, lo que permitió mantener un flujo de trabajo ordenado y colaborativo.

Estas herramientas y frameworks permitieron construir una plataforma web moderna, con capacidad de integración, mantenimiento y evolución continua.

Generación de Informes y Análisis de Datos

Uno de los componentes clave del sistema es la capacidad de generar informes y realizar análisis de datos para apoyar la toma de decisiones. Para ello, se integraron herramientas que permiten procesar, visualizar y exportar información relevante sobre el comportamiento de los usuarios, las vacantes más consultadas, los perfiles más demandados y la efectividad de las recomendaciones.

El sistema permite generar reportes personalizados en formatos descargables, como PDF, que incluyen métricas clave sobre postulaciones, coincidencias entre perfiles y vacantes, y estadísticas de uso. Estos informes son útiles tanto para los administradores del sistema como para las instituciones educativas y empresas que deseen evaluar el impacto de la plataforma.

Herramientas Tecnológicas

Para el desarrollo del sistema de recomendación de empleo y vinculación académica-laboral, se emplearon diversas herramientas tecnológicas que permitieron construir una plataforma robusta, funcional y adaptable a las necesidades del usuario final. Estas herramientas fueron seleccionadas por su compatibilidad, eficiencia y facilidad de integración en entornos de desarrollo web.

Las principales herramientas utilizadas fueron:

- **Java SE 8:** Lenguaje de programación principal del sistema, elegido por su estabilidad, orientación a objetos y amplia adopción en entornos empresariales.
- **NetBeans:** Entorno de desarrollo integrado (IDE) utilizado para la codificación, depuración y gestión del proyecto.
- **HTML y CSS:** Tecnologías empleadas para la construcción de la interfaz gráfica del usuario, permitiendo una experiencia visual clara, responsiva y amigable.
- **SQL y JPQL:** Lenguajes utilizados para la gestión y consulta de bases de datos relacionales. JPQL (Java Persistence Query Language) permitió realizar consultas orientadas a objetos sobre entidades persistentes.
- **Navicat:** Herramienta de administración de bases de datos que facilitó la conexión, diseño y mantenimiento de la base de datos del sistema.
- **TORTOISESVN 1.9.5:** Sistema de control de versiones utilizado para gestionar los cambios en el código fuente, permitiendo un trabajo colaborativo ordenado y seguro.

Metodología

Para el desarrollo del sistema de recomendación de empleo, se adoptó la **metodología ágil Scrum**, una de las más utilizadas en la industria del software por su enfoque iterativo, incremental y centrado en el usuario. Esta metodología permitió organizar el trabajo en ciclos cortos llamados **sprints**, facilitando la entrega continua de funcionalidades y la adaptación a cambios durante el proceso de desarrollo.

Scrum se estructuró en torno a los siguientes elementos clave:

- **Product Backlog:** Se definió una lista priorizada de funcionalidades y requerimientos del sistema, incluyendo módulos como el registro de usuarios, la visualización de vacantes, el motor de recomendaciones y la generación de reportes.
- **Sprint Planning:** Al inicio de cada sprint, se seleccionaron las tareas a desarrollar, estableciendo objetivos claros y alcanzables en un periodo determinado (generalmente de una a dos semanas).
- **Daily Scrum:** Se realizaron reuniones breves diarias para revisar el avance, identificar obstáculos y coordinar esfuerzos entre los miembros del equipo.
- **Sprint Review:** Al finalizar cada sprint, se presentó una versión funcional del sistema para recibir retroalimentación por parte de usuarios o asesores.
- **Sprint Retrospective:** Se evaluó el desempeño del equipo y se propusieron mejoras para los siguientes ciclos de trabajo.

Estándares de nombrado y convenciones

Nombrado de clases, variables y métodos

El nombrado de todo el código debe ser bajo el estándar CamelCase. Este es un estilo de escritura que se aplica a frases o palabras compuestas. El nombre se debe a que las mayúsculas a lo largo de una palabra en CamelCase se asemejan a las jorobas de un camello.

La notación para nombrado de variables y métodos es:

- lowerCamelCase: la primera letra de cada una de las palabras es mayúscula, Excepto la primera letra es minúscula
 - EjemploDeUpperCamelCase.

La notación para el nombrado de clases es:

- UpperCamelCase, cuando la primera letra de cada una de las palabras es mayúscula.
 - EjemploDeUpperCamelCase

Adicional a esto, el nombrado debe ser de acuerdo con la acción. El nombrado de métodos y variables debe expresar el contenido o función que representan. No se deben usar abreviaciones, letras o número en el nombrado. Ejemplo:

Variables:

- Lista<CatUsuario> listaUsuarios,
- String nombreDeUsuario
- Int edadDeUsuario.

Métodos

- guardarCatUsuario(CatUsuario usuario)
- eliminarCatUsuario(CatUsuario usuario)

solo se podrá omitir en el nombrado la palabra correspondiente a la entidad que utiliza el módulo, si este método está dentro de la clase correspondiente a la entidad. Ejemplo:

```
public class CatUsuarioBusiness{  
    guardar(CatUsuario usuario){  
        ...  
    }  
    eliminar (CatUsuario usuario){  
        ...  
    }  
}
```

Restricciones para considerar en clases

La codificación de las distintas actividades de construcción debe ser bajo los siguientes aspectos:

Backend todas las clases .java

- Todas las variables, declaradas y de modo privado
- Declaración de variables e inyección e capa de negocio y persistencia, en el encabezado
- Registro de dependencias complete en archive pomp, cargado en svn
- Cargadas las dependencias que no estén en maven, deben estar en una carpeta en svn

- Base de datos, archive de tipo script y modelo de base datos
- Todos, deben tener get y set, no deben tener is
- No hacer uso redundante de variables condicionantes
- No hacer uso de ciclos for o while y sus derivados
- No hacer uso de impresión a consola, mediante System, en su caso se debe hacer uso de Logger
- Todos los try-cath, deben tener un logger
- Los try-cath, solo deben estar siendo usados en Controller y WebServices
- Las excepciones, no deben usar las genéricas, debe ser la especifica en try-cath
- Solo debe haber un controller de tipo sesión, los demás deben ser de tipo vista
- Si se hace uso de vistas o procedimientos almacenados, debe haber una clase denominada view y dentro debe tener todos los scrips
- En Sesionbussines debe tener ciclado la ejecución de Garbage Collector
- Nombramiento de clases, metodos y variables, debe ser como CamelCase
- Todas las inyecciones de bussines y dao, deben estar como private
- No tener librerías importadas que no se usen

FrontEnd, todas las interfaces .xhtml:

- El login debe de estar separado del témpate de sistema
- No tener código comentado, los comentarios, si se permiten

Controlador de vista: Controller

- No debe haber lógica de negocio
- No debe haber variables declaradas, solo el view
- Variable de view e inyecciones deben estar en modo private
- No debe haber lógica de negocio

Control de variables temporales: View

- Las variables deben estar en modo private

Lógica de negocio: Business.java

- No hacer uso de conectores a base de datos

Persistencia de datos: Dao.

- Debe extender a EntityManager
- No hacer uso de queries, deben ser queries nombradas
- No debe de haber lógica de negocio

Arquitectura de sistema Java

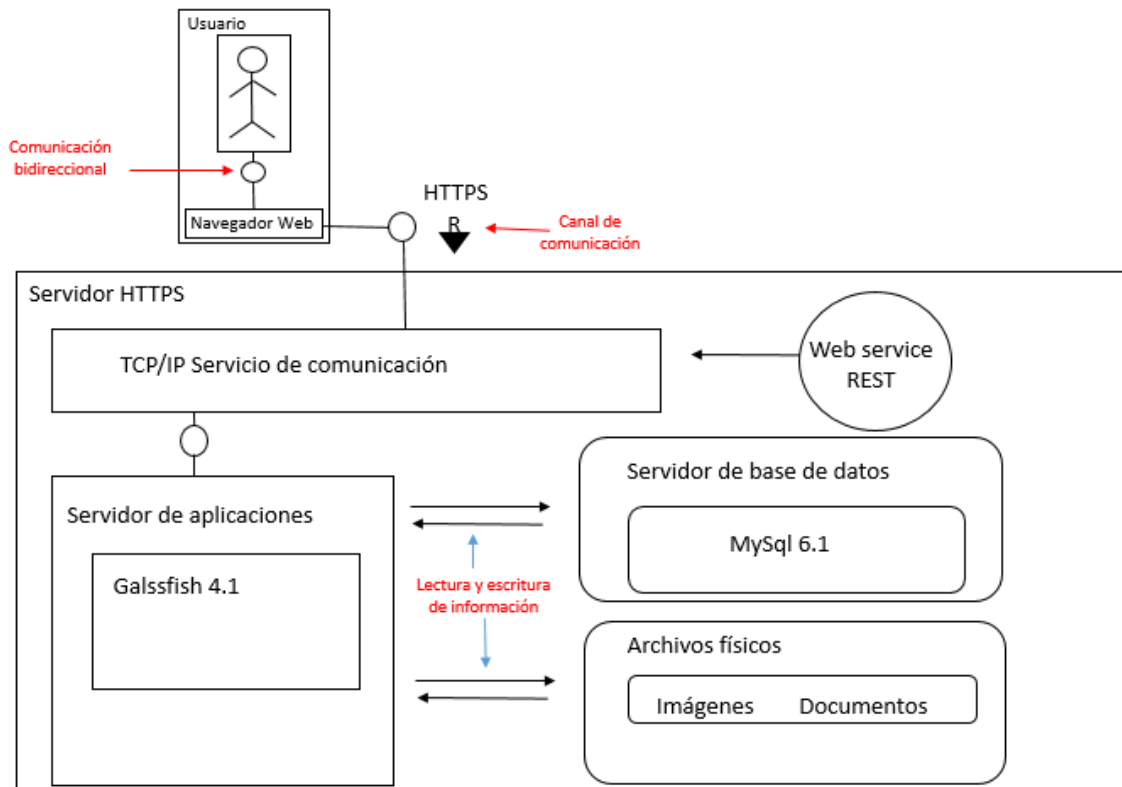


Ilustración 4.Arquitectura de sistema Java

El servicio es desarrollado y publicado en servidores virtuales. Cada sistema publicado se basa en lo siguiente:

- **Servidor de Aplicaciones.** Es el encargado de administrar los recursos informáticos disponibles en el servidor físico para las aplicaciones que aloje. Proporciona un conjunto de servicios a las aplicaciones con la finalidad de facilitar su gestión, monitoreo y control. El servidor de aplicaciones por referencia es Glassfish en su versión 4.1

Dentro del servidor de aplicaciones, se tiene lo siguiente:

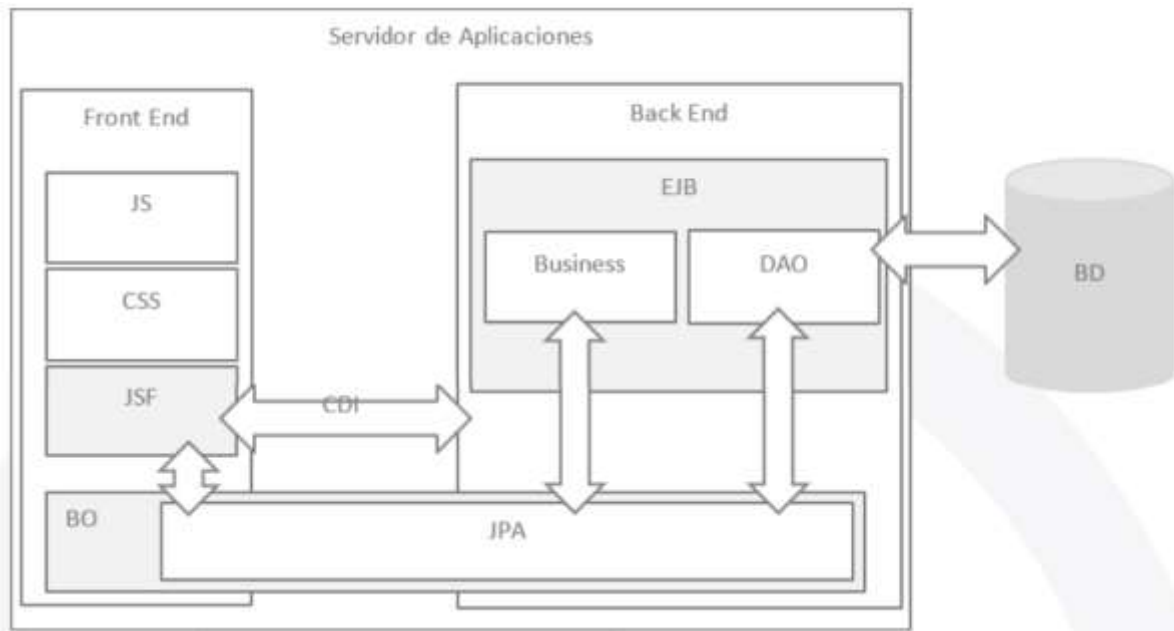


Ilustración 5. Servidor de Aplicaciones

Diagrama de proceso

El diagrama de proceso (BPMN), deberá estar apegado a la notación bizagi, anotado en las referencias de este mismo documento.

El proceso deberá mapear de manera clara y concisa cada uno de los requerimientos a fin de cumplir con todo el proceso requerido por el cliente.

Se debe tener un diagrama con subprocessos para indicar de manera detallada, cada aspecto y su forma de relacionarse entre cada uno de los requerimientos.

El diagrama de proceso deberá estar desarrollado al Nivel 3ª. Modelo Técnico: El modelo técnico es la representación del modelo operacional en un Process Engine, pero adaptando el proceso de negocio a un modelo ejecutable y enriqueciéndolo con aspectos técnicos. Como en realidad no siempre se implementan los modelos de negocio con un Process Engine, dividimos este nivel en un modelo técnico (3a), en el cual en un principio se sigue detallando el modelo de negocio en un BPMS y la opción de un desarrollo propio (3b). En BPMN podemos especificar el modelo técnico directamente en un Process Engine a partir de BPMN 2.0. Para más detalle de puede consultar la referencia: introducción a BPMN indicada en las referencias de este documento.

Diagrama E-R

Para la creación de este diagrama, se debe considerar lo siguiente:

1. Determinar las entidades.
 - a. Son las tablas que se requieren para cumplir con el almacenamiento de información de los requerimientos y funcionamiento del sistema
2. Identificar las relaciones entre cada entidad.
 - a. Se debe identificar el tipo de relación que se encuentra entre la información.
 - b. Nos e debe usar relaciones muchos a muchos, se debe crear una tabla intermedia, pero don llave primaria. Esto para facilitar el proceso de mapeo y manejo de datos por parte de JPA
3. Definir los atributos.
 - a. Esto se refiere a los campos necesarios para el funcionamiento de los requerimientos y o del sistema.
 - b. Los campos que expresen valor monetario deben usar el tipo BigDecimal(19,4)
 - c. Los campos que expresen datos de contenido amplio deben usar el tipo Text()
 - d. Todas las tablas del sistema deberán tener la columna:
 - Institución, tipo int (11)
 - e. Todas las tablas deberán tener las siguientes columnas denominadas "columnas de auditoria":
 - Fecha_registro tipo DateTime
 - Fecha_edicion tipo DateTime
 - Fecha_elimino tipo DateTime
 - Id_usuario_registro tipo int(11)
 - Id_usuario_edito tipo int(11)
 - Id_usuario_elimino tipo int(11)Las cuales deben ser usadas para registrar el dato que es indicado respectivamente.

Se debe considerar que el diagrama, debe estar normalizado a la 3NF forma normal. La Tercera Forma Normal (3FN), consiste en que ningún atributo dato que depende de la PK, dependa de otro atributo dato. Es decir, no debe tener dependencia transitiva.

Codificación

La estructura de codificación deberá ser la siguiente:

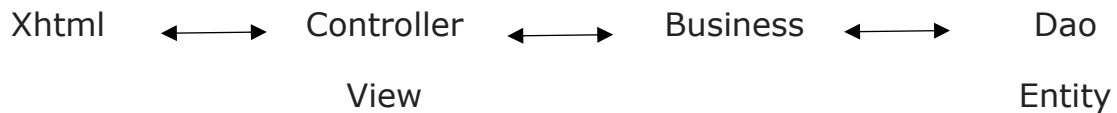


Ilustración 6. estructura de codificación

Cada clase o interfaz, deberá tener por nombre el módulo al que corresponde seguido de su terminación de acuerdo con el diagrama anterior.

El contenido de cada clase será el siguiente:

- XHTML
 - Contendrá la interfaz que se le mostrará al usuario
- Controlar.
 - Contendrá los métodos necesarios para poder controlar la interfaz que se expone al usuario
- View
 - Contendrá las variables o datos necesarios que la interfaz muestre al usuario
- Bussines
 - Debe contener toda la lógica de negocio. Esta es la única clase donde se pueden hacer cálculos o procesamiento de datos.
- Dao
 - Esta clase será la encargada de persistir los datos a la base de datos. No se puede hacer nada de procesamiento. Solo consultas, edición y guardado de datos
 - Se debe considerar en el método de eliminación, que la información jamás deberá ser eliminada físicamente, solo lógicamente.
- Entity
 - Contiene el mapeo de la tabla contra la base de datos.

Comentarios generales

- En las clases "Controller" solo debe haber una sola inyección de business que sería el relacionado al proceso, por ejemplo, `RecepcionCorrespondenciaController` → `RecepcionCorrespondenciaBusiness`
- Puede existir algún otro Controller inyectado como el `SesionController` o `ApplicationController` que fungen como clases de apoyo al sistema general, todo esto va a depender de la necesidad y naturaleza del proyecto. Además de analizar la utilización del método "init" de la clase en caso de ser necesario.
- Antes de integrar lógica de negocio a una clase, analizar las inyecciones con las que ya se cuente para poder reutilizar lo que se pueda y evitar relaciones circulares entre las clases.
EVITAR EN MEDIDA DE LO POSIBLE TENER 2 VECES INYECTADO EL MISMO BUSINESS.
- Antes de crear un nuevo método dependiendo de la naturaleza del cambio buscar primero si no hay alguno otro método que realice esa función, para evitar mantener 2 métodos que hacen lo mismo, pero con diferente nombre.
- Analizar la implementación de datos primitivos y compuestos dependiendo de la naturaleza del cambio.
- Integrar en la lógica de negocios el encapsulamiento de métodos para aligerar la carga de la edición del código, por ejemplo: cuando se crea un objeto en diferentes partes de la clase, este se puede extraer a un método en específico que haga el creado el objeto y solo integrarlo donde sea necesario.
- Evitar subir las impresiones de pantalla al svn, en lugar de eso implementar try catch para el manejo de excepciones, la única excepción sería el rastreo de información de procesos complejos para lo cual una vez terminadas las pruebas las impresiones se deben eliminar y de las cuales se les debe agregar el comentario "//impresión de prueba" con el fin de poder identificarlas y no eliminarlas entre los compañeros.
- Evitar la duplicidad de nombres de named queries o variables en la view con cosas extras como números o caracteres extra,

por ejemplo: En la clase view se tiene el objeto "listaFoliosPendientes" y "listaFoliosPendientes2".

- Cuando se estén trabajando actividades de algún grupo de WhatsApp responder la actividad con un comentario sobre el nombre de la persona que atenderá dicha actividad, todo esto para poder identificar las que se estén trabajando, de igual manera al finalizar indicar que ya se puede validar.
- Eliminar los métodos comentados una vez que se valide su inutilidad y así mismo eliminar de las clases las imports no utilizados.
- Los métodos creados deben de tener como límite de 15 a 20 líneas.
- Clases: Los nombres de clase deben nombrados con sustantivos.
- Métodos: Los métodos deben ser nombrados con verbos.
- Variables: Las variables locales, variables de instancia y variables de clase deben indicar la intención de su uso.
- Constantes: Los nombres de las constantes se deben escribir en mayúsculas separadas por guiones bajos.

Estructura y Relación de Componentes en una Aplicación Java EE utilizando JPQL

Entity (Entidades)

- Qué son: Las entidades son clases Java que representan las tablas de la base de datos. Cada instancia de una entidad corresponde a una fila en la tabla.
- Las entidades son las primeras en ser definidas porque son fundamentales para la estructura de datos de la aplicación.
- Relación: Utilizan anotaciones JPA (Java Persistence API) para mapear las clases a las tablas de la base de datos. Definen consultas nombradas (@NamedQuery) que se pueden utilizar en JPQL para realizar operaciones de búsqueda y manipulación de datos.

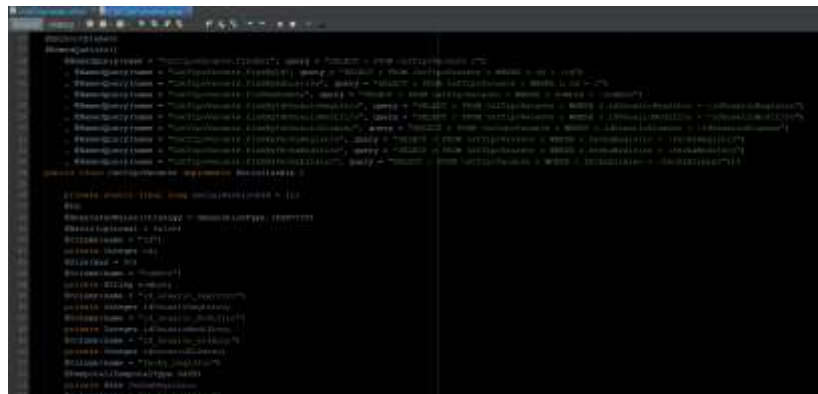


Ilustración 7. captura de entity

DAO (Data Access Objects)

- Qué son: Los DAOs son clases que proporcionan una capa de abstracción para las operaciones de acceso a datos. Encapsulan la lógica necesaria para interactuar con la base de datos.
- Los DAOs se crean después de las entidades. Proporcionan métodos para realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre las entidades.
- Relación: Utilizan EntityManager para ejecutar consultas JPQL y transacciones en la base de datos. Los DAOs llaman a las consultas nombradas definidas en las entidades para realizar operaciones específicas.

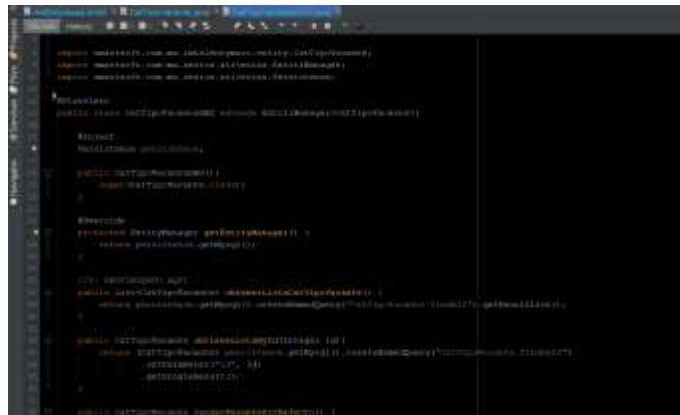


Ilustración 8. captura de DAO

Business (lógica de negocio)

- Qué es: La capa de negocio contiene la lógica de negocio de la aplicación. Implementa las reglas y procesos que no están directamente relacionados con la interfaz de usuario ni con el acceso a datos.
- La capa de negocio se desarrolla después de los DAOs. Implementa la lógica de negocio y las reglas de la aplicación.
- Relación: Utiliza los DAOs para interactuar con la base de datos y realizar operaciones complejas que pueden involucrar múltiples entidades y DAOs. Coordina las operaciones necesarias para cumplir con los requisitos de negocio.

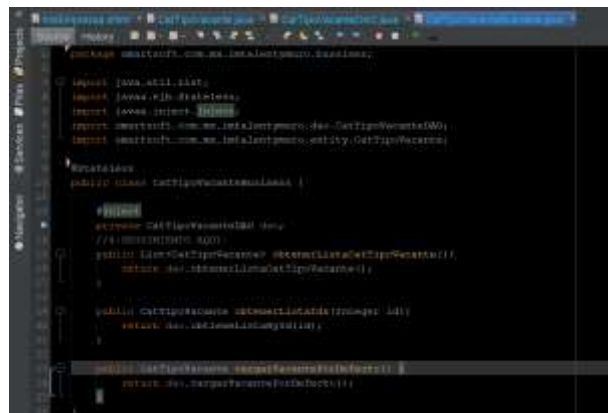


Ilustración 9. captura de business

Controller (Controladores)

- Qué son: Los controladores son clases que manejan la lógica de la interfaz de usuario y la interacción del usuario. Actúan como intermediarios entre la vista y la lógica de negocio.
- Los controladores se crean después de la capa de negocio. Manejan la lógica de la interfaz de usuario y la interacción del usuario.
- Relación: Llamam a los métodos de la capa de negocio para procesar las entradas del usuario y actualizar la vista con los resultados. Gestionan la navegación y el estado de la interfaz de usuario.

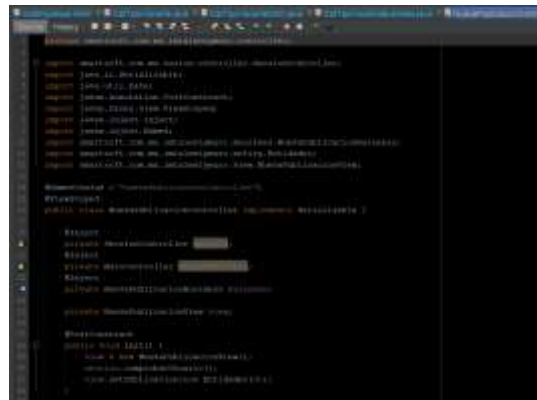


Ilustración 10.captura de Controller

View (Vistas)

- Qué son: Las vistas son componentes que definen la interfaz de usuario de la aplicación. Utilizan tecnologías como JSF (JavaServer Faces) y PrimeFaces para construir las páginas web.
- Las vistas se desarrollan después de los controladores. Definen la interfaz de usuario utilizando tecnologías como JSF y PrimeFaces.
- Relación: Utilizan beans de vista para mostrar datos y proporcionar controles de entrada. Los controladores actualizan estos beans de vista con los datos procesados por la capa de negocio, permitiendo que la interfaz de usuario refleje el estado actual de la aplicación.

Dar de alta un proyecto

- Creas una carpeta con el nombre del proyecto

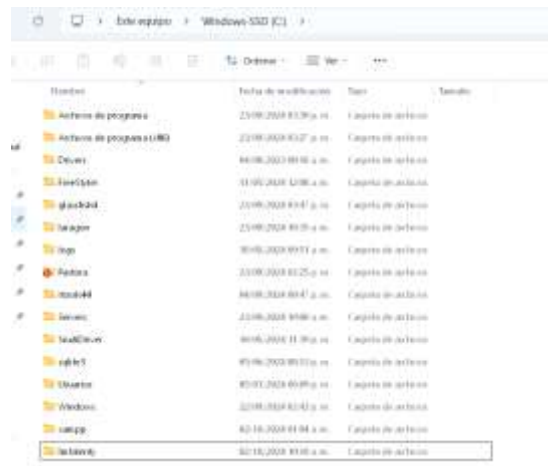


Ilustración 13. creación de carpeta

- Le das click derecho y en la opción de más opciones y en svncheckout

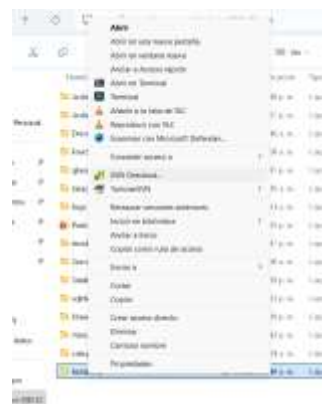


Ilustración 14. descargar de repocitorio

- Pegas la URL del proyecto y le das en ok



Ilustración 15. link de repertotio

- En el navit creas una base de datos con el nombre de la base de datos del proyecto

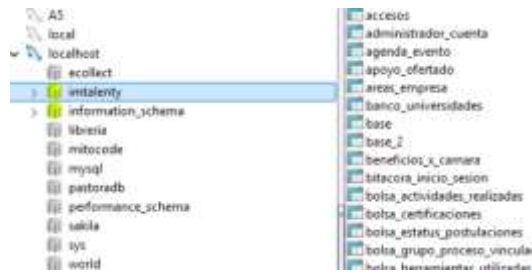


Ilustración 16. nombrado de la base de datos

- Click derecho y le das en execute Sql fie



Ilustración 17. exportación de la base de datos

- En la parte de file buscas el archivo sql del proyecto y le das click en el botón start

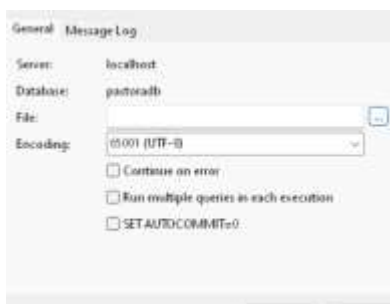


Ilustración 18. selección de la base de datos

- Para añadir el servicio de glasfish tienes que pegar la carpeta de glash fish en el disco local c

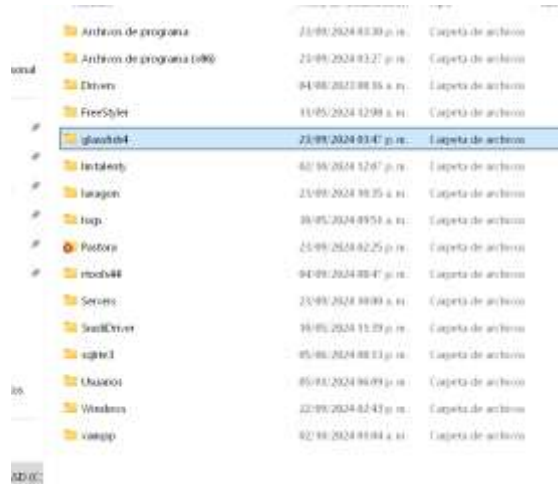


Ilustración 19. añadir el servicio de glassfish

- para importarla desde netbeans tienes que irte al apartado de services y en la sección de servers tienes que dar click derecho y poner add server

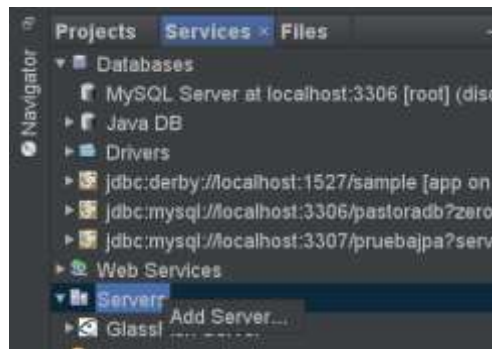


Ilustración 20.add server

- Seguimos la siguiente configuración y damos click en next

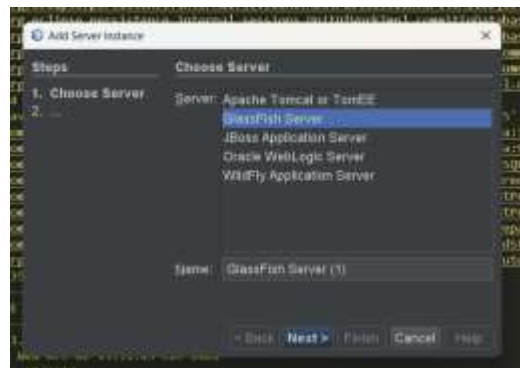


Ilustración 21. Configuración de glassfish

- Damos click en browse y buscamos el archivo que pasamos al disco local.

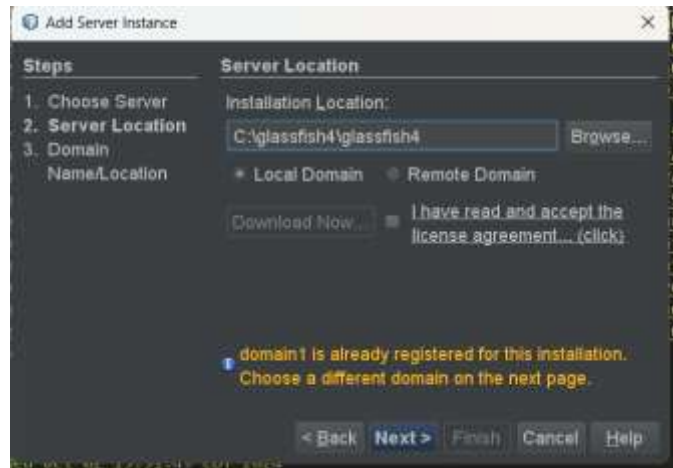


Ilustración 22. selección de glassfish

- En la siguiente ruta se debe pegar el controlador de la base de datos el mysql conector

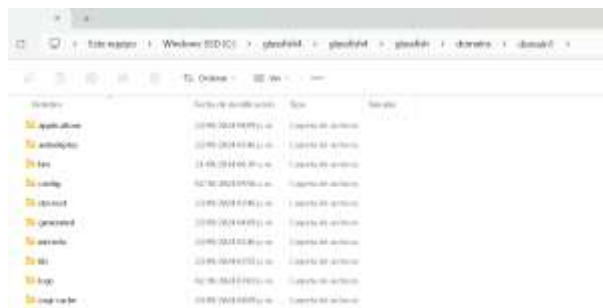


Ilustración 23, ruta para el controlador de bd

- para instalarlo vamos a services, en el apartado databases damos click en la opción de new connection

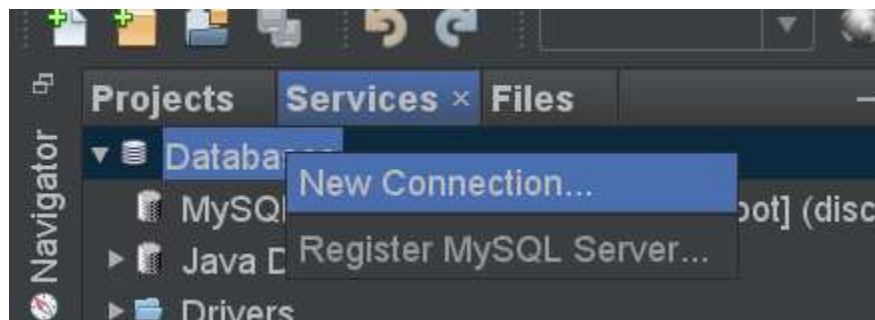


Ilustración 24. nueva conexión

- Escogemos la opción de mysql y seleccionamos el driver por defecto y le damos en remove, después de eso damos click en add

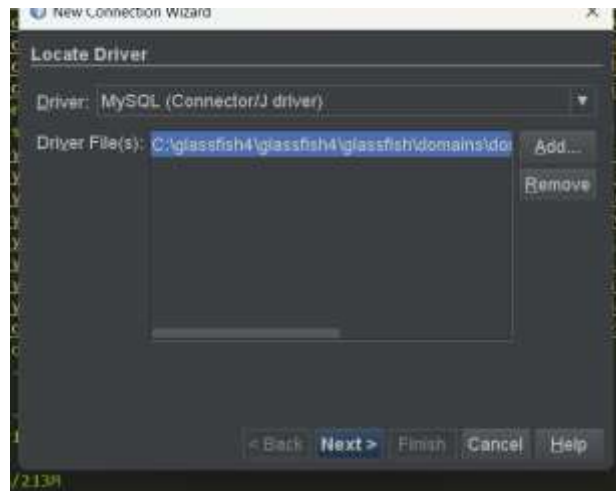


Ilustración 25

- Buscamos el archivo que nosotros movimos al glassfish

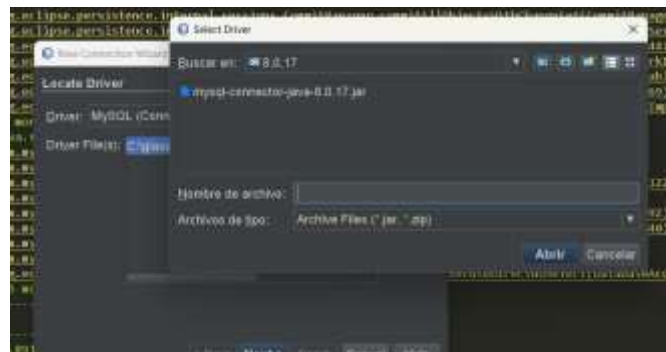
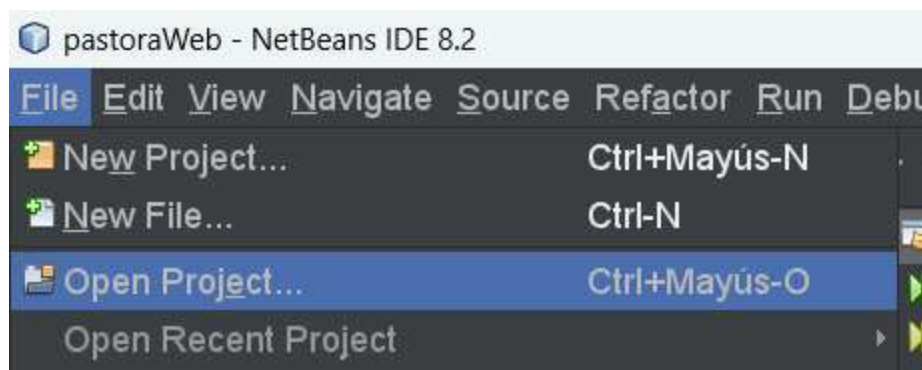


Ilustración 26. selección de ruta para el glassfish

- Damos click en open Project



- Buscamos la dirección de nuestro disco local, en mi caso i´m talenty, trunk, código y seleccionamos el archivo im talenty muro

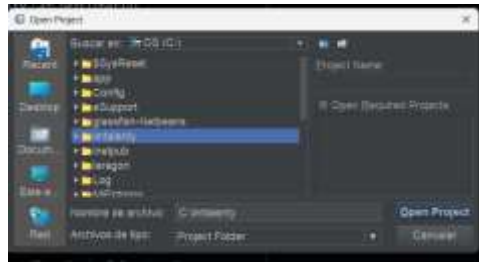


Ilustración 27.abrir proyecto

- Se abre el proyecto

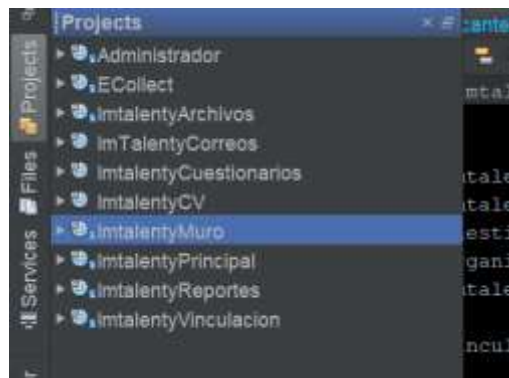


Ilustración 28. proyecto correspondiente

- Dependicis le damos click a la dependencia que falla

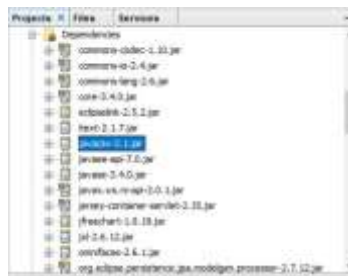


Ilustración 29. selección de dependencias

- Damos click en la siguiente opción

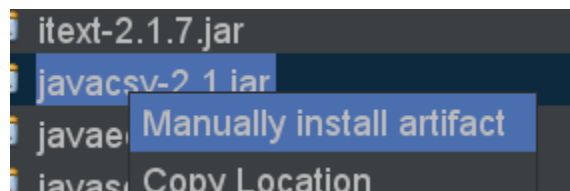


Ilustración 30. manual de instalacion para la dependencias

- Buscamos la dependencia por la que vamos a instalar y le damos en siguiente



Ilustración 31. búsqueda de configuración

nos vamos al archivo de settings para modificarlo



Ilustración 32. configuración modificada para el proyecto

- En la siguiente parte del código pegamos lo siguiente



Ilustración 33. modificación de configuracion

<mirror>

<id>mirrorId</id>

<mirrorOf>repositoryId</mirrorOf>

<name>Human Readable Name for this Mirror.</name>

<url>http://my.repository.com/repo/path</url>

</mirror>

<mirror>

<id>mirror1</id>

<mirrorOf>central</mirrorOf>

<name>mirror1</name>

<url>https://repo.maven.apache.org/maven2/</url>

</mirror>

- En la sección de run nos vamos a la parte que dice set Project configuración y damos click en customize

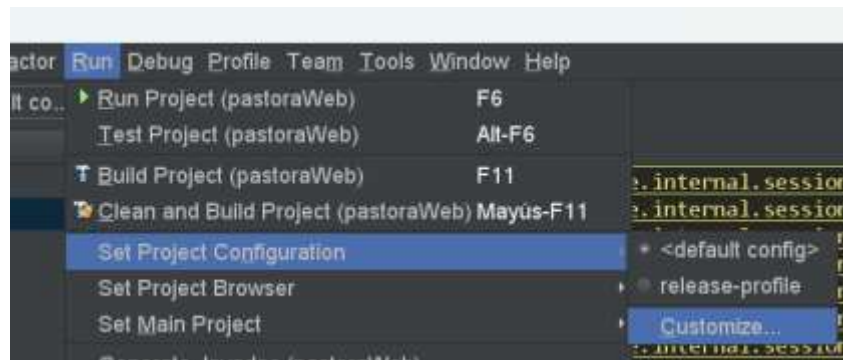


Ilustración 34. Selección para el arranque de proyecto

- Nos vamos a la sección de run para ver si todas las casillas están marcadas, si no están marcadas todas, los marcamos todos

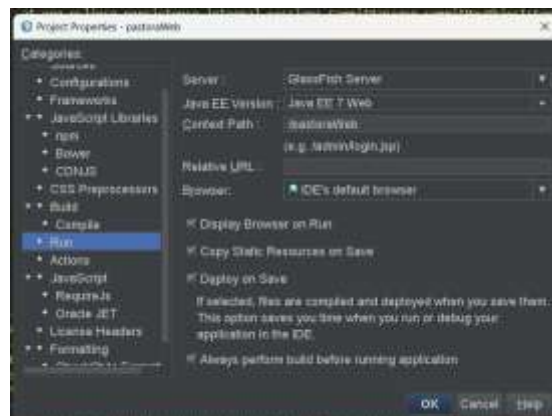


Ilustración 35. configuración de arranque

- Damos clic en clean and build y despues en run para que nos corra nuestro programa

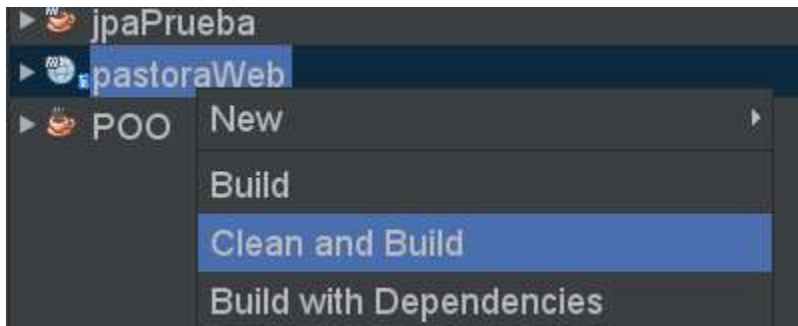


Ilustración 36. levantamiento de proyecto

Notas adicionales

Si el navicat no conecta con el workbench pegar el siguiente pedazo de código en la consola de workbench

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password
BY 'root';
```

flush privileges;

*La configuración del navicat y del workbench debe ser el siguiente



Ilustración 37. configuración de ruta para navicat

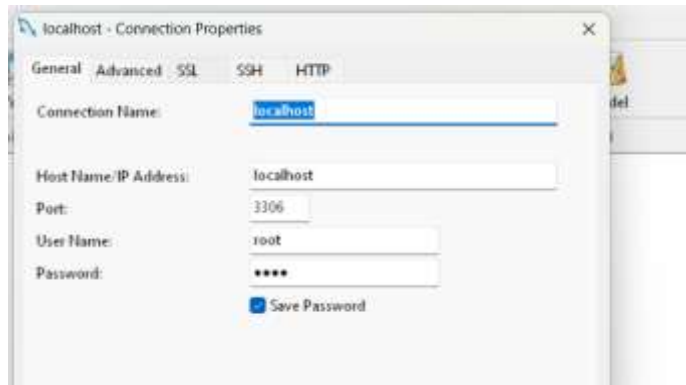


Ilustración 38. configuración de conexión

- para modificar el archivo de setting le debes dar permiso de modificar a todos los usuarios en la parte control total

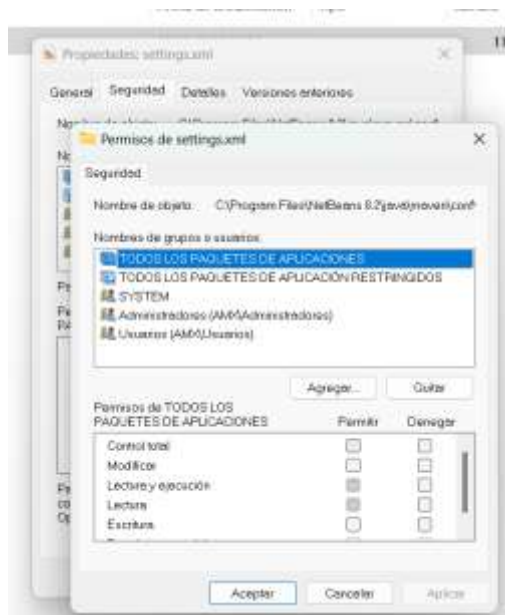


Ilustración 39. Permisos

NetBeans 8.2 (con su JDK y GlassFish)

NetBeans es un entorno de desarrollo integrado (IDE) que facilita el desarrollo de aplicaciones en Java y otros lenguajes. En el contexto de I'm Talenty, puede ser utilizado para:

- Desarrollo de Aplicaciones Backend y Frontend: Permite crear las funcionalidades principales del sistema, como el procesamiento de datos mediante JPQL y el manejo de la lógica de negocio.
- Integración con GlassFish: GlassFish es un servidor de aplicaciones compatible con Java EE que ejecuta aplicaciones empresariales,

permitiendo implementar servicios web, APIs RESTful y otras funciones esenciales para la plataforma.

- Depuración y Testing: Ofrece herramientas para depurar y probar el código del sistema, garantizando la calidad y el rendimiento.
- Manejo de bibliotecas y dependencias: Con soporte integrado para Maven o Ant, facilita la administración de las dependencias necesarias para la implementación de los algoritmos (como inferencia difusa o AHP).



Ilustración 40. logo de NetBeans IDE

Navicat y MySQL 8

Navicat es un administrador de bases de datos que permite interactuar fácilmente con MySQL 8, que sería la base de datos principal del proyecto. Estas herramientas son clave para:

- Diseño de la Base de Datos: Crear y gestionar las tablas necesarias para almacenar datos relacionados con candidatos, habilidades, resultados de pruebas y sugerencias de mejora.
- Consultas Eficientes: Navicat proporciona una interfaz intuitiva para escribir, probar y optimizar consultas SQL que respalden los algoritmos del proyecto.
- Gestión de Datos: Facilita la carga, exportación y respaldo de datos críticos para asegurar su integridad.
- Integración con JPQL: MySQL actúa como el repositorio que JPQL consulta para recuperar datos estructurados que permitan realizar análisis avanzados.



Ilustración 41. logo de navicat

TortoiseSVN

TortoiseSVN es un sistema de control de versiones basado en Subversion que permite gestionar el código fuente y la colaboración en el equipo. Su papel en el proyecto incluye:

- Control de Versiones: Permite registrar cambios en el código fuente del proyecto, asegurando que cualquier modificación sea rastreable y reversible.
- Colaboración en Equipo: Facilita que múltiples desarrolladores trabajen en el mismo proyecto al sincronizar cambios y resolver conflictos de forma eficiente.
- Documentación del Progreso: Con la posibilidad de agregar mensajes a cada commit, se lleva un historial detallado de las actualizaciones realizadas en cada fase del desarrollo.
- Seguridad del Código: Almacena una copia centralizada del proyecto, protegiendo el código ante fallos locales.



Desarrollo de las interfaces del sistema

La correcta aplicación de la marca I'm Talenty es un factor importante para el equipo de la compañía por lo cual se ha establecido en este manual las pautas que se tomarán como base para el uso del logotipo, paleta de colores, iconografía y demás diseños que llegarán a requerirse para el correcto uso y aplicación gráfica de su Identidad visual, en todas sus posibles expresiones para el desarrollo del producto I'm Talenty, propiedad de la empresa SmartSoft America BA.

Se pretende crear un producto homogéneo que transmita formalidad con una combinación fresca y atractiva. En la parte visual sea de fácil acceso al usuario, evitando variaciones que pudieran llegar a poner en riesgo la identidad del producto.

Garantizar que todos los elementos visuales y de comunicación de la marca se utilicen de manera consistente y coherente, esto incluye el logo, colores, tipografía, imágenes.

Facilitar la implementación en diferentes medios para aplicar la identidad de la marca, ya sea en medios impresos, digitales, publicitarios, etc. De esta forma, cualquier miembro del equipo o diseñador externo puede aplicar la identidad de manera correcta.

Composición del logotipo

El logotipo está compuesto por una figura (T) y un nombre comercial (I'm Talenty), se optó por utilizar la figura de la letra T ya que es la inicial de la palabra Talenty que se refiere al talento humano.

A este se le dio una forma abstracta combinando la letra T con los caracteres de signos de admiración (!) así como manejar un degradado con los tonos claros, ya que el verde sugiere frescura, juventud y crecimiento.

Y por último mostrando la palabra que los identifica como I'm talenty por sus siglas en inglés que significa soy un talento.

El diseño del logotipo se maneja en tres presentaciones ya que se utilizan de diversas maneras tanto en documentos físicos, documentos electrónicos y la aplicación del logotipo en el sistema.

A continuación, se muestra la composición de colores para documentos físicos.



Ilustración 42. Composición del logotipo

A continuación, se muestra la composición de colores para el sistema ya que se maneja la interfaz de fondo en tono verde y las letras en tono blanco #FFFFFF.



Ilustración 43.composición de colores para el sistema

A continuación, se muestra la composición de colores para el documento electrónico que genera el sistema.



Ilustración 44.composición de colores para el documento

Paleta de colores para tipografía

La tipografía sugerida para el texto del sistema es Arial.

A continuación, se muestra la paleta de colores con valores en hexadecimal, que serán utilizados por el área de desarrollo en la parte de texto se utilizan los tonos.



Ilustración 45. Paleta de colores para tipografía

Paleta de colores para iconos

Diseños iconográficos para utilizar para expresar con claridad el significado de lo que representa gráficamente en el contexto del proyecto a desarrollar, para este se sugiere los siguientes iconos, así como también los tonos ocupados en cada uno.



Ilustración 46. Paleta de colores para iconos

Paleta de colores para menús y secciones

Para los menús y secciones principales se manejan el siguiente tipo de colores.



Ilustración 47. Paleta de colores para menús y secciones

Generalidades

Descripción del input text de los formularios (Redondeado, color del borde, color de fondo)

Contenemos tres botones para I'm Talenty que es de editar, confirmar y eliminar con las propiedades por default; **button{ width: 160px; height: 36px; border-radius: 4px; border: none; }**

En medidas de teléfonos pequeños se acordó un @media query se le aplicarán en todos los botones un **(min-width:340px) and (max-width: 480px) {** y un **button{ width: 120px; }**

El botón de editar tiene un color azul



Ilustración 48. botón de editar

El botón de confirmar tiene un color verde



Ilustración 49. botón de confirmar

El botón de cancelar tiene un color rojo

(#EF3F61)

Cancelar

Ilustración 50.botón de cancelar

Contenido General

La clase que encierra todo el contenido se llama ContainerMain va tener un with del **100%** y un padding del **0%** arriba y abajo y **4%** derecha e izquierda, aplicando una subclase para las secciones con tarjetas o secciones con fondo oscuro se aplicará una subclase llama **MainGrey** que solo agrega **background-color:**

#f4f4f4.

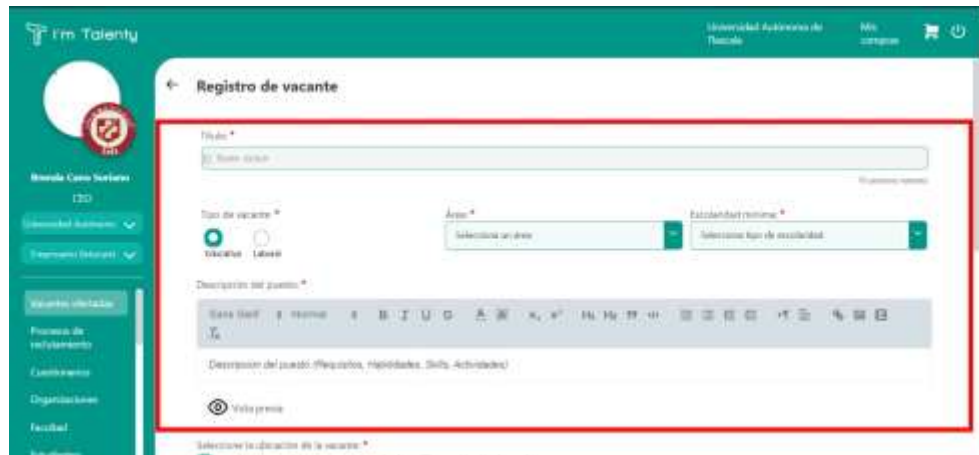


Ilustración 51.ContainerMain

Fondo oscuro gris:

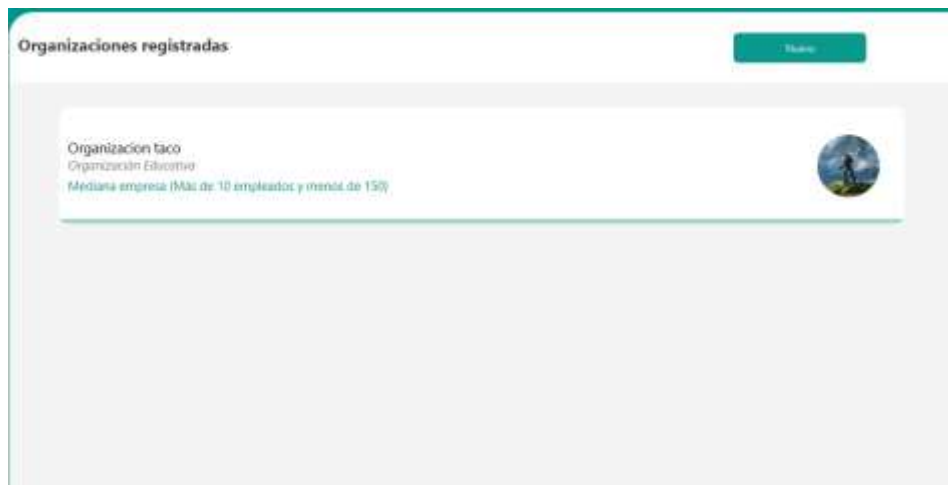


Ilustración 52.Fondo oscuro gris

Para la siguiente clase que controla todos los encabezados de los módulos del sistema se llama **ContainerHeaderOuter**, esta misma tiene como contenido un **width: 100%; padding: 20px 0; display: flex; flex-direction: column; align-items: center; gap: 18px; justify-content: center; position: fixed; top: 0px; left: 0; z-index: 80; background-color: white;** y encierra a sus clases hijas una llamada **encabezado_seccion_modulo** que contiene **display: flex align-items: center; justify-content: space-between; width: 91%;**

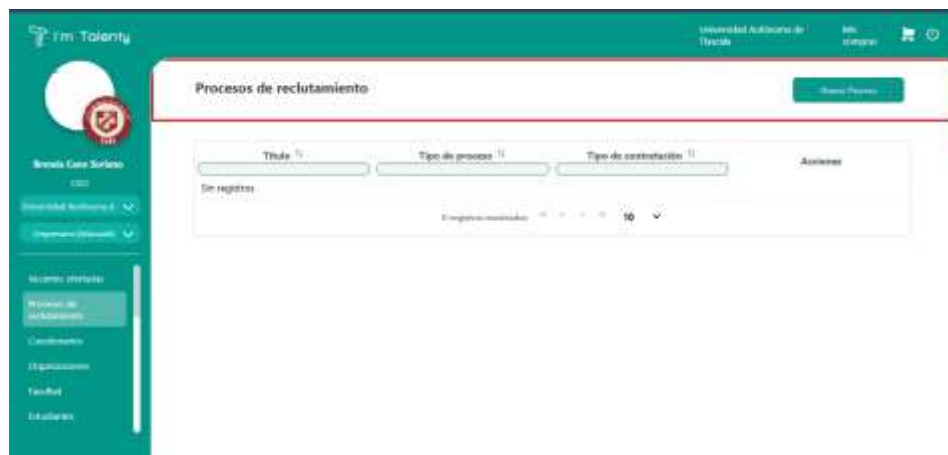


Ilustración 53.ContainerHeaderOuter

Y otra llamada **ContainerBack** que contiene estas propiedades. **ContainerBack{ width:5%; display: flex; justify-content: center; align-items: center; position: fixed; left: 4px; top: 26px;}**

Esta misma se utiliza para el botón de regresar de los módulos.

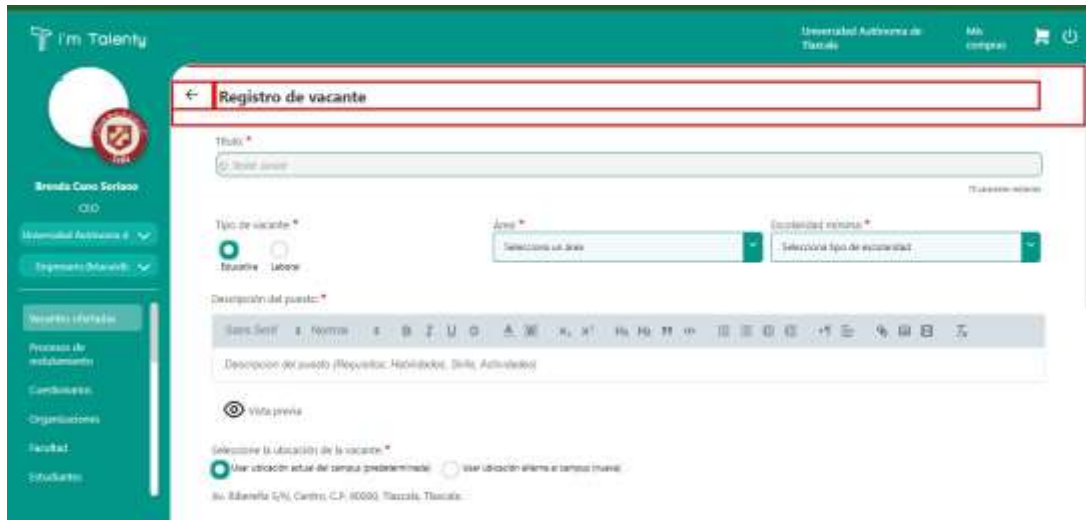


Ilustración 54.ContainerBack

Por lo que es importante tener ambas clases **encabezado_seccion_modulo** y **ContainerBack** dentro de **ContainerHeaderOuter** y **ContainerBack** se renderice o no dentro del mismo **ContainerHeaderOuter**, es decir, el boton de regresar debe estar dentro del mismo **ContainerHeaderOuter** y no en otro div aparte.

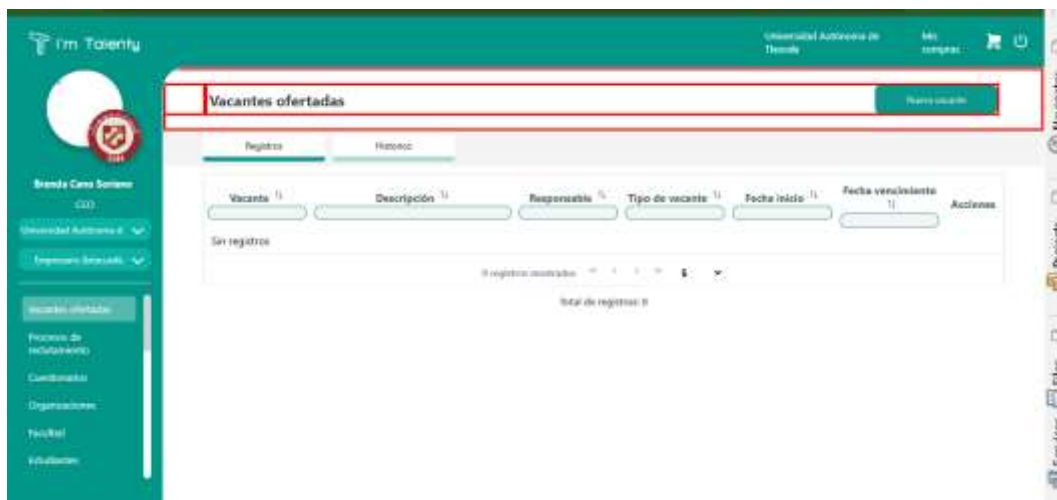


Ilustración 55.ContainerHeaderOuter

Entonces **ContainerHeaderOuter** tendrá su título del módulo y su botón de agregar o regresar en caso de ser necesario. Por la que la regla es, por cada encabezado o título de módulo debe de estar adentro **encabezado_seccion_modulo**, un **ContainerBack**, y a su vez **encabezado_seccion_modulo** va dentro de **ContainerHeaderOuter**.



Ilustración 56. ContainerHeaderOuter y ContainerBack

Como se puede observar en las siguientes capturas de código y vista previa de ejemplo.

Código:

```

</p:dialog>-->
<div class="ContainerHeaderOuter" >
  <div class="ContainerBack">

  </div>

  <div class="encabezado_seccion_modulo">
    <h1>Nombre modulo</h1>
    <p:commandButton value="Nuevo"/>
  </div>

</div>

```

Ilustración 57.vista codigo xhtml

Vista previa módulo inicio:

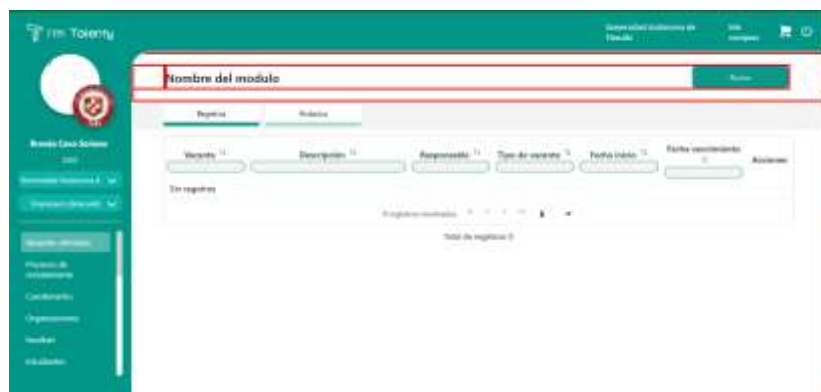


Ilustración 58.Vista previa módulo inicio

Vista previa con flecha de regreso

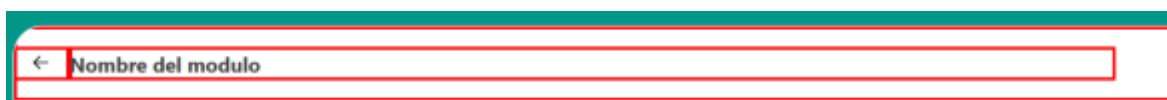


Ilustración 59.Vista previa con flecha de regreso

Para sub-encabezados de tap views artificiales (echo por el equipo de desarrollo) se implementará dentro de **ContainerHeaderOuter** y se colocará debajo de **encabezado_seccion_modulo**, en un div con la clase **contendorOpcionesTablas** con las propiedades: **width: 100%; height: 48px; background-color: #F4F4F4; display: flex; flex-direction: column; justify-content: center; align-items: center;**



Ilustración 60.sub-encabezados

Esta encierra a su clase hijo **optTablasBg** con las siguientes propiedades. **optTablasBg{ width: 91%; margin: 0px; padding: 0px; display: flex; align-items: center; gap: 10px; }**

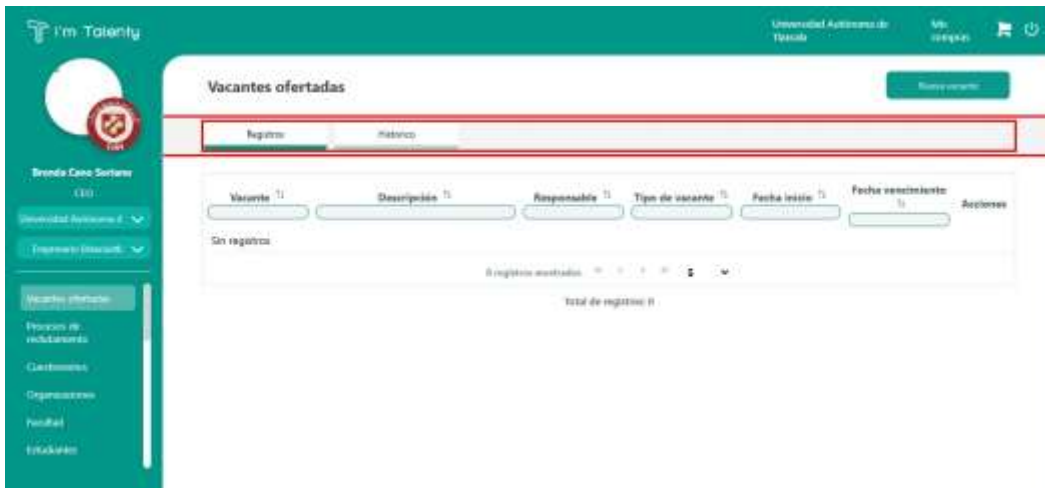


Ilustración 61.optTablasBg

Posteriormente, debajo del **ContainerHeaderOuter** al tener ese caso de uso se debe implementar una **EmptyBox150** con la siguientes clases **EmptyBox150{ display: block; width: 100%; height:150px; overflow:hidden;}** o bien colocar una **EmptyBox80** ajustada a el encabezado.

```
<div class="ContainerHeaderOuter" >
  <div class="ContainerBack">

  </div>

  <div class="encabezado_seccion_modulo">
    <h1>Nombre modulo</h1>
    <p:commandButton value="Nuevo"/>
  </div>

  <div class="contendorOpcionesTablas">
    <div class="optTablasBg">
      <p:commandButton value="Registros" class="tapBotonOn"/>
      <p:commandButton value="Historico" class="tapBotonOff"/>
    </div>
  </div>
</div>

<p:fragment rendered="#{empty nuevaVacanteController.view.entity and nuevaVacanteController.v
  <div class="EmptyBox150"/>
</p:fragment>
```

Ilustración 62. Llamada de estilos

Nota: En caso de que **contendorOpcionesTablas** no funciona con un div al colocarlo dentro de un **<p:Fragment>**, no obstante, este funciona dentro de un **<p:outputPanel/>** con la clase ya antes mencionada sin romper los estilos en algunos casos.

```

<p:outputPanel rendered="#{empty nuevaVacanteController.view.entity and nuevaVacanteController.view.listaEntity != null and nuevaVacanteController.view.listaEntity.size() > 0}" class="contenidoCpinnerTablas">
  <div class="opcTablasBq">
    <p:commandButton value="Registros"
      id="btnRegistros"
      actionListener="#{nuevaVacanteController.mostrarSeccionTap('registros'))}"
      update="contenidoVista btnRegistros btnHistorico"
      styleClass="#{nuevaVacanteController.view.seccionRegistros ? 'tapBotonOn' : 'tapBotonOff'}"
    />
    <p:commandButton value="Historico"
      id="btnHistorico"
      actionListener="#{nuevaVacanteController.mostrarSeccionTap('historico'))}"
      update="contenidoVista btnRegistros btnHistorico"
      styleClass="#{nuevaVacanteController.view.seccionHistorico ? 'tapBotonOn' : 'tapBotonOff'}"
    />
  </div>
</p>
</p:outputPanel>

```

Ilustración 63.ejemplo xhtml

Notas:

Revisar si es necesario un contenedor extra dentro de un **ContainerMain**

Contenido Formularios

Para que el sistema funcione y mostremos ya sea un formulario o sección de vista tenemos que encerrar todo el contenido por medio de **Containers** en específico con un **Container100** ya que con esto se ocupara todo el **100%** de la vista es importante mencionar que esto solo aplica para **fragments** o **panelgroups** donde muestres más de un evento.

Para formularios se debe implementar las siguientes clases, si queremos mostrar dos columnas de campos debemos meterlo dentro de **contenido_dos_columnas{ width: 100%; display: grid; grid-template-columns: repeat(2, 1fr); gap: 6px; align-items: center; }** esto a su vez llevó una clase hijo por cada campo y etiqueta un div con una clase para colocar etiqueta y campo o contenido todo a la izquierda llamada **contenidoColumnaIzquierda{ display: flex; flex-direction: column; justify-content: center; align-items: start; gap: 6px; padding: 6px 0; }**.

Esto aplica para todos los divs con clases de tipo **contenido_[número de columnas]** por ejemplo:

```

contenido_tres_columnas{ width: 100%; display: grid; grid-template-columns: repeat(3, 1fr); gap: 6px; align-items: center; }

```

Ilustración 64. Contenido Formularios

Vista previa del código:

```

<div class="contenido_tres_columnas" style="outline: solid 3px red;" <!-- contenedor para columnas de formulario -->
  <div class="contenidoColumnaIzquierda" <!-- contenedor de campo -->
    <p:outputLabel value="Campo 1"/>
    <p:inputText/>
  </div>

  <div class="contenidoColumnaIzquierda" <!-- contenedor de campo -->
    <p:outputLabel value="Campo 2"/>
    <p:inputText/>
  </div>

  <div class="contenidoColumnaIzquierda" <!-- contenedor de campo -->
    <p:outputLabel value="Campo 3"/>
    <p:inputText/>
  </div>

  <div class="contenidoColumnaIzquierda" <!-- contenedor de campo -->
    <p:outputLabel value="Campo 4"/>
    <p:inputText/>
  </div>
</div>

<div class="contenido_cuatro_columnas" style="outline: solid 3px red;" <!-- contenedor para columnas de formulario -->
  <div class="contenidoColumnaIzquierda" <!-- contenedor de campo -->
    <p:outputLabel value="Campo 1"/>
    <p:inputText/>
  </div>

  <div class="contenidoColumnaIzquierda" <!-- contenedor de campo -->
    <p:outputLabel value="Campo 2"/>
    <p:inputText/>
  </div>

```

Ilustración 65. Vista previa del código

Las opciones disponibles para la disposición de los campos en formularios son las siguientes:

1. **contenido_dos_columnas**{ width: 100%; display: grid; grid-template-columns: repeat(2, 1fr); gap: 6px; align-items: center; }
2. **contenido_tres_columnas**{ width: 100%; display: grid; grid-template-columns: repeat(3, 1fr); gap: 6px; align-items: center; }
3. **contenido_tres_columnasConFilas**{ width: 100%; display: grid; grid-template-columns: repeat(3, 1fr); gap: 6px; align-items: start; } (clase para colocar 3 columnas de campos en formularios con filas de lista de objetos den vertical)

4. **contenido_cuatro_columnas**{ width: 100%; display: grid; grid-template-columns: repeat(4, 1fr); gap: 6px; align-items: center; }
5. **encabezado_seccion**{ /background: red !important; display: flex; align-items: center; justify-content: space-between; width: 100%; } (clase para colocar boton y título en los laterales como por ejemplo en secciones de galería o donde se requiera título y botón en los laterales dentro del formulario).

Nota: no confundir esta clase **encabezado_seccion** con **encabezado_seccion_modulo**, la primera es para formularios o títulos de secciones dentro de los formularios y la otra es para los encabezados de los módulos como se explicó en la sección y regla anterior.

Ilustración 66 muestra la estructura visual de las secciones y campos. Sección 1 tiene un encabezado rojo con el título 'Sección 1' y un botón 'botonEjemplo'. Debajo hay tres campos de texto etiquetados como 'Campo 1', 'Campo 2' y 'Campo 3'. Sección 2 tiene un encabezado rojo con el título 'Sección 2'. Debajo hay un solo campo de texto etiquetado como 'Campo 3'.

Ilustración 66. encabezado_seccion_modulo

Vista previa del código:

```
<div class="encabezado_seccion">
  <h3>Sección 1</h3>
  <p>commandButton value="botonEjemplo"/>
</div>
<div class="contenido_tres_columnas" style="outline: solid 3px red;"> <!-- contenedor para columnas de formulario -->
  <div class="contenidoColumnaIzquierda"> <!-- contenedor de campos -->
    <p>outputLabel value="Campo 1"/>
    <p>inputText/>
  </div>
  <div class="contenidoColumnaIzquierda"> <!-- contenedor de campos -->
    <p>outputLabel value="Campo 2"/>
    <p>inputText/>
  </div>
  <div class="contenidoColumnaIzquierda"> <!-- contenedor de campos -->
    <p>outputLabel value="Campo 3"/>
    <p>inputText/>
  </div>
</div>
<div class="encabezado_seccion">
  <h3>Sección 2</h3>
</div>
<div class="contenidoColumnaIzquierda"> <!-- contenedor de campos -->
  <p>outputLabel value="Campo 3"/>
  <p>inputText/>
</div>
```

Ilustración 67. Vista previa del código

Ahora para botones de guardado, eliminado o opciones de formulario se debe de implementar la siguiente clase **contenidoColumnaDerecha** { **width: 100%; display: flex; justify-content: end;** }.



Ilustración 68. dispersión de botones

Vista previa del código:

```
<div class="contenidoColumnaIzquierda" style="outline: solid green 3px;"> <!--contenedor de campos-->
  <p:outputLabel value="Campo 4"/>
  <p:inputText/>
</div>

<div class="contenidoColumnaDerecha">
  <p:commandButton value="Botón Prueba"/>
</div>
```

Ilustración 69.Vista previa del código

Por lo que dependiendo la disposición del formulario se debe usar un contenedor (div) con algunas de las clases **contenido_[número de columnas]** según sea el caso y este div encierra a sus clases hijas **contenidoColumnaIzquierda** que finaliza con sus opciones de botones con una sola clase o div que tenga contenido columna derecha.

Entonces la regla es primero en cada xhtml debe ir primero un **ContainerMain** éste encierra otro **ContainerHeaderOuter**.

```
<div class="ContainerMain">
  <div class="ContainerHeaderOuter">
  </div>
  <!-- CONTENIDO -->
</div>
```

Ilustración 70.codigo de ContainerMain

En este caso **ContainerHeaderOuter** encierra un div con clase **seccion_modulo** y este mismo dentro tiene un título y botón para los encabezados.

```

<div class="ContainerMain">
  <div class="ContainerHeaderOuter">
    <div class="encabezado_seccion_modulo">
      <h1>Titulo</h1>
      <p:commandButton value="Nuevo"/>
    </div>
  </div>
  <!-- CONTENIDO -->
</div>

```

Ilustración 71.ejemplo Código ContainerHeaderOuter

No olvidar que dentro de ContainerHeaderOuter debe ir su botón de regreso su **ContainerBack** que se renderiza dependiendo de si está dentro de un módulo.

```

<div class="ContainerMain">
  <div class="ContainerHeaderOuter">
    <div class="ContainerBack">
      <p:commandLink
        title="Regresar">
        <i class="pi pi-arrow-left"/>
      </p:commandLink>
    </div>
    <div class="encabezado_seccion_modulo">
      <h1>Titulo</h1>
      <p:commandButton value="Nuevo"/>
    </div>
  </div>
  <!-- CONTENIDO -->
</div>

```

Ilustración 72.ContainerHeaderOuter

Ahora volviendo al ContainerHeaderOuter este igual puede contener dentro un contendor/sub-header que se llama **contendorOpcionesTablas** que a su vez esto encierra **optTablasBg** en las cual este último tendrá las opciones o botones para las diferentes tablas o contenido a mostrar.

```

<div class="ContainerMain">
  <div class="ContainerHeaderOuter">
    <div class="ContainerBack">
      <p:commandLink
        title="Regresar">
        <i class="pi pi-arrow-left"/>
      </p:commandLink>
    </div>
    <div class="encabezado_seccion_modulo">
      <h1>Titulo</h1>
      <p:commandButton value="Nuevo"/>
    </div>
    <div class="contendorOpcionesTablas">
      <div class="optTablasBg">
        <p:commandButton value="Registro"/>
        <p:commandButton value="Historico"/>
      </div>
    </div>
  </div>
  <!-- CONTENIDO -->
</div>

```

Ilustración 73.ejemplo de regla

Para que al final se coloque un **Empybox** para darle el espaciado entre el encabezado y contenido

```

<div class="ContainerMain">
  <div class="ContainerHeaderOuter">
    <div class="ContainerBack">
      <p:commandLink
        title="Regresar">
        <i class="pi pi-arrow-left"/>
      </p:commandLink>
    </div>
    <div class="encabezado_seccion_modulo">
      <h1>Titulo</h1>
      <p:commandButton value="Nuevo"/>
    </div>
    <div class="contendorOpcionesTablas">
      <div class="optTablasBg">
        <p:commandButton value="Registro"/>
        <p:commandButton value="Historico"/>
      </div>
    </div>
  </div>
  <div class="EmptyBox100"/>
  <!-- CONTENIDO -->
</div>

```

Ilustración 74.ejemplo de Emptybox

Y esta será de manera general la estructura que tendrá los diferentes xhtml que tendrá el sistema.

Como, por ejemplo:



Ilustración 75.estructura general

Diseño de pantallas

La identidad visual debe ser reflejada de manera coherente en todas las pantallas del sistema. El logo debe aparecer en la parte superior de la pantalla de inicio, y los botones deben seguir la paleta de colores. Los iconos utilizados en la interfaz deben ser simples, minimalistas y seguir el estilo gráfico del logo.

1. Web

En la pantalla de inicio de sesión se muestra una barra superior de color verde (#0A9A8D), donde, en la parte izquierda, se encuentra el logotipo **"I'm Talenty"** en color blanco (#FFFFFF).

En la interfaz, aparece el ícono de **"I'm Talenty"** en color (#1B9F93) y el de **CEYDT** en los siguientes colores: (#464C6F), (#8A99C8) y (#0A0A0A).

En la parte inferior de la pantalla, se presentan dos campos de entrada: uno para **correo electrónico o usuario** y otro para **contraseña**. Debajo de estos, hay un **botón de "Iniciar sesión"** en color verde (#0A9A8D). Finalmente, se incluyen dos opciones adicionales: **"Registrarse"** y **"¿Olvidaste tu contraseña?"**.



Ilustración 76.pantalla principal

En la **pantalla principal**, se muestra una **barra de navegación superior** de color verde (#0A9A8D). A la izquierda, se encuentra el logotipo "**I'm Talenty**" en color blanco (#FFFFFF), mientras que, a la derecha, aparecen las siguientes opciones:

- "Mis organizaciones"
- "Mis compras"
- Ícono de carrito de compra
- Botón de notificaciones
- Botón de cerrar sesión

En el lado izquierdo, hay una **barra lateral** del mismo color verde (#0A9A8D), donde se muestran los **datos del usuario**, incluyendo su **nombre y foto de perfil**. Debajo de esta sección, se encuentran diferentes **módulos de navegación**. Al seleccionar uno, este cambia a color (#0BB3A8). Entre los módulos disponibles están:

- "Mis organizaciones"
- "Vacantes recomendadas"
- "Mis postulaciones"
- "Vinculación Organizacional"

La **sección principal** de la pantalla tiene un **fondo blanco** (#FFFFFF) y muestra un **mensaje de bienvenida** que dice "**Bienvenido a I'm Talenty**", con una **sombra en color negro** (#000000).

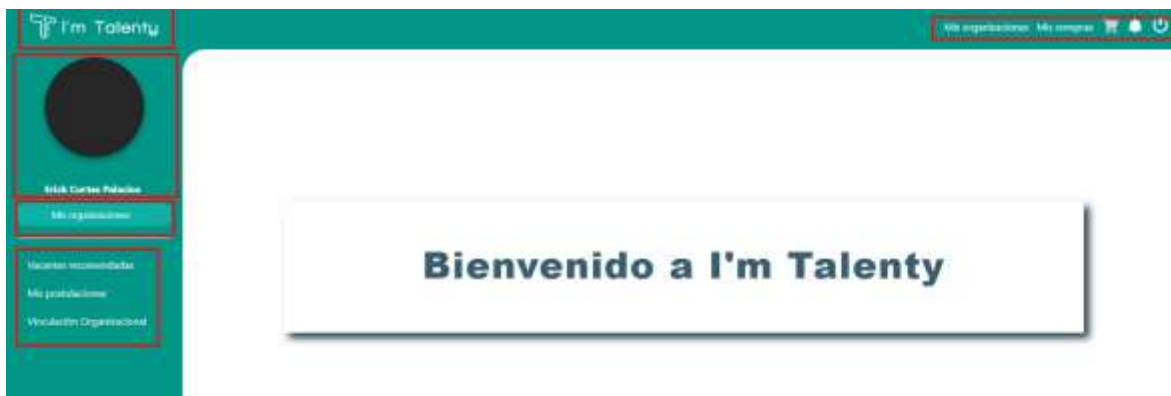


Ilustración 77.página de inicio

En el **módulo de Vacantes Recomendadas**, en la parte superior, se encuentra una **barra de búsqueda** con los siguientes elementos:

- Campo de "Puesto, área laboral o empresa"
- Campo de "Ciudad o estado"

Junto a estos campos, hay un **botón verde (#0A9A8D) de "Buscar"**.

Debajo de la barra de búsqueda, se presentan **filtros desplegados** que permiten precisar la búsqueda según:

- Fecha
- Tipo de vacante
- Tipo de contratación
- Escolaridad
- Modalidad
- Horario

Al seleccionar un filtro, este se resalta en un **cuadro gris (#939393) con su ícono en color blanco (#FFFFFF)**.

También se encuentran dos **pestañas** etiquetadas como **"Para ti"** y **"General"**, que al seleccionarse cambian a color **gris (#939393)**.

La interfaz del **contenido principal** tiene un **diseño con encabezado verde (#0A9A8D) y fondo blanco (#FFFFFF)**. Esta sección se divide en dos partes:

1. **A la izquierda:** Se muestra un **listado de ofertas de trabajo**. Cuando se selecciona una, se resalta con una **línea turquesa (#0A9A8D) debajo del contorno**.
2. **A la derecha:** Se muestra el **detalle de la vacante seleccionada**.

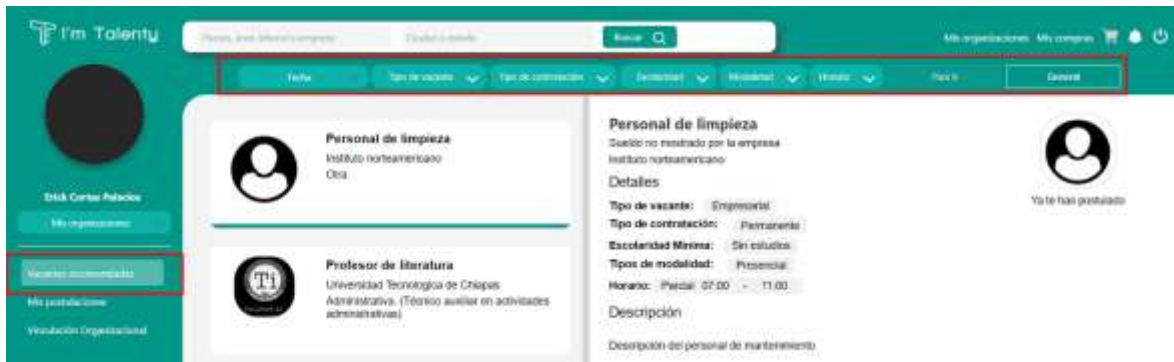


Ilustración 78 distribución de barras de tareas

En el **módulo de Mis Postulaciones**, se muestran todas las **postulaciones realizadas**. Al seleccionar una, esta se resalta con un **sombreado de color verde (#0A9A8D)**.

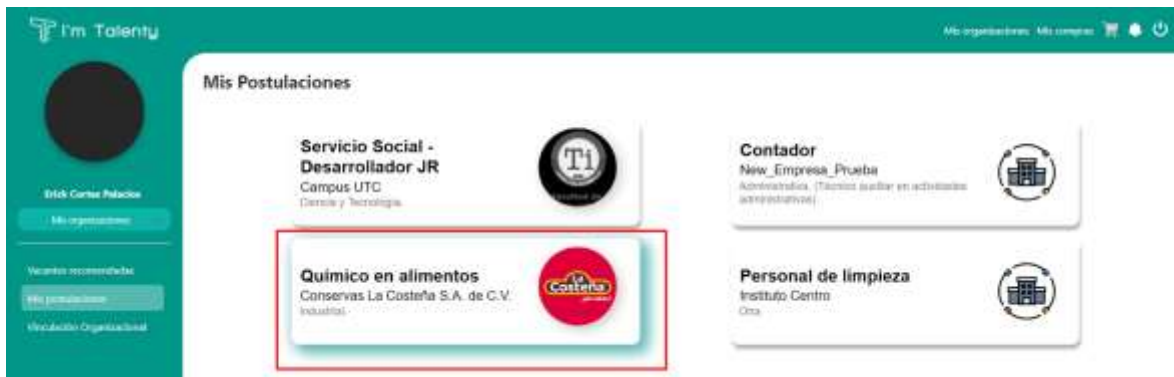


Ilustración 79. módulo de Mis Postulaciones

En el **módulo de Creación de Organización**, al acceder, se presentan diversas opciones que permiten **visualizar las organizaciones ya creadas** o **seleccionar la opción para crear una nueva organización**.



Ilustración 80. módulo de Creación de Organización

Al seleccionar el módulo "**Procesos de Reclutamiento**", este se **resalta con un recuadro verde (#56B9B0)**.

En la parte superior derecha, hay un **botón verde (#0A9A8D)** con el **texto "Nuevo Proceso"**. Cuando se presiona, cambia a **color gris (#939393)**, indicando que la función para crear un nuevo registro de reclutamiento se ha activado.

La interfaz presenta una **tabla con búsqueda rápida**, que contiene tres columnas principales:

- Título
- Tipo de proceso
- Acciones

Al seleccionar una fila, el cuadro cambia a **color turquesa**.

En la parte inferior de la tabla, se muestran los **registros realizados**, junto con un **control de paginación**. El número seleccionado aparece dentro de un **cuadro azul (#E3F2FD)**, resaltando la página activa.



Ilustración 81.resalta con un recuadro verde (#56B9B0).

En el **módulo de Estudiantes**, la interfaz permite **crear un nuevo estudiante**.

Debajo, se encuentran una serie de **filtros desplegados**, que incluyen:

- Nombre
- Correo
- Perfil
- Sucursales/Sedes
- Estatus

Estos filtros muestran la información registrada. En el lado derecho, se encuentran dos **íconos de acción: editar y eliminar**, ambos en **color verde (#0A9A8D)**.

En la parte inferior, hay un **botón de "Carga Masiva"** en **color verde (#0A9A8D)**.

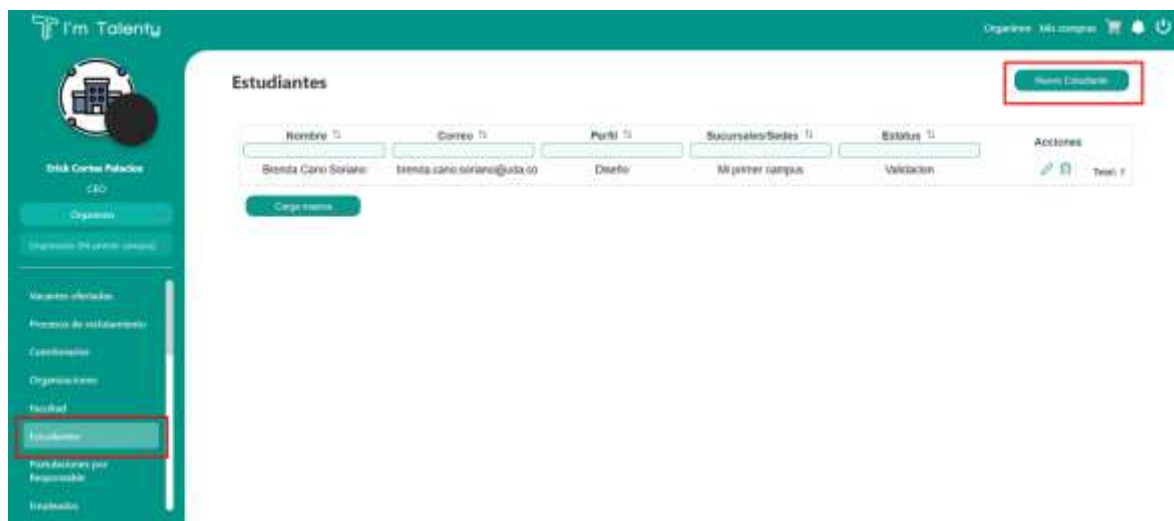


Ilustración 82.botón de Carga Masiva

En el **módulo de Perfiles**, la interfaz muestra un **botón verde (#0A9A8D)** para crear un **"Nuevo Perfil"**.

Debajo, se presentan los **registros realizados**, y en el lado derecho, se encuentran los **íconos de acción** para **editar y eliminar**, ambos en **color (#28A096)**.



Ilustración 83.botón verde (#0A9A8D) para crear un "Nuevo Perfil".

En el **módulo de Organizaciones**, se muestra un **botón** para **seleccionar tu membresía**.

La interfaz presenta las **membresías anuales** junto con sus beneficios, las cuales se presentan en los siguientes paquetes:

- Gratis
- Bronce
- Plata
- Oro
- Diamante

Los nombres de las membresías aparecen en **color azul (#087EFF)**.

Debajo, se encuentra un **botón de "Actualizar Membresía"** en **color verde (#0A9A8D)**.



Ilustración 84.membresías anuales

La interfaz muestra un **resumen del carrito**, incluyendo el **proceso de compra**, el **número de filtro** en **color azul (#E3F2FD)**, así como los **datos de la compra y el costo**.

Al final, se presenta una casilla para **confirmación** del producto dentro de un **recuadro azul (#2196F3)**.

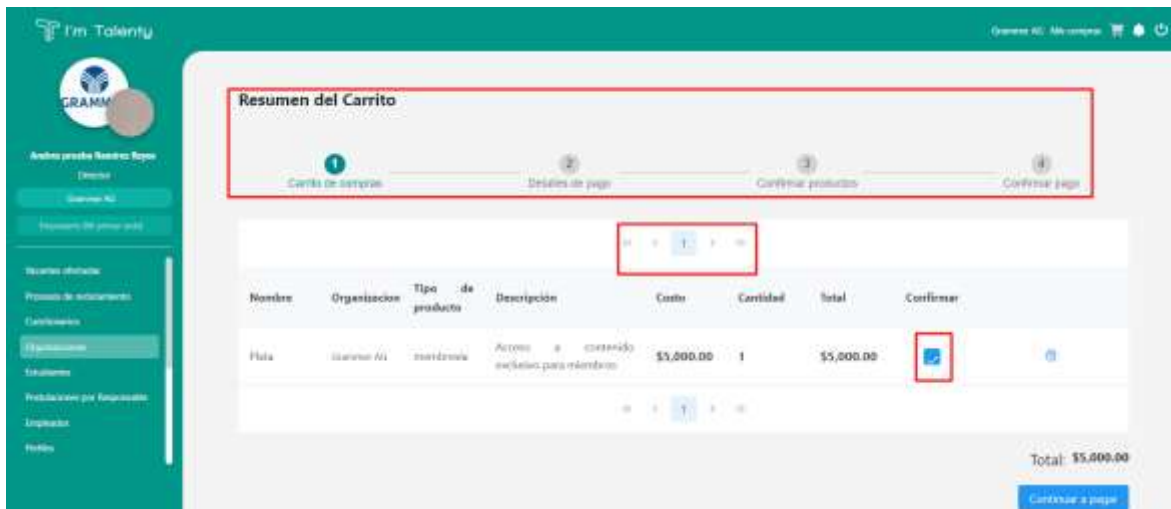


Ilustración 85.resumen del carrito

Al finalizar el proceso, la interfaz mostrará un **mensaje de "Pago en espera"** en **color amarillo (#FFC311)**.

En la parte inferior, se presenta una **descripción del pago**, incluyendo el **monto a pagar**.

A continuación, se muestra la **información de la membresía seleccionada**, resaltando el **precio unitario y el total** en **color azul (#1D8AFF)**.

En la última sección, se ofrecen **tres acciones disponibles**:

- **"Completar Pago"** en **azul fuerte (#004D99)**
- **"Consultar Pago"** en **azul claro (#2E6C9F)**
- **"Cancelar Pago"** en **rojo (#D32F2F)**

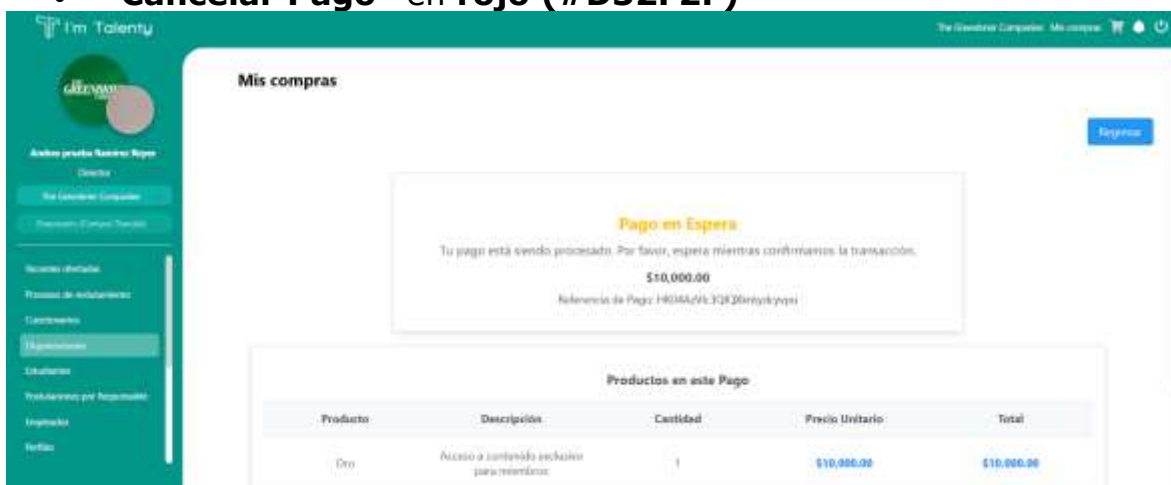


Ilustración 86.Pago en espera"



Ilustración 87. Información de pagos y productos

Implementación

Pantalla de registro

The screenshot shows a registration form for 'Tm Talentu'. The form includes the following fields:

- Nombre***: Text input with 'Diana' entered.
- Apellido Paterno***: Text input with 'Rojas' entered.
- Apellido materno**: Text input with 'Rojas' entered.
- Módulo de estudio***: Dropdown menu with 'Desarrollo de...' selected.
- Número telefónico***: Text input with '2471657448' entered.
- Correo electrónico***: Text input with 'gen00res@talentu.com' entered.
- Confirma tu correo electrónico***: Text input with 'gen00res@talentu.com' entered.
- Contraseña***: Text input with '123456' entered.
- Confirma contraseña***: Text input with '123456' entered.

Ilustración 88. Pantalla de registro

En esta pantalla podremos registrar nuestros datos en el cual los asteriscos en rojo marcan como campos obligatorios los cuales no pueden estar vacío, además con las validación que están implementadas los datos no pueden ser repetidos como es el caso de correo electrónico, número telefónico y nombre de usuario, contando igual con la confirmación de

contraseña y el cual debería de tener mínimo 8 caracteres y los cuales deben de cumplir con una mayúscula, un número y un carácter específico, esto para tener una seguridad alta seguridad, así como evitar la duplicidad de personas.

Cuerpo de XHTML para el login.

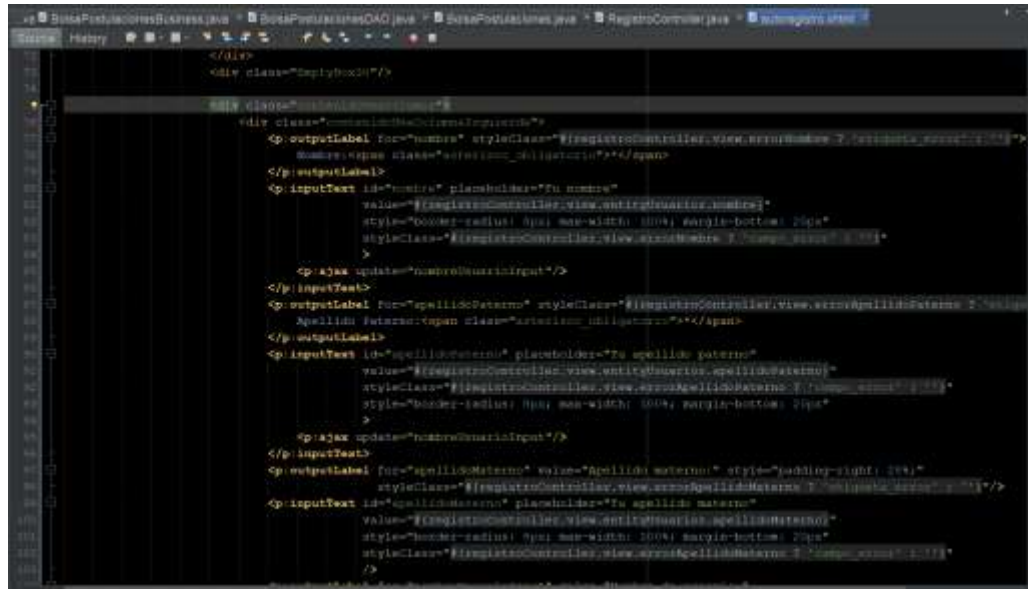
A screenshot of an IDE window showing XHTML code for a login form. The code includes labels and input fields for 'nombre', 'apellidoPaterno', and 'apellidoMaterno'. It also features a 'javax:ajax' update attribute for the 'nombre' input field. The code is styled with various classes and attributes, including 'border-radius', 'margin-bottom', and 'width'.

Ilustración 89.Cuerpo de XHTML para el login

Código para las validaciones de los campos.

A screenshot of an IDE window showing Java code for field validation. The code defines two private methods: 'validarExistenciaCorreo' and 'validarFormatoCorreo'. 'validarExistenciaCorreo' checks if a business object exists for a given email. 'validarFormatoCorreo' uses a regular expression to validate the email format. The code is well-commented and includes a regex pattern for email validation.

Ilustración 90.Código para las validaciones de los campos.

Funcionamiento de las validaciones.

The image shows a registration form with the following fields and validation messages:

- Nombre:** Párra
- Apellido materno:** Aguilera
- Nombre de usuario:** GiovanniParrini (with a green checkmark icon)
- Número telefónico:** 2471067648
- Correo electrónico:** geo03miz@gmail.com (with a red error message: "El correo electrónico ya está en uso por otro usuario.")
- Confirme su correo electrónico:** geo03miz@gmail.com
- Contraseña:** (with a red error message: "La contraseña debe tener al menos 8 caracteres")
- Confirme contraseña:** (with a red error message: "La contraseña debe tener al menos 8 caracteres")

At the bottom, there is a green "Regístrate" button and a link "Regresar al inicio".

Ilustración 91. Funcionamiento de las validaciones.

Recuperación de contraseña mediante correo

En este módulo tenemos la función de poder recuperar la contraseña, pero con la seguridad de ser mediante el correo electrónico, al igual que el link mandado mediante el correo tenga una vida útil de unos 30 minutos y sobre todo que solo pueda usarse una sola vez, con el fin de evitar que alguien más quiera usar el link proporcionado.

Funcionamiento del módulo de recuperar contraseña.



Ilustración 94. vinculo vencido de recuperación de contraseña

Login

Aquí tendremos nuestro formulario para poder ingresar con las credenciales correctas las cuales cumplen con los mismos requerimientos de seguridad, en caso de ingresar varias veces con la contraseña errónea el usuario será bloqueado para evitar que el alguien externo intente ingresar.

Funcionamiento de modulo.



Ilustración 95. funcionamiento del login cuando las credenciales no son validas

Página principal

En esta página contaremos con un mensaje de bienvenida para el usuario, al igual de una barra de lateral y superior, donde tendremos varias

opciones, como desde ingresar crear organizaciones, crear organizaciones, ver las vacantes ofertadas, ver nuestras postulaciones, ver nuestras compras, cerrar sesión.



Ilustración 96. página principal

En caso de que queramos ver las vacantes y no contemos con un perfil completo tendremos un mensaje el cual nos dirá para una mejor experiencia terminemos de configurar nuestro perfil con nuestros datos, como experiencia, estudios...



Ilustración 97. alerta cuando no tenemos información en nuestro perfil

En esta parte podremos visualizar y editar nuestros datos, así con validaciones para evitar campos vacíos y repetidos con el fin de poder tener una mejor experiencia más probabilidad de llamar la atención de alguna institución u organización.



Ilustración 98. perfil de usuario con datos

Ilustración 99. ejemplo de validaciones obligatorias

Vacantes recomendadas

Aquí podremos revisar las postulaciones para los que nuestro perfil esta más adecuado, en cada paso tendremos perfiles de la organización /directivos y de usuarios normales/alumnos, para poder tener una mejor gestión para los procesos de vinculación, así como alertar y validaciones para estar seguros de los movimientos que hagamos para evitar algún movimiento erróneo, alerta de cupo limitado o si no indica alguna información la vacante.

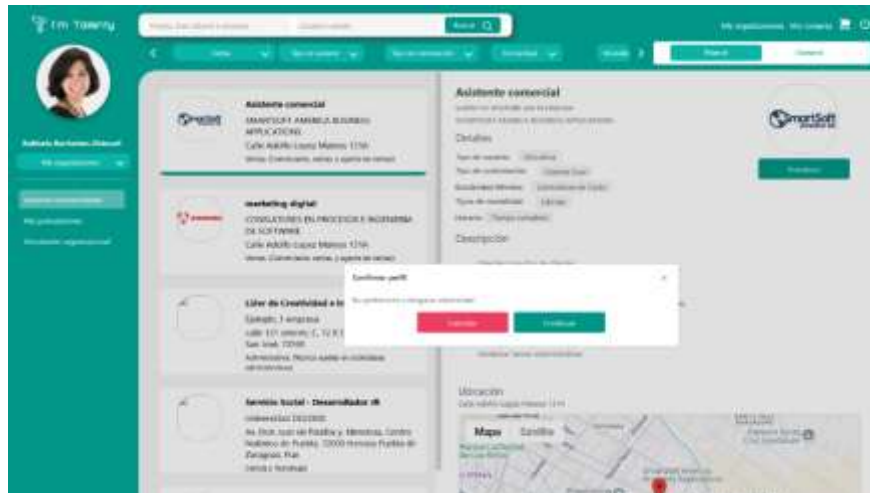


Ilustración 100. confirmación de postulación

Mis postulaciones.

En este módulo podremos visualizar el estatus de nuestras postulaciones para poder estar al tanto de nuestro estatus, así como la misma información de la vacante en la cual nos hemos postulado.

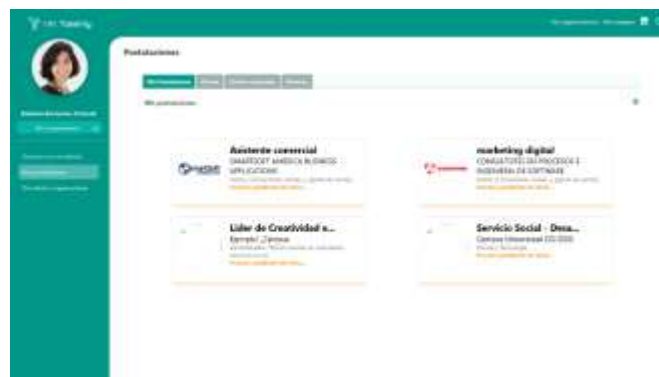


Ilustración 101.ejemplo de estatus de solicitud



Ilustración 102.detalles de vacante

Vacantes ofertadas / publicar vacantes

En este módulo vamos a poder visualizar las nuestras vacantes publicadas, así como poder crear una vacante la cual, así como los demás módulos cuentan con sus propias validaciones para evitar repetir información y falta de esta misma.



Ilustración 103.vacante/ofertas creadas

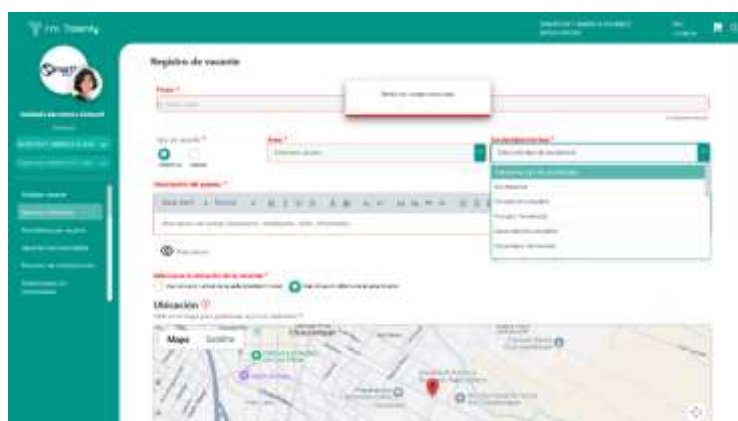


Ilustración 104.ejemplo de campos obligatorios para las postulaciones a crear

Procesos de reclutamiento

Podemos visualizar los procesos que ya hemos generado los cuales tendrán que seguir nuestros postulados dependiendo del tipo de proceso los cuales son Educativo y laboral, así como los tipos de contratación y poder crear nuevos procesos, dependiendo nuestros requerimientos y etapas los cuales consisten en la vinculación de nuestros cuestionarios generados y adecuados a nuestros propios requerimientos.

Tipo de vinculación	Tipo de vinculación 2	Acciones
Procesos de vinculación de tipo 1	Procesos de vinculación de tipo 1	Ver, Editar, Eliminar
Procesos de vinculación de tipo 2	Procesos de vinculación de tipo 2	Ver, Editar, Eliminar
Procesos de vinculación de tipo 3	Procesos de vinculación de tipo 3	Ver, Editar, Eliminar
Procesos de vinculación de tipo 4	Procesos de vinculación de tipo 4	Ver, Editar, Eliminar
Procesos de vinculación de tipo 5	Procesos de vinculación de tipo 5	Ver, Editar, Eliminar
Procesos de vinculación de tipo 6	Procesos de vinculación de tipo 6	Ver, Editar, Eliminar
Procesos de vinculación de tipo 7	Procesos de vinculación de tipo 7	Ver, Editar, Eliminar
Procesos de vinculación de tipo 8	Procesos de vinculación de tipo 8	Ver, Editar, Eliminar
Procesos de vinculación de tipo 9	Procesos de vinculación de tipo 9	Ver, Editar, Eliminar
Procesos de vinculación de tipo 10	Procesos de vinculación de tipo 10	Ver, Editar, Eliminar

Ilustración 105. procesos de vinculación creadas

Ilustración 106. ejemplo de creación para vinculaciones

Modalidades de contratación.

Aquí podremos generar y ver los tipos de contratos que deseamos general, los cuales vincularemos con el módulo anterior.

Modalidad	Tipo de contratación	Acciones
Modalidad 1	Modalidad 1	Ver, Editar, Eliminar
Modalidad 2	Modalidad 2	Ver, Editar, Eliminar
Modalidad 3	Modalidad 3	Ver, Editar, Eliminar
Modalidad 4	Modalidad 4	Ver, Editar, Eliminar
Modalidad 5	Modalidad 5	Ver, Editar, Eliminar
Modalidad 6	Modalidad 6	Ver, Editar, Eliminar
Modalidad 7	Modalidad 7	Ver, Editar, Eliminar
Modalidad 8	Modalidad 8	Ver, Editar, Eliminar
Modalidad 9	Modalidad 9	Ver, Editar, Eliminar
Modalidad 10	Modalidad 10	Ver, Editar, Eliminar

Ilustración 107. modalidad y métodos de contratación



Ilustración 108. ejemplo de creación

Cuestionarios

Con el seguimiento de vinculación, este apartado podremos generar nuestras preguntas para poder analizar a nuestros postulados y evaluar.



Ilustración 109. cuestionarios creados



Ilustración 110. creación de cuestionarios

Podremos revisar nuestras organizaciones, así como la creación, revisión de estas mismas.

Ilustración 111. creación de organizaciones parte 1

Ilustración 112. creación de organizaciones parte 2

Ilustración 113. creación de organizaciones parte 3

Estudiantes

Podremos solicitar el traspaso de los alumnos que estén postulados, así como inactivarlos y revisarlos

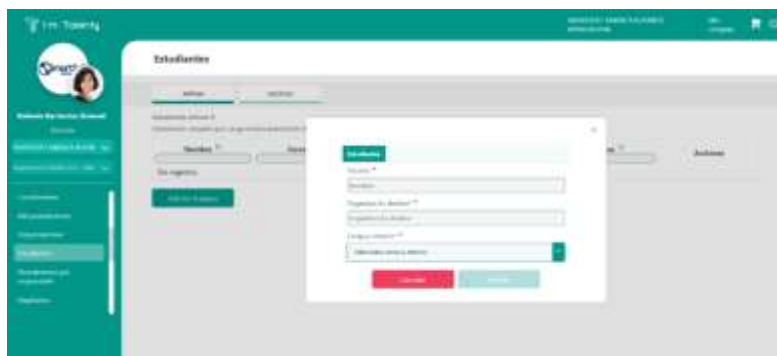


Ilustración 114.solicitud de traspaso para estudiantes

Empleados

La función es la misma que estudiantes con la diferencia que podremos seleccionar un perfil y un área específica en la cual se laborará.



Ilustración 115.área específica en la cual se laborará



Ilustración 116. vinculación de empleado

Perfiles

En este módulo podemos gestionar el tipo de perfil para asignarle a los estudiantes o empleados, así como sus permisos.



Ilustración 117.perfiles creados



Ilustración 118.registro de nuevo perfil

Mis Eventos

Aquí podremos general eventos como congresos, cursos, platicas, visitas, para que podamos llamar la atención de futuros prospectos de socios, colaboradores o estudiantes interesados en residencias o dualidad para la empresa



Ilustración 119.registro de eventos creados

Ilustración 120. creación nuevo evento

Eventos

En este apartado similar a mis eventos, podremos ver y postularnos a los diversos eventos que tengas las demás instituciones.

Ilustración 121. eventos disponibles a los cuales podemos postularnos

Mis pases.

Aquí podremos ver los pases solicitados para los eventos en los que nos hemos registrado.

Ilustración 122. pases de eventos por QR

Conclusiones

El desarrollo del sistema “I’m Talenty” representó una experiencia integral que permitió aplicar conocimientos técnicos, metodológicos y de diseño en un entorno real de desarrollo de software. A lo largo del proyecto se logró implementar una plataforma funcional, intuitiva y escalable que responde a una necesidad concreta: mejorar la vinculación entre egresados universitarios y el mercado laboral.

La integración de tecnologías como Java, HTML, CSS, SQL y JPQL, junto con herramientas como NetBeans, Navicat y TortoiseSVN, permitió construir una solución robusta y eficiente. Asimismo, la aplicación de la metodología ágil Scrum facilitó una gestión ordenada del proyecto, promoviendo la entrega continua de valor y la mejora constante.

El sistema desarrollado no solo automatiza procesos clave como la recomendación de vacantes y la gestión de postulaciones, sino que también ofrece una experiencia de usuario clara, segura y adaptable. Esto contribuye significativamente a reducir los tiempos de búsqueda de empleo, mejorar la calidad de las contrataciones y fortalecer la comunicación entre instituciones educativas, empresas y candidatos.

Recomendaciones

- Continuar con la mejora de la seguridad en el manejo de datos personales y credenciales de acceso.
- Implementar un sistema de retroalimentación por parte de los usuarios para evaluar la calidad de las recomendaciones.
- Integrar nuevas fuentes de datos y vacantes para ampliar la cobertura del sistema.
- Optimizar el rendimiento del sistema en dispositivos móviles.
- Explorar la posibilidad de incorporar inteligencia artificial para mejorar la precisión de las recomendaciones.
- Documentar y versionar continuamente el código para facilitar futuras actualizaciones y mantenimiento.

Experiencia Obtenida

Durante el desarrollo del proyecto de residencias profesionales, la experiencia adquirida fue profundamente enriquecedora tanto en el ámbito técnico como en el personal y profesional. Este proceso representó una oportunidad única para aplicar de forma práctica los conocimientos adquiridos a lo largo de la carrera de Ingeniería en Tecnologías de la Información y Comunicaciones, enfrentando retos reales que exigieron análisis, creatividad, disciplina y trabajo colaborativo.

Desde el punto de vista técnico, se fortalecieron habilidades clave en el desarrollo de software, tales como la programación orientada a objetos con Java, el diseño de interfaces web con HTML y CSS, la gestión de bases de datos con SQL y JPQL, y el uso de herramientas profesionales como NetBeans, Navicat y TortoiseSVN. Estas herramientas no solo facilitaron el desarrollo del sistema, sino que también permitieron comprender la importancia de seguir buenas prácticas de codificación, control de versiones y documentación técnica.

Uno de los aprendizajes más significativos fue la aplicación de la metodología ágil Scrum, que permitió organizar el trabajo en ciclos iterativos, priorizar tareas, adaptarse a cambios y entregar valor de forma continua. Esta experiencia me enseñó a trabajar bajo presión, a comunicarme de manera efectiva con asesores y colaboradores, y a mantener una visión clara del producto final, incluso en entornos cambiantes.

Además, el proyecto me permitió desarrollar competencias en el análisis de requerimientos, modelado de procesos, diseño de arquitectura de software y pruebas funcionales. Aprendí a estructurar un sistema desde cero, considerando aspectos como la seguridad, la escalabilidad, la experiencia del usuario y la integración de múltiples módulos.

En el plano profesional, esta experiencia me ayudó a consolidar mi identidad como ingeniero, a tomar decisiones técnicas fundamentadas y a valorar la importancia del trabajo en equipo, la ética profesional y la responsabilidad en el desarrollo de soluciones tecnológicas que impactan directamente en la vida de las personas.

Finalmente, este proyecto reafirmó mi vocación por el desarrollo de software y me motivó a seguir aprendiendo, investigando y perfeccionando mis habilidades para contribuir de manera significativa en el campo de las tecnologías de la información. Estoy convencido de que los conocimientos y experiencias adquiridas durante esta etapa serán fundamentales para mi desempeño en futuros proyectos profesionales y para mi crecimiento continuo como ingeniero.

Referencias

1. Oracle. (2024). *Documentación oficial de Java SE 8*. Recuperado de <https://docs.oracle.com/javase/8>
2. Apache NetBeans. (2024). *NetBeans IDE*. <https://netbeans.apache.org/>
3. Navicat. (2024). *Herramienta de gestión de bases de datos*. <https://www.navicat.com/>
4. TortoiseSVN. (2024). *Sistema de control de versiones*. <https://tortoisesvn.net/>
5. Schwaber, K., & Sutherland, J. (2020). *Guía oficial de Scrum*. <https://scrumguides.org/>
6. Sommerville, I. (2016). *Ingeniería de Software* (10ª ed.). Pearson Educación.
7. Pressman, R. S. (2014). *Ingeniería del Software: Un enfoque práctico* (7ª ed.). McGraw-Hill.
8. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Patrones de diseño: Elementos de software orientado a objetos reutilizable*. Addison-Wesley.
9. Fowler, M. (2003). *Patterns of Enterprise Application Architecture*. Addison-Wesley.
10. Freeman, E., & Robson, E. (2004). *Head First Design Patterns*. O'Reilly Media.