



INSTITUTO TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE APIZACO

ACCESO DIGITAL CARD-ITA FASE 2 IMPLEMENTACIÓN Y PRUEBAS

TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL

PARA OBTENER EL TÍTULO PROFESIONAL DE:
INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN Y
COMUNICACIONES

PRESENTA:
EDGAR GARCÍA DE LA CADENA ZEBADÚA
18370945

NOMBRE DEL ASESOR:
ING. MERCED PÉREZ MORENO



APIZACO, TLAX., FEBRERO DE 2025

Apizaco, Tlaxcala, **21/agosto /2024**

OFICIO No: SC/OPV/0826/24

ASUNTO: Liberación de proyecto
para titulación integral

CESAR REYNALDO RAMOS GÓMEZ
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:	EDGAR GARCÍA DE LA CADENA ZEBADÚA
CARRERA:	INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES
NO. DE CONTROL:	18370945
NOMBRE DEL PROYECTO:	ACCESO DIGITAL CARD-ITA FASE 2 IMPLEMENTACIÓN Y PRUEBAS
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



JOSÉ ANTONIO CUATEPOTZO VARELA
JEFE DEL DEPTO. DE SISTEMAS Y COMPUTACIÓN

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



MERCEZ PÉREZ MORENO
ASESOR INTERNO

c.c.p. División de Estudios Profesionales
c.c.p. Depto. de Servicios Escolares
c.c.p. Archivo
JAGV/mrol*



Tzompantepec, Tlaxcala, 16/enero/2025

GARCIA DE LA CADENA ZEBADÚA EDGAR.
EGRESADO DE LA CARRERA DE INGENIERÍA
EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES.
CON NÚMERO DE CONTROL 18370945.
PRESENTE.

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 cuyo tema es: "ACCESO DIGITAL CARD – ITA FASE 2: IMPLEMENTACIÓN Y PRUEBAS" 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 Recepcion Profesional.

Sin otro particular por el momento me despido.

ATENTAMENTE

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



SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL
DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO
DIVISIÓN DE
ESTUDIOS PROFESIONALES

ISRAEL JORGE SÁNCHEZ SÁNCHEZ
JEFE DE LA DIVISIÓN DE ESTUDIOS PROFESIONALES

ccp. Archivo
ijss/aehs



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 tecnm.mx |



■ Agradecimientos

Me siento profundamente honrado de poder plasmar mis palabras de agradecimiento en este momento y dedicárselas a todas aquellas personas que creyeron en mí, dando más de lo que merecía para que este momento pueda ser posible. Sacrificaron cosas por mí, se desvelaron por mí, hicieron a un lado lo que estaban haciendo por apoyarme, ellos me moldearon, forjaron mi carácter, le dieron una luz de esperanza a mi vida, y cada uno me brindo sus conocimientos y habilidades más destacadas, que hoy, son las herramientas con las que trabaje para llegar hasta aquí.

Agradezco a mi madre Claudia Zebadúa Velázquez, quien con su amor y cariño siempre vio por el bienestar de todos, es el corazón de la familia, ella me impulso a seguir mis sueños, sin importar que tan imposible parezca.

Agradezco a mi padre Jesús García de la Cadena Zavala, quien fue mi inspiración a buscar siempre una explicación a las cosas, a resolver los problemas y cuya influencia me encamino a estudiar lo que hoy es mi carrera profesional, y siempre nos impulsaba a ser mejores que nuestro previo yo.

Agradezco a mi hermana Alejandra García de la Cadena Zebadúa, quien fue mi guía desde mis primeros recuerdos, mi maestra, cómplice y amiga, cuyo toque artístico me permitió ver atreves del prisma de colores, hacía un mundo diferente.

Agradezco a mi abuelo Víctor Manuel Zebadúa Medrano, quien ya no esta entre nosotros, pero me dejo palabras de enseñanzas y siempre fue mi guía espiritual.

Agradezco a mi pareja la Ing. Stephani Noya Hernández, mi compañera de vida y profesionalismo, quien me acompañó de viva voz en todas y cada una de las etapas de la carrera, juntos formamos el mejor equipo que pude pedir, siempre me alentó, apoyo y se quedó conmigo en todo momento hasta el final.

Agradezco a Lisandro Said Trinidad Zempoalteca, quien es el mejor amigo que uno podría tener, es la familia que elegí, y como toda mi familia, jamás me abandono, siempre estuvo en mis peores momentos y junto a su pareja Ana son mis mas queridos amigos, junto a mi amigo fiel Paul.

Agradezco a todos mis docentes por su Catedra, y por haber compartido sus conocimientos y enseñanzas conmigo, pero quiero rendir un especial agradecimiento a la Ing. Merced Pérez Moreno y al Ing. Joel Quintero Gómez, más allá de su indiscutible labor como docentes, me impulsaron a mejorar personal y profesionalmente, ellos son los asesores de este proyecto y junto con el equipo de Car-Ita, se esforzaron mucho por ver crecer este proyecto, y llevarlo lo mas lejos posible.

Todos son las personas mas importantes en mi vida, estoy muy feliz de tenerlos conmigo, no puedo mas que agradecerles por todo lo que han hecho por mí y honrar el regalo que me han dado. Los amo a todos, Muchas gracias.

Resumen

Las TIC'S están en constante cambio e impactando en todos los ámbitos de la sociedad, incluida la educación. La tecnología es una parte integral de la sociedad actual en constante avance. Como resultado, tenemos acceso a más conocimiento y la demanda de herramientas informáticas en la educación es cada vez más exigente. De esta forma, las TIC'S se han convertido en una herramienta de apoyo a las nuevas tecnologías en la educación.

La implementación de este proyecto implicará varios pasos clave, incluida la digitalización de los registros en papel existentes, el desarrollo de portales en línea y aplicaciones móviles para acceder al instituto y la implementación de medidas de seguridad para proteger los datos confidenciales. Además, el proyecto requerirá la colaboración entre la institución y de estudiantes inscritos en la misma, para garantizar la interoperabilidad de los registros digitales en diferentes sistemas y plataformas.

Índice

Agradecimientos	2
Resumen.....	3
1. Introducción	9
2. Descripción de la empresa u organización y del puesto o área del trabajo el estudiante	10
3. Problemas a resolver, priorizándolos.....	13
4. Objetivos.....	14
5. Justificación	14
6. Descripción del proyecto	15
7. Marco Teórico.....	16
7.1 Bases teóricas	16
7.1 Entorno de trabajo.....	18
7.1.1 Android Studio	18
7.1.2 Visual Studio Code	20
7.1.3 XAMPP	23
7.1.4 Adobe Xtreme Desing (Adobe XD)	24
7.2.1 Instalación de Andoid Studio	25
7.2.2 Instalación de Visual Studio Code	28
7.2.3 Instalación de Xampp.....	31
8.0 Desarrollo	34
8.1 Herramientas de Diseño:	34
8.1.1 Estructura de Pantallas.....	34
AdobeXD(Adobe Extreme Desing):	35
AdobePremierPro:	35
Herramienta de Diseño para Base de Datos:	36
Determinar el propósito de la base de datos.....	36
8.2.1 Modelamiento de base de datos.....	36
8.2.1.1 Defina los atributos.....	36
8.2.1.2 Definir las relaciones.....	37
8.2.1.3 Establezca las restricciones.....	38
8.2.1.4 Diagramas de secuencia.....	39
8.2.1.5 Lista preliminar de clases.....	40
Diagramas de actividad.....	42
8.3 Creación de la aplicación Card-Ita.....	50
8.3.1 Definición de requerimientos funcionales	50
8.3.2 Definición de requerimientos No funcionales	55
8.3.3 Maquetación y Prototipo:	59
8.3.3.1 Creación del logo Card-Ita	59
8.3.3.2 Usos correctos e incorrectos del Logotipo.	62
Funcionalidades Principales:.....	63

8.4 Primeros pasos:	65
8.5 Configuración de la IP para uso de Red Local	67
8.7 Creación del RecoveryActivity	72
1. Creación de la Actividad	72
2. Inicialización de Componentes	72
3. Configuración del Listener del Botón	73
4. Lógica de Recuperación de Usuario	73
5. Uso	73
9.0 Resultados	74
10.0 Pruebas	80
10.1 Pruebas de escritorio por modulo.	80
Numero de modulo 001: Base de datos “register.php”	80
Resultado final:	81
Numero de modulo 002: “myDataBase.sql”	82
Numero de modulo 003: “login.php”	85
Casos adicionales:	90
Numero del modulo 004: “callback.php”	91
Numero de modulo A001: Aplicación Card-Ita “MainFest.xml”	96
Flujo de la Prueba de Escritorio	96
Simulación de la Prueba	98
Numero de la prueba A002: “GalleryFragment.java”	98
Simulación de la Prueba de Escritorio	99
Resultados de la Prueba de Escritorio	100
Descripción del Código	100
Variables de Entrada	100
Simulación de la Prueba de Escritorio	101
Resultados de la Prueba de Escritorio	101
11. Conclusiones	102
12. Competencias desarrolladas	103
13. Referencias	104

Índice de Tablas

Tabla 1.- Requerimiento funcional RF0 muestra Análisis de la información	53
Tabla 2.- Requerimiento funcional RF1 muestra Página web para administradores	53
Tabla 3.- Requerimiento funcional RF2 muestra Herramientas administrativas	54
Tabla 4.- Requerimiento funcional RF3 muestra dministración de usuarios	54
Tabla 5.- Requerimiento funcional RF4 muestra Almacenamiento de credenciales digitales	54
Tabla 6.- Requerimiento funcional RF5	55
Tabla 7.- Requerimiento funcional RF6	55
Tabla 8.- Requerimiento funcional RF7	55
Tabla 9.- Requerimiento funcional RF8	56
Tabla 10.- Requerimiento funcional RF9	56
Tabla 11.- Requerimiento funcional RF10	56
Tabla 12.- Requerimiento funcional RF11	57
Tabla 13.- Requerimiento funcional RF12	57
Tabla 14.- Requerimiento funcional RF13	57
Tabla 15.- Requerimiento funcional RF14	58
Tabla 16.- Requerimiento No funcional RNF1	58
Tabla 17.- Requerimiento No funcional RNF2	58
Tabla 18.- Requerimiento No funcional RNF3	59
Tabla 19.- Requerimiento No funcional RNF4	59
Tabla 20.- Requerimiento No funcional RNF5	60
Tabla 21.- Requerimiento No funcional RNF6	60
Tabla 22.- Requerimiento No funcional RNF7	60
Tabla 23.- Requerimiento No funcional RNF8	61
Tabla 24.- Requerimiento No funcional RNF9	61
Tabla 25.- Requerimiento No funcional RNF10	61
Tabla 26.- Requerimiento No funcional RNF11	61
Tabla 27.- Requerimiento No funcional RNF12	62

Indice de Ilustraciones

Ilustración 30.- Muestra de pantalla del maquetado y prototipado de la aplicación Card-Ita en el programa AdobeXD	62
Ilustración 31.- Captura de Android Studio desarrollando la estructura de la aplicación Card-Ita	63
Ilustración 32.- Captura de Android Studio con esta configuración, <code>MainActivity</code> se mostrará en modo vertical sin una barra de acción y tendrá el título "Card-ITA" en la interfaz de usuario.	64
Ilustración 33.- Captura de Android Studio <code>GalleryFragment.java</code> con la IP local	65
Ilustración 34.- Captura de Android Studio <code>RecoveryActivity.java</code> con la IP local	65
Ilustración 35.- Captura de Android Studio <code>RecoveryActivity.java</code> con la IP local	66
Ilustración 36.- Importaciones de Clases	67
Ilustración 37: Almacenamiento y recuperación de datos del usuario	69
Ilustración 39.- Captura de pantalla del código resultado de la prueba de escritorio 001	74
Ilustración 40.- Captura de pantalla del código a analizar en el módulo 002	75
Ilustración 41.- Captura de pantalla del código a para crear la base de datos en el módulo 002	75
Ilustración 42.- Captura de pantalla del código a para usar la base de datos en el módulo 002	75
Ilustración 43.- Captura de pantalla del código a para crear la tabla usuarios dentro de la base de datos en el módulo 002	76
Ilustración 44 .- Captura de pantalla del código a para la validación de los usuarios en el módulo 003	78
Ilustración 45.- Captura de pantalla del código de valores asignados en el módulo 003	80
Ilustración 46.- Captura de pantalla del código donde la función devuelve el siguiente array en el módulo 003	80
Ilustración 47.- Captura de pantalla del código donde se asignan los datos del usuario al array en el módulo 003	81
Ilustración 48.- Captura de pantalla del código donde se muestra la respuesta JSON con los datos del usuario en el módulo 003	81
Ilustración 49.- Captura de pantalla del código donde se envía una respuesta JSON indicando que las credenciales son incorrectas en el módulo 003	82
Ilustración 50.- Captura de pantalla del código donde se envía una respuesta JSON indigo donde se envía unaditando que falta un parámetro requerido en el módulo 003	82
Ilustración 51.- Captura de pantalla del có respuesta JSON indicando que falta un parámetro requerido en el módulo 004	84

Ilustración 52.- Captura de pantalla del código donde el token obtenido es "abc123" en el modulo 004	85
Ilustración 53.- Captura de pantalla del código donde el token codificado en JSON es "abc123" en el módulo 004	85
Ilustración 54.- Captura de pantalla del código donde se muestra el resultado del token obtenido en el módulo 004	86
Ilustración 55.- Captura de pantalla del código donde se muestra el resultado del token obtenido en el módulo 004	87
Ilustración 56.- Se muestra el código a evaluar en el modulo A002	92

1. Introducción

En la actualidad, las tecnologías de identificación y acceso están en constante evolución, adaptándose a las necesidades de seguridad y eficiencia de diversas instituciones, incluyendo las universidades. Tradicionalmente, las credenciales de acceso han sido físicas, hechas de plástico y equipadas con tecnología RFID (Identificación por Radiofrecuencia). Sin embargo, estas credenciales presentan diversas limitaciones, tales como el costo de producción, el desgaste físico, y problemas de seguridad, lo que ha llevado a explorar alternativas más modernas y eficientes.

En este contexto, la digitalización de las credenciales de acceso mediante el uso de códigos QR (Quick Response) surge como una solución innovadora y efectiva. Los códigos QR, al ser fáciles de generar, leer y utilizar, ofrecen numerosas ventajas sobre las credenciales físicas tradicionales. Este proyecto se enfoca en la transición de un sistema de credenciales de acceso universitario basado en RFID a uno basado en códigos QR, con el objetivo de mejorar la seguridad, reducir costos y aumentar la conveniencia tanto para los estudiantes como para el personal de la universidad.

Las principales ventajas de digitalizar las credenciales escolares son la comodidad, la accesibilidad y la seguridad. Se puede acceder fácilmente a las credenciales digitales desde cualquier lugar, en cualquier momento, usando un dispositivo móvil. Esto elimina la necesidad de que los estudiantes lleven documentos físicos y les permite acceder a la institución educativa de forma más rápida, sencilla y segura.

Las credenciales escolares digitalizadas son más seguras que las que están en plástico, ya que pueden cifrarse y protegerse mediante firmas digitales y otras medidas de seguridad. Esto hace que sea mucho más difícil para los estafadores alterar o falsificar credenciales académicas, lo que reduce el riesgo de robo de identidad y otras formas de fraude.

En general, la digitalización de las credenciales escolares tiene el potencial de simplificar y acelerar el proceso de acceso, además de puede ayudar a modernizar y mejorar la eficiencia del sistema educativo, facilitando que los estudiantes accedan a las oportunidades educativas que necesitan para tener éxito en sus carreras.

2. Descripción de la empresa u organización y del puesto o área del trabajo el estudiante

Datos de la empresa

La historia de la sociedad tlaxcalteca, ha sido marcada e impulsada por el trabajo, el compromiso y la visión de hombres y mujeres que con determinación y ahínco han buscado el desarrollo del estado.

Sin duda, personalidades de gran trascendencia como Don Emilio Sánchez Piedras, entonces Gobernador Constitucional, comprometido con la juventud del estado, contando con el apoyo decisivo del Presidente de la República, el Lic. Luis Echeverría Álvarez, personaje visionario del futuro de la educación y de la juventud de esta región del país, gestiona y funda el Instituto Tecnológico Regional de Apizaco e inicia sus actividades el 1 de octubre del año de 1975.

El Tecnológico de Apizaco, desde hace más de cuatro décadas, se ha dedicado a propiciar alternativas reales y sustentables para la juventud de la región. Es una Institución que pertenece al Tecnológico Nacional de México, del cual formamos parte de 254 Tecnológicos distribuidos en los 32 estados del país.

El ITA ofrece en su formación educativa 9 Ingenierías, 3 Maestrías y un Doctorado, cuyas matrículas abarcan de manera integral una formación general básica que incluye el fortalecimiento científico y metodológico; todas las carreras cuentan con especialidades diseñadas para atender las necesidades de los sectores productivo y social. Los planes de estudio están basados en una estructura genérica, una especialidad y una residencia profesional.



Ilustración 1.- Foto del Tecnológico Nacional de México/Instituto Tecnológico de Apizaco (2022)

Lugar donde se realizará el proyecto

Nombre: Tecnológico Nacional de México/Instituto Tecnológico de Apizaco

Dirección: Carretera Apizaco Tzompantepec, esquina con Av. Instituto Tecnológico S/N, Conurbado Apizaco – Tzompantepec

Teléfono: 2414172010 o 2414180321

Nombre del titular de la empresa: Sergio Octavio Rosales Aguayo

Sitio Web: <https://www.apizaco.tecnm.mx/>

Mapa de localización



Ilustración 2.- Localización del Tecnológico Nacional de México/Instituto Tecnológico de Apizaco

En la ilustración 2.-.- Localización del Tecnológico Nacional de México/Instituto Tecnológico de Apizaco, muestra lo siguiente: Localización de la institución en vista de Google maps

Es una Institución de Servicios de Educación Superior Tecnológica comprometida en la formación de profesionales, que responde a los requerimientos de desarrollo Económico, Científico, Social y Cultural que demanda nuestro entorno, a través de la adecuada integración de nuestros recursos. Tecnológico Nacional de México/Instituto Tecnológico de Apizaco, desde más de cuatro décadas, se ha dedicado a propiciar alternativas reales y sustentables para la juventud de la región. Es una Institución que pertenece al Tecnológico Nacional de México, del cual formamos parte más de 261 Tecnológicos distribuidos en los 32 estados del país.

Misión:

Formar profesionistas que sean líderes visionarios, emprendedores, con capacidad de generar soluciones disruptivas, innovando en la ciencia y la tecnología, comprometidos con el desarrollo económico, responsabilidad social teniendo una visión sostenible y sustentable.

Visión:

Ser una Institución de educación superior tecnológica de vanguardia, formadora de ciudadanos del mundo, con conocimientos intercultural, respetando la diversidad social.

Valores:**Honestidad:**

Se demuestra día a día a través de la transparencia con la ejecución de las actividades laborales cotidianas.

Responsabilidad:

Es la actitud que caracteriza al personal en el cumplimiento de su tarea con eficacia y eficiencia en beneficio de la comunidad.

Lealtad:

Es la actitud que caracteriza al personal del Instituto Tecnológico de Apizaco, aplica sus conocimientos y habilidades en el logro de las metas institucionales, sin olvidar en ningún momento el lema:

“Pensar para servir, servir para triunfar ®”

Área de trabajo del residente

El departamento de sistemas y computación

Resumen de las funciones del departamento:

Planear, coordinar, controlar y evaluar las actividades de docencia, investigación y vinculación en las áreas correspondientes a sistemas y computación que se imparten en el instituto tecnológico, de conformidad con las normas y lineamientos establecidos por la Secretaría de Educación Pública. Coordinar las actividades del departamento con las demás áreas de la Subdirección Académica. Elaborar el programa operativo anual y el anteproyecto de presupuesto del departamento y presentarlos a la Subdirección Académica para lo conducente.

3. Problemas a resolver, priorizándolos

El problema del uso de credenciales plásticas en el Instituto Tecnológico Nacional de México/Apizaco, donde según la División de Estudios Profesionales del Instituto Tecnológico Nacional de México/Apizaco hay 3,877 estudiantes en el periodo ENEJUN2023, donde necesitan llevar documentos físicos llevándonos a un problema de vulnerabilidades de seguridad, las credenciales de plástico se pueden perder, robar o replicar, lo que genera vulnerabilidades de seguridad en el sistema. Esto puede resultar en el acceso no autorizado a instalaciones o recursos, el

proceso de emisión y gestión de credenciales de plástico puede ser ineficiente, lo que genera largos tiempos de espera para los usuarios y cargas administrativas para el personal, puede haber problemas con retrasos en la obtención de credenciales o errores en la información de las credenciales, todo esto nos lleva al Impacto ambiental, la producción de credenciales de plástico tienen un impacto negativo en el medio ambiente, contribuyendo a la generación de desechos y contaminación.

Para abordar estos problemas, el Instituto Tecnológico Nacional de México/Apizaco podría considerar implementar un sistema más seguro y eficiente para emitir y administrar credenciales, como el uso de un sistema de credenciales digitales ya que la mayoría de los estudiantes cuentan con teléfonos móviles y acceso a Internet, estas herramientas se han convertido necesarias para las actividades diarias, debido a que las aplicaciones móviles (apps) son capaces de realizar funciones más específicas, por lo que se ha decidido hacer uso de las TICS.

- Resistencia al cambio: Los estudiantes pueden tener dificultades para adaptarse a la nueva forma de obtener acceso a la universidad. Pueden experimentar resistencia al cambio y preferir el uso de las credenciales de plástico.
- Falta de infraestructura: La implementación de credenciales digitales con código QR puede requerir una infraestructura tecnológica adecuada. Podría haber problemas relacionados con la disponibilidad de dispositivos móviles compatibles con QR, conectividad a Internet, lectores de códigos QR, etc.
- Seguridad: La seguridad de las credenciales digitales es un aspecto crítico para considerar. Podrían surgir preocupaciones sobre la posibilidad de falsificación o manipulación de los códigos QR, lo que comprometería la integridad de las credenciales y la seguridad del sistema.
- Experiencia del usuario: La usabilidad y la experiencia del usuario son aspectos fundamentales a tener en cuenta. Los usuarios pueden enfrentar dificultades al escanear los códigos QR o al acceder a los servicios digitales asociados. La falta de familiaridad con la tecnología o interfaces poco intuitivas podrían generar frustración.
- Disponibilidad de soporte técnico: La implementación de un nuevo sistema puede requerir un soporte técnico eficiente y oportuno. Los problemas técnicos o consultas de los usuarios relacionadas con las credenciales digitales deberán ser atendidos adecuadamente para garantizar un funcionamiento sin problemas.

4. Objetivos

Objetivo general

Lo que se pretende es crear una aplicación para dispositivos móviles que sustituya la credencial de plástico y se genere una credencial digital mediante la utilización de un código QR esto para poder acceder al Instituto Tecnológico Nacional de

México/Apizaco. Para llevar a cabo este objetivo se analizó que la mayoría de los usuarios utilizan un Smartphone, así es que se le facilita al usuario acceder de forma más rápida mediante la aplicación e ingresar a la institución ya que en ocasiones algunos alumnos llegan a olvidar su credencial, como tal la implementación de una aplicación móvil nos dará el rápido acceso al instituto del mismo modo podrá ser una forma de identificación para poder demostrar que son estudiantes del tecnológico.

Objetivos específicos

- Un mejor control de acceso.
- Aumentar la seguridad de información personal de los alumnos.
- Brindar una alternativa al uso del plástico.
- Hacer una mejora con el uso de las Tics.
- “Multiusuario” para maestros y estudiante.

5. Justificación

Actualmente podemos decir que casi el 78% de todo lo que realizamos en el día está respaldado por la tecnología. Con el paso de los años, se ha ido implantando en nuestra vida cotidiana hasta el punto de que ahora resulta complicado realizar algunas tareas o trabajos sin tecnología.

Los móviles han cobrado una gran importancia en nuestra vida y con ellos, han llegado las Apps móviles, que nos ayudan a tener cualquier servicio a la mano.

Una de las ventajas es que las Apps facilitan la extensión de las empresas en el ámbito móvil, ya que estas contribuyen a complementar las estrategias de mercadeo. Permitiendo el aprovechamiento de las funcionalidades en el dispositivo, tales como la cámara de geolocalización y la mayoría de ellas no necesita una conexión a internet demasiado potente, como ocurre con las páginas web adaptadas a terminales móviles.

El desarrollo de una aplicación móvil para el Instituto Tecnológico Nacional de México/Apizaco brindará un valor agregado a los servicios que esta institución brinda a sus alumnos, abriendo camino al desarrollo de nuevas soluciones que permitan acceder a la información en cualquier momento y desde cualquier lugar. La nueva tendencia de la tecnología es dar a los usuarios la posibilidad de tener todo lo que pueden llegar a necesitar en un dispositivo que brinde “movilidad”, definida como la capacidad de acceder a la información y a los servicios en cualquier momento, de cualquier manera y en cualquier parte.

El presente proyecto tiene como finalidad desarrollar una aplicación móvil, como medio viable que facilite el acceso de alumnos en caso de olvidar, extraviar o desgaste de la credencial física. Brindando así una nueva alternativa de acceso a la institución.

6. Descripción del proyecto

El proyecto tiene como objetivo crear una aplicación móvil donde se encuentre en esta una credencial digital que permita a los estudiantes acceder a la universidad por medio de un código QR, el cual será escaneado en un lector de códigos QR en los accesos del instituto, eliminando la necesidad de credenciales plásticas físicas. El proyecto implica identificar las credenciales requeridas, diseñar una plataforma segura con autenticación y encriptación, desarrollar una base de datos y API, probar la plataforma y brindar capacitación y soporte a los usuarios. Este proyecto tiene el potencial de mejorar la seguridad de datos personales, reducir la carga administrativa y mejorar la experiencia del usuario.

Aquí una breve descripción de cómo se ejecutará este proyecto:

Identificar las credenciales requeridas: El primer paso es identificar todas las credenciales de plástico que necesitan ser digitalizadas. Esto se refiere a las tarjetas de identificación de estudiantes.

Diseñar una plataforma digital: El siguiente paso sería diseñar una plataforma digital que pueda almacenar todas las credenciales requeridas de forma segura. La plataforma debe ser fácil de usar y debe ser accesible para todos los estudiantes y administradores.

Integrar medidas de autenticación y seguridad: La plataforma debe incluir fuertes medidas de autenticación y seguridad para garantizar que solo las personas autorizadas puedan acceder a las credenciales y al administrador a estas. Esto podría incluir autenticación multifactor, protección con contraseña y encriptación.

Desarrollar una base de datos: Se debe desarrollar una base de datos para almacenar todas las credenciales digitalmente. Esta base de datos debe ser segura y debe poder manejar grandes volúmenes de datos.

Prueba y lanzamiento: La plataforma debe probarse exhaustivamente para asegurarse de que funciona según lo previsto. Una vez que se completan las pruebas, la plataforma se puede iniciar y poner a disposición de todos los estudiantes.

Brindar capacitación y soporte: Se debe brindar capacitación y soporte para garantizar que todos los usuarios puedan acceder a sus credenciales digitalmente. Esto podría incluir tutoriales en línea, soporte técnico y sesiones de capacitación.

En general, la digitalización de credenciales plásticas para el acceso a la universidad tiene el potencial de mejorar la seguridad de información de los alumnos, reducir la carga administrativa y mejorar la experiencia del usuario para estudiantes.

7. Marco Teórico

El objetivo de este marco teórico es proporcionar una base sólida para el desarrollo del proyecto, describiendo los conceptos fundamentales, las tecnologías involucradas, y los beneficios esperados de la implementación del acceso digital Card-Ita

7.1 Bases teóricas

1. Tecnología RFID

Definición y Principios de Funcionamiento:

- RFID (Radio Frequency Identification) es una tecnología que utiliza ondas de radio para identificar y rastrear etiquetas adheridas a objetos.
- Los componentes principales incluyen el transpondedor (etiqueta RFID) y el lector RFID.

Tipos de RFID:

- Pasivo: No tiene batería y es activado por el lector.
- Activo: Tiene su propia fuente de energía.
- Semipasivo: Tiene una batería pero usa la energía del lector para comunicarse.

Aplicaciones en Identificación y Control de Acceso:

- Uso en credenciales para controlar acceso a edificios y áreas restringidas.
- Implementación en bibliotecas, inventarios y seguimiento de productos.

Ventajas y Desventajas del RFID:

- Ventajas: No requiere línea de visión, puede leer múltiples etiquetas simultáneamente.
- Desventajas: Costos más altos, susceptibilidad a interferencias y problemas de seguridad.

2. Tecnología de Códigos QR

Definición y Principios de Funcionamiento:

- QR (Quick Response) es un tipo de código de barras bidimensional que puede almacenar información en un formato legible por dispositivos móviles.
- Generación mediante algoritmos específicos que convierten datos en un patrón de puntos.

Estructura y Tipos de Información Codificable:

- Puede contener URL, texto, datos de contacto, ubicación geográfica, etc.

- Consiste en módulos (cuadrados negros) distribuidos en una cuadrícula sobre un fondo blanco.

Aplicaciones de los Códigos QR en la Identificación y Control de Acceso:

- Uso en credenciales digitales para acceso a eventos, edificios y sistemas de transporte.
- Escaneo mediante cámaras de smartphones o lectores específicos.

Ventajas y Desventajas de los Códigos QR:

- Ventajas: Bajo costo, fácil de generar y escanear, amplio uso en dispositivos móviles.
- Desventajas: Requiere línea de visión, menor capacidad de datos comparado con RFID.

3. Comparación entre RFID y Códigos QR

Costo de Implementación y Mantenimiento:

- Análisis de costos de producción, mantenimiento y reemplazo de ambas tecnologías.

Seguridad:

- Evaluación de la seguridad de datos en RFID y códigos QR.
- Métodos de encriptación y autenticación utilizados en códigos QR.

Eficiencia y Usabilidad:

- Facilidad de uso y accesibilidad para los usuarios finales.
- Velocidad y precisión en la lectura de credenciales.
- 4. Seguridad en la Identificación Digital

Protocolos de Seguridad para Códigos QR:

- Métodos de autenticación, encriptación y verificación de la información.
- Prevención de ataques como phishing y falsificación de códigos QR.

Comparación de Seguridad entre RFID y Códigos QR:

- Vulnerabilidades comunes y cómo mitigarlas.
- Ejemplos de brechas de seguridad en ambos sistemas y lecciones aprendidas.

4. Marco Legal y Normativo

Regulaciones sobre Protección de Datos:

- Leyes y normativas aplicables a la gestión y protección de datos personales en la digitalización de credenciales.

- Cumplimiento de GDPR (Reglamento General de Protección de Datos) y otras regulaciones locales e internacionales.

Aspectos Éticos:

- Consideraciones éticas en la recolección, almacenamiento y uso de datos personales.
- Importancia de la privacidad y consentimiento informado de los usuarios.

7.1 Entorno de trabajo

1. Software por utilizar

- Android Studio
- Visual Studio Code
- XAMPP
- Adobe XtremeDesing

7.1.1 Android Studio

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones Android. Fue lanzado por Google en 2013 y está basado en IntelliJ IDEA, un entorno de desarrollo de software para Java. Aquí hay algunas características clave de Android Studio:

1. **Editor de Código Inteligente:** Ofrece autocompletado de código, detección de errores en tiempo real, y sugerencias para mejorar el código.
2. **Diseñador de Interfaces Gráfico:** Permite arrastrar y soltar elementos de interfaz de usuario (UI) y ver cómo se verán en diferentes dispositivos y tamaños de pantalla.
3. **Emulador de Android:** Permite probar aplicaciones en una variedad de dispositivos virtuales y configuraciones de Android sin necesidad de hardware físico.
4. **Integración con GitHub:** Facilita la gestión del código fuente y la colaboración con otros desarrolladores.
5. **Herramientas de Depuración y Pruebas:** Incluye herramientas avanzadas para depurar aplicaciones, realizar pruebas unitarias y de integración, y optimizar el rendimiento de las aplicaciones.
6. **Compatibilidad con Gradle:** Utiliza Gradle como sistema de construcción, lo que permite una configuración flexible y potente del proyecto.

7. **Compatibilidad con Kotlin y Java:** Aunque inicialmente solo soportaba Java, ahora también es compatible con Kotlin, que es el lenguaje de programación preferido por Google para el desarrollo de aplicaciones Android.

¿Por qué Android Studio es la mejor elección para realizar el proyecto?

1. **Soporte Oficial de Google:** Al ser el IDE oficial de Google para el desarrollo de aplicaciones Android, recibe actualizaciones y soporte directo de Google, garantizando compatibilidad y acceso a las últimas herramientas y APIs.
2. **Integración Completa:** Ofrece una integración completa con el ecosistema de Android, incluyendo el acceso a Google Play, herramientas de monetización, y servicios de Google como Firebase.
3. **Compatibilidad con Kotlin:** Kotlin es el lenguaje preferido para el desarrollo de aplicaciones Android. Android Studio soporta completamente Kotlin, ofreciendo ventajas como un código más conciso y seguro.
4. **Editor de Interfaces de Usuario:** Permite diseñar interfaces de usuario con un editor gráfico que facilita la creación y modificación de layouts, soportando también el diseño responsivo para diferentes tamaños de pantalla y dispositivos.
5. **Emulador de Android:** El emulador de Android integrado permite probar aplicaciones en múltiples versiones de Android y dispositivos sin necesidad de hardware físico, acelerando el proceso de desarrollo y prueba.
6. **Herramientas de Depuración y Perfilado:** Incluye herramientas avanzadas para depurar y perfilar aplicaciones, ayudando a identificar y corregir problemas de rendimiento y errores.
7. **Integración con Herramientas de Construcción y Pruebas:** Utiliza Gradle como sistema de construcción, lo que permite automatizar y gestionar el proceso de construcción de la aplicación. Además, soporta pruebas unitarias y de interfaz de usuario (UI) para asegurar la calidad del código.
8. **Documentación y Comunidad:** Al ser la herramienta oficial, cuenta con una amplia documentación y una gran comunidad de desarrolladores, facilitando la resolución de problemas y el aprendizaje continuo.

- 9. Compatibilidad con Librerías y Plugins:** Tiene una vasta cantidad de librerías y plugins disponibles que extienden sus funcionalidades, permitiendo personalizar y mejorar el entorno de desarrollo según las necesidades del proyecto

7.1.2 Visual Studio Code

Visual Studio Code (VS Code) es un editor de código fuente desarrollado por Microsoft. Es conocido por ser ligero pero potente, y está diseñado para ser extensible y personalizable. Aquí hay algunas de sus características clave:

- 1. Multiplataforma:** Disponible para Windows, macOS y Linux, lo que permite a los desarrolladores trabajar en cualquier sistema operativo.
- 2. Soporte para Múltiples Lenguajes:** Admite una amplia variedad de lenguajes de programación y tecnologías, como JavaScript, Python, Java, C++, C#, PHP, Go, TypeScript, y muchos más. Los desarrolladores pueden agregar soporte adicional para otros lenguajes a través de extensiones.
- 3. Extensiones y Plugins:** Tiene una vasta biblioteca de extensiones disponibles en el Marketplace, que permite a los desarrolladores agregar características y funcionalidades según sus necesidades. Esto incluye soporte para frameworks específicos, herramientas de depuración, linters, y más.
- 4. IntelliSense:** Ofrece características avanzadas de autocompletado de código, incluyendo sugerencias de sintaxis, definiciones de funciones y métodos, y autocompletado de variables y módulos.
- 5. Control de Versiones Integrado:** Integración directa con sistemas de control de versiones como Git y otros, permitiendo a los desarrolladores manejar sus repositorios directamente desde el editor.
- 6. Depuración Integrada:** Permite depurar aplicaciones directamente desde el editor, con soporte para puntos de interrupción, inspección de variables, y seguimiento de la ejecución del código.
- 7. Terminal Integrada:** Incorpora una terminal en el editor, lo que permite a los desarrolladores ejecutar comandos sin salir de su entorno de desarrollo.
- 8. Editor Ligero y Rápido:** Diseñado para ser rápido y eficiente, incluso con archivos grandes y múltiples proyectos abiertos simultáneamente.

9. **Personalización:** Ofrece una amplia gama de opciones de personalización, incluyendo temas, iconos, atajos de teclado, y configuración del editor para adaptarse a las preferencias individuales de los desarrolladores.
10. **Soporte para Contenedores y Desarrollo Remoto:** Facilita el desarrollo en contenedores Docker y entornos remotos, lo que es útil para trabajar en proyectos que requieren entornos específicos.

¿Por qué es la mejor elección para el proyecto?

1. **Ligereza y Rapidez:** VS Code es conocido por ser rápido y ligero, lo que permite a los desarrolladores trabajar de manera eficiente incluso en sistemas con recursos limitados.
2. **Multiplataforma:** Funciona en Windows, macOS y Linux, permitiendo a los desarrolladores mantener un entorno de desarrollo consistente sin importar el sistema operativo que utilicen.
3. **Extensibilidad:** Posee una vasta biblioteca de extensiones en el Marketplace, lo que permite personalizar y ampliar sus funcionalidades según las necesidades del desarrollador. Las extensiones cubren una amplia gama de herramientas y lenguajes de programación, desde soporte para nuevos lenguajes hasta integraciones con servicios y herramientas de desarrollo.
4. **IntelliSense:** Ofrece autocompletado inteligente de código, proporcionando sugerencias basadas en el contexto del código, lo que mejora la productividad y reduce errores.
5. **Depuración Integrada:** Incluye potentes herramientas de depuración que permiten establecer puntos de interrupción, inspeccionar variables, y seguir el flujo de ejecución del código directamente desde el editor.
6. **Control de Versiones:** La integración con Git y otros sistemas de control de versiones está incorporada, lo que facilita la gestión del código fuente y la colaboración con otros desarrolladores.
7. **Terminal Integrada:** Incorpora una terminal que permite ejecutar comandos sin salir del entorno de desarrollo, lo que es conveniente para ejecutar scripts, comandos de compilación, y otras tareas.

8. **Soporte para Desarrollo Remoto:** Permite el desarrollo en entornos remotos y contenedores Docker, facilitando el trabajo en proyectos que requieren configuraciones específicas de entorno.
9. **Comunidad y Soporte:** VS Code tiene una gran comunidad de usuarios y desarrolladores que contribuyen con extensiones, documentación y soporte, lo que hace que encontrar soluciones a problemas comunes sea más sencillo.
10. **Actualizaciones Regulares:** Microsoft proporciona actualizaciones frecuentes con nuevas características, mejoras de rendimiento y correcciones de errores, asegurando que los desarrolladores tengan acceso a las herramientas más recientes y eficientes.

7.1.3 XAMPP

XAMPP es una distribución de software libre que contiene Apache, MySQL (MariaDB en versiones más recientes), PHP y Perl. Es una herramienta muy popular utilizada para crear y probar entornos de desarrollo web localmente antes de desplegarlos en un servidor de producción. Aquí tienes un desglose de lo que incluye:

1. **Apache:** Es el servidor web que permite alojar y servir aplicaciones web.
2. **MySQL/MariaDB:** Es el sistema de gestión de bases de datos relacional que permite almacenar y gestionar los datos de las aplicaciones web.
3. **PHP:** Es un lenguaje de programación del lado del servidor ampliamente utilizado para el desarrollo web dinámico.
4. **Perl:** Es otro lenguaje de programación que se incluye en XAMPP, aunque es menos comúnmente utilizado en el desarrollo web moderno comparado con PHP.
5. **Herramientas adicionales:** XAMPP también incluye otras herramientas útiles como phpMyAdmin (para la gestión de bases de datos MySQL a través de una interfaz web), FileZilla FTP Server, Mercury Mail Server y más.

Ventajas de usar XAMPP:

1. **Instalación Sencilla:** Proporciona una manera fácil y rápida de instalar y configurar todos los componentes necesarios para un entorno de desarrollo web.
2. **Multiplataforma:** Está disponible para Windows, macOS y Linux, lo que permite a los desarrolladores trabajar en cualquier sistema operativo.
3. **Ambiente de Desarrollo Local:** Permite probar aplicaciones web localmente sin necesidad de un servidor en línea, lo que facilita el desarrollo y la prueba antes de desplegar en producción.
4. **Flexibilidad y Configuración:** Incluye una interfaz de control que facilita la administración de los diferentes componentes (como iniciar y detener Apache y MySQL).

Usos Comunes de XAMPP:

1. **Desarrollo y Pruebas:** Ideal para desarrollar y probar aplicaciones web en un entorno local.
2. **Aprendizaje y Experimentación:** Es una excelente herramienta para aprender a desarrollar aplicaciones web y experimentar con diferentes tecnologías web sin necesidad de un servidor en línea.
3. **Prototipos y Demostraciones:** Permite crear prototipos rápidos y realizar demostraciones de aplicaciones web.

7.1.4 Adobe Xtreme Desing (Adobe XD)

Adobe XD es una herramienta de diseño y prototipado de experiencia de usuario (UX) y de interfaz de usuario (UI) desarrollada por Adobe Inc. Está diseñada para facilitar el proceso de creación de interfaces de usuario y experiencias de usuario para aplicaciones móviles, sitios web y otras plataformas digitales. Aquí hay algunas características y beneficios clave de Adobe XD:

Diseño de Interfaz de Usuario: Adobe XD permite crear diseños de alta fidelidad para aplicaciones móviles y sitios web. Ofrece herramientas para dibujar y diseñar elementos de interfaz, como botones, menús, formularios y más.

Prototipado Interactivo: Permite a los diseñadores crear prototipos interactivos de sus diseños, lo que significa que pueden simular la navegación y las interacciones dentro de una aplicación o sitio web. Esto es útil para probar y validar el diseño antes de la implementación.

Colaboración en Tiempo Real: Adobe XD facilita la colaboración entre diseñadores y equipos de desarrollo. Los proyectos pueden ser compartidos y editados en tiempo real, lo que permite una colaboración más efectiva y rápida.

Integración con Otros Productos de Adobe: Se integra perfectamente con otras herramientas de Adobe, como Photoshop e Illustrator, lo que permite importar y exportar recursos fácilmente entre diferentes aplicaciones.

Componentes y Símbolos: Permite crear componentes reutilizables (llamados símbolos) que pueden ser usados en múltiples lugares dentro de un proyecto. Esto facilita la consistencia y la eficiencia en el diseño.

Responsive Resize: Ofrece herramientas para redimensionar de manera automática los elementos de la interfaz para adaptarse a diferentes tamaños de pantalla y dispositivos, lo que es crucial para el diseño responsive.

Plugins y Extensiones: Adobe XD tiene un ecosistema de plugins y extensiones que amplían sus capacidades. Los diseñadores pueden agregar funcionalidades adicionales según sus necesidades específicas.

Herramientas de Animación: Incluye herramientas para crear animaciones y transiciones entre pantallas, lo que ayuda a visualizar mejor cómo se comportará la interfaz en la vida real.

Previsualización y Compartición: Los diseñadores pueden previsualizar sus prototipos en dispositivos móviles y compartir enlaces de sus proyectos con clientes o equipos para recibir retroalimentación.

Accesibilidad y Sistemas de Diseño: Adobe XD soporta la creación y gestión de sistemas de diseño, lo que ayuda a mantener la coherencia visual y funcional en proyectos grandes.

a. Lenguajes de programación

b. Java

c. Php my admin

d. HTML5

e. C++

7.2 Instalación de software requerido

7.2.1 Instalación de Andoid Studio

Entrar al navegador de nuestra preferencia y buscamos “Descargar Android Studio”

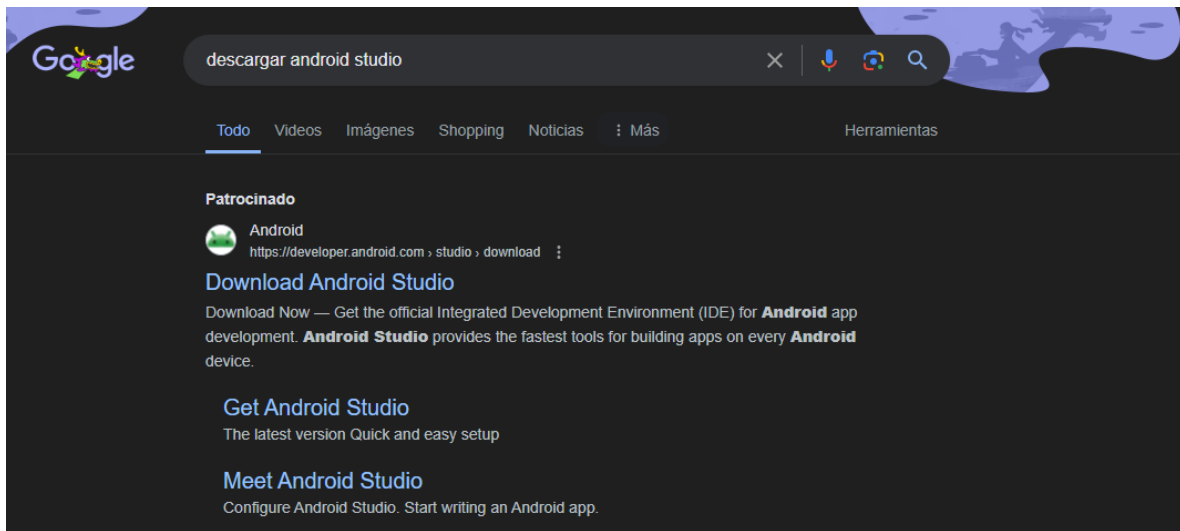


Ilustración 3.- Captura de pantalla de Google arrojando la búsqueda “Descargar Android Studio”

Entramos al primer link que aparece en la ilustración 3, es un enlace directo a la pagina de Android.

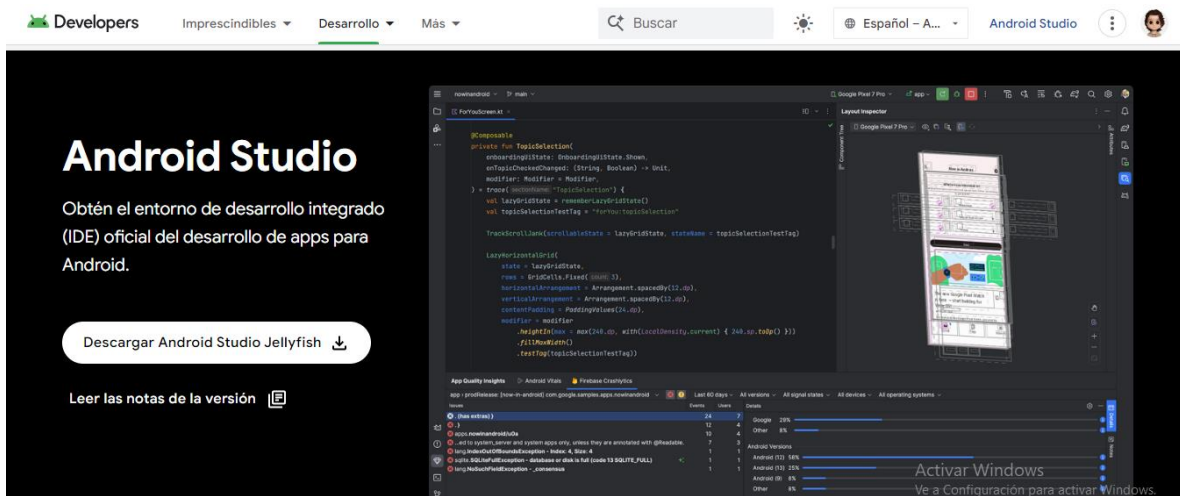
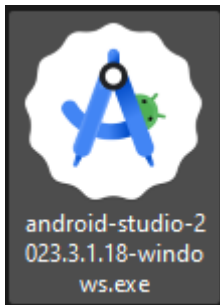


Ilustración 4.- Pagina principal de Android

Hacemos click en el botón “Descargar Android Studio: JallyFish”



- Se descargara un archivo en nuestro ordenador con este nombre y la extensión .exe, que será el paquete instalador del programa.
- Solo tenemos que hacer doble click para iniciar la instalación

Ilustración 5.- Logo del archivo creado al iniciar la descarga de Android studio Jelly Fish

Ejecutamos el archivo .exe de Android Studio y hacemos click en Next.

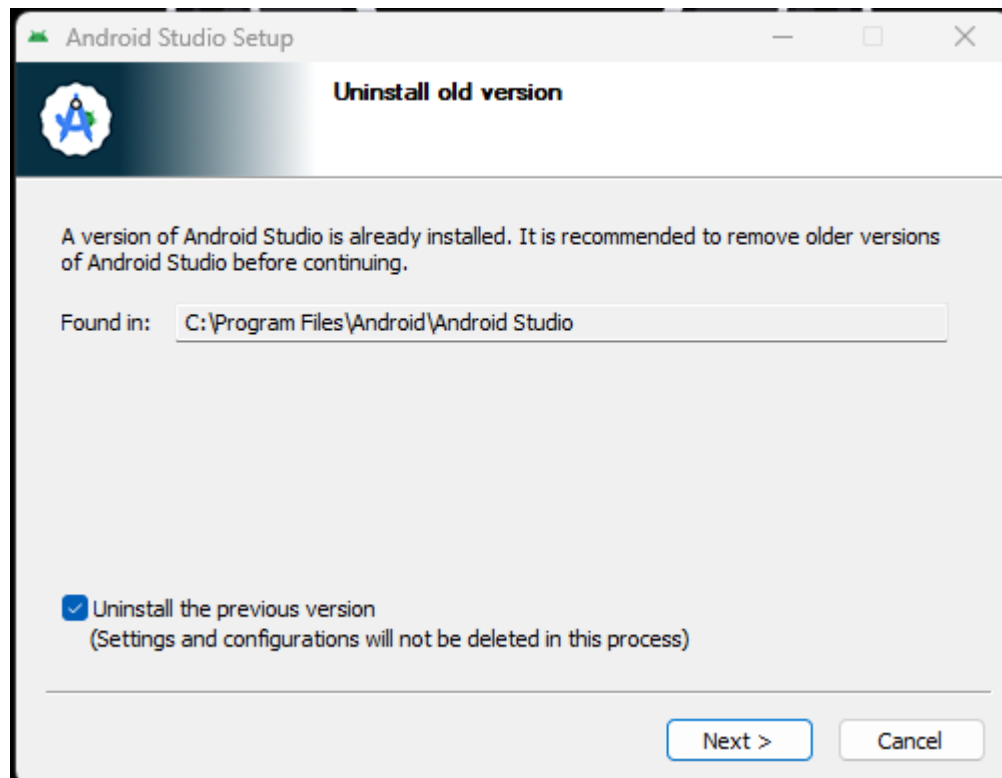


Ilustración 6.- Instalador de Android Studio con la ruta donde se alojaran los archivos

7.2.2 Instalación de Visual Studio Code

En el navegador buscar la pagina de Visual Studio Code y seleccionar el sistema operativo para descargar, en esta ocasión descargare la versión de Windows 11.

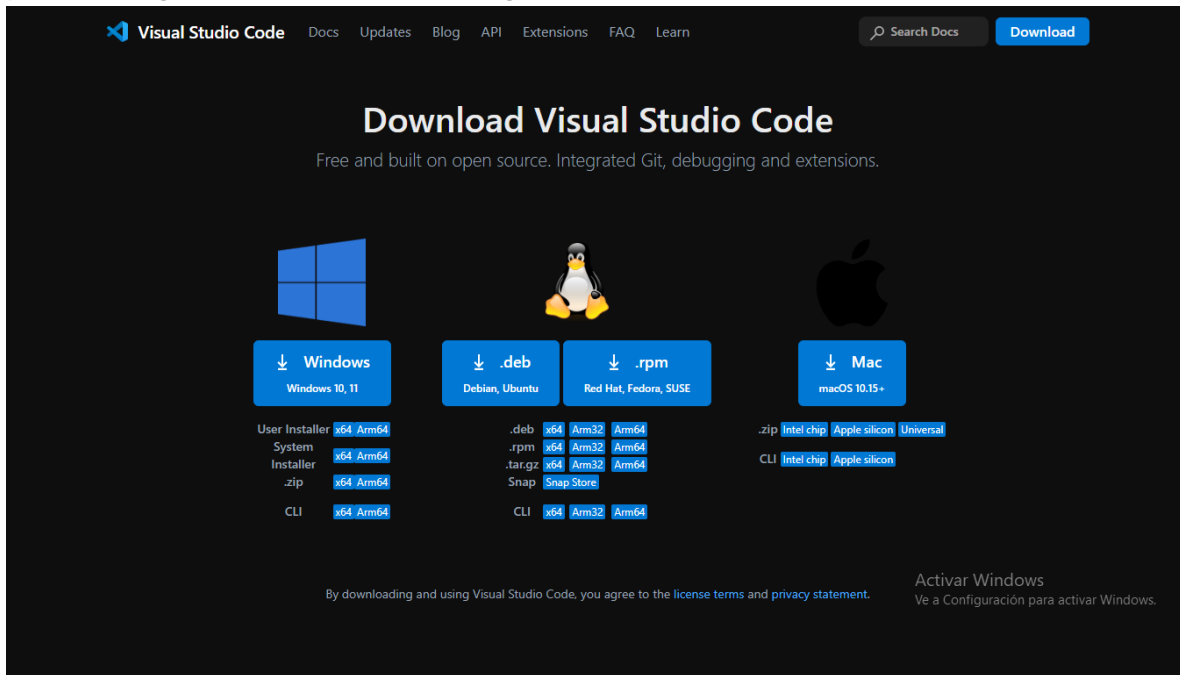


Ilustración 7.- En la ilustración podemos ver el sitio web oficial de visual studio code

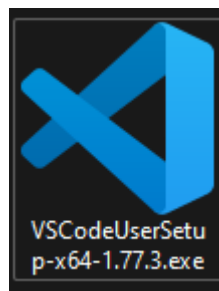


Ilustración 8.- En la ilustración podemos ver el instalador formato .exe de visual studio code

Al ejecutar el instalador daremos click en acepto el acuerdo

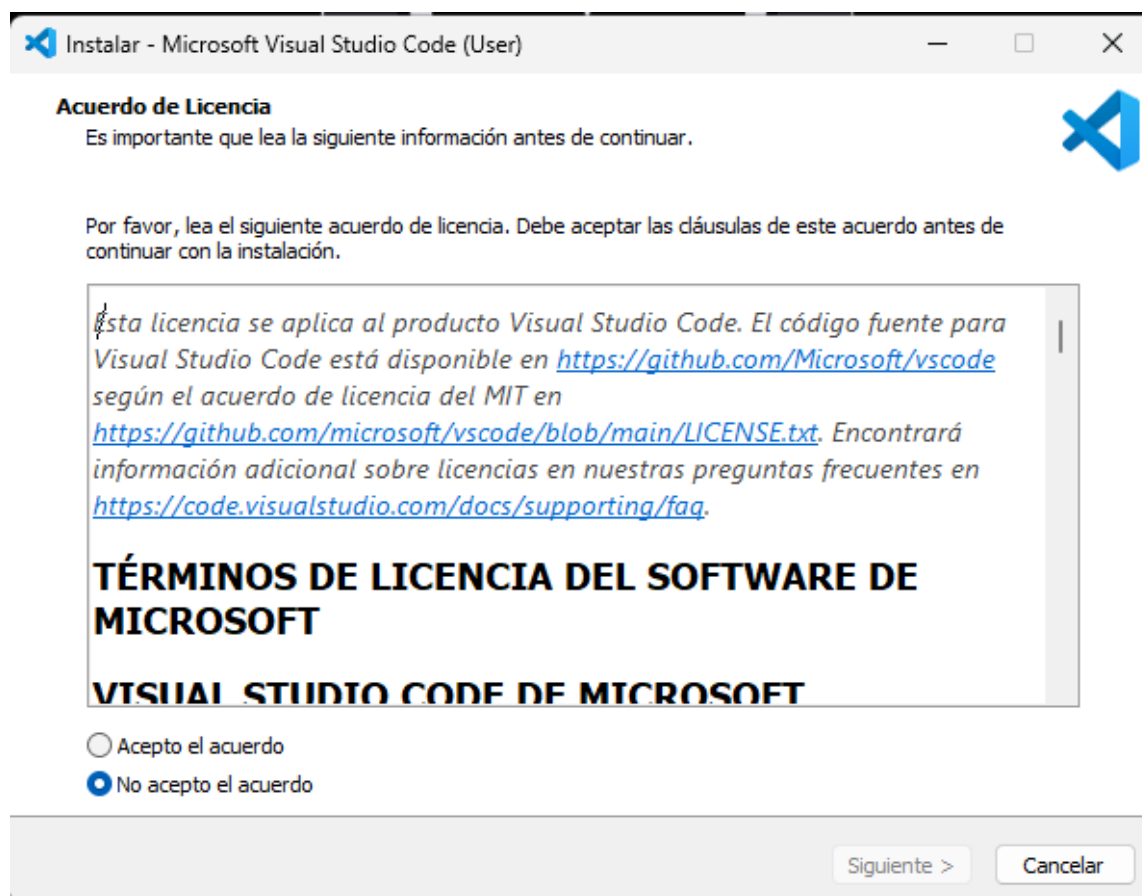


Ilustración 9.- En la ilustración podemos ver el software de instalación

Después de aceptar los acuerdos, nos pedirá seleccionar las tareas adicionales que deseamos agregar como extensión a Visual Studio Code. damos en siguiente para que comience la instalación.

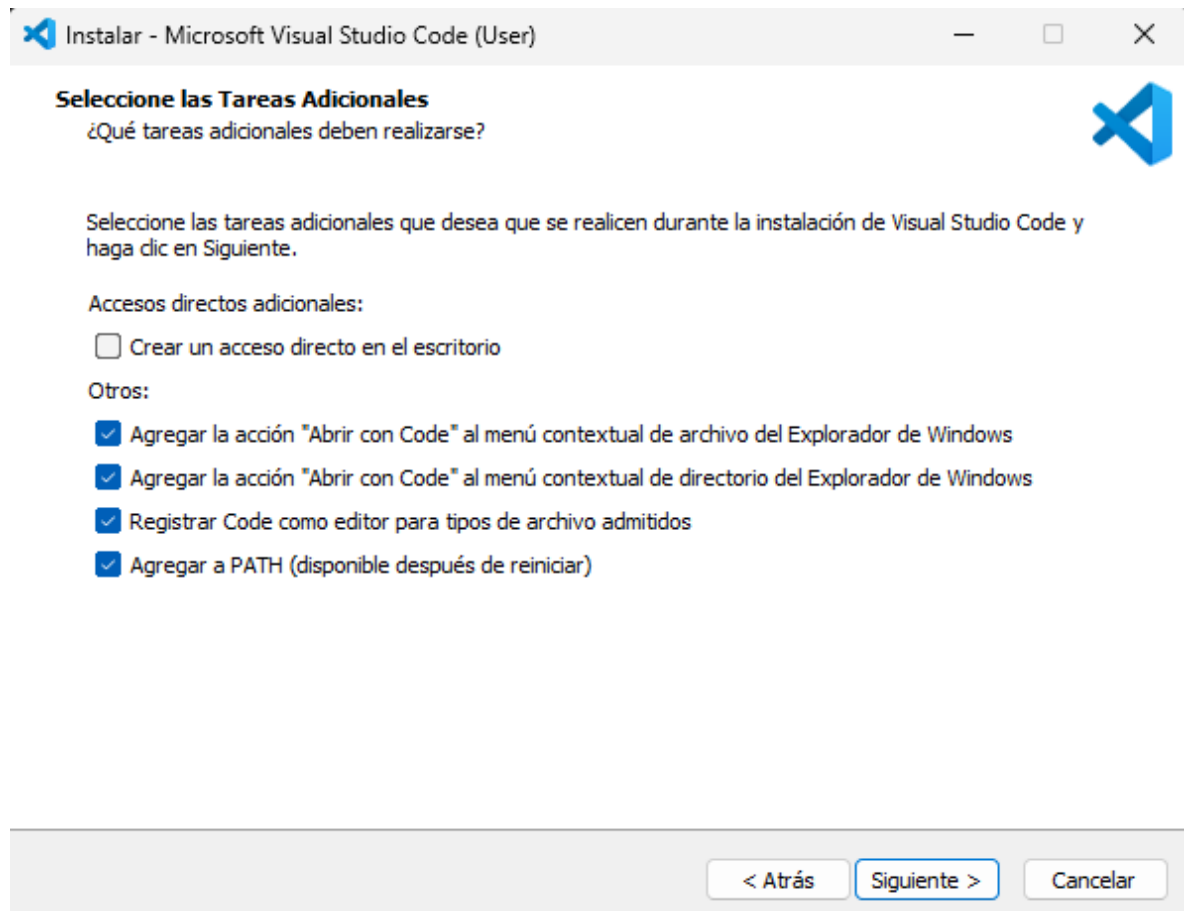
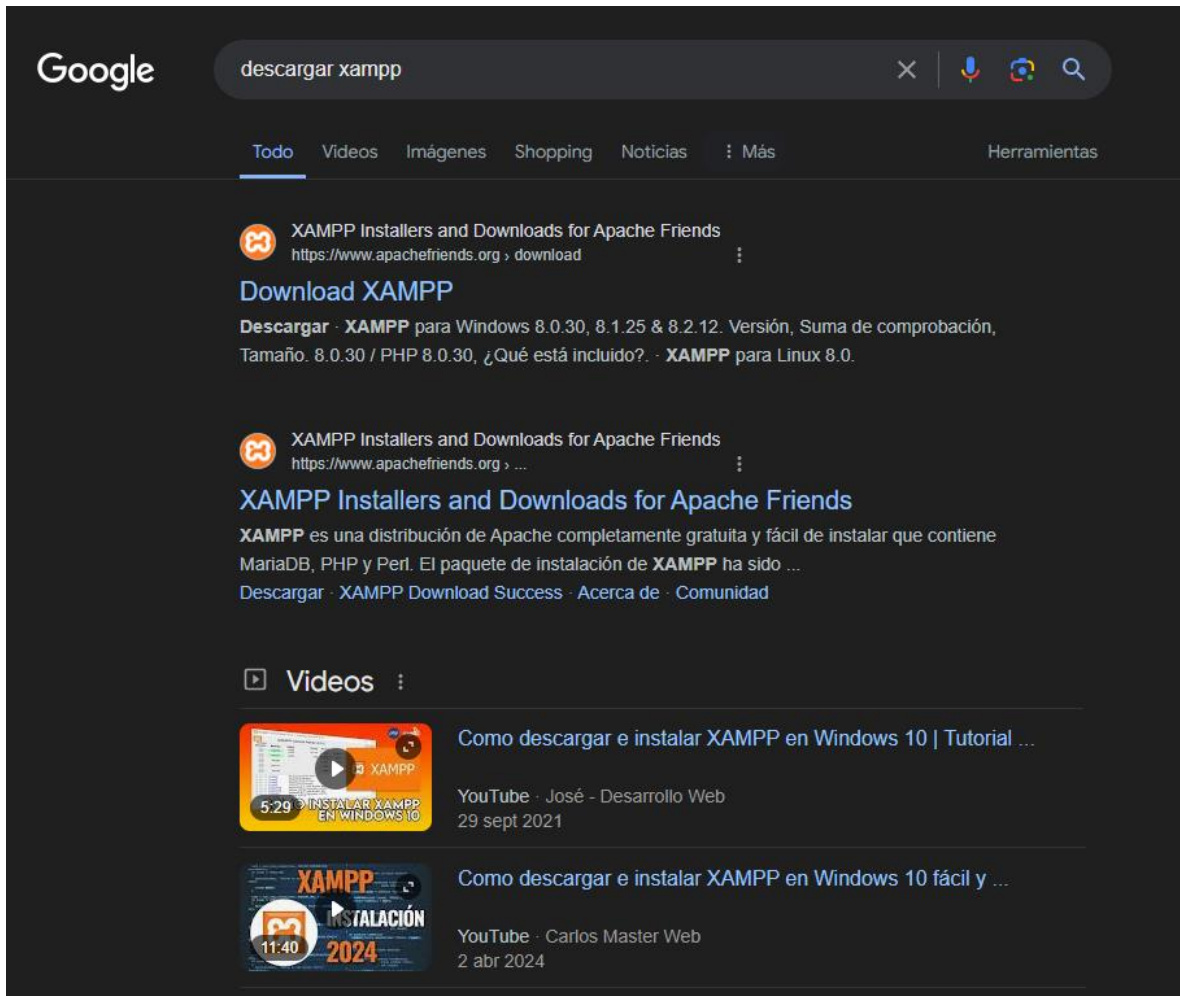


Ilustración 10.- En la ilustración podemos ver el software de instalación, en el apartado de selección de tareas adicionales

7.2.3 Instalación de Xampp

Vamos al navegador y buscamos “Descargar Xampp” y entramos al primer enlace que aparece en la búsqueda.



Ilustracion 11.- En la ilustración podemos ver el resultado de la búsqueda en Google “Descargar Xampp”

Después debemos ubicar la versión que va con nuestro sistema operativo, en este caso Windows, y hacemos click en la 3ra opción, la Suma de comprobación “MD5 SHA1”.

[Apache Friends](#) [Descargar](#) [Alojamiento](#) [Comunidad](#) [Acerca de](#) [Buscar](#) [ES](#)

Descargar

XAMPP es una distribución de Apache fácil de instalar que contiene MariaDB, PHP y Perl. Simplemente descarga y ejecuta el instalador. ¡Es así de fácil!

 **XAMPP para Windows 8.0.30, 8.1.25 & 8.2.12**

Versión		Suma de comprobación		Tamaño
8.0.30 / PHP 8.0.30	¿Qué está incluido?.	md5 sha1	Descargar (64 bit)	144 Mb
8.1.25 / PHP 8.1.25	¿Qué está incluido?.	md5 sha1	Descargar (64 bit)	148 Mb
8.2.12 / PHP 8.2.12	¿Qué está incluido?.	md5 sha1	Descargar (64 bit)	149 Mb

[Requisitos](#) [Más Descargas »](#)

Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms here.

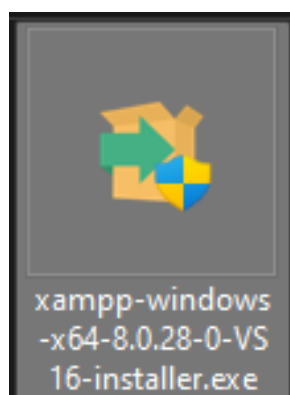
Documentación/FAQs

No hay un manual para XAMPP. Escribimos la documentación en forma de preguntas frecuentes (FAQs). ¿Tienes una pregunta que no está respondida? Prueba los Foros o Stack Overflow.

- Linux Preguntas frecuentes
- Windows Preguntas frecuentes
- OS X Preguntas frecuentes

[Activar Windows](#)
Ve a Configuración para activar Windows.

Ilustracion 12.- En la ilustración podemos ver el sitio oficial de XAMPP para descargar el software



Ilustracion 13.- En la ilustración podemos ver el archivo .exe de instalación del servicio de XAMPP

Una vez ejecutemos el archivo .exe se abrirá el software de instalación de XAMPP, daremos click en “Next” para que empiece la instalación.

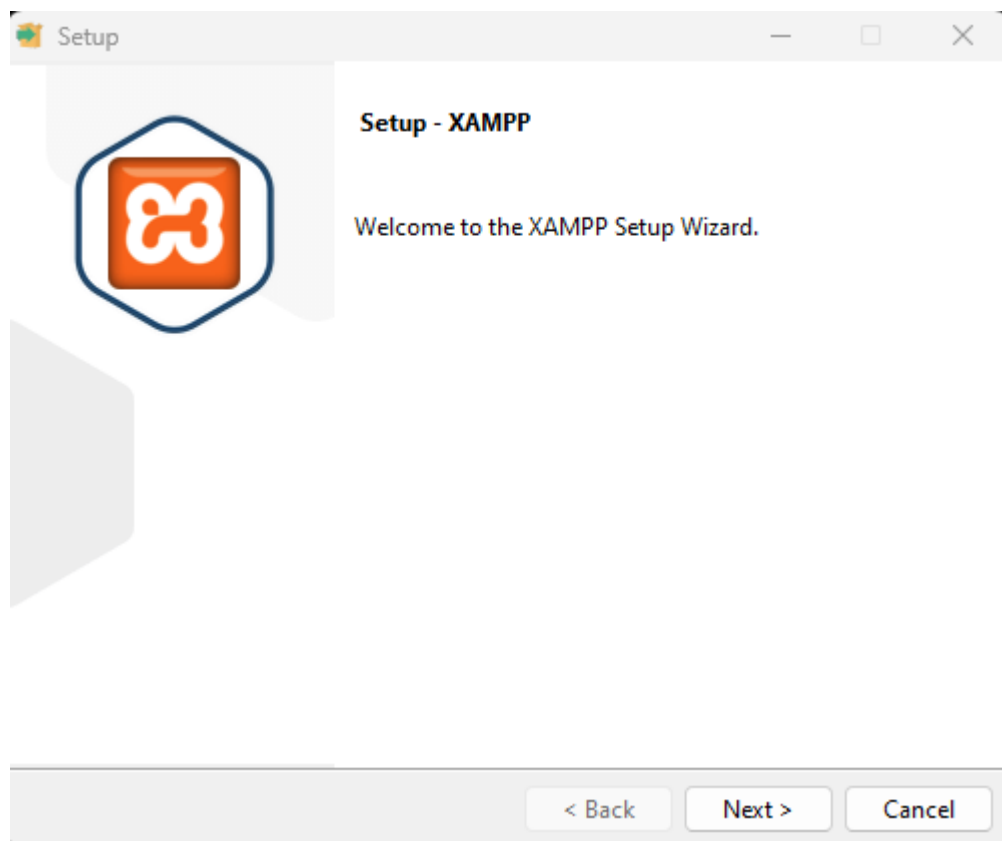


Ilustración 14.- En la ilustración podemos ver el programa instalador de XAMPP

8.0 Desarrollo

8.1 Herramientas de Diseño:

Adobe Illustrator: Utilicé Adobe Illustrator para el diseño de las credenciales digitales y la creación de la paleta de colores. Esta herramienta proporciona una amplia gama de funciones y herramientas para la creación de ilustraciones vectoriales y diseño gráfico. Con Adobe Illustrator, pude diseñar las credenciales digitales de manera precisa, utilizando formas, colores, tipografías y otros elementos gráficos para lograr un diseño atractivo y profesional. Además, aproveché las capacidades de Adobe Illustrator para crear una paleta de colores coherente y armoniosa que reflejara la identidad visual del proyecto.

8.1.1 Estructura de Pantallas

Adobe Illustrator: Utilicé Adobe Illustrator para crear la estructura de pantallas de la aplicación móvil. Esta herramienta me permitió diseñar y organizar los elementos de interfaz de usuario de manera visual, lo que resultó en un flujo de navegación intuitivo y una experiencia de usuario coherente. Con Adobe Illustrator, pude diseñar y colocar los diferentes elementos de la interfaz de usuario, como botones, iconos, campos de entrada y etiquetas, de acuerdo con las mejores prácticas de diseño y usabilidad.

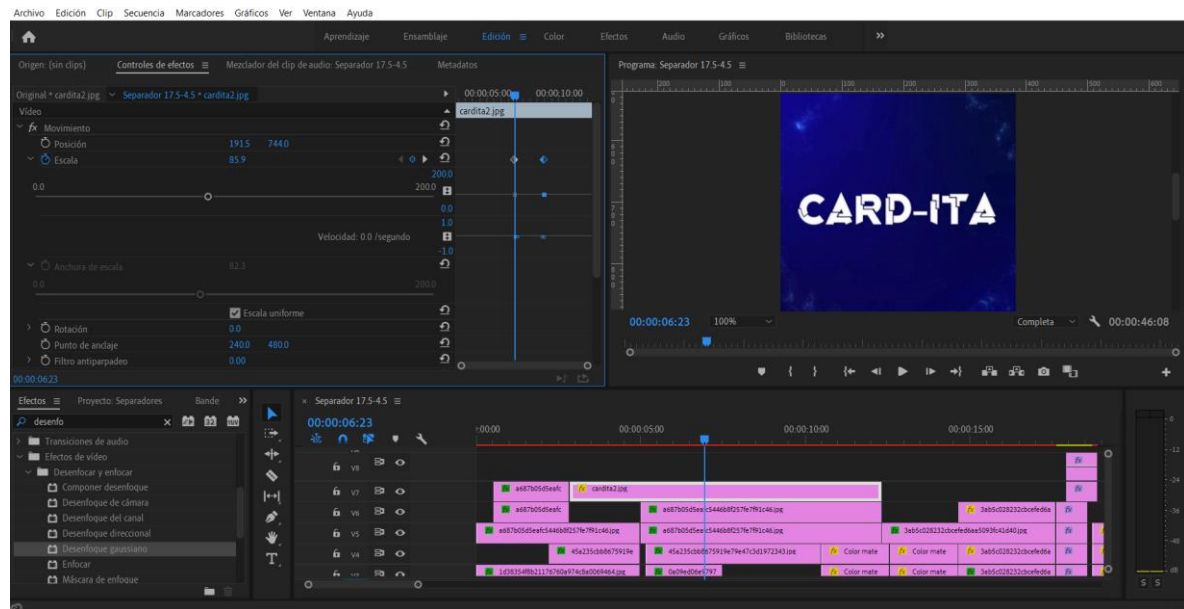
AdobeXD(Adobe Extreme Desing):

Además de Adobe Illustrator, también utilicé otras herramientas de diseño gráfico y prototipado, como Adobe XD, para crear prototipos interactivos y realizar pruebas

de usabilidad con los usuarios. Estas herramientas me permitieron simular la experiencia de uso de la aplicación antes de su implementación, lo que ayudó a identificar posibles mejoras y realizar ajustes en el diseño.

AdobePremierPro:

Utilice Adobe premier pro, un software de edición de video profesional, como complemento para el video de pantalla de carga de la aplicación Card- Ita, así como gif que se necesitaban para el funcionamiento y prototipado de la aplicación.



Herramienta de Diseño para Base de Datos:

MySQL Workbench: Utilicé MySQL Workbench como herramienta para el desarrollo de la estructura y elementos de la base de datos del proyecto. MySQL Workbench es una herramienta visual que proporciona una interfaz gráfica intuitiva para diseñar, modelar y administrar bases de datos MySQL.

Con MySQL Workbench, pude crear y definir la estructura de la base de datos para el proyecto de la aplicación móvil. Esto incluyó la creación de tablas, la definición de campos y tipos de datos, la especificación de claves primarias y secundarias, y el establecimiento de relaciones entre las tablas. Utilicé el lenguaje de modelado de bases de datos (como el Modelo Entidad-Relación) para representar visualmente la estructura de la base de datos y asegurarme de que cumpliera con los requisitos del proyecto.

Además, MySQL Workbench ofreció funcionalidades para generar scripts SQL a partir del diseño del modelo de base de datos, lo que facilitó la implementación y migración de la base de datos en el entorno de desarrollo.

La elección de MySQL Workbench se basó en su amplia adopción y popularidad como herramienta de administración de bases de datos MySQL, su interfaz visual

8.2 Creación de la base de datos

Determinar el propósito de la base de datos

El propósito de la base de datos es almacenar información relacionada con las credenciales de los estudiantes. Esto incluiría el nombre del estudiante, número de identificación, estado de inscripción y cualquier otra información relevante necesaria para verificar su identidad y acceso a la universidad. La base de datos también almacenara información sobre cada código QR, como cuándo y dónde se emitió, con fines de seguimiento y auditoría. El personal autorizado, como administradores, tendrían acceso seguro a la base de datos para verificar las credenciales de los estudiantes en los puntos de entrada.

8.2.1 Modelamiento de base de datos

8.2.1.1 Defina los atributos

Estudiantes

- **Id_estudiante**
- Nombre
- Apellido Paterno
- Apellido Materno
- No. Control
- Imagen generativa de QR

Credencial Digital

- **Id_credencial**
- QR
- **Id_estudiante**

Registro de Acceso

- **Id_Acceso**
- Ubicación_de_acceso
- Fecha
- Hora
- **Id_credencial**

8.2.1.2 Definir las relaciones

Tabla de estudiantes: Esta tabla almacenaría información sobre cada estudiante, como su nombre, carrera e identificación. Cada estudiante tendría una identificación única, que sería la clave principal de la tabla.

Tabla de credenciales digitales: Esta tabla almacenaría información sobre cada credencial digital, como su identificación única y la identificación del estudiante al que corresponde. Cada estudiante tendría una relación de uno a uno con una credencial digital, lo que significa que cada estudiante tendría solo una credencial digital y cada credencial digital correspondería a un solo estudiante. Para modelar esta relación, la tabla Credenciales digitales incluiría una clave foránea que haga referencia al ID del alumno en la tabla Alumnos.

Tabla de registros de acceso: esta tabla almacenaría información sobre cada vez que se escanea una credencial digital, como la fecha y la hora del escaneo y la identificación de la credencial digital que se escaneó. Cada registro de acceso correspondería a una credencial digital y cada credencial digital correspondería a un estudiante. Para modelar esta relación, la tabla de registros de acceso incluiría una clave externa que hace referencia al ID de la credencial digital en la tabla de credenciales digitales.

La tabla Credenciales digitales contiene información sobre cada credencial digital, incluida su identificación, la identificación del estudiante al que pertenece y la fecha en que se emitió y caduca.

La tabla de registros de acceso contiene información sobre cada registro de acceso, incluida su identificación, la identificación de la credencial digital que se utilizó para acceder a la universidad, la fecha y hora del acceso y la ubicación de la entrada donde se produjo el acceso.

8.2.1.3 Establezca las restricciones

1. Cada alumno solo puede tener una credencial digital, y cada credencial digital solo puede corresponder a un alumno. Esta relación uno a uno asegura que cada estudiante solo pueda acceder a la universidad utilizando su propia credencial digital única, y que cada credencial digital corresponda a un solo estudiante.

2. Cada registro de acceso debe estar asociado a una credencial digital válida. Esto garantiza que los registros de acceso solo se registren cuando se escanee una credencial digital válida.
3. Cada registro de acceso debe estar asociado con una fecha y hora únicas. Esto garantiza que no se puedan crear múltiples registros de acceso para la misma credencial digital al mismo tiempo.
4. Cada registro de acceso debe estar asociado con una credencial digital válida que pertenezca a un estudiante válido. Esto asegura que solo los estudiantes que tengan credenciales digitales válidas puedan acceder a la universidad y que solo se creen registros de acceso para estudiantes válidos.

8.2.1.4 Diagramas de secuencia

Diagrama de secuencia de inicio de sesión Alumno a la aplicación

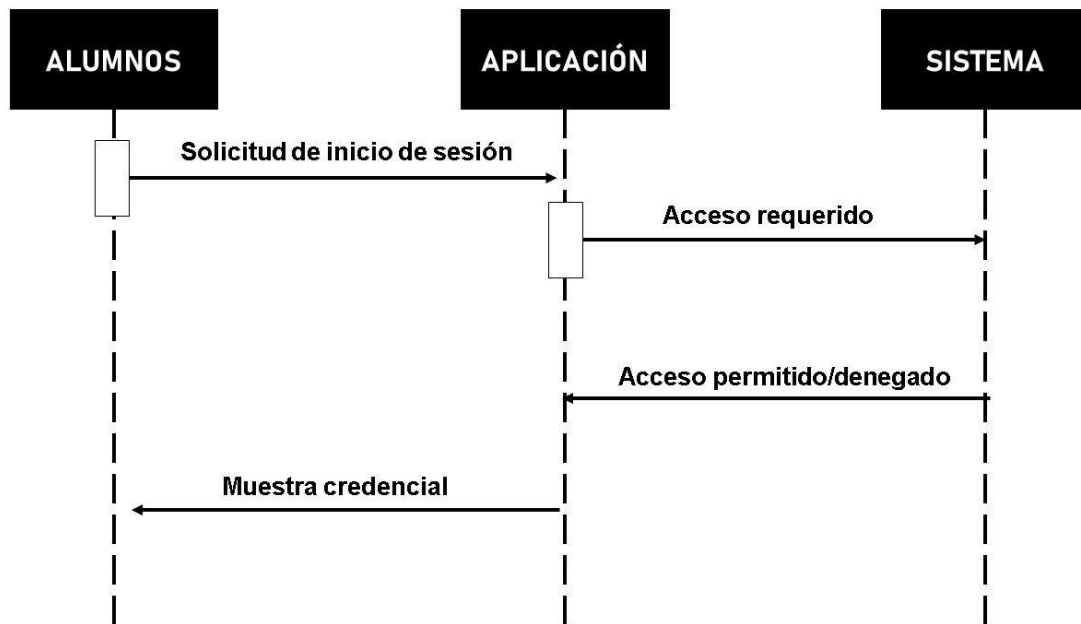


Ilustración 15.- Diagrama de secuencia de inicio de sesión Alumno a la aplicación

En la ilustración 22.- Diagrama de secuencia de inicio de sesión alumnos a la aplicación, muestra lo siguiente:

En este diagrama de secuencia, el usuario inicia una solicitud de inicio de sesión ingresando sus credenciales (por ejemplo, número de control y contraseña) en la aplicación móvil. La aplicación móvil envía la solicitud de acceso requerido, que verifica las credenciales y envía una respuesta que indica si el inicio de sesión fue permitido o denegado.

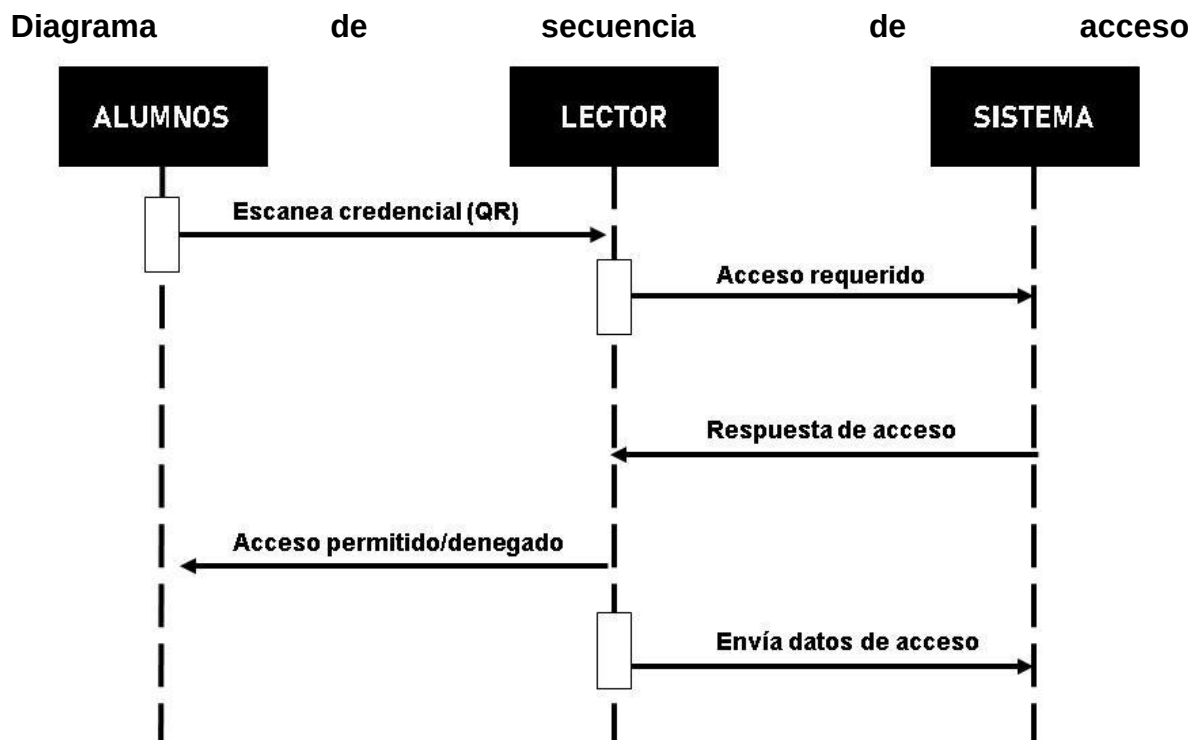


Ilustración 16.- Diagrama de secuencia de acceso

En este diagrama de secuencia, el estudiante escanea su credencial digital (el código QR) al lector en la entrada del instituto. El lector escanea el código QR y envía una solicitud de acceso al sistema. El sistema verifica la credencial y devuelve una respuesta que indica si se debe otorgar o denegar el acceso. Luego, el lector permite o deniega el acceso según la respuesta del servidor. El lector envía los datos de acceso de entrada y salida del estudiante al sistema.

8.2.1.5 Lista preliminar de clases

Estudiante: una clase que representa a un estudiante que usa la aplicación móvil para acceder al instituto.

Administrador: una clase que representa a un administrador que administra la base de datos de estudiantes y el sistema de control de acceso.

Credencial digital: una clase que representa la credencial digital del estudiante, que contiene el código QR utilizado para el acceso.

Access Reader: una clase que representa el lector de códigos QR en la entrada del instituto, que escanea las credenciales digitales.

Sistema de control de acceso: una clase que representa el sistema principal que controla el acceso al instituto.

Registro de acceso: una clase que representa el registro de acceso de los estudiantes al instituto, incluidas las horas y ubicaciones de entrada y salida.

Credenciales de usuario: una clase que representa las credenciales de inicio de sesión para estudiantes y administradores.

Base de datos de estudiantes: una clase que representa la base de datos que contiene información del estudiante, incluido el nombre, el número de control y otros detalles relevantes.

Estas clases pueden tener atributos y métodos asociados, según los requisitos específicos de este proyecto. Por ejemplo, la clase Estudiante podría tener atributos como nombre, número de control y contraseña, así como métodos para iniciar sesión y acceder al instituto. La clase Registro de acceso podría tener atributos como ID de estudiante, fecha y hora de entrada, fecha y hora de salida y ubicación, así como métodos para actualizar el registro y recuperar datos de acceso.

Es importante identificar las clases y sus relaciones al principio del proceso de desarrollo para garantizar un sistema funcional y bien diseñado.

Diagramas de actividad

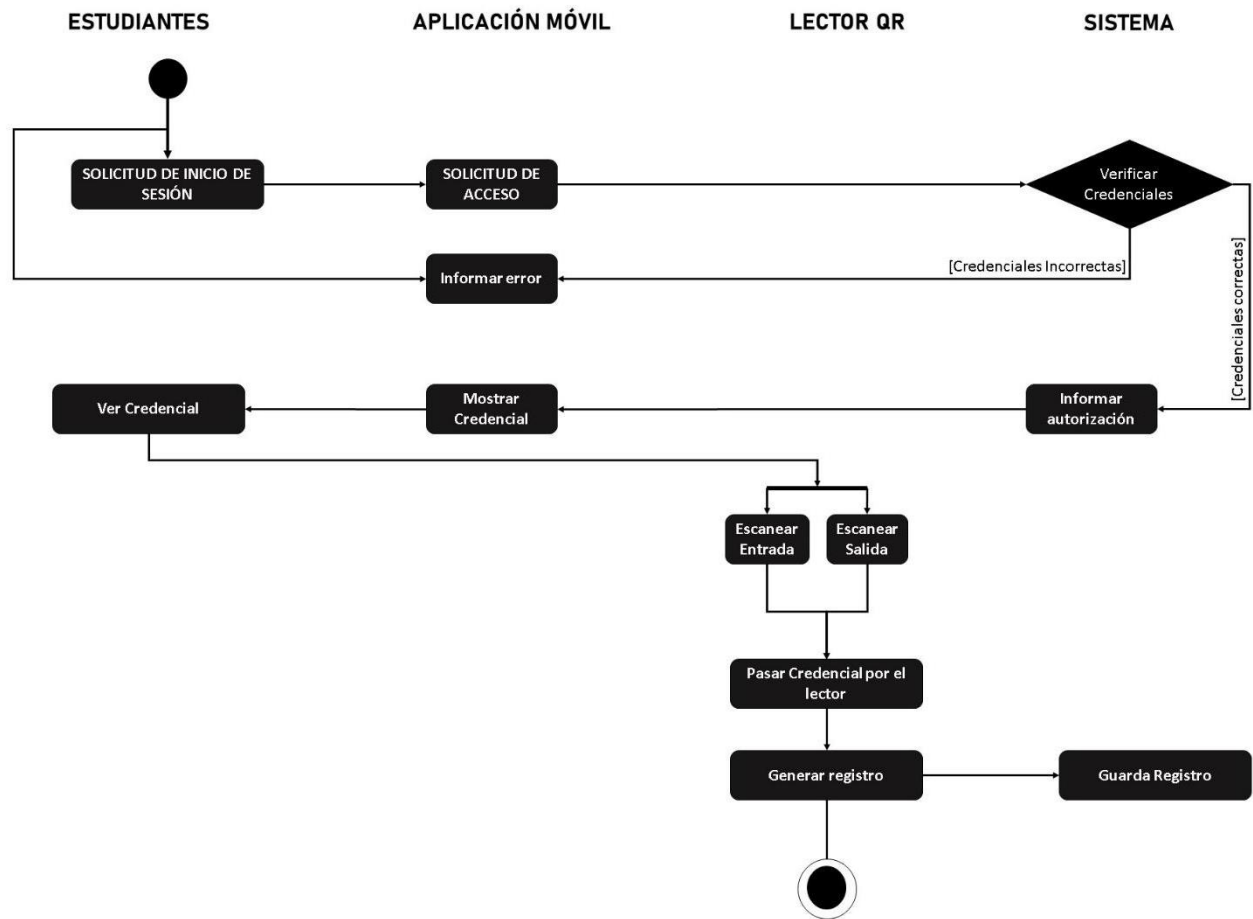


Ilustración 17.- Diagramas de actividad del sistema

Si las credenciales son válidas, el sistema responde con un mensaje de "Inicio de sesión permitido" y el estudiante es redirigido a la actividad "Acceso", donde presenta su credencial digital (código QR) al lector en la entrada del instituto. Luego, el lector escanea el código QR y envía una solicitud al sistema para otorgar acceso. El sistema verifica la credencial y devuelve un mensaje de "Conceder acceso" o "Denegar acceso", dependiendo de si la credencial es válida o no.

Si se otorga el acceso, el estudiante puede ingresar al instituto y el sistema registra los datos de acceso, incluida la hora, la fecha y la ubicación de la entrada. Luego, el sistema actualiza el registro de acceso del estudiante, representado por la actividad "Actualizar registro de acceso".

Abrimos el Programa de XAMPP y activamos los servidores de Apache y MySQL

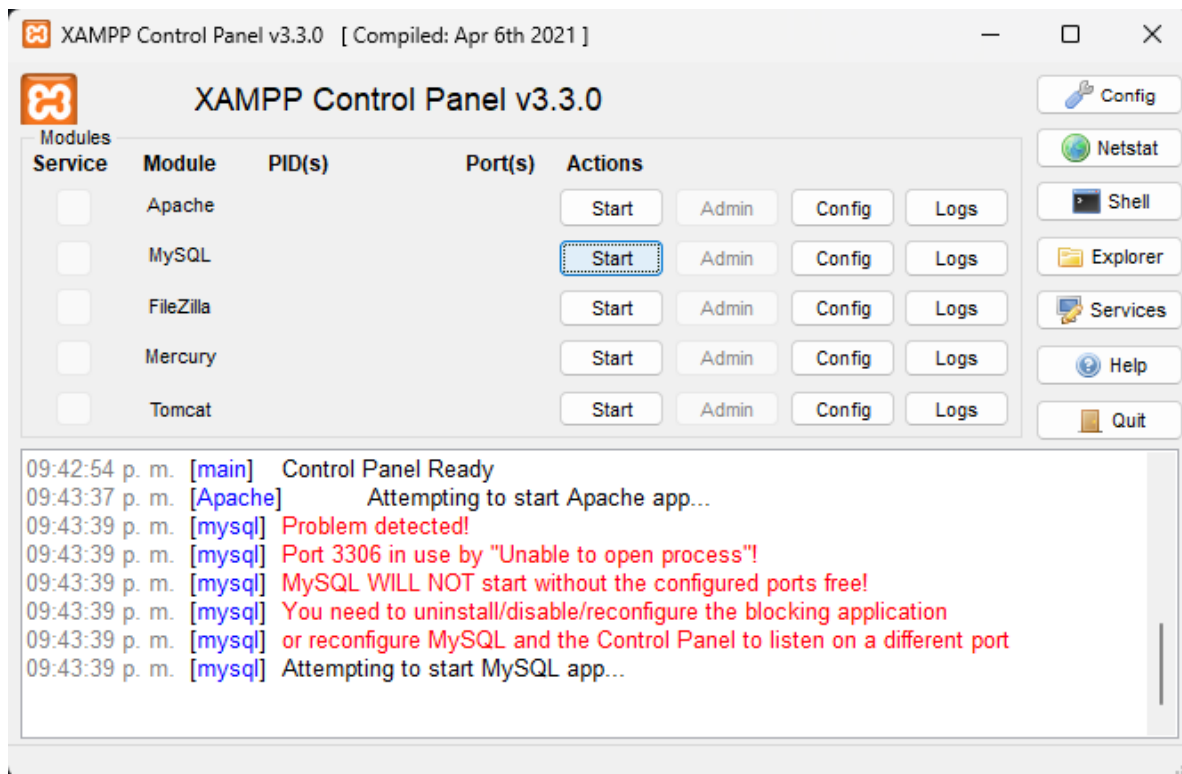


Ilustración 18.- Se muestra el programa XAMPP en ejecución iniciando el servidor MySQL

En el programa Visual Studio Code vamos a crear un nuevo proyecto ubicado dentro de una carpeta en el ordenador, dentro de esta carpeta crearemos un nuevo archivo llamado “db.sql” en dicho documento crearemos la base de datos con los atributos necesarios, como los son (id, provider, provider_value) así como también su primary key, esta tabla nos servirá para crear la base de datos de los alumnos en la aplicación.

```
db.sql
Email > db.sql
1 CREATE TABLE `google_oauth` (
2   `id` int(11) NOT NULL AUTO_INCREMENT,
3   `provider` varchar(255) NOT NULL,
4   `provider_value` text NOT NULL,
5   PRIMARY KEY (`id`)
6 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
7
```

Ilustración 19.- Captura del código escrito en visual studio code para la creación de la base de datos

Entramos a nuestro servidor local digitando nuestra ruta de acceso o dando click en el panel de XAMPP, en el botón llamado “Admin”



Welcome to XAMPP for Windows 8.0.28

You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the FAQs section or check the HOW-TO Guides for getting started with PHP applications.

XAMPP is meant only for development purposes. It has certain configuration settings that make it easy to develop locally but that are insecure if you want to have your installation accessible to others.

Start the XAMPP Control Panel to check the server status.

Community

XAMPP has been around for over 10 years. It has a large community of users and developers. You can find more information about the community in the FAQs section or check the HOW-TO Guides for getting started with PHP applications.

Ilustración 20.- Captura de pantalla de el sitio web de nuestro servidor local de XAMPP

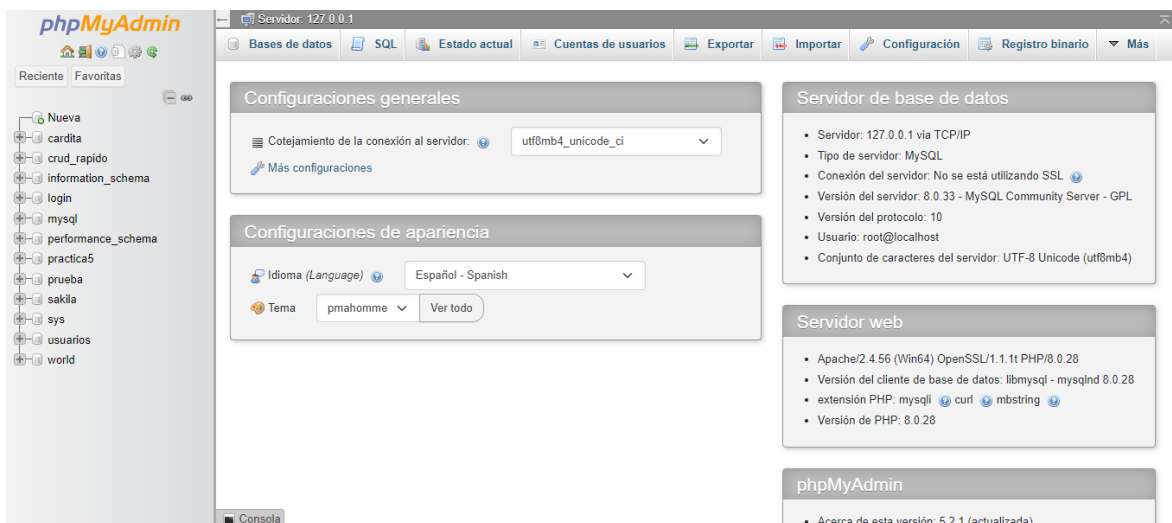


Ilustración 21.- Captura de pantalla del administrador phpMyAdmin, donde se muestran las bases de datos que tenemos en el servidor local

Una vez dentro del sitio web nos dirigiremos al apartado que dice phpMyAdmin, en donde nos mostrará nuestro panel de control de las bases de datos que tengamos en el servidor local, iremos a la parte de “importar” para cargar nuestra base de datos, esta debe ser el documento que antes llamamos “db.sql”

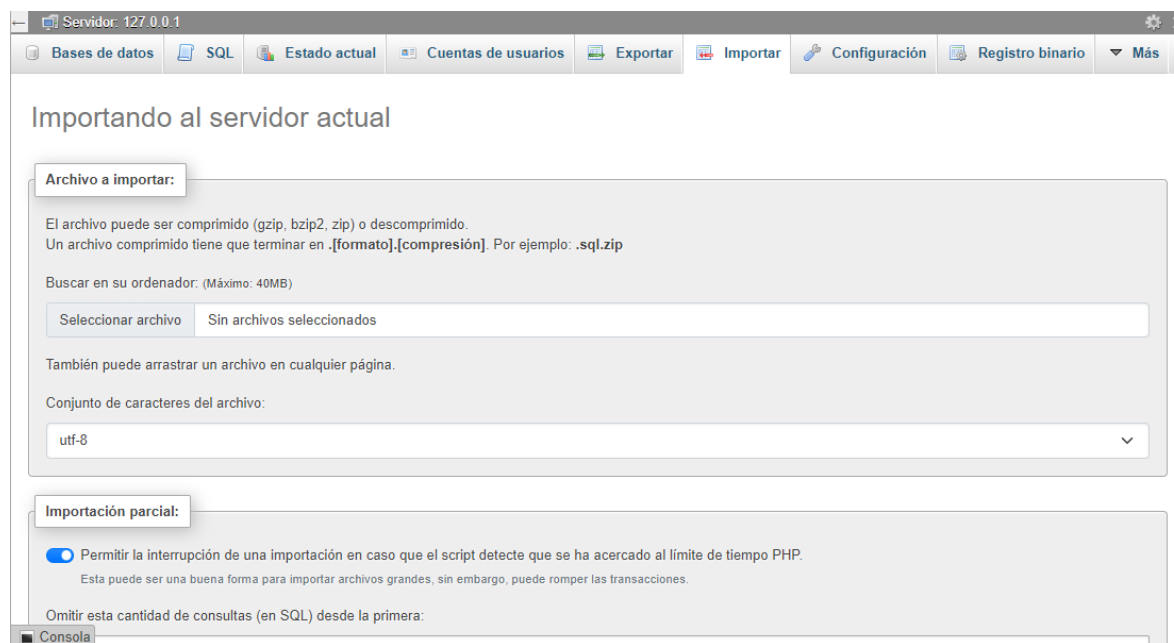


Ilustración 22.- Captura de pantalla del administrador phpMyAdmin, donde se muestran la sección de Importar

Una vez importada la base de datos cargaran 2 tablas creadas, una llamada “google_nauth”, y la otra llamada “usuarios”

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
☐ google_nauth	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8mb4_0900_ai_ci	16.0 KB	-
☐ usuarios	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8mb4_0900_ai_ci	16.0 KB	-
2 tablas	Número de filas	2	InnoDB	utf8mb4_0900_ai_ci	32.0 KB	0 B

Ilustración 23.- Captura de pantalla de la base de datos “db.sql” con sus tablas creadas.

La tabla “google_nauth”, tiene un código interno en uno de sus atributos que crea los códigos QR para cada uno de la IP tomadas de la tabla usuarios, para posteriormente almacenar la imagen de dicho QR en otro campo dentro de la tabla usuarios

La tabla “usuarios” almacena datos de los estudiantes, los cuales el código de la tabla “google_nauth”, toma y codifica en dicho QR, y luego toma la imagen resultante en un formato .jpg

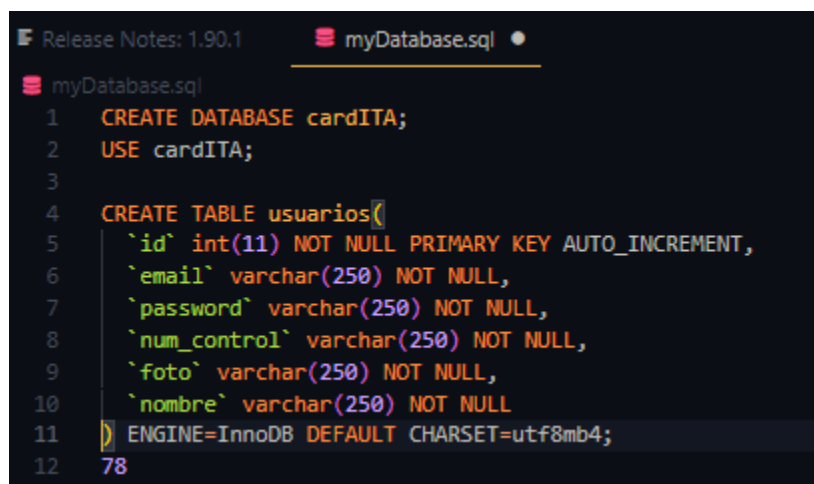
Crearemos la base de datos del login a la aplicación en un documento aparte, pero estando ubicado dentro de la misma carpeta del proyecto y las bases de datos, a esta nueva base de datos la llamaremos “myDatabase.sql” la llenaremos con atributos tales como:

- Id
- Email
- Password
- Numero de control (num_control)
- Fotografía del estudiante (foto)
- Nombre

Le asignaremos valores a cada atributo como:

- ``id` int(11) NOT NULL PRIMARY KEY AUTO_INCREMENT,`
- ``email` varchar(250) NOT NULL,`
- ``password` varchar(250) NOT NULL,`
- ``num_control` varchar(250) NOT NULL,`
- ``foto` varchar(250) NOT NULL,`
- ``nombre` varchar(250) NOT NULL`

Modulo 002

A screenshot of a code editor with a dark background. The editor shows a file named 'myDatabase.sql' with the following SQL code:

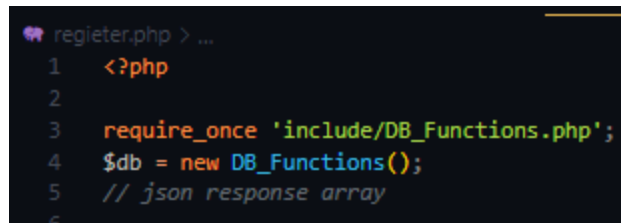
```
1 CREATE DATABASE cardITA;
2 USE cardITA;
3
4 CREATE TABLE usuarios(
5     `id` int(11) NOT NULL PRIMARY KEY AUTO_INCREMENT,
6     `email` varchar(250) NOT NULL,
7     `password` varchar(250) NOT NULL,
8     `num_control` varchar(250) NOT NULL,
9     `foto` varchar(250) NOT NULL,
10    `nombre` varchar(250) NOT NULL
11 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
12 78
```

Ilustración 24.- Captura de pantalla del código escrito para la creación de una nueva base de datos llamada "myDatabase.sql"

Esta base de datos también deberá cargarse en phpMyAdmin, será importante al momento de crear la aplicación.

Lo siguiente será crear una base de datos para el recover de la contraseña en caso de olvidar su contraseña de acceso a la aplicación.

Esta nueva base de datos se llamará “register.php” en esta base haremos la función de que, el usuario reciba su confirmación de su nueva contraseña en su correo electrónico



```
register.php > ...
1  <?php
2
3  require_once 'include/DB_Functions.php';
4  $db = new DB_Functions();
5  // json response array
```

Ilustración 25.- Captura de pantalla del código escrito para la creación de una nueva base de datos llamada "register.php" que servirá como recover de contraseña

La función `require_once` incluye y evalúa el archivo especificado durante la ejecución del script. En este caso, está incluyendo un archivo llamado `DB_Functions.php` desde el directorio `include`. El `require_once` se asegura de que el archivo solo se incluya una vez, incluso si se llama múltiples veces.

Instancia un objeto de la clase `DB_Functions`:

Aquí, se crea una nueva instancia de la clase `DB_Functions` y se asigna a la variable `$db`. Esto permite utilizar los métodos y propiedades definidos en la clase `DB_Functions`.

Modulo 001



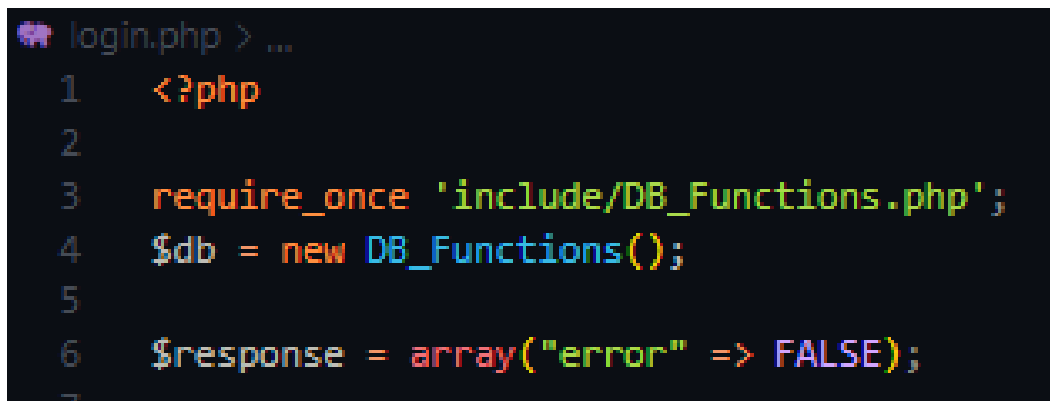
```
$response = array("error" => FALSE);
if (isset($_GET['correo']) && isset($_GET['contrasena']) && isset($_GET['nombre'])) {

    $correo = $_GET['correo'];
    $contrasena = $_GET['contrasena'];
    $nombre = $_GET['nombre'];

    $user = $db->storeUsuario($correo, $contrasena, $nombre);
    if (true) {
        $response["error"] = FALSE;
        $response["user"]["id"] = $user["id"];
        $response["user"]["nombre"] = $user["nombre"];
        $response["user"]["foto"] = $user["foto"];
        $response["user"]["num_control"] = $user["num_control"];
        $response["user"]["email"] = $user["email"];
        $response["user"]["password"] = $user["password"];
        echo json_encode($response);
    } else {
        $response["error"] = TRUE;
        $response["error_msg"] = "Unknown error occurred in registration!";
        echo json_encode($response);
    }
}
?>
```

Ilustración 26.- Captura de pantalla del código escrito en visual studio code para la recuperación de contraseñas en la app CARD-ITA

La creación de esta base de datos será utilizada para el acceso de los usuarios a la aplicación Card-Ita.



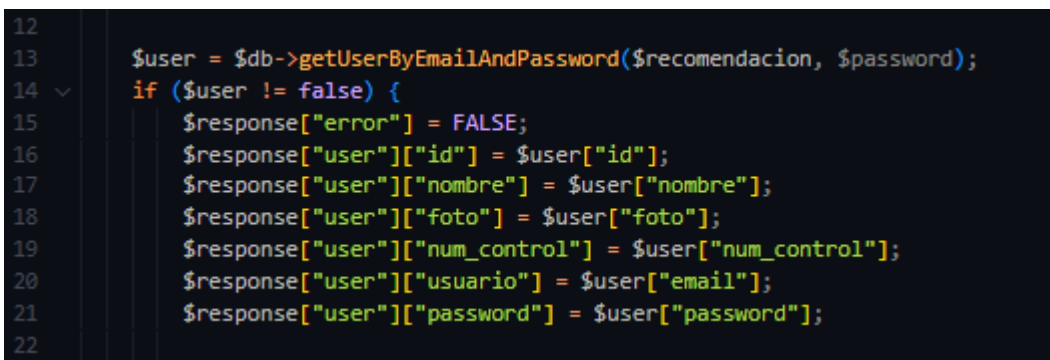
```
login.php > ...
1  <?php
2
3  require_once 'include/DB_Functions.php';
4  $db = new DB_Functions();
5
6  $response = array("error" => FALSE);
7
```

Ilustración 27.- Captura de pantalla del código escrito en visual studio code para el login de la app CARD-ITA

Objetivo del código: Autenticar a un usuario mediante su correo electrónico y contraseña.

Funcionamiento:

- Llama a la función `getUserByEmailAndPassword` para buscar un usuario en la base de datos.
- Si encuentra un usuario que coincide, prepara una respuesta JSON que incluye los datos del usuario y un indicador de que no hubo error.
- Si no encuentra un usuario que coincida, no se ejecuta el bloque de código dentro del `if`.



```
12
13  $user = $db->getUserByEmailAndPassword($recomendacion, $password);
14  if ($user != false) {
15      $response["error"] = FALSE;
16      $response["user"]["id"] = $user["id"];
17      $response["user"]["nombre"] = $user["nombre"];
18      $response["user"]["foto"] = $user["foto"];
19      $response["user"]["num_control"] = $user["num_control"];
20      $response["user"]["usuario"] = $user["email"];
21      $response["user"]["password"] = $user["password"];
22
```

Ilustración 28.- Captura de pantalla del código escrito en visual studio code para la validación de credenciales en la app CARD-ITA

Este código hace la función de autenticación de usuarios, verificando credenciales y devolviendo la información del usuario autenticado si las credenciales son correctas.

```
mail > 🐞 callback.php > ...
1  <?php
2  require_once 'config.php';
3
4  try {
5      $adapter->authenticate();
6      $token = $adapter->getAccessToken();
7      $db = new DB();
8      $db->update_access_token(json_encode($token));
9      echo "Access token inserted successfully.";
10 }
11 catch( Exception $e ){
12     echo $e->getMessage() ;
13 }
14
```

Ilustración 29.- Captura de pantalla del código escrito para autenticar un servicio, obtener un token de acceso y almacenar ese token en una base de datos para su uso posterior en la app CARD-ITA

- Inicia incluyendo configuraciones necesarias desde config.php.
- Intenta autenticar el adaptador (\$adapter->authenticate()).
- Obtiene un token de acceso después de la autenticación (\$adapter->getAccessToken()).
- Crea una instancia de la clase DB.
- Actualiza el token de acceso en la base de datos usando el método update_access_token de la clase DB.
- Imprime un mensaje de éxito si todo el proceso se realiza sin problemas.
- Captura y muestra cualquier excepción que ocurra durante la ejecución.

8.3 Creación de la aplicación Card-Ita

Plataforma: Android

Propósito: Card-Ita es una innovadora aplicación móvil diseñada específicamente para el Instituto Tecnológico de Apizaco (ITA). Su principal objetivo es facilitar y modernizar el sistema de acceso a las instalaciones del ITA mediante el uso de códigos QR. Esta aplicación busca mejorar la seguridad y la eficiencia en el control de acceso, ofreciendo una solución tecnológica avanzada y fácil de usar tanto para los estudiantes como para el personal administrativo.

8.3.1 Definición de requerimientos funcionales

Los requisitos funcionales son un conjunto de especificaciones que definen las funciones o capacidades específicas que debe poseer un sistema o producto para satisfacer las necesidades de sus usuarios.

Requerimiento funcional	
Identificación del requerimiento	RF0
Nombre del requerimiento	Análisis de la información (estructura)
Descripción del requerimiento	El estudiante proporciona información importante como color, propósito, diseño correspondiente a su profesión.
Prioridad del requerimiento	Alta

Tabla 1.- Requerimiento funcional RF0 muestra Análisis de la información

Requerimiento funcional	
Identificación del requerimiento	RF1
Nombre del requerimiento	Página web para administradores
Descripción del requerimiento	El software debe incluir una página web que permita a los administradores administrar el sistema lo que incluye agregar o eliminar cuentas de usuario, actualizar credenciales, generar informes y monitorear registros de acceso. La página web debe ser segura y fácil de usar.
Prioridad del requerimiento	Alta

Tabla 2.- Requerimiento funcional RF1 muestra Página web para administradores

Requerimiento funcional	
Identificación del requerimiento	RF2

Nombre del requerimiento	Herramientas administrativas
Descripción del requerimiento	La página web para administradores debe proporcionar herramientas para administrar las cuentas de los usuarios, generar registros de acceso y establecer políticas de acceso para las diferentes áreas de la universidad.
Prioridad del requerimiento	Alta

Tabla 3.- Requerimiento funcional RF2 muestra Herramientas administrativas

Requerimiento funcional	
Identificación del requerimiento	RF3
Nombre del requerimiento	Administración de usuarios
Descripción del requerimiento	El administrador debe poder administrar las cuentas de los usuarios, lo que incluye agregar, modificar, eliminar cuentas, cambiar su contraseña o ver sus registros de acceso. A través de la página web.
Prioridad del requerimiento	Alta

Tabla 4.- Requerimiento funcional RF3 muestra dministración de usuarios

Requerimiento funcional	
Identificación del requerimiento	RF4
Nombre del requerimiento	Almacenamiento de credenciales digitales
Descripción del requerimiento	El software debe permitir a los administradores almacenar las credenciales digitales de cada estudiante en una base de datos segura. Las credenciales digitales deben incluir información del estudiante, como nombre, foto, programa de estudio, fechas de inscripción y cualquier otra información relevante.
Prioridad del requerimiento	Alta

Tabla 5.- Requerimiento funcional RF4 muestra Almacenamiento de credenciales digitales

Requerimiento funcional	
Identificación del requerimiento	RF5
Nombre del requerimiento	Integración con los sistemas existentes

Descripción del requerimiento	El sistema debe poder integrarse con los sistemas universitarios existentes.
Prioridad del requerimiento	Alta

Tabla 6.- Requerimiento funcional RF5

Requerimiento funcional	
Identificación del requerimiento	RF6
Nombre del requerimiento	Generación de códigos QR
Descripción del requerimiento	El software debe permitir a los administradores generar códigos QR para la credencial digital de cada estudiante.
Prioridad del requerimiento	Alta

Tabla 7.- Requerimiento funcional RF6

Requerimiento funcional	
Identificación del requerimiento	RF7
Nombre del requerimiento	Digitalización de credenciales
Descripción del requerimiento	El software debe poder capturar la información relevante de las credenciales de plástico y convertirla a un formato digital que pueda almacenarse y escanear con un escáner y usarse para el acceso. Luego, el sistema debe almacenar las credenciales digitalizadas en una base de datos segura, luego, el sistema debe almacenar las credenciales digitalizadas en una base de datos segura.
Prioridad del requerimiento	Alta

Tabla 8.- Requerimiento funcional RF7

Requerimiento funcional	
Identificación del requerimiento	RF8
Nombre del requerimiento	Emisión de tarjetas
Descripción del requerimiento	El administrador debe poder emitir nuevas credenciales digitales a los usuarios según sea necesario a través de la página web.
Prioridad del requerimiento	Alta

Tabla 9.- Requerimiento funcional RF8

Requerimiento funcional	
Identificación del requerimiento	RF9
Nombre del requerimiento	Aplicación Móvil
Descripción del requerimiento	El software debe incluir una aplicación móvil que permita a los usuarios acceder a las instalaciones y servicios de la universidad utilizando su dispositivo móvil. La aplicación móvil debe estar disponible para dispositivos Android, y debe ser fácil de usar e intuitiva.
Prioridad del requerimiento	Alta

Tabla 10.- Requerimiento funcional RF9

Requerimiento funcional	
Identificación del requerimiento	RF10
Nombre del requerimiento	Escaneo de credenciales
Descripción del requerimiento	El sistema debe poder escanear y leer la información de las credenciales digitales para verificar y luego validar la identidad del usuario comparándola con una base de datos de credenciales válidas.
Prioridad del requerimiento	Alta

Tabla 11.- Requerimiento funcional RF10

Requerimiento funcional	
Identificación del requerimiento	RF11
Nombre del requerimiento	Control de acceso
Descripción del requerimiento	El sistema debe controlar y habilitar el acceso, permitiendo a los usuarios acceder únicamente a las áreas o servicios que están autorizados a utilizar y mostrar los mensajes de error apropiados si el usuario no está autorizado para acceder a un área en particular.
Prioridad del requerimiento	Alta

Tabla 12.- Requerimiento funcional RF11

Requerimiento funcional	
Identificación del requerimiento	RF12
Nombre del requerimiento	Autenticación de usuarios

Descripción del requerimiento	El sistema debe autenticar la identidad de los usuarios y verificar que está autorizado para acceder a las instalaciones o servicios de la universidad a través de la aplicación móvil o página web.
Prioridad del requerimiento	Alta

Tabla 13.- Requerimiento funcional RF12

Requerimiento funcional	
Identificación del requerimiento	RF13
Nombre del requerimiento	Validación de credenciales
Descripción del requerimiento	El software debe poder validar la autenticidad de la credencial digitalizada realizando una suma de verificación u otro procedimiento de validación para garantizar que los datos no hayan sido alterados.
Prioridad del requerimiento	Alta

Tabla 14.- Requerimiento funcional RF13

Requerimiento funcional	
Identificación del requerimiento	RF14
Nombre del requerimiento	Informes y análisis
Descripción del requerimiento	El software debe proporcionar informes detallados y herramientas de análisis para ayudar a los administradores a comprender el comportamiento de los usuarios y realizar un seguimiento de los patrones de uso en toda la universidad.
Prioridad del requerimiento	Alta

Tabla 15.- Requerimiento funcional RF14

8.3.2 Definición de requerimientos No funcionales

Los requisitos no funcionales son el rendimiento, la usabilidad, la seguridad y otras características que debe poseer el software para satisfacer las necesidades y expectativas de sus usuarios. Estos son algunos requisitos no funcionales para un proyecto de software que implica la digitalización de credenciales de plástico para acceder a una universidad.

Requerimiento No Funcional	
Identificación del requerimiento	RNF1

Nombre del requerimiento	Rendimiento
Descripción del requerimiento	La aplicación móvil y la página web deben cargarse rápidamente y responder a las entradas del usuario sin demoras significativas. El proceso de escaneo de credenciales debe ser rápido y preciso, y el software debe poder manejar una gran cantidad de solicitudes simultáneas sin ralentizarse.
Prioridad del requerimiento	Alta

Tabla 16.- Requerimiento No funcional RNF1

Requerimiento No Funcional	
Identificación del requerimiento	RNF2
Nombre del requerimiento	Seguridad y privacidad del sistema
Descripción del requerimiento	El sistema debe implementar las medidas de seguridad adecuadas para proteger los datos, la privacidad del usuario, incluido el cifrado de datos en tránsito y en reposo, la autenticación y autorización del usuario y los controles de acceso para evitar el acceso no autorizado.
Prioridad del requerimiento	Alta

Tabla 17.- Requerimiento No funcional RNF2

Requerimiento No Funcional	
Identificación del requerimiento	RNF3
Nombre del requerimiento	Escalabilidad
Descripción del requerimiento	El software debe estar diseñado para escalar horizontal y verticalmente, para adaptarse a un número creciente de usuarios y solicitudes. El sistema debe poder agregar nuevos servidores o recursos para manejar el tráfico y la carga adicionales.
Prioridad del requerimiento	Alta

Tabla 18.- Requerimiento No funcional RNF3

Requerimiento No Funcional	
Identificación del requerimiento	RNF4
Nombre del requerimiento	Confiabilidad

Descripción del requerimiento	El software debe ser confiable y estable, con un tiempo de inactividad o errores mínimos. El proceso de escaneo de credenciales debe ser preciso y consistente, y el software debe poder recuperarse rápidamente de las fallas.
Prioridad del requerimiento	Alta

Tabla 19.- Requerimiento No funcional RNF4

Requerimiento No Funcional	
Identificación del requerimiento	RNF5
Nombre del requerimiento	Usabilidad
Descripción del requerimiento	La aplicación móvil y la página web deben diseñarse para que sean fáciles de usar y navegar, con una interfaz limpia e intuitiva que sea accesible para usuarios con diferentes niveles de experiencia y conocimientos. El proceso de escaneo de credenciales debe ser sencillo y requerir una intervención mínima del usuario.
Prioridad del requerimiento	Alta

Tabla 20.- Requerimiento No funcional RNF5

Requerimiento No Funcional	
Identificación del requerimiento	RNF6
Nombre del requerimiento	Accesibilidad
Descripción del requerimiento	El software debe ser accesible para usuarios con discapacidades y cumplir con estándares de accesibilidad como WCAG 2.1. El proceso de escaneo de credenciales debe diseñarse para adaptarse a usuarios con diferentes capacidades y necesidades.
Prioridad del requerimiento	Alta

Tabla 21.- Requerimiento No funcional RNF6

Requerimiento No Funcional	
Identificación del requerimiento	RNF7
Nombre del requerimiento	Seguridad
Descripción del requerimiento	El software debe diseñarse teniendo en cuenta la seguridad e implementar las medidas adecuadas para proteger los datos y la privacidad del usuario, como el cifrado, la autenticación y los controles de acceso. El software

	también debe cumplir con los estándares de seguridad relevantes, como ISO 27001.
Prioridad del requerimiento	Alta

Tabla 22.- Requerimiento No funcional RNF7

Requerimiento No Funcional	
Identificación del requerimiento	RNF8
Nombre del requerimiento	Cumplimiento
Descripción del requerimiento	El software debe cumplir con las leyes y reglamentaciones pertinentes, como las leyes de protección de datos, y poder demostrar el cumplimiento mediante auditorías y otras medidas.
Prioridad del requerimiento	Alta

Tabla 23.- Requerimiento No funcional RNF8

Requerimiento No Funcional	
Identificación del requerimiento	RNF9
Nombre del requerimiento	Mantenibilidad
Descripción del requerimiento	El software debe diseñarse para que sea fácil de mantener y actualizar, con una documentación clara y un código bien organizado que sea fácil de entender y modificar. La base de datos de credenciales debe ser fácil de actualizar y administrar.
Prioridad del requerimiento	Alta

Tabla 24.- Requerimiento No funcional RNF9

Requerimiento No Funcional	
Identificación del requerimiento	RNF10
Nombre del requerimiento	Interoperabilidad
Descripción del requerimiento	El software debe poder funcionar sin problemas con otros sistemas y plataformas, y estar diseñado con estándares abiertos y API que permitan la integración con otras herramientas de software.
Prioridad del requerimiento	Alta

Tabla 25.- Requerimiento No funcional RNF10

Requerimiento No Funcional	
Identificación del requerimiento	RNF11
Nombre del requerimiento	Compatibilidad
Descripción del requerimiento	La aplicación móvil debe ser compatible con una amplia gama de dispositivos móviles, diversas versiones del sistema operativo Android, y la página web debe

	ser compatible con una variedad de navegadores web.
Prioridad del requerimiento	Alta

Tabla 26.- Requerimiento No funcional RNF11

Requerimiento No Funcional	
Identificación del requerimiento	RNF12
Nombre del requerimiento	Facilidad de uso
Descripción del requerimiento	El software debe diseñarse con una interfaz fácil de usar para garantizar la facilidad de uso y minimizar los errores del usuario. La aplicación móvil y la página web deben ser intuitivas y fáciles de navegar, con instrucciones y avisos claros.
Prioridad del requerimiento	Alta

Tabla 27.- Requerimiento No funcional RNF12

8.3.3 Maquetación y Prototipo:

8.3.3.1 Creación del logo Card-Ita

Introducción

La identidad visual de una aplicación es un elemento clave para su reconocimiento y diferenciación en el mercado. detallaré el proceso de diseño del logo de "Card-Ita", la aplicación móvil que funciona como credencial digital para los alumnos del Tecnológico Nacional de México/Instituto Tecnológico de Apizaco.

Concepto y Diseño

El logo de Card-Ita fue diseñado con el objetivo de representar la modernidad, la seguridad y la accesibilidad de la aplicación. Se optó por un diseño minimalista con una tipografía que represente modernidad y confianza, que refuerza la idea de innovación tecnológica.

Elementos del Logo

Color: Se eligió el color blanco para transmitir limpieza, sencillez y profesionalismo. Este color también facilita la integración del logo en distintos fondos y materiales, respetando así la paleta de colores establecida para el TecNM y todos sus planteles, siendo el Azul, Blanco y Gris, los colores a representar.

Tipografía: Se utilizó una tipografía futurista para reflejar el carácter tecnológico de la aplicación.

Simplicidad: El diseño evita elementos excesivos para garantizar claridad y escalabilidad en diferentes formatos y dispositivos. Un diseño responsivo de la aplicación nos lleva a ajustar los elementos mas básicos de la misma, como es el caso del logotipo, en donde la futura implementación de un Isotipo, jugará un papel clave a futuras actualizaciones.

Proceso de Creación

Investigación: Se analizaron tendencias de diseño en aplicaciones tecnológicas y credenciales digitales.

Bocetado: Se realizaron varios bocetos explorando distintas opciones de tipografía y disposición.



Ilustración 30.- Captura de pantalla de los primeros bosquejos del logo para la aplicación CARD-ITA, mostrando la opción número 1.



Ilustración 31.- Captura de pantalla de los primeros bosquejos del logo para la aplicación CARD-ITA, mostrando la opción número 2.



Ilustración 32.- Captura de pantalla de los primeros bosquejos del logo para la aplicación CARD-ITA, mostrando la opción número 3.

Selección de Tipografía: Se eligió una fuente moderna con un estilo geométrico y limpio.

Pruebas de Visibilidad: Se verificó la legibilidad del logo en distintos tamaños y fondos.



Ilustración 33.- Captura de pantalla del logotipo preliminar, haciendo ajustes de compatibilidad con la paleta de colores establecida por el TecNM.

Ajustes Finales: Se realizaron optimizaciones para mejorar la armonía visual y la versatilidad del logo.

Aplicaciones del Logo.

El logo de Card-Ita se implementará en:

- La interfaz de la aplicación móvil.
- Materiales promocionales y de difusión de la credencial digital en las instalaciones.
- Documentación oficial y credenciales digitales.

8.3.3.2 Usos correctos e incorrectos del Logotipo.

El diseño del logo de Card-Ita cumple con los principios de claridad, modernidad y representatividad. Su estilo futurista y minimalista refuerza la identidad de la aplicación como una herramienta tecnológica innovadora para la comunidad estudiantil, representando de la manera más leal la identidad del Tecnológico Nacional de México/Instituto Tecnológico de Apizaco.

Para mantener la coherencia visual y garantizar el reconocimiento de la marca, es fundamental seguir las siguientes recomendaciones sobre el uso del logotipo de Card-Ita.

Usos Correctos

Escalabilidad adecuada: Se debe mantener la proporción original del logo sin distorsionarlo al redimensionarlo. Siendo su escala de proporción **[9:16]**

Fondos apropiados: Usar el logotipo en fondos que garanticen su visibilidad, preferentemente en contrastes adecuados, para la aplicación; su incorporación será únicamente en fondos Azul marino (**233d7b**) y Gris (484848).

Espaciado mínimo: Mantener un área de seguridad alrededor del logo para evitar que otros elementos interfieran con su lectura.

Uso de colores oficiales: Mantener el color blanco del logo, respetando su identidad visual.

Formatos adecuados: Utilizar el logotipo en formatos de alta calidad (PNG, SVG) para evitar pérdida de resolución.

Usos Incorrectos

Distorsionar el logo: No estirar, comprimir ni alterar la proporción del diseño.

Cambiar los colores: No modificar el color blanco original del logo ni aplicar degradados o efectos no autorizados.

Agregar sombras o contornos excesivos: No aplicar efectos adicionales que alteren la estética del diseño.

Usarlo sobre fondos inapropiados: No colocar el logo en fondos que dificulten su lectura, como colores muy claros o estampados complejos.

Rotar o inclinar el logo: No girarlo ni modificar su orientación original.

Modificar la tipografía: No alterar la fuente del logo, ya que afecta su identidad visual.

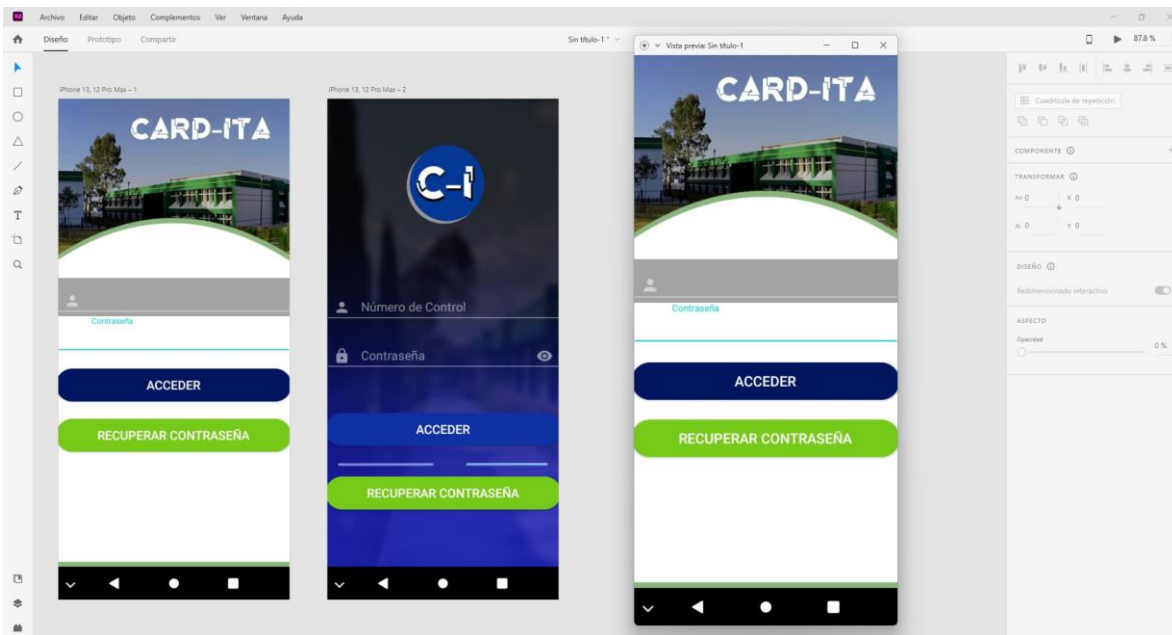


Ilustración 34- Muestra de pantalla del maquetado y prototipado de la aplicación Card-Ita en el programa AdobeXD

El maquetado tiene como prioridad, organizar el texto y los elementos gráficos de diferentes formas, de tal manera que estas ayuden a materializar ideas y concentrarlas en un mismo bosquejo antes de pasar a la parte de la programación en el backend y frontend.

El prototipo se hizo en base a los requerimientos no funcionales, paleta de colores y tipografía empleada en el manual de uso del logotipo “Card-Ita”

Se utilizo AdobeXD(Extreme Desing) por su amplia capacidad de crear maquetados a raíz de figuras geométricas que pueden ser vectorizadas, transformadas y deformadas, dando así una libertad creativa mas optima para el maquetado, además de poder transformar un maquetado a prototipo en muy pocos pasos.

Funcionalidades Principales:

1. **Generación de Códigos QR:**
 - Cada usuario registrado en la aplicación recibe un código QR único, el cual sirve como su credencial digital de acceso.
2. **Escaneo de Códigos QR:**
 - A través de la cámara del dispositivo móvil, la aplicación permite escanear los códigos QR para verificar la identidad del usuario y conceder acceso a las instalaciones del ITA.
3. **Registro y Autenticación de Usuarios:**
 - Los usuarios pueden registrarse en la aplicación utilizando su correo electrónico institucional y una contraseña segura.
 - La aplicación incluye un sistema de autenticación para garantizar que solo usuarios autorizados puedan acceder a sus funcionalidades.
4. **Gestión de Usuarios:**
 - Los administradores pueden gestionar los perfiles de los usuarios, incluyendo la asignación y revocación de permisos de acceso.
 - La aplicación permite la actualización de información personal y de contacto del usuario.

8.4 Primeros pasos:

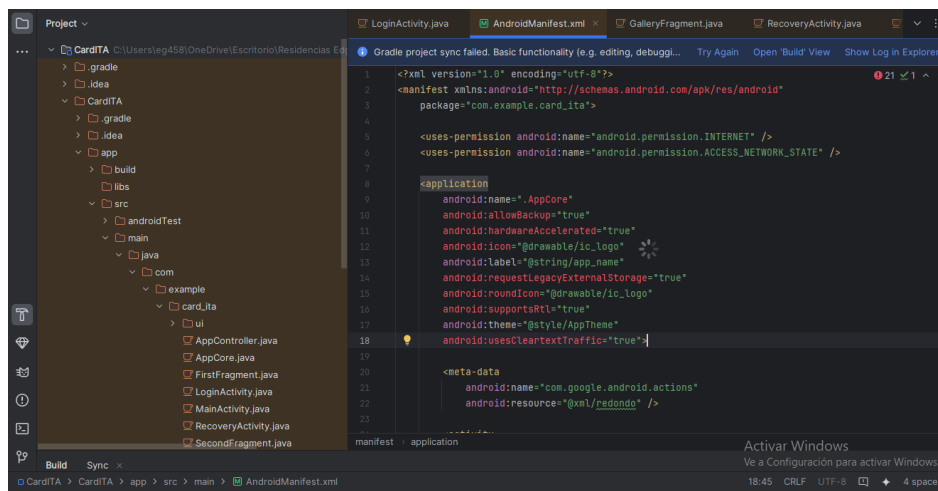


Ilustración 35.- Captura de Android Studio desarrollando la estructura de la aplicación Card-Ita

Especifica la clase Application personalizada que se utilizará para mantener el estado global de la aplicación.

Modulo A001

```

37     <activity
38         android:name=".LoginActivity"
39         android:label="Card-ITA"
40         android:screenOrientation="portrait"
41         android:theme="@style/AppTheme.NoActionBar" />
42     <activity
43         android:name=".MainActivity"
44         android:label="Card-ITA"
45         android:screenOrientation="portrait"
46         android:theme="@style/AppTheme.NoActionBar" />
47     <activity
48         android:name=".splash"
49         android:exported="true"
50         android:label="Card-ITA"
51         android:screenOrientation="portrait"
52         android:theme="@style/AppTheme.NoActionBar">
53         <intent-filter>
54             <action android:name="android.intent.action.MAIN" />
55

```

Ilustración 36.- Captura de Android Studio con esta configuración, *MainActivity* se mostrará en modo vertical sin una barra de acción y tendrá el título "Card-ITA" en la interfaz de usuario.

Propiedades:

1. **android:name=".MainActivity":**

- Especifica el nombre completo de la clase de la actividad. En este caso, MainActivity es la actividad principal de la aplicación. El prefijo . indica que la clase está en el mismo paquete que la aplicación.

2. **android:label="Card-ITA":**

- Establece el nombre que se mostrará en la barra de título (si está presente) cuando esta actividad esté en primer plano. En este caso, se mostrará "Card-ITA".

3. **android:screenOrientation="portrait":**

- Fuerza a la actividad a mostrarse solo en modo vertical (portrait). Esto es útil si tu aplicación está diseñada para ser utilizada solo en orientación vertical.

4. **android:theme="@style/AppTheme.NoActionBar":**

- Especifica el tema que se aplicará a esta actividad en particular. En este caso, se está utilizando un estilo llamado AppTheme.NoActionBar, que típicamente se define en res/values/styles.xml. Este tema probablemente elimina la barra de acción (action bar) de la actividad.

- **android:name=".splash"**: Define el nombre completo de la clase de la actividad.
- **android:exported="true"**: Indica que esta actividad puede ser llamada desde otras aplicaciones.
- **android:label="Card-ITA"**: Especifica el título de la actividad.
- **android:screenOrientation="portrait"**: Fuerza la actividad a mostrarse solo en orientación vertical.
- **android:theme="@style/AppTheme.NoActionBar"**: Aplica el tema AppTheme.NoActionBar a esta actividad.
- **<intent-filter>**: Define un filtro de intentos, indicando que esta actividad responde a la acción principal (MAIN), que normalmente es la primera actividad que se muestra cuando la aplicación se inicia.

8.5 Configuración de la IP para uso de Red Local

Establecemos la ip en la cual funcionara la aplicación ya que será una app local para la institución

modulo A002

```

39
40     ImageView imageView = (ImageView) root.findViewById(R.id.foto);
41     Glide.with(this).load("http://192.168.1.89/card-ITA/fotos/"+vfoto).into(imageView);
42     texto.setText(vnombre);
43     return root;
44 }
```

Ilustración 37.- Captura de Android Studio GalleryFragment.java con la IP local

Este código utiliza Glide para cargar una imagen en un ImageView y establece un texto en un TextView.

Obtener una referencia a un ImageView:

- Se obtiene la referencia a un ImageView con el ID foto desde la vista root.

Cargar una imagen usando Glide:

- Se usa Glide para cargar una imagen desde una URL, que incluye el valor de vfoto, y se coloca en el ImageView obtenido anteriormente.

```

52     private void RecoveryUser(final String correo) {
53         String tag_string_req = "req_register";
54         progressDialog.setMessage("Enviando ...");
55         progressDialog.show();
56         StringRequest strReq = new StringRequest(Request.Method.POST,
57             "http://192.168.1.89/Card-ITA/Email/index.php", new Response.Listener<String>() {

```

Ilustración 38.- Captura de Android Studio RecoveryActivity.java con la IP local

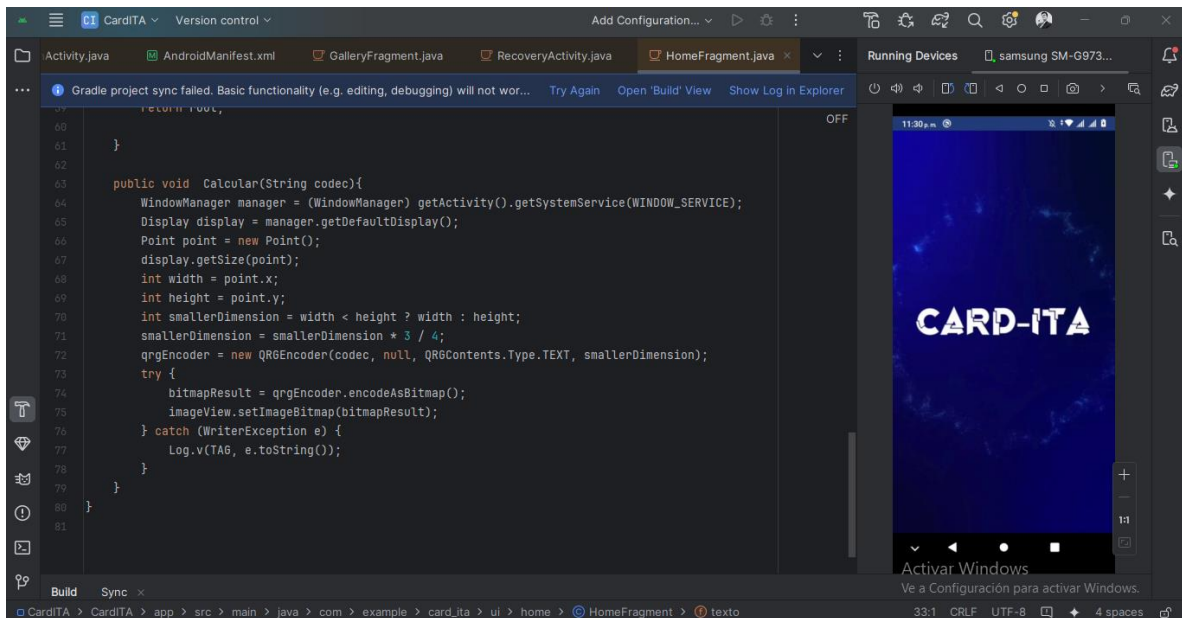


Ilustración 39.- Captura de Android Studio RecoveryActivity.java con la IP local

Entrada:

- **Argumento:** El método Calcular recibe una cadena de texto codec. Supongamos que el valor de codec es "TextoEjemplo".

Paso 1: Obtener el WindowManager

- El código obtiene una instancia de WindowManager a través del método getSystemService(WINDOW_SERVICE) de la actividad actual (getActivity()).
- Esto es necesario para acceder a la información de la pantalla del dispositivo.

Paso 2: Obtener el Display

- Usando manager.getDefaultDisplay(), el código obtiene la pantalla principal del dispositivo (Display display).

Paso 3: Obtener el Tamaño de la Pantalla

- Se crea un objeto Point llamado point.

- Luego, `display.getSize(point)` llena `point` con el ancho (`point.x`) y alto (`point.y`) de la pantalla.
- **Suposición:** Si la pantalla tiene un ancho de 1080 píxeles y una altura de 1920 píxeles:
 - `width = 1080`
 - `height = 1920`

Resultado Esperado:

- **Pantalla del Dispositivo:** Se espera que el `ImageView` en la interfaz de usuario muestre un código inicio de sesión que representa el texto "TextoEjemplo".
- **Dimensión del Inicio de sesión:** El tamaño del código `contain` se ajustará a un máximo de 810 píxeles (en el lado más pequeño).
- **Posible Error:** Si ocurre algún error durante el inicio de sesión, se registrará en los logs con el TAG definido previamente.

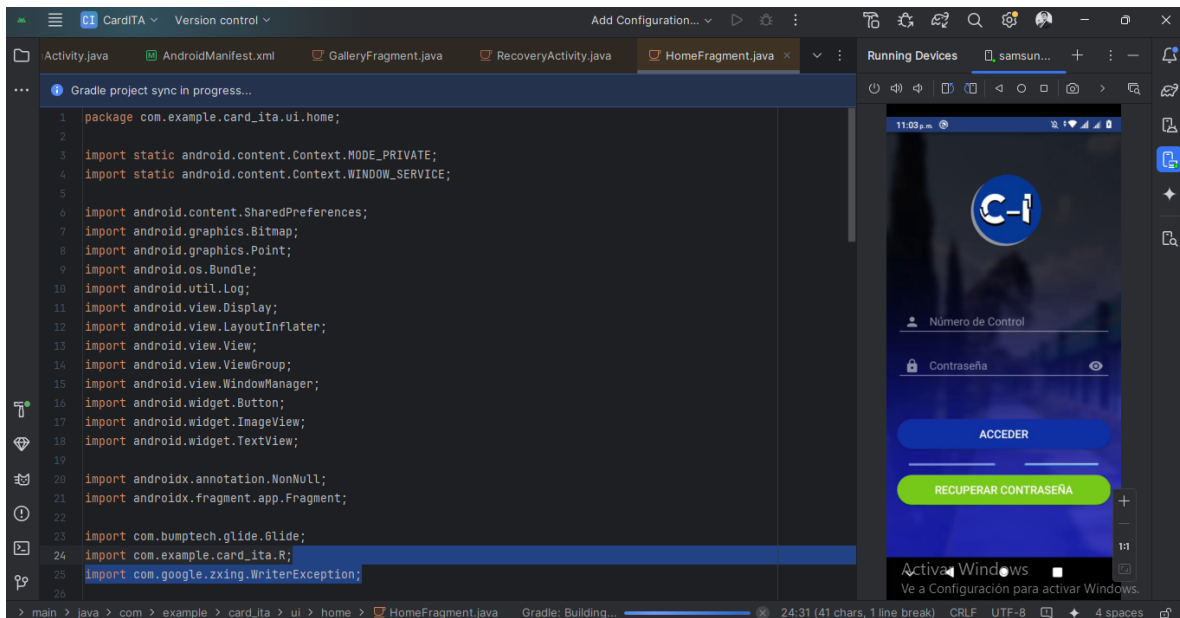


Ilustración 40.- Importaciones de Clases

SharedPreferences:

- Permite almacenar datos en pares clave-valor de manera persistente en el dispositivo, como configuraciones o información del usuario.

Bitmap:

- Representa una imagen en memoria, comúnmente utilizada para manipular gráficos o mostrar imágenes en la interfaz de usuario.

Point:

- Representa una coordenada (x, y). Se usa para obtener el tamaño de la pantalla o una vista específica.

Bundle:

- Contenedor de datos que se utiliza para pasar información entre actividades, fragmentos o métodos dentro de la aplicación.

Log:

- Se utiliza para generar mensajes de registro (logs) que ayudan a depurar y entender el comportamiento del código.

Display:

- Representa la pantalla física del dispositivo. Se usa para obtener información sobre la pantalla, como su tamaño o resolución.

LayoutInflater:

- Se utiliza para inflar (crear) vistas desde archivos XML de diseño (layouts) en un fragmento o actividad.

View:

- Clase base para todos los componentes de la interfaz de usuario en Android. Puede representar botones, cuadros de texto, imágenes, etc.

ViewGroup:

- Un contenedor que agrupa otras vistas (hijos) y las organiza en la pantalla. Ejemplo: LinearLayout, RelativeLayout.

WindowManager:

- Se utiliza para gestionar la ventana en la que se ejecuta una actividad. Puedes cambiar aspectos como el tamaño y la orientación de la ventana.

Button:

- Un componente de la interfaz de usuario que se puede presionar para ejecutar una acción.

ImageView:

- Se utiliza para mostrar imágenes en la pantalla.

TextView:

- Un componente de la interfaz de usuario para mostrar texto.

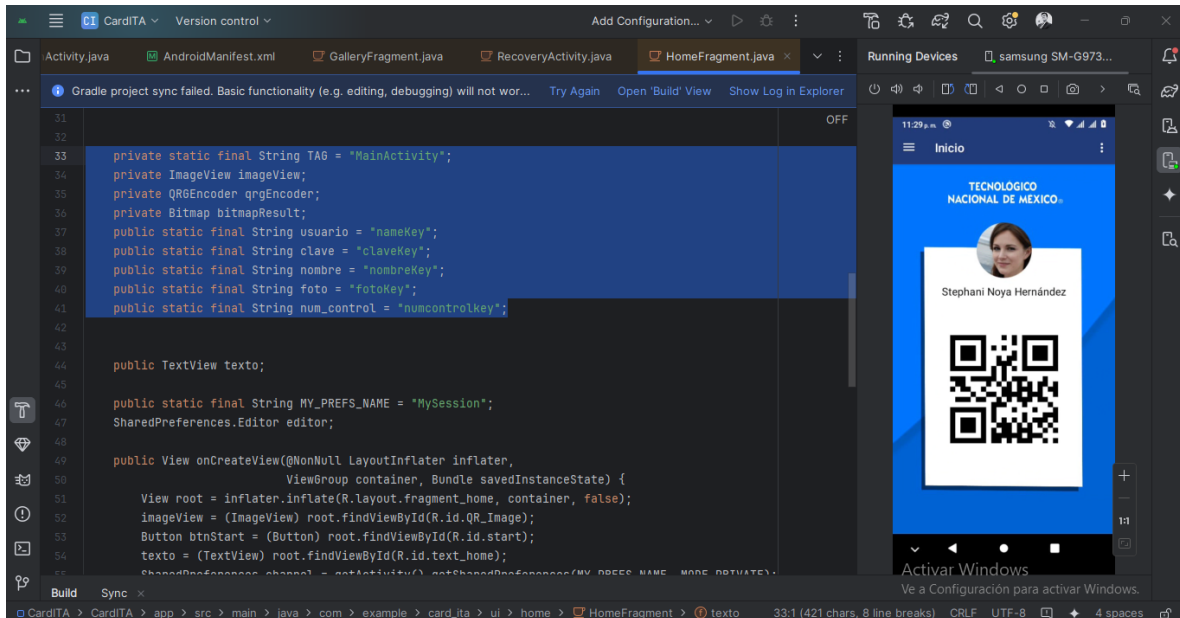


Ilustración 41: Almacenamiento y recuperación de datos del usuario

Este código está preparado para generar un código QR usando QRLEncoder, luego almacenarlo como un Bitmap en bitmapResult y mostrarlo en un ImageView.

Las constantes usuario, clave, nombre, foto, y num_control toman estos datos que se manejan dentro de la actividad, para personalizar el contenido y generar el código QR con información específica del usuario.

```
public static final String usuario = "nameKey";
public static final String clave = "claveKey";
public static final String nombre = "nombreKey";
public static final String foto = "fotoKey";
public static final String num_control = "numcontrolkey";
```

- Estas son claves (Keys) que se usan para almacenar o recuperar valores de un SharedPreferences o algún otro almacenamiento persistente.
- Usuario, clave, nombre, foto, y num_control están vinculados con los datos de un usuario, como su nombre, contraseña, nombre completo, foto de perfil, y número de control.

```
private QRLEncoder qrgEncoder;
```

- QRLEncoder una instancia de una clase usada para generar códigos QR basados en ciertos datos.

8.7 Creación del RecoveryActivity

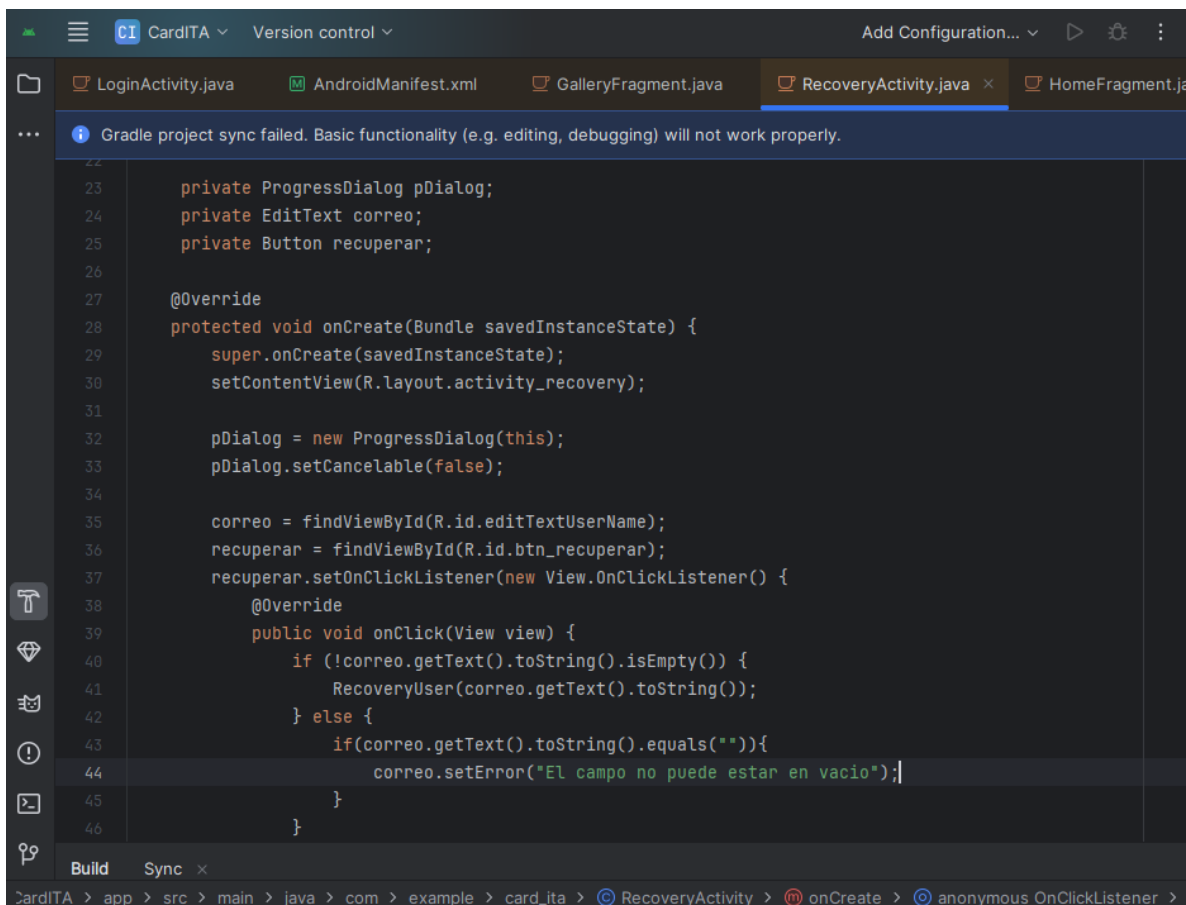


Ilustración 42: código petición y recuperación de datos del usuario

1. Creación de la Actividad

- **onCreate(Bundle savedInstanceState):**
 - Cuando se crea la actividad RecoveryActivity, se llama al método onCreate.
 - Se configura el diseño de la actividad utilizando `setContentView(R.layout.activity_recovery)`, que infla el archivo de diseño XML `activity_recovery.xml`.

2. Inicialización de Componentes

- **ProgressDialog pDialog:**
 - Se crea e inicializa un ProgressDialog para mostrar una ventana de progreso, que por defecto no es cancelable (`pDialog.setCancelable(false)`).

- **EditText correo:**
 - Se asigna a la variable correo un campo de texto (EditText) que se encuentra en el diseño y tiene el ID editTextUserName.
- **Button recuperar:**
 - Se asigna a la variable recuperar un botón (Button) con el ID btn_recuperar del diseño.

3. Configuración del Listener del Botón

- Se establece un OnClickListener para el botón recuperar. Cuando el usuario hace clic en el botón:
 - **Verificación del Campo correo:**
 - **Condición 1:** Si el campo de texto correo no está vacío (`!correo.getText().toString().isEmpty()`):
 - Se llama al método `RecoveryUser(correo.getText().toString())`, pasando el correo electrónico como parámetro.
 - **Condición 2:** Si el campo de texto correo está vacío (`correo.getText().toString().equals("")`):
 - Se muestra un error en el campo de texto (`correo.setError("El campo no puede estar en vacío");`).

4. Lógica de Recuperación de Usuario

- **RecoveryUser(String email):**
 - Este método es llamado cuando el usuario introduce un correo electrónico válido y hace clic en el botón de recuperación. La lógica específica para recuperar el usuario utilizando el correo electrónico debería implementarse aquí.

5. Uso

Escenario 1: Usuario Ingresa un Correo Electrónico

- El usuario escribe "usuario@ejemplo.com" en el campo correo.
- El usuario hace clic en el botón recuperar.
- Como el campo correo no está vacío, el método `RecoveryUser("usuario@ejemplo.com")` es llamado, iniciando la lógica de recuperación de usuario.

Escenario 2: Usuario No Ingresa un Correo Electrónico

- El usuario no escribe nada en el campo correo.

- El usuario hace clic en el botón recuperar.
- Como el campo correo está vacío, se muestra un mensaje de error en el campo que dice "El campo no puede estar en vacío".

Resultados Esperados:

- Si el usuario ingresa un correo válido, se llamará a la función RecoveryUser y se iniciará el proceso de recuperación.
- Si el usuario no ingresa un correo, se mostrará un error en el campo de texto indicándole que el campo no puede estar vacío.

9.0 Resultados

Resultados del Proyecto

La creación del sistema de credencial digital basada en códigos QR para el acceso a la universidad se llevó a cabo exitosamente, cumpliendo con los objetivos establecidos al inicio del proyecto. Los resultados obtenidos fueron los esperados y se pueden resumir en los siguientes puntos:

Digitalización del Acceso

Se logró crear un nuevo sistema de credenciales digitales, cumpliendo la función de las credenciales físicas de plástico con tecnología RFID por un sistema basado en códigos QR generados desde una aplicación móvil.

La digitalización permitirá una mayor accesibilidad y comodidad para los usuarios, eliminando la necesidad de portar una credencial física. Sin mencionar los increíbles beneficios ecológicos y económicos que esta traerá.

Funcionalidad y Rendimiento del Sistema

La aplicación móvil Card-Ita fue desarrollada con éxito en Android Studio, permitiendo generar y mostrar un código QR desde dispositivos móviles.

El escaneo del código QR en los puntos de acceso simulados funcionó de manera rápida y eficiente, sincronizando los sistemas distribuidos.

Seguridad y Autenticación

La implementación de medidas de seguridad adecuadas para evitar fraudes o el uso indebido de códigos QR, asegurando que cada código fuera único y generado de forma segura.

La integración con la base de datos de la universidad permitió verificar en tiempo real la validez de cada código QR antes de conceder el acceso.

Satisfacción del Usuario

Las pruebas realizadas con estudiantes y personal administrativo reflejaron una alta aceptación del nuevo sistema debido a su facilidad de uso y eficiencia.

Los usuarios destacaron la comodidad de no depender de una credencial física y la rapidez en el proceso de ingreso.

Optimización y Reducción de Costos

La eliminación de credenciales físicas representaría una reducción en los costos de producción y reposición de credenciales perdidas o dañadas. El mantenimiento del sistema basado en códigos QR resultó ser más económico y sostenible a largo plazo, ahorrando así hasta \$50,000 anuales en credenciales físicas de plástico.

Integración con la Infraestructura Existente

El sistema podría ser implementado sin afectar la operatividad de otros procesos administrativos de la universidad.

Resultados de la aplicación

La aplicación se ejecuto en diversos modelos de celulares y marcas, algunos con distintas versiones de los sistemas operativos Android y IOs. Para calificar su comportamiento y que los usuarios tengan un correcto uso de la aplicación.

En primera instancia ejecutaremos la aplicación con las especificaciones de desarrollo creadas como lo son la pantalla de carga, la pantalla de inicio de sesión con datos como “No. Control” y “Contraseña”, seguido de la visualización del código QR que dará acceso al estudiante a la institución.



Ilustración 43: Captura animación del logo Card-Ita en la primera pantalla, esperando la conexión a la segunda pantalla.

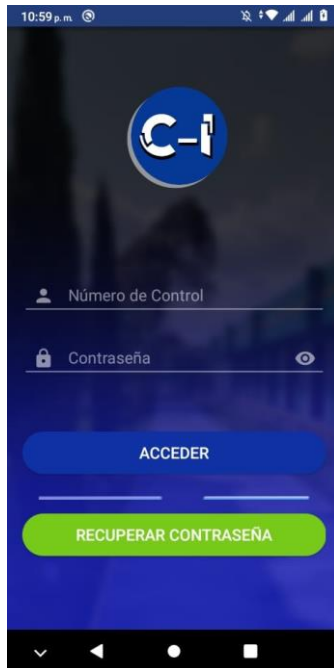


Ilustración 44: Captura de la segunda pantalla donde el usuario deberá iniciar sesión con su numero de control y contraseña

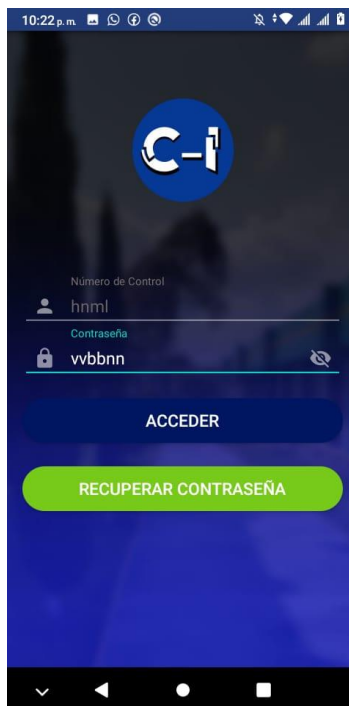


Ilustración 45: Captura de la segunda pantalla donde el usuario ingresa sus datos para iniciar sesión.

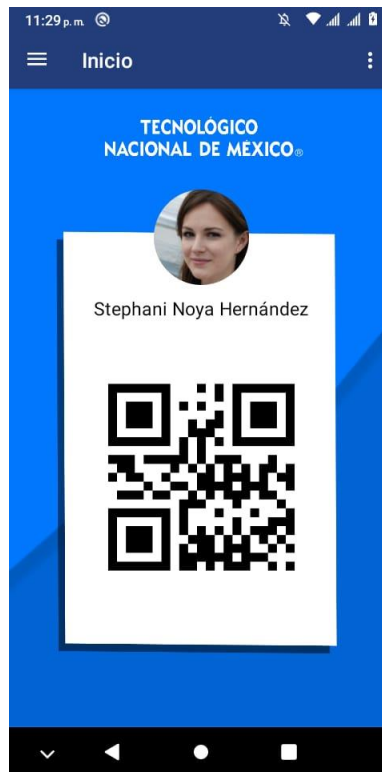


Ilustración 46: Captura de la tercera pantalla donde se visualiza el código QR con el que el usuario ingresará a la institución.

Posibilidad de implementaciones futuras

La aplicación como herramienta para los estudiantes, tiene un gran potencial de crecimiento en desarrollo que facilitará aun mas el uso de los servicios que la institución tiene para con sus estudiantes. Como lo son:

Sistema SIA: Es una pagina web donde los alumnos pueden visualizar su información académica, generar fichas de pago, y consultar su carga academica.

Solución de integraciones futuras: Una posible implementación futura sería integrar este sistema dentro de la aplicación Card-Ita para que los alumnos tengan su información de manera más rápida, sin necesidad de ingresar a un navegador web o recordar un usuario y contraseña diferente.

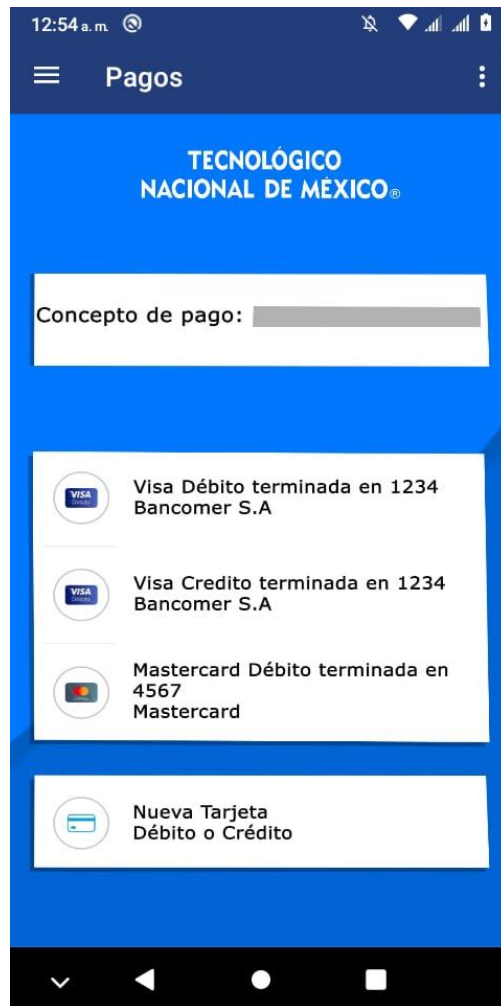


Ilustración 49: Captura de pantalla del apartado “Pagos” donde se visualiza la generación de un concepto de pago generado por el usuario, y dando opciones de pago.



Ilustración 50: Captura de pantalla del apartado “Directorios” donde se visualiza el apartado con directorios de contacto con departamentos de “Servicios Escolares” y la división de “Estudios profesionales”.

10.0 Pruebas

10.1 Pruebas de escritorio por modulo.

Se realizará una prueba de escritorio para comprobar la lógica al final de cada módulo programado.

Numero de modulo 001: Base de datos “register.php”

Modulo creado para el recover de contraseña de los usuarios en la aplicación Card-Ita en phpMyAdmin

Paso 1: Identificar las variables y los datos de entrada

- correo: eg458434@gmail.com
- contrasena: a2d
- nombre: Edgar Garcia de la Cadena Zebadúa

Paso 2: Dibujar una tabla para la prueba de escritorio

Línea	Acción	Variable	Valor
4	Incluye el archivo DB_Functions.php		
5	Crea una instancia de DB_Functions	\$db	Instancia de DB_Functions
8	Inicializa la respuesta JSON	\$response	array("error" => FALSE)
9	Verifica si existen los parámetros correo, contraseña, nombre		TRUE
11	Asigna el valor del parámetro correo	\$correo	eg458434@gmail.com
12	Asigna el valor del parámetro contraseña	\$contrasena	a2d
13	Asigna el valor del parámetro nombre	\$nombre	Edgar Garcia de la Cadena Zebadúa
15	Llama a la función storeUsuario con los valores	\$user	Array con datos del usuario registrado
16	Siempre es true (condición fija)		TRUE
17	Establece \$response["error"] a FALSE	\$response["error"]	FALSE
18	Asigna id del usuario al array response	\$response["user"]["id"]	id del usuario (ej. 1)
19	Asigna nombre del usuario al array response	\$response["user"]["nombre"]	Edgar Garcia de la Cadena Zebadúa
20	Asigna foto del usuario al array response	\$response["user"]["foto"]	URL de la foto
21	Asigna num_control del usuario al array response	\$response["user"]["num_control"]	123456
22	Asigna email del usuario al array response	\$response["user"]["email"]	eg458434@gmail.com
23	Asigna password del usuario al array response	\$response["user"]["password"]	a2d
24	Muestra la respuesta JSON		{ "error": false, "user": { "id": 1, "nombre": "Edgar Garcia de la Cadena Zebadúa", "foto": "URL de la foto", "num_control": "123456", "email": "eg458434@gmail.com", "password": "a2d" } }

25-28	No se ejecuta porque la condición siempre es true		
-------	---	--	--

Resultado final:

```
{
  "error": false,
  "user": {
    "id": 1,
    "nombre": "Edgar Garcia de la Cadena Zebadúa",
    "foto": "URL de la foto",
    "num_control": "123456",
    "email": "eg458434@gmail.com",
    "password": "a2d"
  }
}
```

Ilustración 51.- Captura de pantalla del código resultado de la prueba de escritorio 001

1. **Verificación de parámetros:** El código verifica si los parámetros correo, contraseña y nombre están presentes en la solicitud GET.
2. **Asignación de variables:** Asigna los valores de estos parámetros a las variables \$correo, \$contrasena y \$nombre.
3. **Almacena el usuario:** Llama a la función storeUsuario del objeto \$db, que guarda el usuario en la base de datos y devuelve un array con los datos del usuario.
4. **Construcción de la respuesta JSON:** Crea una respuesta JSON con los datos del usuario y un indicador de error (false).
5. **Salida de la respuesta JSON:** Muestra la respuesta JSON al cliente.

Numero de modulo 002: “myDataBase.sql”

Modulo para la creación de una base de datos de registro de usuarios en la aplicación Card-Ita.

```
CREATE DATABASE cardITA;
USE cardITA;

CREATE TABLE usuarios (
  `id` int(11) NOT NULL PRIMARY KEY AUTO_INCREMENT,
  `email` varchar(250) NOT NULL,
  `password` varchar(250) NOT NULL,
  `num_control` varchar(250) NOT NULL,
  `foto` varchar(250) NOT NULL,
  `nombre` varchar(250) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Ilustración 52.- Captura de pantalla del código a analizar en el módulo 002

1.- Creación de la base de datos

```
CREATE DATABASE cardITA;
```

Ilustración 53.- Captura de pantalla del código a para crear la base de datos en el módulo 002

- Acción: Crea una base de datos llamada cardITA.
- Resultado: Se crea una nueva base de datos en el servidor.

2.- Seleccionar la base de datos:

```
USE cardITA;
```

Ilustración 54.- Captura de pantalla del código a para usar la base de datos en el módulo 002

- Acción: Cambia el contexto a la base de datos cardITA.
- Resultado: Cualquier operación posterior se realizará en la base de datos cardITA.

3.- Crear la tabla usuarios:

```
CREATE TABLE usuarios (  
  `id` int(11) NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  `email` varchar(250) NOT NULL,  
  `password` varchar(250) NOT NULL,  
  `num_control` varchar(250) NOT NULL,  
  `foto` varchar(250) NOT NULL,  
  `nombre` varchar(250) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Ilustración 55.- Captura de pantalla del código a para crear la tabla usuarios dentro de la base de datos en el módulo 002

Acción: Crea una tabla llamada usuarios con las siguientes columnas:

- id: Entero, no nulo, clave primaria, autoincremental.
- email: Cadena de caracteres de hasta 250 caracteres, no nulo.
- password: Cadena de caracteres de hasta 250 caracteres, no nulo.
- num_control: Cadena de caracteres de hasta 250 caracteres, no nulo.
- foto: Cadena de caracteres de hasta 250 caracteres, no nulo.
- nombre: Cadena de caracteres de hasta 250 caracteres, no nulo.
 - Motor de almacenamiento: InnoDB.
 - Conjunto de caracteres: utf8mb4.

Resultados esperados:

Paso 1: Crear la base de datos

Acción	Resultado
CREATE DATABASE cardITA;	Base de datos cardITA creada exitosamente.

Paso 2: Seleccionar la base de datos

Acción	Resultado
--------	-----------

USE cardITA;	Contexto cambiado a la base de datos cardITA.
---------------------	---

Paso 3: Crear la tabla usuarios

Acción	Resultado
CREATE TABLE usuarios (...);	Tabla usuarios creada con las columnas especificadas.

Estado final de la base de datos cardITA:

- **Base de datos:** cardITA
- **Tablas:** usuarios
- **Estructura de la tabla usuarios:**

Columna	Tipo	Atributos
id	int(11)	NOT NULL, PRIMARY KEY, AUTO_INCREMENT
email	varchar(250)	NOT NULL
password	varchar(250)	NOT NULL
num_control	varchar(250)	NOT NULL
foto	varchar(250)	NOT NULL
nombre	varchar(250)	NOT NULL

Esta prueba de escritorio muestra que el código SQL crea una base de datos cardITA y una tabla usuarios dentro de esa base de datos con las columnas especificadas y sus respectivas restricciones y tipos de datos.

Numero de modulo 003: “login.php”

Modulo creado para la validación de los usuarios en la app Card-Ita

```

<?php

require_once 'include/DB_Functions.php';
$db = new DB_Functions();

$response = array("error" => FALSE);

if (isset($_POST['correo']) && isset($_POST['password'])) {

    $recomendacion = $_POST['correo'];
    $password = $_POST['password'];

    $user = $db->getUserByEmailAndPassword($recomendacion, $password);
    if ($user != false) {
        $response["error"] = FALSE;
        $response["user"]["id"] = $user["id"];
        $response["user"]["nombre"] = $user["nombre"];
        $response["user"]["foto"] = $user["foto"];
        $response["user"]["num_control"] = $user["num_control"];
        $response["user"]["usuario"] = $user["email"];
        $response["user"]["password"] = $user["password"];

        echo json_encode($response);
    } else {
        $response["error"] = TRUE;
        $response["error_msg"] = "Login credentials are wrong. Please try again!";
        echo json_encode($response);
    }
} else {
    $response["error"] = TRUE;
    $response["error_msg"] = "Required parameters email or password is missing!";
    echo json_encode($response);
}
?>

```

Ilustración 56.- Captura de pantalla del código a para la validación de los usuarios en el módulo 003

Datos de entrada:

Vamos a asumir los siguientes datos de entrada para la prueba:

- correo: example@example.com
- password: password123

1. Incluir el archivo DB_Functions.php:

- Acción: require_once 'include/DB_Functions.php';
- Resultado: Se incluye el archivo que contiene la clase DB_Functions.

2. Crear una instancia de DB_Functions:

- Acción: `$db = new DB_Functions();`
- Resultado: Se crea una instancia de la clase `DB_Functions`.

3. Inicializar el array de respuesta:

- Acción: `$response = array("error" => FALSE);`
- Resultado: Se inicializa un array llamado `$response` con el valor `"error" => FALSE`.

4. Verificar si los parámetros POST existen:

- Acción: `if (isset($_POST['correo']) && isset($_POST['password'])) {`
- Resultado: Se verifica si existen los parámetros `correo` y `password` en la solicitud POST. En este caso, sí existen.

5. Asignar los valores de los parámetros POST a las variables:

- Acción:

```
$recomendacion = $_POST['correo'];  
$password = $_POST['password'];
```

Ilustración 57.- Captura de pantalla del código de valores asignados en el módulo 003

- Resultado: Se asignan los valores de `$_POST['correo']` y `$_POST['password']` a las variables `$recomendacion` y `$password`, respectivamente.

Llamar a la función getUserByEmailAndPassword:

- Acción: `$user = $db->getUserByEmailAndPassword($recomendacion, $password);`
- Resultado: Se llama a la función `getUserByEmailAndPassword` con los valores de `$recomendacion` y `$password`. Esta función busca un usuario en la base de datos con el correo y la contraseña proporcionados.

```
$user = array(  
    "id" => 1,  
    "nombre" => "John Doe",  
    "foto" => "photo_url",  
    "num_control" => "123456",  
    "email" => "example@example.com",  
    "password" => "password123"  
);
```

Ilustración 58.- Captura de pantalla del código donde la función devuelve el siguiente array en el módulo 003

Verificar si se encontró un usuario:

- Acción: `if ($user != false) {`
- Resultado: Se verifica si la variable `$user` no es `false`. En este caso, `$user` contiene los datos del usuario, por lo que la condición es `true`.

Asignar los datos del usuario al array de respuesta:

- Acción

```

$response["error"] = FALSE;
$response["user"]["id"] = $user["id"];
$response["user"]["nombre"] = $user["nombre"];
$response["user"]["foto"] = $user["foto"];
$response["user"]["num_control"] = $user["num_control"];
$response["user"]["usuario"] = $user["email"];
$response["user"]["password"] = $user["password"];

```

Ilustración 59.- Captura de pantalla del código donde se asignan los datos del usuario al array en el módulo 003

Enviar la respuesta JSON:

- Acción: `echo json_encode($response);`

```

{
  "error": false,
  "user": {
    "id": 1,
    "nombre": "John Doe",
    "foto": "photo_url",
    "num_control": "123456",
    "usuario": "example@example.com",
    "password": "password123"
  }
}

```

Ilustración 60.- Captura de pantalla del código donde se muestra la respuesta JSON con los datos del usuario en el módulo 003

Casos adicionales:

Caso 1: Credenciales incorrectas

- Si `$db->getUserByEmailAndPassword($recomendacion, $password)` devuelve false, el código ejecuta:

```
$response["error"] = TRUE;
$response["error_msg"] = "Login credentials are wrong. Please try again!";
echo json_encode($response);
```

Ilustración 61.- Captura de pantalla del código donde se envía una respuesta JSON indicando que las credenciales son incorrectas en el módulo 003

Caso 2: Falta un parámetro

Si no se proporcionan los parámetros correo y password, el código ejecuta:

```
$response["error"] = TRUE;
$response["error_msg"] = "Required parameters email or password is missing!";
echo json_encode($response);
```

Ilustración 62.- Captura de pantalla del código donde se envía una respuesta JSON indicando que falta un parámetro requerido en el módulo 003

Estado final de la prueba de escritorio:

Línea	Acción	Resultado
2	<code>require_once 'include/DB_Functions.php';</code>	Se incluye el archivo DB_Functions.php.
3	<code>\$db = new DB_Functions();</code>	Se crea una instancia de DB_Functions.
5	<code>\$response = array("error" => FALSE);</code>	Se inicializa \$response con array("error" => FALSE).
7	<code>if (isset(\$_POST['correo']) && isset(\$_POST['password'])) {</code>	Se verifica si existen los parámetros correo y password en \$_POST. En este caso, sí existen.
9-10	<code>\$recomendacion = \$_POST['correo']; \$password = \$_POST['password'];</code>	Se asignan los valores de \$_POST['correo'] y \$_POST['password'] a \$recomendacion y \$password.
12	<code>\$user = \$db->getUserByEmailAndPassword(\$recomendacion, \$password);</code>	Se llama a la función getUserByEmailAndPassword y se obtiene el array \$user con los datos del usuario.
13	<code>if (\$user != false) {</code>	Se verifica si \$user no es false. En este caso, la condición es true.

14-22	Se asignan los datos del usuario al array \$response y se envía la respuesta JSON.	Se envía la respuesta JSON: {"error": false, "user": {"id": 1, "nombre": "John Doe", "foto": "photo_url", "num_control": "123456", "usuario": "example@example.com", "password": "password123"}}.
-------	--	---

Este análisis muestra cómo el código maneja la autenticación del usuario y construye una respuesta basada en los resultados de la búsqueda en la base de datos.

Numero del modulo 004: “callback.php”

Este modulo fue creado para autenticar un servicio, obtener un token de acceso y almacenar ese token en una base de datos para su uso posterior

```
<?php
require_once 'config.php';

try {
    $adapter->authenticate();
    $token = $adapter->getAccessToken();
    $db = new DB();
    $db->update_access_token(json_encode($token));
    echo "Access token inserted successfully.";
}
catch( Exception $e ){
    echo $e->getMessage() ;
}
?>
```

Ilustración 63.- Captura de pantalla del código donde se envía una respuesta JSON indicando que falta un parámetro requerido en el módulo 004

Datos de entrada y condiciones iniciales:

- config.php contiene la configuración necesaria y define las clases DB y la instancia \$adapter.

- \$adapter->authenticate() y \$adapter->getAccessToken() son métodos válidos del objeto \$adapter.
- La clase DB tiene un método update_access_token(\$token) que actualiza el token de acceso en la base de datos.

Incluir el archivo config.php:

- Acción: require_once 'config.php';
- Resultado: Se incluye el archivo config.php, lo que configura las dependencias necesarias y define la instancia \$adapter y la clase DB.

Inicio del bloque try:

- Acción: try {
- Resultado: Se inicia el bloque try para manejar posibles excepciones.

Autenticar usando el adaptador:

- Acción: \$adapter->authenticate();
- Resultado: Se llama al método authenticate del objeto \$adapter. Asumimos que la autenticación es exitosa y no lanza una excepción.

Obtener el token de acceso:

- Acción: \$token = \$adapter->getAccessToken();
- Resultado: Se llama al método getAccessToken del objeto \$adapter y se obtiene un token de acceso.

```
$token = array(
    "access_token" => "abc123",
    "expires_in" => 3600,
    "token_type" => "Bearer"
);
```

Ilustración 648 Captura de pantalla del código donde el token obtenido es "abc123" en el modulo 004

Crear una instancia de la clase DB:

- Acción: `$db = new DB();`
- Resultado: Se crea una nueva instancia de la clase DB.

Actualizar el token de acceso en la base de datos:

- Acción: `$db->update_access_token(json_encode($token));`
- Resultado: Se llama al método `update_access_token` del objeto `$db`, pasando el token de acceso codificado en JSON.

```
{
  "access_token": "abc123",
  "expires_in": 3600,
  "token_type": "Bearer"
}
```

Ilustración 65.- Captura de pantalla del código donde el token codificado en JSON es "abc123" en el módulo 004

Mensaje de éxito:

- Acción: `echo "Access token inserted successfully.";`
- Resultado: Se imprime el mensaje "Access token inserted successfully."

Captura de excepciones:

- Acción: `catch(Exception $e) { echo $e->getMessage(); }`
- Resultado: Si ocurre una excepción en el bloque try, se captura y se imprime el mensaje de la excepción. En este caso, no se produce ninguna excepción, por lo que esta sección no se ejecuta.

Resultado de la prueba de escritorio:

1. Inicio del script:

- Incluir el archivo `config.php`:
 - Acción: `require_once 'config.php';`
 - Resultado: Archivo `config.php` incluido correctamente.

Autenticación del adaptador:

- Acción: \$adapter->authenticate();
- Resultado: Autenticación exitosa.

Obtener el token de acceso:

- Acción: \$token = \$adapter->getAccessToken();

```
$token = array(
    "access_token" => "abc123",
    "expires_in" => 3600,
    "token_type" => "Bearer"
);
```

Ilustración 66.- Captura de pantalla del código donde se muestra el resultado del token obtenido en el módulo 004

Crear una instancia de DB:

- Acción: \$db = new DB();
- Resultado: Instancia de DB creada correctamente.

Actualizar el token de acceso en la base de datos:

- Acción: \$db->update_access_token(json_encode(\$token));
- Resultado: Token actualizado en la base de datos con el valor:

```
{
    "access_token": "abc123",
    "expires_in": 3600,
    "token_type": "Bearer"
}
```

Ilustración 67.- Captura de pantalla del código donde se muestra el resultado del token obtenido en el módulo 004

Mensaje de éxito:

- Acción: echo "Access token inserted successfully.";
- Resultado: Se imprime el mensaje "Access token inserted successfully.".

Captura de excepciones:

- Acción: catch(Exception \$e) { echo \$e->getMessage(); }

- Resultado: No se produce ninguna excepción, por lo que esta sección no se ejecuta.

Estado final de la prueba de escritorio:

Línea	Acción	Resultado
2	<code>require_once 'config.php';</code>	Archivo config.php incluido correctamente.
4	<code>try {</code>	Inicio del bloque try.
5	<code>\$adapter->authenticate();</code>	Autenticación exitosa.
6	<code>\$token = \$adapter->getAccessToken();</code>	Token obtenido: {"access_token": "abc123", "expires_in": 3600, "token_type": "Bearer"}.
7	<code>\$db = new DB();</code>	Instancia de DB creada correctamente.
8	<code>\$db->update_access_token(json_encode(\$token));</code>	Token actualizado en la base de datos con el valor: {"access_token": "abc123", "expires_in": 3600, "token_type": "Bearer"}.
9	<code>echo "Access token inserted successfully.";</code>	Se imprime el mensaje "Access token inserted successfully.".
11	<code>catch(Exception \$e) { echo \$e->getMessage(); }</code>	No se produce ninguna excepción, por lo que esta sección no se ejecuta.

Numero de modulo A001: Aplicación Card-Ita “MainFest.xml”

Descripción del Código

El código define tres actividades (LoginActivity, MainActivity, y splash) para la aplicación "Card-ITA". Aquí se detallan las configuraciones clave de cada actividad y el flujo de la aplicación:

1. LoginActivity:

- Nombre de la actividad: .LoginActivity
- Título: "Card-ITA"
- Orientación: vertical
- Tema: AppTheme.NoActionBar

2. MainActivity:

- Nombre de la actividad: .MainActivity
- Título: "Card-ITA"
- Orientación: vertical
- Tema: AppTheme.NoActionBar

3. splash:

- Nombre de la actividad: .splash
- Exportada: sí
- Título: "Card-ITA"
- Orientación: vertical
- Tema: AppTheme.NoActionBar
- Filtro de intentos: actividad principal y lanzador

Flujo de la Prueba de Escritorio

Escenario 1: Inicio de la Aplicación

1. Inicio de la aplicación

- **Acción:** El usuario toca el ícono de la aplicación en el lanzador.
- **Resultado esperado:** La actividad splash se inicia debido al filtro de intentos configurado para esta actividad.

2. Verificar configuración de splash

- **Propiedades:**
 - Título: "Card-ITA"
 - Orientación: vertical
 - Tema: AppTheme.NoActionBar
- **Resultado esperado:** La actividad splash se muestra en orientación vertical sin una barra de acción, y el título de la actividad es "Card-ITA".

Escenario 2: Navegación a LoginActivity

1. Navegación a LoginActivity desde splash

- **Acción:** Después de mostrar la pantalla splash, la aplicación navega a LoginActivity.

- **Resultado esperado:** La actividad LoginActivity se inicia.
- 2. **Verificar configuración de LoginActivity**
 - **Propiedades:**
 - Título: "Card-ITA"
 - Orientación: vertical
 - Tema: AppTheme.NoActionBar
 - **Resultado esperado:** La actividad LoginActivity se muestra en orientación vertical sin una barra de acción, y el título de la actividad es "Card-ITA".

Escenario 3: Navegación a MainActivity

- 1. **Inicio de sesión exitoso en LoginActivity**
 - **Acción:** El usuario inicia sesión correctamente desde LoginActivity.
 - **Resultado esperado:** La actividad MainActivity se inicia.
- 2. **Verificar configuración de MainActivity**
 - **Propiedades:**
 - Título: "Card-ITA"
 - Orientación: vertical
 - Tema: AppTheme.NoActionBar
 - **Resultado esperado:** La actividad MainActivity se muestra en orientación vertical sin una barra de acción, y el título de la actividad es "Card-ITA".

Simulación de la Prueba

- **Inicio de la aplicación:**
 - La pantalla de splash se muestra en orientación vertical, sin barra de acción, y con el título "Card-ITA".
- **Navegación a LoginActivity:**
 - Tras unos segundos, la pantalla de splash navega a LoginActivity, que también se muestra en orientación vertical, sin barra de acción, y con el título "Card-ITA".

- **Inicio de sesión y navegación a MainActivity:**
 - El usuario inicia sesión exitosamente y navega a MainActivity, que se muestra en orientación vertical, sin barra de acción, y con el título "Card-ITA".

Numero de la prueba A002: "GalleryFragment.java"

Este modulo fue creado para el funcionamiento local de la aplicación Card-Ita dentro del Instituto Tecnológico de Apizaco

Simulación de la Prueba de Escritorio

Estado Inicial

- root es una vista raíz válida que contiene un ImageView con el ID foto y un TextView.
- vfoto tiene un valor como "imagen.jpg".
- vnombre tiene un valor como "Nombre del Usuario".
- R.id.foto es un ID válido en el layout.
- texto es una referencia válida a un TextView dentro de root.

Paso a Paso

1. **Obtener la referencia al ImageView:**

- **Acción:** `ImageView imageView = (ImageView) root.findViewById(R.id.foto);`
- **Resultado Esperado:** imageView ahora referencia al ImageView con el ID foto dentro de root.

2. **Cargar la imagen usando Glide:**

- **Acción:** `Glide.with(this).load("http://192.168.1.89/card-ITA/fotos/" + vfoto).into(imageView);`

- Supongamos que vfoto = "imagen.jpg".
 - La URL completa será "http://192.168.1.89/card-ITA/fotos/imagen.jpg".
 - **Resultado Esperado:** Glide carga la imagen desde la URL y la muestra en imageView.
 - **Verificación:** La imagen debería aparecer en el ImageView.
3. **Establecer el texto en el TextView:**
- **Acción:** texto.setText(vnombre);
 - Supongamos que vnombre = "Nombre del Usuario".
 - **Resultado Esperado:** texto muestra el texto "Nombre del Usuario".
 - **Verificación:** El TextView debería mostrar el nombre del usuario.
4. **Devolver la vista root:**
- **Acción:** return root;
 - **Resultado Esperado:** El método devuelve la vista root.

Resultados de la Prueba de Escritorio

- **Estado Final:**
 - imageView muestra la imagen cargada desde "http://192.168.1.89/card-ITA/fotos/imagen.jpg".
 - texto muestra el texto "Nombre del Usuario".
 - La vista root es devuelta correctamente.

```
ImageView imageView = (ImageView) root.findViewById(R.id.foto);
Glide.with(this).load("http://192.168.1.89/card-ITA/fotos/" + vfoto).into(imageView);
```

Ilustración 62.- Se muestra el código a evaluar en el modulo A002

Descripción del Código

1. **Obtener una referencia a un ImageView:**
 - Se obtiene la referencia a un ImageView con el ID foto desde la vista root.
2. **Cargar una imagen usando Glide:**

- Se usa Glide para cargar una imagen desde una URL, que incluye el valor de vfoto, y se coloca en el ImageView obtenido anteriormente.

Variables de Entrada

- root: La vista raíz que contiene el ImageView.
- vfoto: La variable que contiene el nombre de la imagen que se debe cargar.
- R.id.foto: El ID del ImageView dentro del layout.

Simulación de la Prueba de Escritorio

Estado Inicial

- root es una vista raíz válida que contiene un ImageView con el ID foto.
- vfoto tiene un valor como "imagen.jpg".
- R.id.foto es un ID válido en el layout.

Paso a Paso

1. **Obtener la referencia al ImageView:**

- **Acción:** `ImageView imageView = (ImageView) root.findViewById(R.id.foto);`
- **Resultado Esperado:** imageView ahora referencia al ImageView con el ID foto dentro de root.

2. **Cargar la imagen usando Glide:**

- **Acción:** `Glide.with(this).load("http://192.168.1.89/card-ITA/fotos/" + vfoto).into(imageView);`
 - Supongamos que vfoto = "imagen.jpg".
 - La URL completa será "http://192.168.1.89/card-ITA/fotos/imagen.jpg".
- **Resultado Esperado:** Glide carga la imagen desde la URL y la muestra en imageView.
- **Verificación:** La imagen debería aparecer en el ImageView.

Resultados de la Prueba de Escritorio

- **Estado Final:**

- imageView muestra la imagen cargada desde "<http://192.168.1.89/card-ITA/fotos/imagen.jpg>".

11. Conclusiones

El desarrollo e implementación del sistema de credencial digital basado en códigos QR representó un avance significativo en la modernización del control de acceso a la universidad. A lo largo del proyecto, se logró sustituir el uso de credenciales físicas con tecnología RFID por una solución digital accesible desde dispositivos móviles, brindando mayor comodidad y eficiencia para los estudiantes.

Entre los principales logros esta el echo de una alternativa viable para la reducción en los costos de producción y reposición de credenciales, la optimización del tiempo de acceso a las instalaciones y la mejora en la seguridad mediante la autenticación en tiempo real. Además, el sistema fue bien recibido por los usuarios de prueba, quienes valoraron su facilidad de uso y la eliminación de la necesidad de portar una credencial física.

La implementación de este sistema podría ser viable y sostenible a largo plazo, alineándose con la tendencia de transformación digital en instituciones educativas. Asimismo, sentó las bases para futuras mejoras, como la integración con otros servicios académicos o el uso de nuevas tecnologías de seguridad.

En conclusión, el proyecto cumplió con los objetivos planteados, proporcionando una solución innovadora y eficiente para el acceso a la universidad. Su desarrollo evidencia el impacto positivo de la digitalización en los procesos administrativos y abre la puerta a nuevas oportunidades de optimización en el ámbito académico.

12. Competencias desarrolladas

Durante el desarrollo e implementación del sistema de credencial digital con código QR, se fortaleció diversas competencias técnicas y profesionales, que fueron clave para el éxito del proyecto. Entre las que destacan:

Competencias Técnicas

Desarrollo de Software: Implementación de una aplicación móvil en Android Studio utilizando Java/Kotlin, incluyendo el manejo de interfaces gráficas y almacenamiento de datos.

Gestión de Bases de Datos: Diseño e integración de una base de datos para el almacenamiento y validación de credenciales digitales.

Seguridad en Aplicaciones: Implementación de medidas de seguridad para garantizar la autenticidad y privacidad de los códigos QR generados.

Arquitectura Cliente-Servidor: Desarrollo de un sistema basado en una comunicación eficiente entre la aplicación móvil y el servidor de la universidad.

Optimización de Procesos: Implementación de técnicas para mejorar la rapidez y eficiencia en la validación del acceso mediante QR.

Competencias de Gestión y Planificación

Gestión de Proyectos de Software: Aplicación de metodologías para la planificación, desarrollo y prueba del sistema.

Documentación Técnica: Elaboración de documentos detallados sobre la arquitectura, funcionamiento y resultados del proyecto.

Trabajo en Equipo: Coordinación con asesores y personal administrativo para asegurar la viabilidad e integración del sistema.

Competencias Profesionales

Resolución de Problemas: Identificación y solución de desafíos técnicos y operativos durante la implementación del sistema.

Adaptabilidad y Aprendizaje Continuo: Uso de nuevas herramientas y tecnologías para mejorar el rendimiento del proyecto.

Comunicación Efectiva: Presentación y justificación del proyecto ante asesores y autoridades de la universidad.

13. Referencias

- Biblioguías: Qué son los Códigos QR: Qué es Código QR. (n.d.). [https://biblioguías.cepal.org/QR#:~:text=%C2%BFQu%C3%A9%20es%20el%20c%C3%B3digo%20QR%3F&text=Los%20c%C3%B3digos%20QR%20\(Quick%20Response,est%C3%A1n%20actualmente%20muy%20de%20moda.](https://biblioguías.cepal.org/QR#:~:text=%C2%BFQu%C3%A9%20es%20el%20c%C3%B3digo%20QR%3F&text=Los%20c%C3%B3digos%20QR%20(Quick%20Response,est%C3%A1n%20actualmente%20muy%20de%20moda.)
- Instituto de manejo proyectos. (2017). Una guía de los fundamentos de la dirección de proyectos (Guía del PMBOK), sexta edición.

- International Organization for Standardization (ISO). (2019). ISO 9241-11:2018 Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts. <https://www.iso.org/standard/63500.html>
- International Organization for Standardization (ISO). (2019). ISO 9000:2015 Quality management systems — Fundamentals and vocabulary. <https://www.iso.org/standard/63712.html>
- International Organization for Standardization. (2015). ISO/IEC 18004:2015 - Information technology - Automatic identification and data capture techniques - QR Code 2005 bar code symbology specification. Retrieved from <https://www.iso.org/standard/62021.html>
- Investopedia. (2022). Blockchain. Retrieved from <https://www.investopedia.com/terms/b/blockchain.asp>
- ISO/IEC 27001. (2013). Information technology - Security techniques - Information security management systems - Requirements. International Organization for Standardization.
- Kahneman, D. (2011). Thinking, fast and slow. Farrar, Straus and Giroux.
- Lantrip, L. (2021). The Psychology of Color: A Designer's Guide to Color Association & Meaning. Retrieved from <https://99designs.com/blog/tips/the-psychology-of-color/>
- Laudon, K. C. y Laudon, J. P. (2016). Sistemas de Información Gerencial: Gestión de la Empresa Digital. Pearson Educación limitada.
- Li, X., Li, W., Li, Y., & Zhu, H. (2016). A ubiquitous campus information service system based on QR code. Journal of Ambient Intelligence and Humanized Computing, 7(2), 189-200.
- Liu, H., & Wu, Y. (2017). Design and implementation of campus smart card system based on QR code. Journal of Physics: Conference Series, 946(1), 012010.
- Makridakis, S., Lontou, C. y Barlas, P. (2018). Blockchain: una solución a los problemas de transparencia y confianza en la acreditación y la acreditación. En R. Koussouris, D. Gouscos y E. Bacao (Eds.), Gobierno Abierto: Oportunidades y Desafíos para la Gobernanza Pública (pp. 289-306). Saltador.
- Makridakis, S., Lontou, C., & Barlas, P. (2018). Blockchain: A Solution to Transparency and Trust Issues in Accreditation and Credentialing. In R. Koussouris, D. Gouscos, & E. Bacao (Eds.), Open Government: Opportunities and Challenges for Public Governance (pp. 289-306). Springer.
- Moon, J. W., & Kim, Y. G. (2001). Extending the TAM for a World-Wide-Web context. Information & management, 38(4), 217-230.