****

**ControlPC**

**PROYECTO INTEGRADOR**

**TÉCNICO SUPERIOR UNIVERSITARIO EN TECNOLOGÍAS DE LA INFORMACIÓN ÁREA DESARROLLO DE SOFTWARE MULTIPLATAFORMA**

*P R E S E N T A*

**RIAÑO ISHIWARA YOSHIRO PABLO ARMANDO**

ASESORA DE LA ORGANIZACIÓN: INGA. CLAUDIA VANESSA SÁNCHEZ

ASESORA ACADÉMICA: LDA. SANDRA RAQUEL LÓPEZ ARCE SOROA

ORGANIZACIÓN: UNIVERSIDAD TECNOLÓGICA DE TECÁMAC  
PERIODO CUATRIMESTRAL: MAYO-AGOSTO 2025

**ÍNDICE**

[RESUMEN 1](#_Toc204846612)

[ABSTRACT 2](#_Toc204846613)

[INTRODUCCIÓN 3](#_Toc204846614)

[OBJETIVOS 5](#_Toc204846615)

[PROGRAMA Y CRONOGRAMA 6](#_Toc204846616)

[MARCO TEÓRICO 11](#_Toc204846617)

[METODOLOGÍA 20](#_Toc204846618)

[CAPÍTULO 1 RECOLECCIÓN Y REFINAMIENTO DE REQUISITOS 23](#_Toc204846619)

[1.1. Necesidades del cliente 23](#_Toc204846620)

[1.2. Identificación y definición de roles de usuario 24](#_Toc204846621)

[1.3. Requerimientos funcionales 24](#_Toc204846622)

[1.4. Requerimientos no funcionales 25](#_Toc204846623)

[1.5. Requerimientos de hardware 27](#_Toc204846624)

[1.6. Requerimientos de software 27](#_Toc204846625)

[1.7. Diagrama general de casos de uso 28](#_Toc204846626)

[1.8. Especificación de los casos de uso principales 30](#_Toc204846627)

[CAPÍTULO 2 DISEÑO Y DESARROLLO DEL PROYECTO 35](#_Toc204846628)

[2.1 Diseño de la base de datos (modelo relacional) 35](#_Toc204846629)

[2.2 Diccionario de datos 36](#_Toc204846630)

[2.3 Diseño de interfaces 39](#_Toc204846631)

[2.4 Mapa de navegación 45](#_Toc204846632)

[2.5 Primer avance del proyecto 46](#_Toc204846633)

[CAPÍTULO 3 IMPLEMENTACIÓN Y PRUEBAS 48](#_Toc204846634)

[3.1 Documentación de interfaces 48](#_Toc204846635)

[3.2 Página web informativa y de contacto 63](#_Toc204846636)

[CONCLUSIONES 69](#_Toc204846637)

[ANEXOS Y/O APÉNDICES 70](#_Toc204846638)

[LISTADO DE SIGLAS Y ACRÓNIMOS 74](#_Toc204846639)

[GLOSARIO 75](#_Toc204846640)

[REFERENCIAS 78](#_Toc204846641)

**RESUMEN**

El presente proyecto tiene como finalidad el desarrollo de una aplicación móvil denominada “Control PC”, que tiene como objetivo gestionar de manera eficiente el préstamo de laptops dentro de instituciones educativas, específicamente en la Universidad Tecnológica de Tecámac. Esta aplicación responde a la necesidad de modernizar y optimizar procesos que anteriormente se realizaban de forma manual, provocando pérdida de información, inconsistencias en los registros y falta de control sobre los equipos prestados.

Pensada especialmente para entornos donde no se dispone de acceso a internet, ControlPC opera completamente offline mediante una base de datos local, lo que garantiza trazabilidad, confiabilidad y disponibilidad en todo momento. La aplicación permite al personal encargado, principalmente el laboratorista, realizar tareas clave como registrar usuarios, gestionar el inventario de laptops, administrar préstamos y devoluciones, así como generar reportes sobre el estado y uso de los equipos.

Se utilizaron herramientas como Visual Studio, Java y SQLite, enfocándose en una interfaz intuitiva y un diseño minimalista, con el objetivo de facilitar el uso del sistema y reducir la curva de aprendizaje del personal. Se definieron tanto requerimientos funcionales (registro de usuarios, control de inventario, gestión de préstamos, generación de reportes) como no funcionales (seguridad, usabilidad, rendimiento, escalabilidad y funcionamiento sin conexión), además de especificaciones de hardware y software necesarias para su correcta operación.

Esta aplicación móvil no solo mejora el control y seguimiento del préstamo de equipos de cómputo, sino que también promueve una administración más ordenada y segura.

**ABSTRACT**

The purpose of this project is the development of a mobile application called “Control PC”, aimed at efficiently managing the lending of laptops within educational institutions, specifically at the Universidad Tecnológica de Tecámac. This application addresses the need to modernize and optimize processes that were previously carried out manually, which led to information loss, inconsistencies in records, and a lack of control over loaned equipment.

Designed especially for environments where there is no internet access, ControlPC operates fully offline using a local database, ensuring traceability, reliability, and continuous availability. The application allows the personnel responsible, mainly the laboratory technician, to perform key tasks such as registering users, managing the laptop inventory, handling loan and return operations, and generating reports on the status and usage of the equipment.

Tools such as Visual Studio, Java, and SQLite were used, focusing on an intuitive interface and a minimalist design to make the system easy to use and reduce the learning curve for staff. Both functional requirements (user registration, inventory control, loan management, report generation) and non-functional requirements (security, usability, performance, and scalability, and offline functionality) were defined, along with the necessary hardware and software specifications for proper operation.

This mobile application not only improves the control and tracking of laptop loans but also promotes a more organized and secure administrative process.

INTRODUCCIÓN

La Universidad Tecnológica de Tecámac (UTTEC), fundada en 1996, es una institución pública de nivel superior ubicada en el kilómetro 37.5 de la carretera federal México-Pachuca, en el predio Sierra Hermosa del municipio de Tecámac, Estado de México. La universidad ofrece programas educativos organizados en diversas divisiones académicas, entre ellas la División de Tecnologías de la Información y Comunicación (TIC), que contempla carreras como Desarrollo de Software Multiplataforma (DSM) e Infraestructura de Redes Digitales (IRD), con posibilidad de continuidad hacia las ingenierías en Desarrollo y Gestión de Software o en Redes Inteligentes y Ciberseguridad.

El presente proyecto integrador se llevará a cabo dentro del programa de Desarrollo de Software Multiplataforma (DSM) durante el cuatrimestre mayo agosto 2025. Este proyecto tendrá como objetivo que los estudiantes demuestren las competencias adquiridas a lo largo de su formación, mediante la planeación, desarrollo y documentación de una solución tecnológica a una problemática real detectada en su entorno. Como parte de este proceso, los estudiantes trabajarán en equipo, desarrollarán una aplicación móvil funcional, y generarán una identidad empresarial compuesta por nombre, logotipo y eslogan. También se documentará el desarrollo en un reporte técnico estructurado por capítulos.

Descripción del problema:

Actualmente, el personal encargado del préstamo de laptops dentro de la Universidad Tecnológica de Tecámac enfrenta diversas dificultades para registrar y dar seguimiento a dichos préstamos. El proceso se realiza de manera manual, utilizando hojas de papel, lo genera pérdida de información, falta de control en los equipos entregados y poca eficiencia en la gestión del inventario. Esta situación representa un obstáculo en la administración adecuada de los recursos tecnológicos disponibles, por lo que se requiere implementar una solución digital que permita optimizar este proceso y minimizar errores.

Propuesta del proyecto:

Para resolver esta problemática, se desarrollará una aplicación móvil denominada ControlPC, diseñada para registrar usuarios, gestionar el inventario de laptops disponibles y realizar el seguimiento de los préstamos en tiempo real. Esta aplicación se conectará a una base de datos relacional local y permitirá al personal administrativo consultar el estado físico, las características técnicas, el número de inventario y la disponibilidad de cada equipo. Los dispositivos en préstamo podrán identificarse fácilmente junto con su fecha de salida y posible devolución, lo cual facilitará el seguimiento y permitirá mantener un historial de uso confiable.

Se implementarán además requisitos no funcionales como seguridad de los datos, compatibilidad con distintos sistemas operativos móviles, portabilidad, alta disponibilidad y facilidad de mantenimiento. Con ello se buscará garantizar una solución tecnológica eficiente, segura y accesible.

El reporte técnico del proyecto se estructurará en tres capítulos principales:

En el Capítulo 1: Recolección y Refinamiento de Requisitos, se recopilarán las necesidades del cliente y se definirán los roles de usuario. También se establecerán los requerimientos funcionales y no funcionales, así como los recursos de hardware y software necesarios. Este capítulo incluirá un diagrama general de casos de uso y la especificación de los principales casos de uso.

En el Capítulo 2: Diseño y Desarrollo del Proyecto, se presentará el diseño de la base de datos mediante un modelo relacional, junto con su respectivo diccionario de datos. Se realizará el diseño de interfaces de usuario, se desarrollará el mapa de navegación de la aplicación y se documentará el primer avance del sistema.

En el Capítulo 3: Implementación y Pruebas, se documentará el proceso de construcción del sistema, incluyendo la implementación de las interfaces, el desarrollo de una página web informativa con las funcionalidades CRUD de la aplicación, y la realización de pruebas que permitirán validar el correcto funcionamiento del sistema.

OBJETIVOS

Objetivo general:

Desarrollar una solución móvil multiplataforma que permita administrar de forma eficiente el préstamo de laptops, mediante una aplicación, que registre, organice y supervise la disponibilidad, uso y estado de los equipos, garantizando un control preciso y actualizado, orientado a mejorar la gestión del personal administrativo.

Objetivos específicos:

1. Diseñar una estructura de datos adecuada que permita gestionar usuarios, equipos y registros de préstamo de forma eficiente.
2. Implementar estrategias para una manipulación eficiente de datos, que respalden el manejo de información sin conexión a internet.
3. Analizar factores que influyen en el rendimiento y usabilidad de aplicaciones móviles nativas en contextos escolares, a fin de optimizar la experiencia del usuario.
4. Realizar pruebas funcionales y de usabilidad en distintos dispositivos móviles para garantizar una experiencia estable.
5. Garantizar una aplicación que facilite el control, monitoreo y trazabilidad de los equipos prestados en instituciones educativas, mejorando la eficiencia del proceso y reduciendo errores humanos en el registro manual.

PROGRAMA Y CRONOGRAMA

|  |
| --- |
| ***UNIVERSIDAD TECNOLÓGICA DE TECÁMAC*** |
| ***DIVISIÓN TIC*** |
| ***PROGRAMA DE ESTADÍAS PROFESIONALES*** |

|  |  |  |
| --- | --- | --- |
| ***PROGRAMA DE TRABAJO*** |  | |
|  | | *FECHA: 06/05/2025* |

*DATOS DEL O DE LA ESTUDIANTE*

|  |  |
| --- | --- |
| NOMBRE: | * Flores Zarate Edwin * Florez Reynoso Martha * Gómez Fuentes Kevyn Axel * Riaño Ishiwara Yoshiro Pablo Armando |
| DIVISIÓN: | Tecnologías de la Información y Comunicación |
| CARRERA: | Técnico Superior Universitario en Tecnologías de la Información Área Desarrollo de Software Multiplataforma |
| MATRÍCULA: | * 2524160037 * 2524260015 * 2524260101 * 2524260029 |
| GENERACIÓN: | Mayo-agosto 2025 |

*ASESOR ACADÉMICO O ASESORA ACADÉMICA*

|  |  |
| --- | --- |
| NOMBRE: | López Arce Soroa Sandra Raquel |
| CARGO: | Profesora de asignatura |

*DATOS DE LA ORGANIZACIÓN*

|  |  |
| --- | --- |
| NOMBRE DE LA ORGANIZACIÓN: | Universidad Tecnológica de Tecámac |
| DEPARTAMENTO: | División de Tecnologías de la información y comunicación |
| ÁREA: | Desarrollo de Software Multiplataforma |
| DIRECCIÓN: | Carretera Federal México-Pachuca km 37.5, Predio Sierra Hermosa, Tecámac, Estado de México |
| TELÉFONO: | 59388400 extensión 446 |
| E-MAIL: | [dtic@uttecamac.edu.mx](mailto:dtic@uttecamac.edu.mx) |

*ASESOR DE LA ORGANIZACIÓN O ASESORA DE LA ORGANIZACIÓN*

|  |  |
| --- | --- |
| NOMBRE: | Tamariz Sánchez Claudia Vanessa |
| CARGO: | Profesor de asignatura |

*PERÍODO*

|  |  |
| --- | --- |
| DURACIÓN: | 15 semanas |
| FECHA DE INICIO: | 06 de mayo de 2025 |
| FECHA DE TERMINACIÓN: | 23 de agosto 2025 |
| HORARIO: | 09-18 horas |

*PROYECTO*

|  |  |
| --- | --- |
| NOMBRE: | ControlPC |
| DESCRIPCIÓN: | Diseñar y desarrollar una aplicación móvil multiplataforma conectada a una base de datos relacional local, que permita al personal administrativo llevar el control de los préstamos de los equipos de cómputo, optimizando la gestión de los recursos tecnológicos mediante un sistema digital confiable, eficiente y accesible. |
| OBJETIVO GENERAL: | Desarrollar una solución móvil multiplataforma que permita administrar de forma eficiente el préstamo de laptops, mediante una aplicación que registre, organice y supervise la disponibilidad, uso y estado de los equipos, garantizando un control preciso y actualizado, orientado a mejorar la gestión del personal administrativo. |
| OBJETIVOS ESPECÍFICOS: | 1. Diseñar una estructura de datos adecuada que permita gestionar usuarios, equipos y registros de préstamo de forma eficiente. 2. Implementar estrategias para una manipulación eficiente de datos, que respalden el manejo de información sin conexión a internet. 3. Analizar factores que influyen en el rendimiento y usabilidad de aplicaciones móviles nativas en contextos escolares, a fin de optimizar la experiencia del usuario. 4. Realizar pruebas funcionales y de usabilidad en distintos dispositivos móviles para garantizar una experiencia estable. 5. Garantizar una aplicación que facilite el control, monitoreo y trazabilidad de los equipos prestados en instituciones educativas, mejorando la eficiencia del proceso y reduciendo errores humanos en el registro manual. |
| ALCANCE(S): | Implementar una aplicación móvil multiplataforma que funcione de forma local, sin requerir conexión a internet, permitiendo al personal administrativo registrar y consultar el préstamo de laptops disponibles, gestionar los datos de los usuarios y mantener un historial de préstamos. La duración estimada para el desarrollo completo es de quince semanas. |
| META(S): | Contar con una aplicación móvil multiplataforma completamente funcional, capaz de operar de manera óptima, sin conexión a internet, que permitirá gestionar de forma segura, confiable y eficiente el préstamo de laptops, con una base de datos local que proteja la integridad y confidencialidad de los datos. |
| RECURSOS: | Hardware:   * Laptop Lenovo Ideapad 3 15ITL6 * Pantalla Foxconn 24 pulgadas * Celular Xiaomi Redmi Note 10S * Celular Cubot Note 50 * Mouse Steren Mouse USB * Bocinas Bluetooth mmii BoomBox TWS * USB Kingston DataTraveler Exodia   Software:   * PowerPoint * Word * Visual Studio 2022 * C# * SQLite * Xamarin * GitHub |

*PLAN DE TRABAJO*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ACTIVIDAD | | DESCRIPCIÓN | SEMANA | | FECHAS | |
| INICIO | TÉRMINO | INICIO | TÉRMINO |
| 1 | 1.1 | Necesidades del cliente | 3 | 3 | |  |  | | --- | --- | |  | | |  |  |  | | --- | | 19/05/2025 | | |  | | --- | |  |  |  | | --- | | 19/05/2025 | |
|  | 1.2 | Identificación y definición de roles de usuario | 3 | 3 | 19/05/2025 | 19/05/2025 |
|  | 1.3 | Requerimientos funcionales | 3 | 3 | 20/05/2025 | 20/05/2025 |
|  | 1.4 | Requerimientos No funcionales | 3 | 3 | 20/05/2025 | 20/05/2025 |
|  | 1.5 | Requerimientos de hardware | 3 | 3 | 21/05/2025 | 21/05/2025 |
|  | 1.6 | Requerimientos de software | 3 | 3 | 21/05/2025 | 21/05/2025 |
|  | 1.7 | Diagrama general de casos de uso | 4 | 4 | 26/05/2025 | 26/05/2025 |
|  | 1.8 | Especificación de los casos de uso principales | 4 | 4 | 26/05/2025 | 26/05/2025 |
| 2 | 2.1 | Diseño de la base de datos (modelo relacional) | 4 | 4 | 27/05/2025 | 29/05/2025 |
|  | 2.2 | Diccionario de datos | 4 | 4 | |  | | --- | |  |  |  | | --- | | 03/06/2025 | | |  | | --- | |  |  |  | | --- | | 03/06/2025 | |
|  | 2.3 | Diseño de interfaces | 6 | 6 | 10/06/2025 | 12/06/2025 |
|  | 2.4 | Mapa de navegación | 7 | 8 | 17/06/2025 | 28/06/2025 |
|  | 2.5 | Primer avance del proyecto | 9 | 10 | |  | | --- | |  |  |  | | --- | | 01/07/2025 | | 12/07/2025 |
| 3 | 3.1 | Documentación de interfaces | 11 | 11 | 15/07/2025 | 19/07/2025 |
|  | 3.2 | Página web informativa y de contacto | 12 | 13 | 22/07/2025 | 02/08/2025 |

*CRONOGRAMA DE ACTIVIDADES (PROGRAMA)*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | ACTIVIDADES | CONTROL | Mes 1 | | | | | Mes 2 | | | | Mes 3 | | | | Mes 4 | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 1 | RECOLECCIÓN DE DATOS | PROG. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| REAL |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | RECOLECCIÓN Y REFINAMIENTO DE REQUISITOS | PROG. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| REAL |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | DISEÑO Y DESARROLLO DEL PROYECTO | PROG. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| REAL |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | IMPLEMENTACIÓN Y PRUEBAS | PROG. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| REAL |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

*FIRMAS*

|  |  |
| --- | --- |
| Inga. Tamariz Sánchez Claudia Vanessa  *ASESORA O ASESOR DE LA ORGANIZACIÓN* | |
| Riaño Ishiwara Yoshiro Pablo Armando  *ESTUDIANTE* | *Lda. López Arce Soroa Sandra Raquel*  *ASESORA ACADÉMICA O ASESOR ACADÉMICO* |

MARCO TEÓRICO

Es una herramienta analítica con variaciones y contexto que sirve para realizar investigaciones, también se puede aplicar en diferentes categorías de trabajo donde se necesita una imagen general.

Las características del marco teórico son una identificación de fuentes primarias y secundarias sobre las cuales se podrá investigar y diseñar la propuesta, así como, la lectura de textos, libros especializados, revistas y trabajos anteriores en la modalidad de tesis de grado son fundamentales en su formación.

El uso del móvil se ha multiplicado en los últimos años. Por eso, no es de extrañar que muchas empresas cuenten ya con una aplicación móvil. De hecho, existen aplicaciones para casi todo: salud y fitness, educación, entretenimiento, gestión de facturas.

Tipos de aplicaciones móviles:

Aplicación Nativa:

Las aplicaciones nativas son aplicaciones desarrolladas para un sistema operativo móvil concreto (iOS o Android), en el lenguaje de programación específico de cada plataforma. Esto quiere decir que una aplicación nativa creada para Android no puede ser utilizada en un dispositivo iOS y viceversa.

Es el tipo de aplicación móvil más conocida. Para que funcione, debemos descargarla desde los markets de aplicaciones, como App Store o Google Play e instalarla en nuestro teléfono.

Ventajas:

* Tienen el mejor rendimiento. Las aplicaciones nativas son las más rápidas y tienen un rendimiento superior a otros tipos de aplicaciones, ya que han sido optimizadas específicamente para el hardware y el sistema operativo del dispositivo.
* Acceso completo e integración con las funciones hardware del dispositivo. Las aplicaciones nativas permiten aprovechar al máximo las funcionalidades móviles: cámara, micrófono, lector biométrico de huella, sensores, redes inalámbricas (Wi-Fi, bluetooth…).
* Pueden funcionar sin acceso a Internet (funcionamiento offline) si han sido diseñadas para ello.

Desventajas:

* Costes de desarrollo altos. Si queremos tener nuestra aplicación disponible para los dos sistemas, necesitaremos dos líneas de desarrollo diferentes, ya que el código utilizado para un sistema no es reutilizable para otro.
* Complejidad de desarrollo. Necesitamos equipos expertos en el lenguaje específico de cada sistema. Por ejemplo, en Kotlin para Android y en Swift para iOS.
* Tiempo de desarrollo superior. De 4 a 6 meses.

Aplicación híbrida:

Las aplicaciones híbridas o multiplataforma combinan elementos de las aplicaciones nativas y las aplicaciones web. Estas aplicaciones se desarrollan utilizando tecnologías web como HTML, CSS y JavaScript, pero se empaquetan en un formato que puede ser instalado en un dispositivo móvil como cualquier otra aplicación nativa. Por tanto, podemos obtener una aplicación para varias plataformas con un único desarrollo.

Ventajas:

* Menor coste gracias al uso de lenguajes de programación más conocidos, con una mayor disponibilidad de profesionales en el mercado.
* Carácter multiplataforma, con una sola línea de desarrollo.
* Acceso a algunas funcionalidades del móvil.
* Reducción de los tiempos de desarrollo a 3 meses.
* Se pueden subir a los markets de aplicaciones (App Store y Google Play), obteniendo así las ventajas que ello supone, como la [opción de monetizar cobrando por descarga](https://emma.io/blog/como-monetizar-app/) o la visibilidad y la accesibilidad.

Desventajas:

* Rendimiento inferior a una aplicación nativa. Suelen tener un tamaño considerable y, además, ser más lentas.
* El acceso a las funciones del dispositivo es limitado.

Web App:

En este caso, las web app o aplicaciones web en realidad son webs especialmente diseñadas para navegadores móviles. A diferencia de las aplicaciones nativas o híbridas, no necesitan ser descargadas, ya que se accede a ellas desde un navegador web.

Emplea las mismas tecnologías de desarrollo que una web, como sería HTML, CSS o JavaScript. Así, estaríamos hablando de una web con apariencia de aplicación, por lo que presentaría sus mismas limitaciones.

Ventajas:

* Carácter multiplataforma, con una sola línea de desarrollo.
* Fácil desarrollo, ya que se emplean tecnologías ampliamente conocidas.
* Tiempo y coste de desarrollo bajo.

Desventajas:

* Acceso limitado a las funciones del dispositivo.
* No se pueden subir a las tiendas de aplicaciones.
* Diferentes experiencias de usuario en función del navegador utilizado.
* Necesidad de contar con conexión a Internet incluso si se cuenta con un modo pensado para ello. Esto es necesario para acceder a las posibles actualizaciones o para entrar por primera vez.

Aplicaciones Progresivas (PWA):

Las aplicaciones progresivas son un reciente avance de las Web Apps. Al igual que las web apps, son webs diseñadas para móviles, pero esta vez, sí pueden ser descargadas en el móvil como una aplicación más, aunque no es necesario para que ofrezcan un comportamiento similar al de una aplicación nativa a través del navegador.

Las PWA adoptan un comportamiento más propio de aplicaciones nativas que de web, como el funcionamiento sin Internet, un mayor rendimiento o su funcionamiento en segundo plano. Sin embargo, como desventaja, seguimos contando con la imposibilidad de subirlas a los markets de aplicaciones.

Herramientas de software:

* PowerPoint: **Microsoft PowerPoint** es un software de presentación desarrollado por Microsoft, ampliamente utilizado en contextos educativos, empresariales y profesionales para diseñar y mostrar contenido visual de forma estructurada.
* Word: **Microsoft Word** es un procesador de texto desarrollado por Microsoft, ampliamente utilizado para la creación, edición, formato y gestión de documentos digitales.
* Visual Studio 2022: Herramienta eficaz para desarrolladores que puede usar para completar todo el ciclo de desarrollo en un solo lugar. Es un entorno de desarrollo integrado (IDE) completo que se puede usar para escribir, editar, depurar y compilar código.

Características de Visual Studio 2022:

* + Instalador basado en cargas de trabajo: instale solo lo que necesita.
  + Herramientas y características de codificación eficaces: todo lo que necesita para compilar sus aplicaciones en un solo lugar
  + Compatibilidad con varios lenguajes: código en C++, C#, JavaScript, TypeScript, Python.
  + Desarrollo multiplataforma: compilación de aplicaciones para cualquier plataforma
  + Integración del control de versiones: colaboración en el código con compañeros de equipo
  + Desarrollo asistido por ia: escritura de código de forma más eficaz con ayuda de inteligencia artificial.

Lenguaje de programación:

C#: Es un lenguaje de programación moderno, orientado a objetos y de propósito general, desarrollado por Microsoft como parte de su plataforma .NET. Su diseño busca combinar la expresividad de lenguajes como C++ con la simplicidad y seguridad de lenguajes como Java, ofreciendo una sintaxis clara y un modelo de programación robusto.

* + Orientación a objetos: C# permite la creación de aplicaciones modulares y reutilizables mediante clases, herencia y polimorfismo.
  + Tipado fuerte y seguro: El lenguaje implementa un sistema de tipos que ayuda a detectar errores en tiempo de compilación, mejorando la fiabilidad del código.
  + Manejo automático de memoria: A través del recolector de basura (garbage collector), C# gestiona la asignación y liberación de memoria, reduciendo errores comunes como fugas de memoria.
  + Interoperabilidad: C# puede interactuar con otros lenguajes y tecnologías dentro del ecosistema .NET, facilitando la integración de componentes diversos.
  + Soporte para programación asincrónica: Con palabras clave como async y await, C# simplifica la escritura de código asincrónico, esencial para aplicaciones modernas que requieren operaciones no bloqueantes.
  + LINQ (Language Integrated Query): Permite realizar consultas a colecciones de datos de manera declarativa, integrando capacidades de consulta directamente en el lenguaje.
  + Estas características hacen de C# una opción popular para el desarrollo de aplicaciones de escritorio, web y móviles, especialmente cuando se utiliza en conjunto con herramientas como Xamarin y Visual Studio.

Gestor de base de datos:

SQLite: es una base de datos relacional de código abierto que se destaca por su simplicidad y eficiencia.

A diferencia de los sistemas de gestión de bases de datos tradicionales que requieren un servidor separado, SQLite funciona como una biblioteca que se integra directamente en la aplicación.

Características de SQLite:

* + Soporta consultas que van desde las más sencillas a las más difíciles del lenguaje SQL.
  + Gran velocidad, ya que ofrece opciones como su uso enteramente en memoria.
  + No necesita el soporte de un servidor.
  + Implementa un archivo para el esquema completo de una determinada base de datos, lo que evita preocupaciones de seguridad.
  + Ofrece simpleza, rapidez y [usabilidad](https://keepcoding.io/blog/que-es-la-usabilidad-web/), con lo que contribuye a mantener un desarrollo amigable.

Plataforma de desarrollo de software:

Xamarin: Plataforma de desarrollo móvil que permite a los desarrolladores crear aplicaciones nativas móviles para iOS, Android y Windows mediante un có[digo.NET](http://digo.net).

Características:

* + Las aplicaciones desarrolladas con Xamarin son 100% nativas.
  + El código de programación se comparte entre todas las plataformas.
  + Proporciona un acceso total a la API de cada plataforma.
  + Código Open Source y en constante innovación.
  + Sólida biblioteca de clases y documentación.
  + Permite utilizar código existente programado en otra plataforma.
  + Las interfaces de usuario se pueden desarrollar mediante XAML.

Lenguaje de marcado para el diseño de interfaces:

XAML: Es un lenguaje de marcado basado en XML desarrollado por Microsoft, utilizado principalmente para definir interfaces de usuario en aplicaciones desarrolladas con tecnologías .NET, como WPF, UWP y Xamarin.Forms. Este lenguaje permite a los desarrolladores separar la lógica de presentación del código de negocio, mejorando la organización y mantenibilidad del proyecto. En Xamarin.Forms, XAML se emplea para diseñar pantallas de manera declarativa, facilitando la creación de elementos visuales como botones, listas, formularios, y su disposición en la interfaz gráfica.

Además, XAML permite el uso de enlaces de datos (data binding), estilos, y plantillas que mejoran la experiencia del usuario y agilizan el desarrollo. Su integración con C# hace posible que los desarrolladores trabajen simultáneamente en la parte visual y en la funcionalidad de la aplicación móvil de forma ordenada y coherente.

Patrón de diseño:

MVVM (Model-View-ViewModel): Es el patrón de diseño más utilizado en aplicaciones Xamarin.Forms. Ayuda a separar la lógica de negocio de la interfaz de usuario. El patrón MVVM es una arquitectura que separa la representación visual de la lógica de negocio. Consiste en tres componentes: el Modelo (Model), que representa los datos; la Vista (View), que representa la interfaz de usuario; y el ViewModel, que actúa como intermediario entre ambos. MVVM facilita la reutilización del código, las pruebas unitarias y el mantenimiento de las aplicaciones móviles.

Controlador de versiones:

GitHub y el Control de Versiones: GitHub es una plataforma de desarrollo colaborativo basada en la web que utiliza el sistema de control de versiones Git. Esta herramienta permite a los desarrolladores gestionar el historial de cambios en el código fuente, colaborar en proyectos en equipo, y mantener una trazabilidad clara del progreso del desarrollo. El control de versiones es fundamental en cualquier proceso de desarrollo de software, ya que permite:

* Registrar cada modificación del código.
* Restaurar versiones anteriores en caso de errores.
* Facilitar la colaboración entre múltiples desarrolladores.
  + Crear ramas (branches) para desarrollar nuevas funcionalidades sin afectar la versión estable del proyecto.

GitHub no solo proporciona alojamiento para los repositorios Git, sino también funcionalidades adicionales como:

* + Revisión de código mediante pull requests.
  + Seguimiento de tareas con issues.
  + Integración con herramientas de desarrollo como Visual Studio y CI/CD.

Framework:

Microsoft .NET:.NET es un [framework](https://es.wikipedia.org/wiki/Framework) de [Microsoft](https://es.wikipedia.org/wiki/Microsoft) que hace un énfasis en la [transparencia](https://es.wikipedia.org/wiki/Transparencia_de_red) de [redes](https://es.wikipedia.org/wiki/Red_de_computadoras), con independencia de plataforma de [hardware](https://es.wikipedia.org/wiki/Hardware) y que permite un rápido desarrollo de [aplicaciones](https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_inform%C3%A1tica).

Componentes:

Los principales componentes del marco de trabajo son:

* + El conjunto de [lenguajes de programación](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n).
  + La biblioteca de clases base o BCL.
  + El entorno común de ejecución para lenguajes, o CLR (Common Language Runtime) por sus siglas en inglés.

CLI:

Debido a la publicación de la norma para la infraestructura común de lenguajes (*CLI* por sus siglas en inglés), el desarrollo de lenguajes se facilita, por lo que el marco de trabajo .NET soporta ya más de 20 lenguajes de programación y es posible desarrollar cualquiera de los tipos de aplicaciones soportados en la plataforma con cualquiera de ellos, lo que elimina las diferencias que existían entre lo que era posible hacer con uno u otro lenguaje.

Algunos de los lenguajes desarrollados para el marco de trabajo .NET son: [C#](https://es.wikipedia.org/wiki/C_Sharp), [Visual Basic .NET](https://es.wikipedia.org/wiki/Visual_Basic_.NET), [Delphi](https://es.wikipedia.org/wiki/Embarcadero_Delphi) ([Object Pascal](https://es.wikipedia.org/wiki/Object_Pascal)), [C++](https://es.wikipedia.org/wiki/C%2B%2B), [F#](https://es.wikipedia.org/wiki/F_Sharp), [J#](https://es.wikipedia.org/wiki/J_Sharp), [Perl](https://es.wikipedia.org/wiki/Perl), [Python](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Python), [Fortran](https://es.wikipedia.org/wiki/Fortran), [Prolog](https://es.wikipedia.org/wiki/Prolog) (existen al menos dos implementaciones, el [P#](https://es.wikipedia.org/wiki/P_Sharp)[[1]](https://es.wikipedia.org/wiki/Microsoft_.NET#cite_note-1)​ y el Prolog.NET[[2]](https://es.wikipedia.org/wiki/Microsoft_.NET#cite_note-2)​), [Cobol](https://es.wikipedia.org/wiki/Cobol) y [PowerBuilder](https://es.wikipedia.org/wiki/PowerBuilder).

CLR:

El CLR es el verdadero núcleo del framework de .NET, entorno de ejecución en el que se cargan las aplicaciones desarrolladas en los distintos lenguajes, ampliando el conjunto de servicios del sistema operativo (W2k y W2003). Permite integrar proyectos en distintos lenguajes soportados por la plataforma .Net, como C++, Visual Basic, C#, entre otros.

Características de CLR:

Es el encargado de proveer lo que se llama código administrado, es decir, un entorno que provee servicios automáticos al código que se ejecuta.

Los servicios son variados:

* Cargador de clases: permite cargar en memoria las clases.
* Compilador MSIL a nativo: transforma código intermedio de alto nivel independiente del hardware que lo ejecuta a código de máquina propio del dispositivo que lo ejecuta.
  + Administrador de código: coordina toda la operación de los distintos subsistemas del Common Language Runtime.
  + Recolector de basura: elimina automáticamente de memoria objetos no utilizados.
  + Motor de seguridad: administra la seguridad del código que se ejecuta.
  + Motor de [depuración](https://es.wikipedia.org/wiki/Depuraci%C3%B3n_de_programas): permite hacer un seguimiento de la ejecución del código aun cuando se utilicen lenguajes distintos.
  + Verificador de tipos: controla que las variables de la aplicación usen el área de memoria que tienen asignado.
  + Administrador de excepciones: maneja los errores que se producen durante la ejecución del código.
  + Soporte de [multiproceso](https://es.wikipedia.org/wiki/Multiproceso) ([hilos](https://es.wikipedia.org/wiki/Hilo_(inform%C3%A1tica))): permite desarrollar aplicaciones que ejecuten código en forma paralela.
  + Empaquetador de COM: coordina la comunicación con los componentes COM para que puedan ser usados por el .NET Framework.
  + Biblioteca de Clases Base que incluye soporte para muchas funcionalidades comunes en las aplicaciones.

METODOLOGÍA

La metodología representa el conjunto de técnicas y procesos organizados que permiten desarrollar un proyecto de forma eficiente, estructurada y sistemática, con el objetivo de cumplir las metas planteadas. Para el presente proyecto, se adopta la metodología tradicional en cascada, la cual se basa en una secuencia lineal de fases que deben completarse una tras otra, permitiendo una planificación clara y un control riguroso del avance del desarrollo.

Esta metodología, es adecuada, dado que, la aplicación móvil de préstamo de laptops dentro del edificio no requiere cambios constantes ni interacción continua con los usuarios durante el proceso de desarrollo, lo cual encaja con el enfoque lineal y estructurado que propone el modelo en cascada. Cada etapa del proyecto debe finalizar por completo antes de iniciar la siguiente, lo que permite una mayor claridad en los requerimientos, facilita la documentación y mejora la calidad del producto final.

El desarrollo de la aplicación móvil se realiza a través de las siguientes etapas definidas en la metodología en cascada:

Recolección de Requisitos:

En esta primera fase se identifican y documentan las necesidades del personal administrativo respecto al control de préstamo de laptops. Se detallan las funcionalidades requeridas, como el registro de usuarios, la gestión de equipos disponibles y el control del historial de préstamos. Se establecen restricciones técnicas, como la necesidad de funcionar sin conexión a internet y el uso de una base de datos local en el dispositivo móvil.

Análisis:

Con base en los requisitos recopilados, se realiza un análisis detallado para definir la estructura funcional de la aplicación móvil. Se diseñan los modelos de datos, se determinan las relaciones entre entidades como usuarios, préstamos y equipos, y se establece cómo será el flujo lógico de la aplicación móvil. También se consideran los posibles riesgos, como la pérdida de datos o errores de registro, y se proponen mecanismos de prevención, como respaldos manuales.

Diseño de la aplicación móvil:

En esta fase se define la arquitectura general de la aplicación móvil. Se diseña la interfaz de usuario con base en criterios de usabilidad y se estructura la base de datos local, considerando tecnologías como SQLite o Realm (modo local). Se especifican los módulos de la aplicación móvil, como inicio de sesión, gestión de préstamos y consultas de historial, además del diseño de pantallas, botones y navegación.

Implementación:

Una vez definido el diseño, se procede a la codificación de la aplicación móvil. Se desarrollan los distintos módulos de acuerdo con los requerimientos y el diseño planteado. Se integran las funciones CRUD (crear, leer, actualizar, eliminar) sobre la base de datos local, asegurando que la aplicación móvil funcione completamente sin conexión a internet.

Pruebas:

Durante esta etapa se realizan pruebas funcionales y de usabilidad para verificar que cada componente de la aplicación móvil opere correctamente. Se simulan diferentes escenarios de uso, como registrar préstamos simultáneos, modificar datos de usuarios o consultar historial, para identificar y corregir posibles errores. También se validan los mecanismos de respaldo y recuperación de datos en el entorno local.

Implementación y Despliegue:

Finalizadas las pruebas, se procede a la instalación de la aplicación móvil en los dispositivos móviles destinados al personal encargado del préstamo de laptops. Se capacita brevemente al personal en el uso de la aplicación móvil y se documentan las funciones principales.

Mantenimiento:

Después del despliegue, se planifican actividades periódicas de mantenimiento. Esto incluye mejoras en la interfaz, corrección de errores que puedan surgir en el uso diario y actualizaciones para garantizar el buen funcionamiento de la aplicación móvil a lo largo del tiempo.

Ventajas del modelo en cascada aplicado al proyecto:

Este enfoque permite una definición clara de objetivos desde el inicio, facilita el seguimiento del avance mediante documentación constante y garantiza una estructura lógica en el desarrollo de la aplicación móvil. Además, al tratarse de un proyecto con requerimientos bien definidos y poco cambiantes, reduce los riesgos de desviación del alcance y facilita el cumplimiento de los plazos establecidos.

Desventajas consideradas:

El principal reto del modelo en cascada es su rigidez ante cambios imprevistos. Cualquier modificación importante en los requisitos una vez que se ha superado la etapa de análisis puede implicar rehacer varias fases del proceso. Por ello, es fundamental que los requerimientos se definan con claridad y precisión desde el inicio del proyecto. También puede representar un mayor consumo de tiempo si se presentan errores en fases avanzadas que no fueron detectados oportunamente.

CAPÍTULO 1 RECOLECCIÓN Y REFINAMIENTO DE REQUISITOS

* 1. Necesidades del cliente

Una necesidad del cliente consiste en aquello que motiva a un cliente a comprar un producto o servicio. La necesidad puede ser conocida (es decir, el cliente puede expresarla con palabras) o desconocida, y es el factor que determina qué solución compra el cliente.

Necesidad: En la Universidad Tecnológica de Tecámac, particularmente en la división de Tecnologías de la Información y Comunicación (TIC), es frecuente el préstamo de laptops a los estudiantes para que puedan desarrollar sus actividades académicas dentro de la institución. Esta dinámica requiere una gestión precisa y actualizada que garantice la disponibilidad, el uso adecuado y el seguimiento de los equipos prestados. Por lo cual el personal encargado necesita una herramienta tecnológica que le permita controlar este proceso de forma eficiente y organizada.

Problema: El personal encargado del préstamo de laptops en la Universidad Tecnológica de Tecámac enfrenta diversas dificultades en el registro y control de esta actividad. Entre los principales problemas se pueden encontrar la pérdida de información, la falta de seguimiento de los préstamos de los equipos y la ausencia de un control adecuado sobre los usuarios. Estos problemas se deben en gran parte a que el proceso se realiza de forma manual, utilizando hojas de papel para manejar el registro, lo cual puede resultar ineficiente y poco práctico para una gestión adecuada.

Solución: El proyecto ControlPC propone el desarrollo de una aplicación móvil que funcione sin conexión a internet, capaz de almacenar y gestionar localmente toda la información relacionada con los equipos, los usuarios y los préstamos realizados. Esta aplicación permitirá al personal administrativo registrar, consultar y eliminar datos sobre los préstamos, optimizando el proceso de control del préstamo de laptops permitiendo reducir errores, mediante una interfaz amigable y funcional.

* 1. Identificación y definición de roles de usuario

Se identifican diferentes roles a los usuarios y luego se asignan tareas específicas a los distintos usuarios que interaccionen con la aplicación móvil.

Dado que la aplicación móvil ControlPC está diseñada para ser utilizada en un entorno educativo específico y gestionada únicamente por el personal encargado del préstamo de equipos de cómputo, se ha definido un único rol de usuario: el usuario final, quien interactuará completamente con la aplicación móvil. Este rol corresponde al laboratorista o administrador de los equipos de cómputo.

Este usuario será responsable de:

* Registrar los préstamos de laptops a estudiantes.
* Consultar la disponibilidad de los equipos.
* Reportar fallas o daños en los dispositivos.
* Llevar el historial de mantenimiento y cobros.
* Controlar el tiempo de uso de cada equipo.
  1. Requerimientos funcionales

Los requerimientos funcionales de un sistema son aquellos que describen cualquier actividad que este deba realizar, en otras palabras, el comportamiento o función particular de un sistema o software cuando se cumplen ciertas condiciones.

RF-01. Registro de usuario.

El sistema permite únicamente al laboratorista o encargado registrar nuevos usuarios (como alumnos y docentes) que pueden solicitar préstamos. Estos son dados de alta manualmente por el encargado, quien ingresa sus datos personales y académicos necesarios.

RF-02. Gestión de inventario de equipos.

La plataforma móvil permite al administrador registrar, modificar, eliminar y consultar los equipos disponibles para préstamo, incluyendo datos como nombre, tipo, número de serie, estado y ubicación.

RF-03. Solicitud y autorización de préstamos.

Los usuarios pueden solicitar el préstamo de un equipo, indicando la fecha y hora de inicio y fin.

RF-04. Registro de entregas y devoluciones.

La solución móvil lleva un control del estado de cada préstamo, registrando las fechas de entrega y devolución, y actualizando el estado del equipo.

RF-05. Generación de reportes.

La aplicación genera reportes periódicos (semanal y mensual) sobre el uso de los equipos, historial de préstamos por usuario, equipos más solicitados, préstamos vencidos y daños reportados.

* 1. Requerimientos no funcionales

Los requisitos no funcionales (NFR) son las restricciones impuestas a un sistema que definen sus atributos de calidad. Por lo general, se denotan con adjetivos como seguridad, rendimiento y escalabilidad. Los requisitos no funcionales son importantes porque ayudan a garantizar que el sistema satisfaga las necesidades del usuario.

RNF-01. Usabilidad e interfaz intuitiva.

La interfaz de la aplicación móvil es clara y fácil de usar, permitiendo que cualquier encargado opere el sistema sin requerir conocimientos técnicos avanzados. La navegación es coherente desde cualquier pantalla o módulo.

RNF-02. Mantenimiento y escalabilidad.

La herramienta está diseñada de manera modular, facilitando futuras actualizaciones y mantenimiento sin afectar las funciones actuales. Además, permite agregar más equipos o usuarios sin necesidad de rediseñar toda la estructura.

RNF-03. Rendimiento y eficiencia.

La aplicación móvil se carga de manera rápida en menos de 5 segundos desde su apertura y responde a las acciones del usuario en menos de 3 segundos, incluso en dispositivos de gama media.

RNF-04. Disponibilidad sin conexión a internet.

Dado que la base de datos que se usa es local, la aplicación móvil funciona completamente sin requerir conexión a internet, permitiendo a los laboratoristas acceder y registrar información en todo momento.

RNF-05. Seguridad de Datos Locales.

Todos los datos almacenados localmente están protegidos mediante cifrado (por ejemplo, AES-256) para garantizar la confidencialidad de la información en caso de pérdida o robo del dispositivo.

* 1. Requerimientos de hardware

Los requerimientos de hardware son las especificaciones técnicas básicas que debe cumplir un dispositivo para poder instalar y ejecutar correctamente una aplicación o sistema. Estos requisitos aseguran que el software funcione de manera estable, sin errores ni interrupciones.

Requerimientos del dispositivo Android (usuario final)

Requerimientos mínimos:

* Procesador: 1.4 GHz (Quad-core recomendado)
* Memoria RAM: 2 GB
* Almacenamiento interno disponible: 2 GB libres
* Pantalla: 5 pulgadas con resolución mínima de 720p

Requerimientos recomendados:

* Procesador: Octa-core 2.0 GHz o superior
* Memoria RAM: 4 GB
* Almacenamiento interno disponible: 5 GB
* Pantalla: 6 pulgadas o más
  1. Requerimientos de software

Se describe el software que es necesario para ejecutar el proyecto.

* Sistema operativo: Android 7.0 (Nougat) o superior:

Se recomienda el sistema operativo Android 7.0 Nougat porque introduce mejoras significativas en multitarea, notificaciones y seguridad. Entre sus características destacan:

* + Multiventana: permite ejecutar dos aplicaciones simultáneamente en pantalla dividida.
  + Respuestas directas: posibilidad de responder mensajes directamente desde las notificaciones.
  + Doze mejorado: optimización del ahorro de batería incluso cuando el dispositivo está en movimiento.
* Espacio libre mínimo recomendado: 150 MB

Este espacio permite no solo la instalación inicial de la aplicación móvil, sino también una gestión fluida de los datos conforme se utilice el sistema.

Este espacio contempla:

* La instalación de la aplicación y sus librerías.
* El almacenamiento de datos generados por el usuario final, como registros de préstamos, equipos, reportes y usuarios.
* El funcionamiento óptimo de la aplicación, incluyendo el manejo de archivos temporales y caché.
* Permisos del sistema

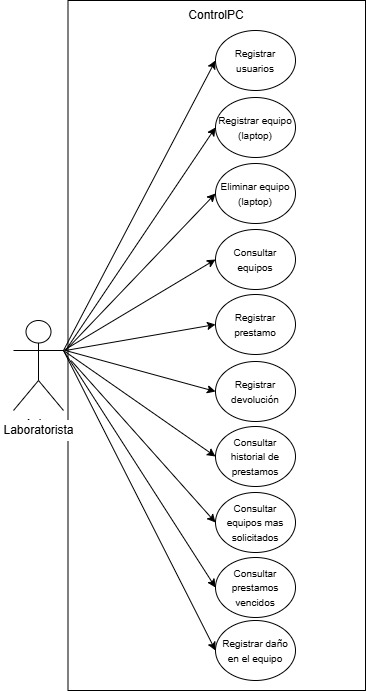
Para operar correctamente, la aplicación requiere:

* Acceso al almacenamiento interno: para guardar la base de datos SQLite dentro del espacio asignado a la aplicación.
  1. Diagrama general de casos de uso

Un diagrama general de casos de uso es una representación gráfica que forma parte del lenguaje de modelado UML. Su propósito es mostrar de manera clara las funcionalidades principales de un sistema desde el punto de vista del usuario final, también conocidos como actores.

Este tipo de diagrama ilustra qué hace el sistema, no cómo lo hace. Describe las interacciones entre los actores y los casos de uso, es decir, las acciones o servicios que el sistema debe ofrecer.

En la figura 1.7 se muestra el diagrama de casos de uso del sistema "ControlPC", en el cual el único actor es el Laboratorista, quien interactúa con el sistema para realizar diversas funciones relacionadas con la gestión de equipos de cómputo. Entre las acciones que puede ejecutar se encuentran: registrar usuarios, registrar y eliminar equipos (laptops), consultar el inventario de equipos, registrar préstamos y devoluciones, consultar el historial de préstamos, verificar cuáles son los equipos más solicitados, consultar los préstamos vencidos y registrar daños en los equipos. Cada una de estas funciones está representada como un caso de uso vinculado al actor, indicando que el Laboratorista tiene acceso a todas estas operaciones dentro del sistema.



**Figura 1.7** Diagrama general de casos de uso de ControlPC.

* 1. Especificación de los casos de uso principales

La especificación de casos de uso principales es una descripción detallada de las funciones más importantes que ofrece un sistema desde el punto de vista del usuario. Estos casos de uso representan las acciones fundamentales que deben estar correctamente definidas y ejecutadas para que el sistema cumpla con su propósito principal.

A través de esta especificación, se describen de manera clara y estructurada los requisitos funcionales clave, incluyendo los actores involucrados, los pasos a seguir, las condiciones necesarias y los posibles flujos alternativos. De esta forma, se proporciona una guía precisa para el diseño, desarrollo y prueba del sistema.

En la figura 1.8 se detalla el caso de uso de registro de usuarios en el cual su objetivo es registrar manualmente a nuevos usuarios en el sistema (como alumnos o docentes), ingresando datos personales y académicos esenciales. Esta acción garantiza que únicamente personas autorizadas puedan solicitar préstamos, y permite llevar un control adecuado de quiénes utilizan los equipos tecnológicos.

|  |  |
| --- | --- |
| Nombre: | Registro de usuario |
| Autor: | Flores Zarate Edwin |
| Fecha: | 29/05/2025 |
| Descripción: | El laboratorista utiliza la aplicación para registrar manualmente nuevos usuarios (alumnos o docentes) este proceso implica ingresar datos personales y académicos. |
| Actores: | Laboratorista |
| Precondiciones: | El laboratorista necesita tener acceso a la aplicación. |
| Flujo Normal: | 1. Iniciar la aplicación en el dispositivo móvil. 2. Acceder a la sección de registro de usuario. 3. Registrar al usuario con sus datos personales y académicos. |
| Flujo Alternativo: | 3.1 Si los datos ingresados están incompletos o inválidos, el sistema muestra error al intentar registrar el usuario. |
| Postcondiciones: | El usuario queda registrado en la aplicación. |

**Figura 1.8** Caso de uso, registro de usuario de ControlPC.

En la figura 1.8.1 se presenta el caso de uso de gestión de inventario, donde permite al laboratorista registrar, consultar, modificar o eliminar información sobre los equipos disponibles para préstamo. Este módulo garantiza un control riguroso del inventario, incluyendo datos como número de serie, marca, modelo, estado y otros detalles relevantes del equipo, lo que facilita su localización y mantenimiento.

|  |  |
| --- | --- |
| Nombre: | Gestión de inventario |
| Autor: | Flores Zarate Edwin |
| Fecha: | 29/05/2025 |
| Descripción: | El administrador utiliza la aplicación para registrar manualmente un nuevo equipo disponible para préstamo, ingresando datos como nombre, tipo, número de serie, etc. |
| Actores: | Laboratorista |
| Precondiciones: | Tener acceso a la aplicación. |
| Flujo Normal: | 1.Iniciar la aplicación en el dispositivo móvil.  2.Seleccionar la opción “registrar un nuevo equipo”.  3.Ingresar los datos del equipo que se quiere registrar. |
| Flujo Alternativo: | 3.1 Si los datos ingresados están incompletos o inválidos, la aplicación muestra error al intentar registrar un nuevo equipo. |
| Postcondiciones: | El nuevo equipo queda registrado y disponible para préstamo. |

**Figura 1.8.1** Caso de uso, gestión de inventario de ControlPC.

En la figura 1.8.2 se describe el caso de uso de registro de solicitud de préstamo este caso de uso que facilita al laboratorista la creación de un nuevo registro de préstamo a favor de un usuario previamente dado de alta. Este proceso implica seleccionar al usuario, el equipo disponible, y definir la fecha y horario de préstamo. El sistema valida automáticamente la disponibilidad del equipo antes de confirmar el registro, con el fin de evitar duplicidades o conflictos de uso.

|  |  |
| --- | --- |
| Nombre: | Registrar solicitud de préstamo |
| Autor: | Florez Reynoso Martha |
| Fecha: | 28 de mayo de 2025 |
| Descripción: | Este caso de uso permite al laboratorista registrar una solicitud de préstamo realizada por un usuario (alumno o docente), especificando fecha y hora del préstamo. |
| Actores: | Laboratorista |
| Precondiciones: | El usuario debe estar previamente registrado en la aplicación. Debe haber al menos un equipo disponible para préstamo. |
| Flujo Normal: | 1. El laboratorista selecciona “Registrar préstamo”.  2. Elige al usuario solicitante.  3. Selecciona el equipo a prestar.  4. Ingresa la fecha, hora de inicio y fin del préstamo.  5. La aplicación valida disponibilidad.  6. La aplicación guarda el préstamo.  7. Se muestra confirmación. |
| Flujo Alternativo: | 2.1 Si el usuario no está registrado, no puede continuar el proceso.  3.1 Si no hay equipos disponibles, la aplicación muestra un mensaje de error. |
| Postcondiciones: | El préstamo queda registrado en la base de datos y el equipo cambia su estado a “Prestado”. |

**Figura 1.8.2** Caso de uso, registro de solicitud de préstamo de ControlPC.

En la figura 1.8.3 se presenta el caso de uso de registro de entrega y devolución de equipo. Permite registrar la devolución de un equipo previamente prestado. Involucra verificar el estado físico del equipo, registrar la hora de entrega y actualizar el estado del préstamo como finalizado. Además, el equipo se reintegra al inventario como disponible, asegurando un historial preciso del uso de los dispositivos.

|  |  |
| --- | --- |
| Nombre: | Registrar entrega y devolución de equipo |
| Autor: | Riaño Ishiwara Yoshiro Pablo Armando |
| Fecha: | 28 de mayo de 2025 |
| Descripción: | Este caso de uso permite al laboratorista registrar la devolución de un equipo previamente prestado, actualizando el estado del préstamo y del equipo. |
| Actores: | Laboratorista |
| Precondiciones: | Debe existir un préstamo activo previamente registrado. El equipo debe estar marcado como "prestado" en la aplicación. |
| Flujo Normal: | 1. El laboratorista accede al módulo de “Devoluciones”.  2. Selecciona el préstamo activo a finalizar.  3. Revisa el estado del equipo.  4. Registra la fecha y hora de devolución.  5. La aplicación actualiza el estado del préstamo a “finalizado”.  6. El equipo vuelve a estar disponible. |
| Flujo Alternativo: | 5.1 Si el préstamo no existe o ya está finalizado, la aplicación mostrará un mensaje de error. |
| Postcondiciones: | El equipo regresa al inventario como “disponible” y el préstamo queda cerrado con su historial actualizado. |

**Figura 1.8.3** Caso de uso, registrar entrega y devolución de equipo de ControlPC.

En la figura 1.8.4 se muestra el caso de uso de generación de reportes, este caso de uso permite al laboratorista visualizar y analizar información consolidada sobre el uso de los equipos tecnológicos. Los reportes pueden incluir métricas como historial de préstamos, equipos más solicitados, registros vencidos y fallos reportados. Esta funcionalidad facilita la toma de decisiones, la planeación del mantenimiento y el seguimiento del uso del inventario a lo largo del tiempo.

|  |  |
| --- | --- |
| Nombre: | Generación de reportes. |
| Autor: | Gómez Fuentes Kevyn Axel |
| Fecha: | 28/05/2025 |
| Descripción: | La aplicación deberá generar reportes periódicos sobre el uso de los equipos, historial de préstamos por usuario, equipos más solicitados, prestamos vencidos y daños reportados. |
| Actores: | Laboratorista |
| Precondiciones: | Deberá contar con datos almacenados sobre préstamos, uso de equipos y reportes de daños |
| Flujo Normal: | 1.El usuario accede al apartado de préstamos.  2.Ingresa a la sección de reportes.  3.Visualiza los reportes. |
| Flujo Alternativo: | 3.1 Si no existen datos suficientes para generar los reportes, la aplicación mostrará un mensaje indicando que no hay información disponible. |
| Postcondiciones: | El usuario podrá visualizar el reporte y utilizar la información obtenida para su análisis |

**Figura 1.8.4** Caso de uso, Generación de reporte.

# **CAPÍTULO 2 DISEÑO Y DESARROLLO DEL PROYECTO**

## Diseño de la base de datos

Una base de datos relacional es una colección de información que organiza datos en relaciones predefinidas, en la que los datos se almacenan en una o más tablas (o "relaciones") de columnas y filas, lo que facilita la visualización y la comprensión de cómo se relacionan las diferentes estructuras de datos entre sí. Las relaciones son conexiones lógicas entre las diferentes tablas y se establecen a partir de la interacción entre ellas.

En la figura 2.1 se presenta el modelo entidad-relación para gestionar préstamos de laptops. El sistema está compuesto por cinco entidades principales: Usuario, Laptop, Préstamo, Historial de Mantenimiento y Reporte de Falla. La entidad Usuario tiene como llave primaria la matrícula e incluye datos personales como nombre, apellidos, rol y contacto. Laptop, por su parte, utiliza id\_equipo como llave primaria y almacena detalles técnicos (marca, modelo, número de serie, etc.). Los Préstamos vinculan a usuarios y equipos mediante las llaves foráneas id\_usuario e id\_equipo, registrando fechas de inicio, fin y duración. Adicionalmente, el Historial de Mantenimiento y el Reporte de Falla dependen de la entidad Laptop, documentando acciones de mantenimiento y fallos reportados, respectivamente. Este diseño permite un control estructurado y eficiente de los recursos tecnológicos.

Diagrama

El contenido generado por IA puede ser incorrecto.

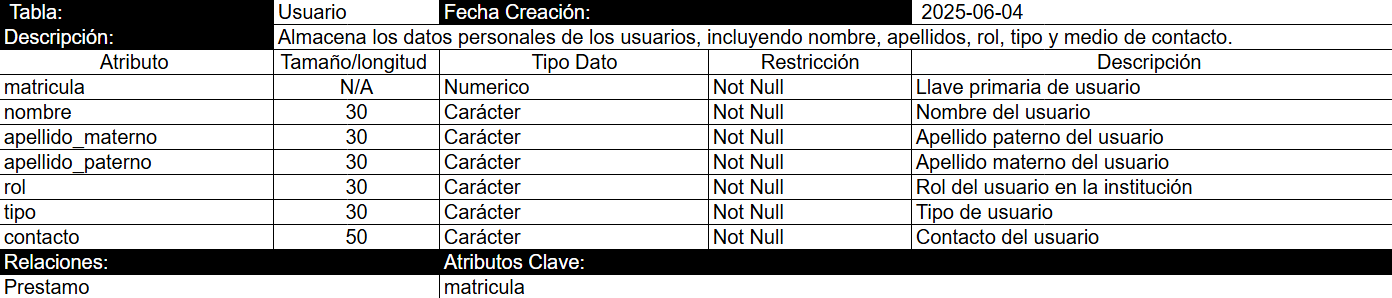
**Figura 2.1** Modelo relacional de la base de datos

## Diccionario de datos

Un diccionario de datos mapea los elementos de datos, aclara las relaciones y garantiza una interpretación homogénea de las bases de datos.

Los diccionarios de datos suelen implementarse como repositorio de metadatos de descripciones que aclaran el significado, las características y las relaciones de los elementos de datos dentro de una base de datos. Describe los objetos de datos de un modelo de datos que el usuario final puede consultar.

En la figura 2.2 se muestra la tabla usuario que se conforma por distintos los atributos que requiere llenar la persona que de quiera solicitar un préstamo de una laptop. Se incluyen datos personales, escolares y un medio de contacto, cada uno de ellos incluye su tipo de dato que es cada uno de ellos, en esta tabla solo se usan de tipo carácter o numérico, también se incluye una descripción sobre cada atributo que incluye la tabla, además se indica con que tablas tiene relación y por último muestra los atributos clave, que se refiere a los datos que son necesarios de llenar y no pueden quedar vacíos, además de las llaves foráneas.



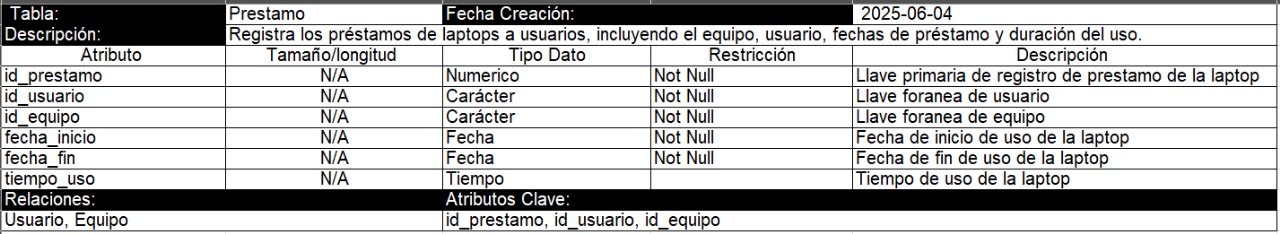
**Figura 2.2** Diccionario de datos, Tabla de Usuario, ControlPC.

En la figura 2.2.1 se muestra la tabla laptop la cual está conformada por sus atributos, en este caso acerca de la información sobre la laptop. Cada atributo incluye su tipo de dato en esta tabla se usan el tipo de dato carácter, este tipo de dato incluye su longitud, también se hace uso del tipo de dato numérico y fecha, cada dato muestra la restricción que requiere. Se incluye una descripción de la función que realiza cada atributo. Incluye la relación que tiene con otras tablas y por ultimo los atributos clave que se refiere a los atributos que no pueden quedar vacíos y por lo tanto son importantes en los cuales se incluye la información del equipo.



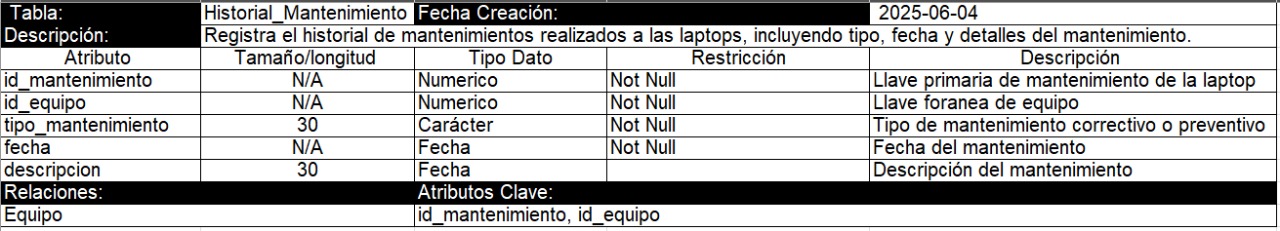
**Figura 2.2.1** Diccionario de datos, Tabla de Laptop, ControlPC.

En la figura 2.2.2 se muestra la tabla préstamo, esta está compuesta por varios atributos esenciales para registrar las laptops que se prestan durante las actividades académicas de estudiantes y docentes. Entre estos atributos se incluyen referencias directas a los usuarios “id\_usuario” y a los equipos “id\_equipo”, así como el registro de la fecha de inicio y fin del préstamo, y el total de tiempo de uso. Esta información permite llevar un control preciso y un seguimiento detallado del uso de cada equipo.



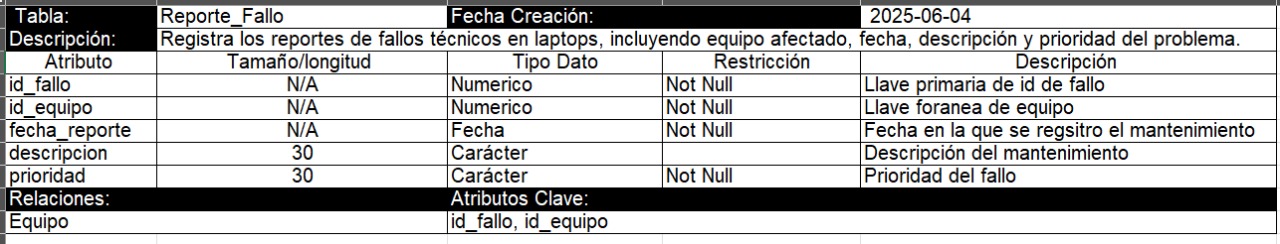
**Figura 2.2.2** Diccionario de datos, Tabla de Préstamo, ControlPC.

En la figura 2.2.3 se muestra la tabla de historial de mantenimiento, que está compuesta por varios atributos esenciales para registrar los mantenimientos realizados a las laptops con las que cuenta la institución. Entre estos atributos se incluyen el identificador del mantenimiento “id\_mantenimiento”, la referencia al equipo correspondiente “id\_equipo”, el tipo de mantenimiento que será realizado, la fecha en que se realizó y una breve descripción del procedimiento llevado a cabo facilitando el control detallado del historial de mantenimiento de cada equipo.



**Figura 2.2.3** Diccionario de datos, Tabla de Historial de Mantenimiento, ControlPC.

En la figura 2.2.4 se muestra la tabla reporte de fallo, compuesta por varios atributos esenciales para registrar los fallos técnicos presentados en las laptops de la institución. Entre estos atributos se incluyen el identificador del fallo “id\_fallo”, incluyendo referencias directas del equipo afectado “id\_equipo”, la fecha en que se reportó el problema, una descripción del fallo y su nivel de prioridad permitiendo llevar un control de los incidentes técnicos.



**Figura 2.2.4** Diccionario de datos, Tabla de Reporte de Fallo, ControlPC.

## Diseño de interfaces

Una interfaz es el punto donde las personas y los dispositivos electrónicos se comunican entre sí. Esto puede incluir hardware como pantallas, teclados y ratones. En un contexto menos tangible, una interfaz abarca las interfaces gráficas de usuario para aplicaciones y sitios web, elementos de diseño, estructuras de navegación y otros componentes interactivos que permiten a los usuarios interactuar fácilmente con las plataformas digitales.

En la figura 2.3 se muestra el diseño de la interfaz de inicio de ControlPC contiene una etiqueta de imagen centrada en la parte superior<Image>, en la parte del centro se ubican dos etiquetas de imagen <Image> de forma centrada una a lado de la otra y debajo de cada etiqueta de imagen se ubica una etiqueta de texto <Label> y debajo de estas etiquetas de texto se encuentra otra etiqueta de imagen <Image> de forma centrada e igualmente debajo de la imagen se encuentra una etiqueta de texto <Label>.

Diagrama, Forma

El contenido generado por IA puede ser incorrecto.

**Figura 2.3** Interfaz de inicio de ControlPC

En la figura 2.3.1 se presenta la interfaz de registro de usuario la cual, en la parte superior, centrada, se ubica una etiqueta de texto <Label>. Debajo de esta etiquetada, hay alineados a la izquierda, cinco etiquetas de texto <Label>, cada una acompañada debajo por un campo de entrada <Entry>. Antes de la sección final, se incluye una etiqueta de entrada adicional <Picker> debajo de su respectivo <Label>. Finalmente, en la parte inferior y centrada, se encuentran dos botones (<Button>).

Diagrama

El contenido generado por IA puede ser incorrecto.

**Figura 2.3.1** Interfaz de registro de usuario de ControlPC

En la figura 2.3.2 se muestra la interfaz de registro de laptop compuesta en la parte superior de forma centrada por una etiqueta de texto <Label>, continuando en la parte superior derecha se encuentran dos etiquetas de imagen <Image>, posteriormente alineadas a la izquierda secuencialmente se incluyen tres etiquetas de texto <Label> cada una de ellas acompañadas por una etiqueta de entrada de texto <Entry>, así mismo alineadas a la izquierda hay dos etiquetas de texto <Label>, la primera de ellas acompañada por una caja de entrada de texto <Editor>, y la otra está acompañada por un seleccionador de objetos <Picker>, ya por último en la parte inferior de forma centrada se encuentra dos botones <Button>.

Diagrama

El contenido generado por IA puede ser incorrecto.

**Figura 2.3.2** Interfaz de registro de laptop de ControlPC

En la figura 2.3.3 se visualiza la interfaz de registro de préstamo, en la parte superior de forma centrada se incluye una etiqueta de texto <Label>, debajo de esta se ubican dos etiquetas de texto <Label> alineados a la izquierda, ambas acompañadas por una etiqueta de entrada de texto <Entry>, después se incluyen otras dos etiquetas de texto <Label> las cuales en la misma fila una está alineada a la izquierda y la otra alineada a la derecha, debajo de cada una de estas dos se incluyen dos etiquetas de seleccionador de fecha <DatePicker>, por último incluye otra etiqueta de texto <Label> acompañada por una etiqueta de entrada de texto <Entry> , para finalizar se incluye dos botones <Button> de forma centrada.

Diagrama

El contenido generado por IA puede ser incorrecto.

**Figura 2.3.3** Interfaz de registro de préstamo de ControlPC.

En la figura 2.3.4 se visualiza el diseño de la interfaz de registro de mantenimiento compuesta inicialmente en la parte superior izquierda por una imagen <Image> la cual está acompañada a su derecha por una etiqueta <entry>, y debajo en la parte superior se ubican tres etiquetas de texto <Label> una centrada, y dos debajo de esta alineadas a la izquierda, debajo de esto se ubican dos etiquetas de texto <Label> alineadas a la izquierda, debajo de estas etiquetas se ubican tres etiquetas de texto <Label> la primera de ellas acompañada por un seleccionador de objetos <Picker>, la segunda de ellas se acompaña por un seleccionador de fecha <DatePicker> y la última de estas etiquetas de texto se acompaña por una caja de texto <Entry>, ya por último de forma centrada y equidistantes en la parte inferior se ubican dos botones <Button>.

Diagrama

El contenido generado por IA puede ser incorrecto.

**Figura 2.3.4** Interfaz de registro de mantenimiento de ControlPC

En la figura 2.3.5 se observa la interfaz de registro de fallo, compuesta en la parte superior izquierda por una imagen <Image>, en la misma parte superior a su derecha se ubica una etiqueta de texto <Label>, debajo de estas etiquetas se ubican tres etiquetas de texto <Label> una de forma centrada, debajo de esta dos y alineadas a la izquierda una debajo de otra, se ubican dos etiquetas de texto <Label> alineadas a la izquierda, debajo de cada una de estas etiquetas se ubican tres etiquetas de texto <Label> la primera de ellas acompañada por una entrada de texto <Entry> , la segunda de ellas se acompaña por un seleccionador de fecha <DatePicker> y la última de estas etiquetas por un seleccionador de objetos <Picker>, y ya por último de forma centrada en la parte inferior se ubican dos botones <Button>.

Diagrama

El contenido generado por IA puede ser incorrecto.

**Figura 2.3.5** Interfaz de registro de fallo de ControlPC

## Mapa de navegación

El mapa de navegación es un esquema, que se puede pensar como un árbol jerárquico, que representa la arquitectura de las páginas de un sitio web. Esta representación puede ser, como un mapa conceptual, de forma gráfica para visualizar y entender sobre el recorrido de visitante, es decir su navegación y cómo este llega a lo que está buscando.

En la figura 2.4. se visualiza el mapa de navegación ControlPC el cual presenta la navegación de la aplicación, comenzando en la pantalla de inicio con la opción de redirigir al registro de usuario, registro de equipos o registro de préstamo de laptop. Desde la interfaz de registro de usuarios, se puede acceder a una interfaz de lista de usuarios de esta lista se permite acceder a una interfaz para editar un usuario y desde esta permite regresar a la interfaz de inicio. Al acceder a la interfaz de registro de préstamo, permite navegar a la interfaz de lista de préstamos y desde esta permite acceder a la interfaz de editar préstamo y desde esta misma se puede acceder de nuevo al inicio. Cuando se accede a la interfaz de registro de laptop se puede ir directamente a la interfaz de lista de laptops y desde esta misma se puede acceder a distintas interfaces como: Editar laptop que esta permite regresar al inicio, también se accede a registro de mantenimiento y desde esta se puede acceder a lista de mantenimiento y desde ahí permite acceder a editar mantenimiento que a su vez permite acceder de nuevo al inicio, ya por último de desde editar laptop se puede acceder a registro de fallo y de esta directamente a lista de fallo que también permite acceder a la interfaz de editar fallo, esta misma permite regresar al inicio.

Diagrama

El contenido generado por IA puede ser incorrecto.

**Figura 2.4** Mapa de navegación de ControlPC.

## Primer avance del proyecto

En la figura 2.5 se visualiza el primer avance de la aplicación ControlPC, donde se puede ver la interfaz de inicio de la aplicación.

Imagen de la pantalla de un celular con letras

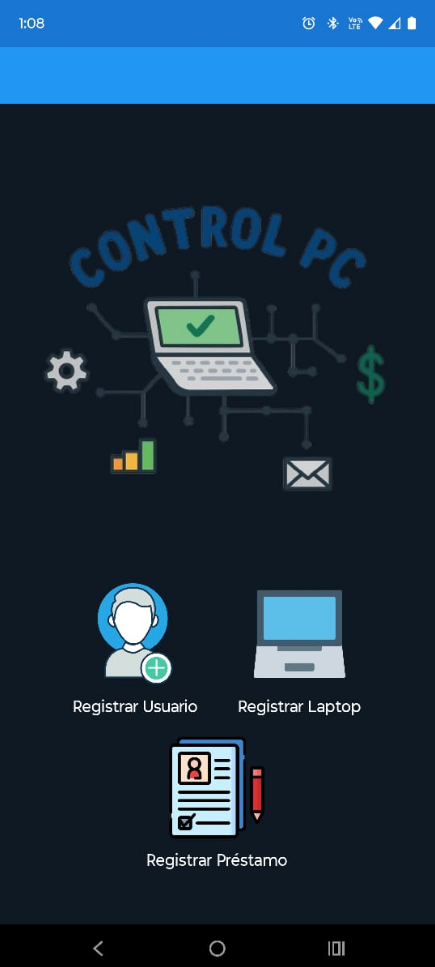
El contenido generado por IA puede ser incorrecto.

**Figura 2.5** Página de inicio de ControlPC.

# **CAPÍTULO 3 IMPLEMENTACIÓN Y PRUEBAS**

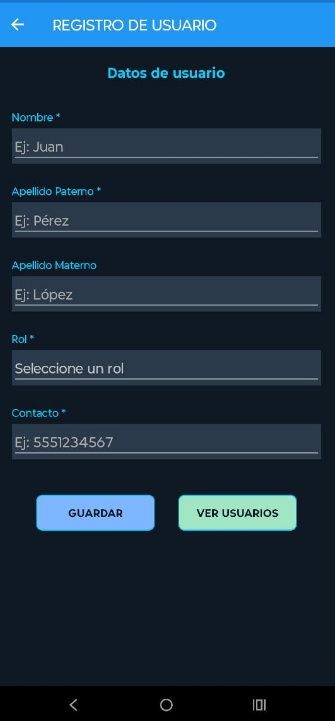
## 3.1 Documentación de interfaces

En la figura 3.1 se muestra la Interfaz de inicio de ControlPC, esta interfaz es la primera en visualizarse al abrir la aplicación, tiene la funcionalidad de que al presionar sobre la imagen de “Registrar usuario”, “Registrar laptop” o “Registrar préstamo” se accede a su interfaz correspondiente.



**Figura 3.1** Interfaz de inicio de ControlPC

En la figura 3.1.1 se visualiza la Interfaz de registro de usuario, esta interfaz tiene como funcionalidad el registrar un nuevo usuario, permitiendo ingresar datos personales, rol dentro de la institución y su número de contacto. Los botones tienen la funcionalidad de guardar un usuario al previamente llenar cada uno de los datos requeridos, también se incluye la opción de poder acceder a otra interfaz de lista de usuarios, mediante el botón “Ver usuarios”.



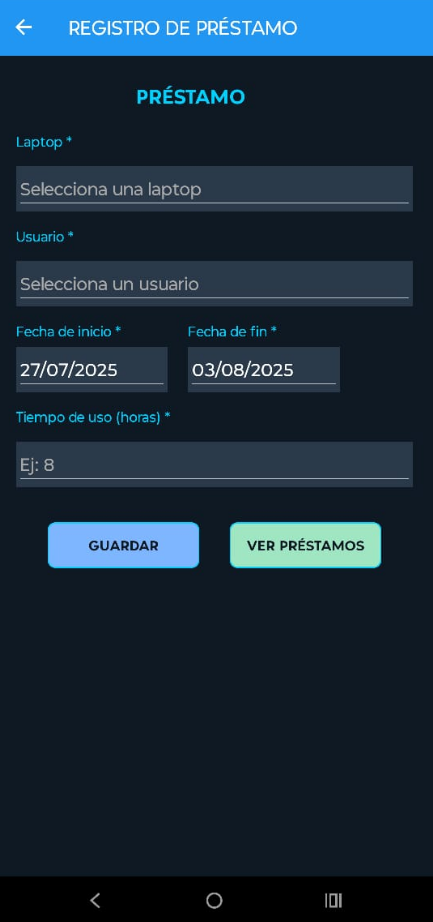
**Figura 3.1.1** Interfaz de registro de usuario de ControlPC.

En la figura 3.1.2 se visualiza la Interfaz de registro de laptop, esta interfaz tiene como funcionalidad el registrar una nueva laptop, permitiendo ingresar datos y características del equipo. Los botones tienen la funcionalidad de guardar una laptop al previamente llenar cada uno de los datos requeridos, también se incluye la opción de poder acceder a otra interfaz de lista de laptops, mediante el botón “Ver laptops”. Las imágenes colocadas en la parte superior derecha permiten respectivamente redireccionar a una lista de historial de mantenimiento y de reporte de fallos de equipos.



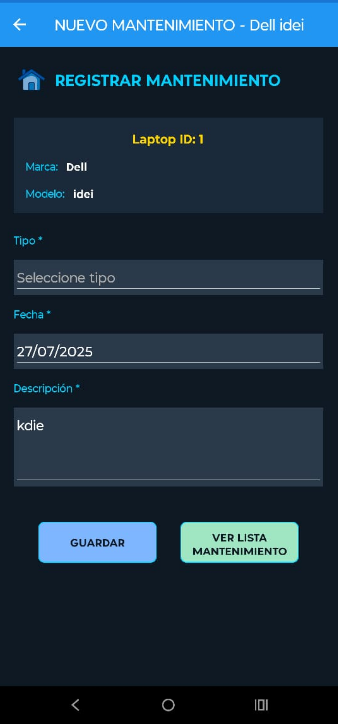
**Figura 3.1.2** Interfaz de registro de laptop de ControlPC.

En la figura 3.1.3 se visualiza la Interfaz de registro de préstamo, esta interfaz tiene la funcionalidad de registrar un nuevo préstamo al solicitar un equipo y seleccionarlo de una lista al ser previamente registrado así mismo se selección el usuario previamente registrado, además de esto se ingresan datos del préstamo como fecha de inicio y fin de préstamo, como las horas de uso aproximado que tendrá el equipo. También se incluye la funcionalidad para que desde esta interfaz se permita acceder a una lista de préstamos previamente registrados.



**Figura 3.1.3** Interfaz de registro de préstamo de ControlPC.

En la figura 3.1.4 se visualiza la Interfaz de registro de mantenimiento, desde esta interfaz se permite registrar un nuevo mantenimiento a un equipo que lo requiera y que este previamente registrado, permite seleccionar el tipo de mantenimiento que va necesitar, su fecha y descripción del mantenimiento. Así mismo permite acceder a la lista de mantenimientos.



**Figura 3.1.4** Interfaz de registro de mantenimiento de ControlPC.

En la figura 3.1.5 se visualiza la Interfaz de reporte de fallo de laptop, desde esta interfaz se permite hacer el registro de fallo de una laptop que ya este registrada, se puede añadir una descripción del fallo, su hora de uso previo, la fecha del reporte de fallo y la prioridad que requiere el equipo para su reparación, así mismo permite mediante el botón de ver lista de fallos el poder visualizar las laptops que tengan algún reporte de fallo. La imagen de la esquina superior izquierda permite acceder a la interfaz de inicio.



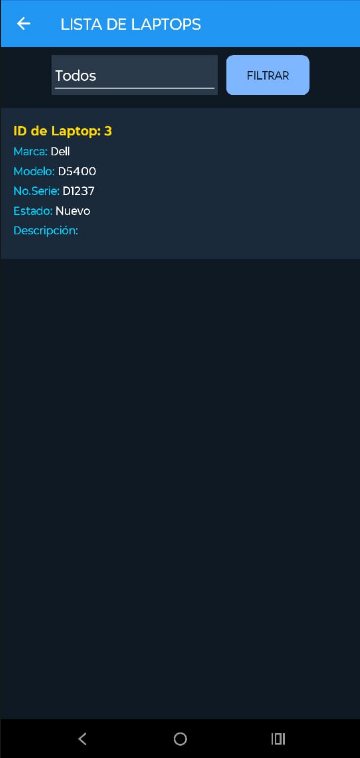
**Figura 3.1.5** Interfaz de reporte de fallo de ControlPC.

En la figura 3.1.6 se visualiza la Interfaz de lista de usuarios, esta interfaz tiene como función el mostrar todos los usuarios registrados, además mediante un filtro se puede seleccionar el tipo de usuario, también permite editar un usuario o eliminarlo al seleccionarlo.



**Figura 3.1.6** Interfaz de lista de usuarios de ControlPC.

En la figura 3.1.7 visualiza la Interfaz de lista de laptops, esta interfaz tiene como función mostrar todas las laptops registradas, además permite filtrar por el estado de la laptop, también permite editar o eliminar una laptop.



**Figura 3.1.7** Interfaz de lista de laptops de ControlPC.

En la figura 3.1.8 se visualiza la Interfaz de lista de préstamos, esta interfaz tiene como función mostrar todos los prestamos registrados, además permite filtrar por el estado del préstamo si está activo o finalizado, también permite editar o eliminar un préstamo.



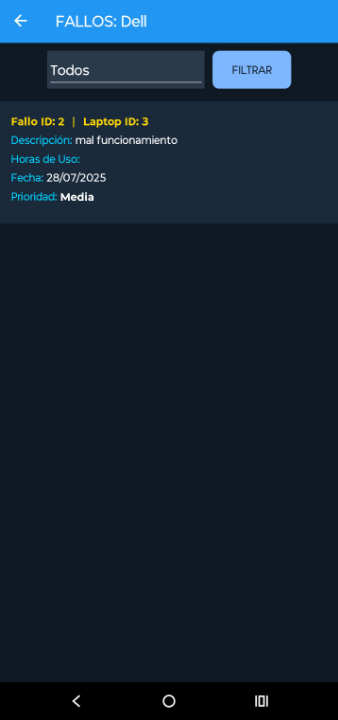
**Figura 3.1.8** Interfaz de lista de préstamos de ControlPC.

En la figura 3.1.9 se visualiza la Interfaz de lista de mantenimiento, esta interfaz tiene como función mostrar todos los mantenimientos registrados, además permite filtrar por tipo de mantenimiento, también permite editar o eliminar un mantenimiento.



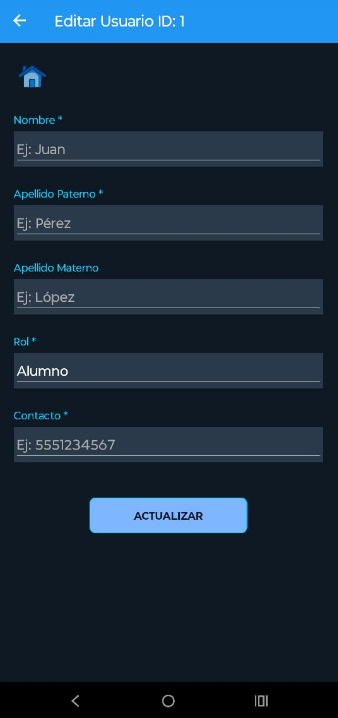
**Figura 3.1.9** Interfaz de lista de mantenimientos de ControlPC.

En la figura 3.1.10 se visualiza la Interfaz de lista de fallos, esta interfaz tiene como función mostrar todos los reportes registrados, además permite filtrar por prioridad de reporte de fallo, también permite editar o eliminar un reporte.



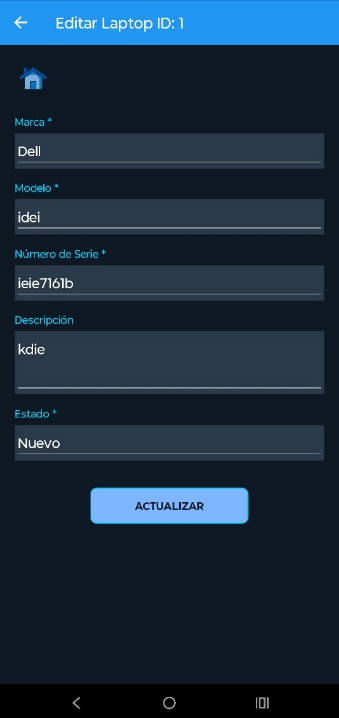
**Figura 3.1.10** Interfaz de lista de reporte de fallos de ControlPC.

En la figura 3.1.11 se visualiza la Interfaz de editar usuario, esta interfaz tiene como función editar un usuario previamente registrado, permitiendo modificar cada uno de los datos del usuario y mediante el botón permite actualizar el cambio y con la imagen de la esquina superior izquierda permite redireccionar al inicio de la aplicación.



**Figura 3.1.11** Interfaz de editar usuario de ControlPC.

En la figura 3.1.12 se visualiza la Interfaz de editar laptop, esta interfaz tiene como función editar una laptop previamente registrada, permitiendo modificar cada uno de los datos de la laptop y mediante el botón permite actualizar el cambio y con la imagen de la esquina superior izquierda permite redireccionar al inicio de la aplicación.



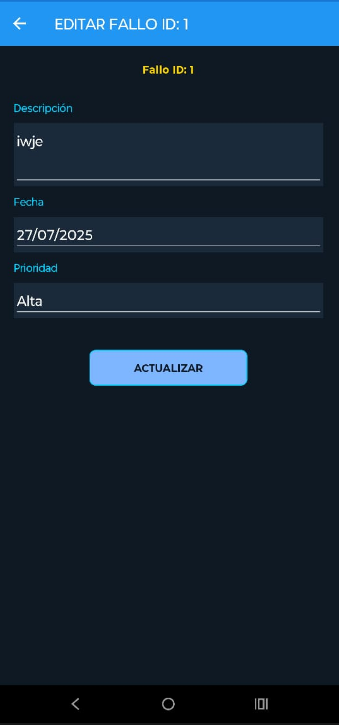
**Figura 3.1.12** Interfaz de editar laptop de ControlPC.

En la figura 3.1.13 se visualiza la Interfaz de editar mantenimiento, esta interfaz tiene como función el modificar un mantenimiento previamente registrado, al dar en el botón actualiza el cambio y con la imagen de la esquina superior izquierda redirecciona al inicio de la aplicación.



**Figura 3.1.13** Interfaz editar mantenimiento de ControlPC.

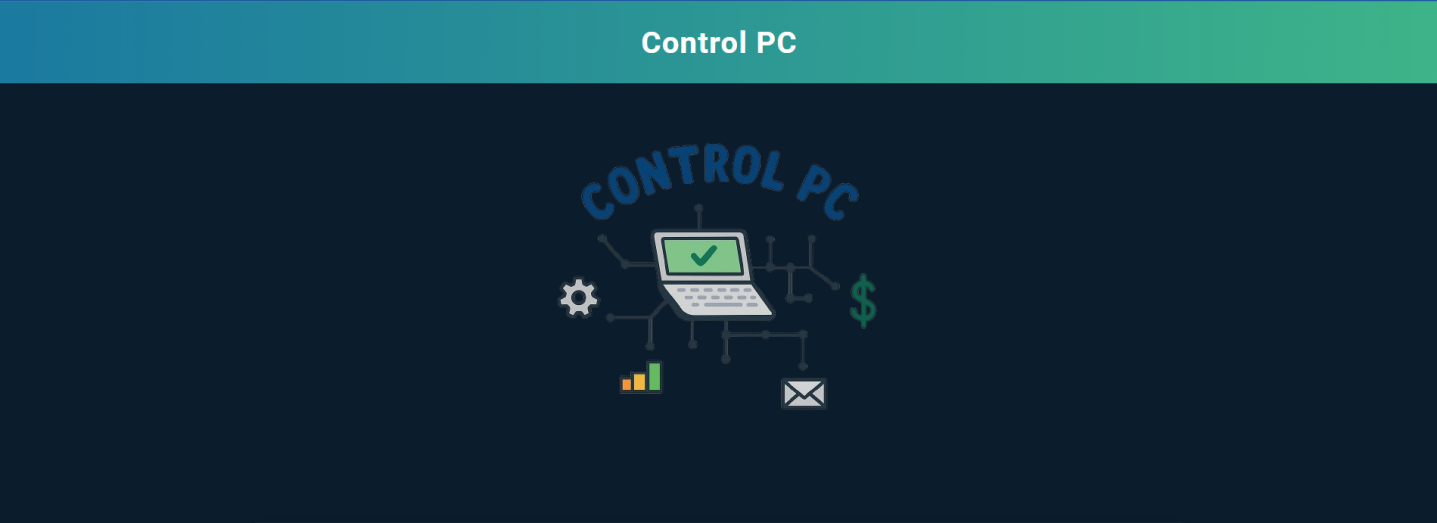
En la figura 3.1.14 se visualiza la Interfaz de editar fallo, esta interfaz tiene como función el modificar un reporte de fallo previamente registrado, al dar en el botón actualizar el cambio



**Figura 3.1.14** Interfaz editar fallo de ControlPC.

3.2 Página web informativa y de contacto

En la figura 3.2 se muestra la interfaz del encabezado de la página web informativa del sistema ControlPC. En la parte superior se encuentra el nombre del proyecto “Control PC” y un encabezado con el logotipo del proyecto.



**Figura 3.2** Sección de Encabezado de ControlPC

En la figura 3.2.1 se presenta la sección "Objetivo General", donde se detalla cómo se soluciona la necesidad de los laboratoristas (encargados de las laptops institucionales), por la tanto se explica cómo la APP móvil multiplataforma ControlPC busca resolverla.

Texto

El contenido generado por IA puede ser incorrecto.

**Figura 3.2.1** Sección Objetivo General de ControlPC

En la figura 3.2.2 se presenta la sección "Misión, Visión y Valores” que son los aspectos más relevantes que una empresa tiene, en este caso la presente empresa NovaWare desarrolladora de la aplicación móvil ControlPC..

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

**Figura 3.2.2** Sección Misión y Visión de ControlPC

En la figura 3.2.3 se presenta la sección "Organigrama” que es la representación gráfica de la estructura interna de la empresa ControlPC, en el que se muestran las relaciones jerárquicas y las interconexiones entre los diferentes roles dentro de cada miembro de la empresa.

Interfaz de usuario gráfica, Sitio web

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de un celular con la pantalla encendida

El contenido generado por IA puede ser incorrecto.

**Figura 3.2.3** Sección Organigrama de ControlPC

En la figura 3.2.4 se presenta la sección "Descripción, Alcance y Meta” donde se muestra el conjunto de cuestiones principales y especificas del proyecto ControlPC.

Texto

El contenido generado por IA puede ser incorrecto.

**Figura 3.2.4** Sección Organigrama de ControlPC

En la figura 3.2.5 se presenta la sección "Metodología y Requerimientos Funcionales” donde se muestra el conjunto de procedimientos por los cuales fue desarrollado el proyecto de ControlPC.

**Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.**

**Figura 3.2.5** Sección Metodología y Requerimientos Funcionales de ControlPC

En la figura 3.2.6 se presenta la sección "Tecnologías y Herramientas Utilizadas” donde se muestra el conjunto de Tecnologías con el que fue desarrollado el proyecto de ControlPC.

**Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.**

**Figura 3.2.6** Sección Tecnologías y Herramientas Utilizadas de ControlPC

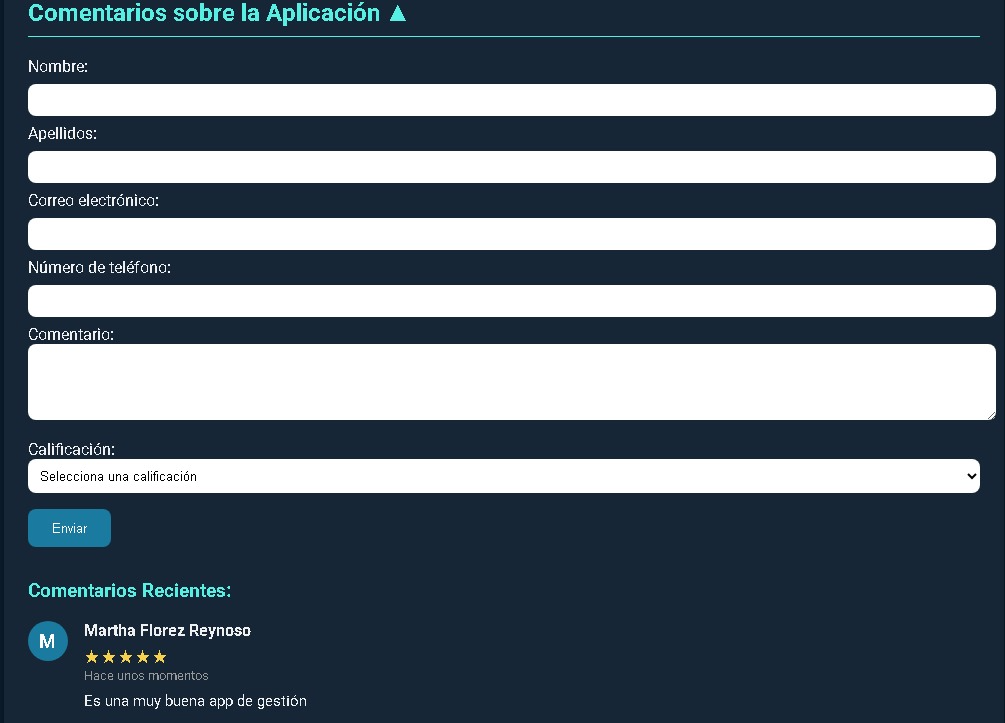
En la figura 3.2.7 se presenta la sección "Coste del Proyecto” donde se muestra el conjunto precios con el que fue costeado el proyecto de ControlPC.

**Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.**

**Figura 3.2.7** Sección Coste del Proyecto ControlPC

En la figura 3.2.8 se presenta la sección "Comentarios de la Aplicación” donde se muestra un pequeño formulario en donde se requiere datos personales como nombre, apellido ,correo electrónico, y número de teléfono del usuario, así mismo el formulario cuenta con un apartado de comentarios para que el usuario nos de su opinión acerca de la aplicación del proyecto ControlPc y la califique.



**Figura 3.2.8** Sección Comentarios de la Aplicación del Proyecto ControlPC

En la figura 3.2.9 se presenta la sección "Comentarios Recientes ” donde los usuarios podrán visualizar los comentarios de las demás personas acerca de la aplicación del proyecto ControlPc y la calificación que le han dado así como podrán ver la fecha en la que se realizo su comentario.

**Figura 3.2.9** Sección Comentarios Recientes del Proyecto ControlPC

CONCLUSIONES

El desarrollo del proyecto *ControlPC* representa una solución tecnológica efectiva para optimizar el proceso de préstamo de laptops dentro de la Universidad Tecnológica de Tecámac. A través del diseño e implementación de una aplicación móvil multiplataforma, se logró automatizar y centralizar el registro, seguimiento y control del inventario de equipos de cómputo, eliminando el uso de métodos manuales propensos a errores e ineficiencias.

Por lo tanto, a través del diseño e implementación de una aplicación móvil multiplataforma, se logró automatizar y centralizar el registro, seguimiento y control del inventario de equipos de cómputo, eliminando el uso de métodos manuales propensos a errores e ineficiencias.

ANEXOS Y/O APÉNDICES

Figura A. Se muestra la firma de autorización del diagrama general de casos de uso por parte de la Mtra. Carrada Mariscal Jocabed, asignada a la materia de Base de Datos para Computo en la Nube. Capítulo 1, Recolección y Refinamiento De Requisitos, Subtema 1.7.1 Diagrama general de casos de uso.

Diagrama

El contenido generado por IA puede ser incorrecto.

**Figura A.** Diagrama general de casos de uso de ControlPC.

Figura B. Se muestra la firma de autorización del modelo relacional de la base de datos de ControlPC por parte de la Mtra. Carrada Mariscal Jocabed, asignada a la materia de Base de Datos para Computo en la Nube. Capítulo 2, Diseño y Desarrollo del Proyecto, Subtema 2.1 Diseño de la base de datos.

Diagrama

El contenido generado por IA puede ser incorrecto.

**Figura B.** Modelo relacional de ControlPC.

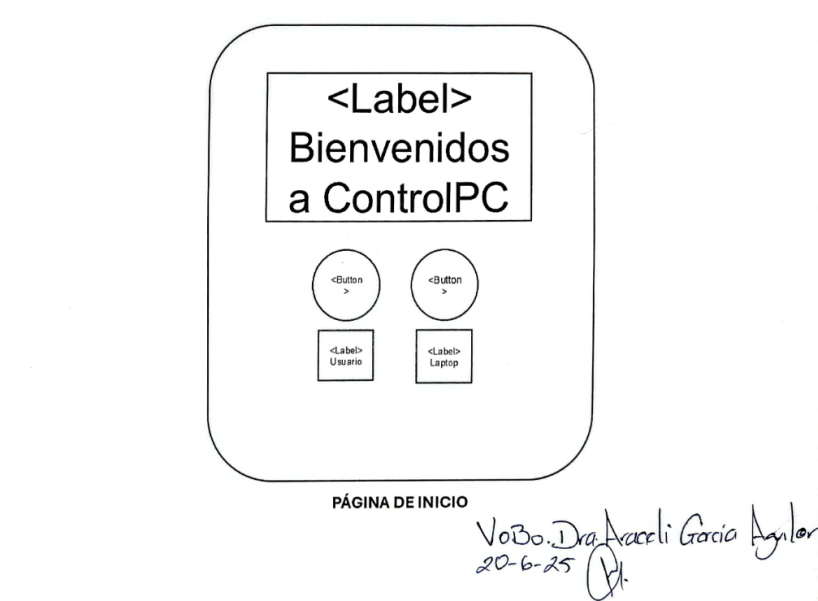
Figura C. Se muestra la firma de autorización del diccionario de datos de la base de datos de ControlPC por parte de la Mtra. Carrada Mariscal Jocabed, asignada a la materia de Base de Datos para Computo en la Nube. Capítulo 2, Diseño y Desarrollo del Proyecto, Subtema 2.2 Diccionario de datos.

Un conjunto de letras negras en un fondo blanco

El contenido generado por IA puede ser incorrecto.

**Figura C.** Diccionario de datos de ControlPC.

Figura D. Se muestra la firma de autorización del diseño de interfaces de ControlPC por parte de la Mtra. García Aguilar Araceli, asignada a la materia de Desarrollo móvil multiplataforma. Capítulo 2, Diseño y desarrollo del proyecto, subtema 2.3, Diseño de interfaces.



**Figura D.** Diseño de interfaces de ControlPC.

Figura E. Se muestra la firma de autorización del mapa de navegación de ControlPC por parte de la Mtra. García Aguilar Araceli, asignada a la materia de Desarrollo móvil multiplataforma. Capítulo 2, Diseño y desarrollo del proyecto, subtema 2.3, Mapa de navegación.

Diagrama

El contenido generado por IA puede ser incorrecto.

**Figura E.** Mapa de navegación de ControlPC.

LISTADO DE SIGLAS Y ACRÓNIMOS

* RF: Functional Requirement - Requerimiento funcional.
* RNF: Non-Functional Requirement - Requerimiento no funcional.
* CRUD: Crear, Leer, Actualizar, Eliminar - Create, Read, Update, Delete.
* iOS: Apple’s mobile operating system for iPhone and iPad - Sistema operativo móvil de Apple para iPhone y iPad.
* C++: High-performance object-oriented programming language - Lenguaje de programación de alto rendimiento orientado a objetos.
* IA: Artificial Intelligence - Inteligencia Artificial.
* UTTEC: Public university located in the State of Mexico - Universidad pública que se encuentra en el Estado de México.
* TIC: Information and Communication Technologies - Tecnologías de la información y comunicación.
* DSM: Cross-platform Software Development - Desarrollo de Software Multiplataforma.
* IRD: Digital Network Infrastructure- Infraestructura de redes digitales.
* IDE: Entorno de Desarrollo Integrado - Integrated Development Environment.
* BCL: Biblioteca de Clases Base - Base Class Library.
* CLR: Entorno de Ejecución del Lenguaje Común - Common Language Runtime.
* CLI: Infraestructura de Lenguaje Común - Common Language Infrastructure.
* PWA: Aplicación Web Progresiva - Progressive Web App.
* UML: Infraestructura de Modelado Unificado - Unified Modeling Language.
* MVVM: Modelo-Vista-VistaModelo - Model-View-ViewModel.
* SQL: Lenguaje de Consulta Estructurada - Structured Query Language.
* XAML: Lenguaje de Marcado de Aplicación Extensible - Extensible Application Markup Language.
* APK: Android Application Package - Paquete de Aplicación para Android.
* API: Interfaz de Programación de Aplicaciones - Application Programming Interface.
* C#: Lenguaje de programación desarrollado por Microsoft - Programming language developed by Microsoft.
* LINQ: Consulta Integrada en el Lenguaje - Language Integrated Query.

GLOSARIO

* Android: Sistema operativo para móviles diseñado por la compañía estadounidense Google. Basado en el sistema operativo Linux
* Aplicación móvil: También llamada aplicación móvil, es un tipo de aplicación diseñada para ejecutarse en un dispositivo móvil, que puede ser un teléfono inteligente o una tableta.
* Aplicaciones nativas: Programas diseñados y desarrollados específicamente para un sistema operativo particular, como iOS o Android.
* Base de datos Realm: Realm es una base de datos orientada a objetos que proporciona a los desarrolladores una interfaz de objetos en múltiples lenguajes de programación para resolver el desajuste entre la capa de objetos y la capa de base de datos en el desarrollo orientado a objetos.
* Base de datos relacional: Una base de datos relacional (RDB) es una forma de estructurar información en tablas, filas y columnas.
* Control de versiones: También conocido como gestión de código fuente, utiliza herramientas para realizar un seguimiento de las modificaciones o los cambios realizados en el código fuente a lo largo del tiempo.
* Desarrollo de software: Proceso de crear y diseñar programas o aplicaciones informáticas mediante la escritura de código, diseño, pruebas y mantenimiento.
* Eslogan: Frase corta que busca representar una marca para promover la rápida identificación y memorización de sus productos y servicios por los consumidores.
* Garbage Collection: o recolección de basura es una característica clave del lenguaje de programación Java que gestiona automáticamente la asignación y designación de memoria para los objetos que se crean en un espacio edén.
* GitHub: GitHub es una plataforma basada en la nube donde puedes almacenar, compartir y trabajar junto con otros usuarios.
* JavaScript: Lenguaje de programación que los desarrolladores utilizan para hacer páginas web interactivas.
* Logo: Los logos son imágenes, textos, formas o una combinación de los tres que representan el nombre y el propósito de un negocio.
* Normatividad aplicable: Conjunto de directrices y reglas que los equipos de desarrollo siguen al escribir código.
* Open source: (Código Abierto): Se refiere al software cuyo código fuente es accesible para cualquier usuario.
* Polimorfismo: Capacidad de un objeto para adoptar más de una forma.
* Python: Lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el machine learning (ML).
* Requisitos funcionales: Descripción específica de lo que un sistema, software, producto o servicio debe hacer para cumplir con los objetivos establecidos y satisfacer las necesidades de los usuarios o del negocio.
* Requisitos no funcionales: También conocidos como "requisitos de calidad" o "requisitos de atributos del sistema", son características y criterios que describen cómo debe ser el rendimiento, la seguridad, la usabilidad y otros aspectos de un sistema o software más allá de su funcionalidad básica.
* Sistema operativo móvil: Un sistema operativo móvil (SO) es una plataforma de software especializada que permite el funcionamiento de teléfonos inteligentes, tabletas y otros dispositivos móviles.
* TypeScript: Lenguaje basado en JavaScript, pero con algunas mejoras que permiten convertir la tarea de crear una web con JavaScript en una nueva experiencia de programación.
* Xamarin: Plataforma de desarrollo móvil que permite a los desarrolladores crear aplicaciones móviles para iOS, Android y Windows; utilizando un único conjunto de herramientas y lenguajes de programación.
* Interfaz: Una interfaz es el punto de contacto o medio a través del cual dos entidades interactúan entre sí. En tecnología y desarrollo de software, el término puede tener distintos significados dependiendo del contexto.
* Hardware: El hardware es la parte física y tangible de una computadora o dispositivo electrónico. Es todo aquello que puedes ver y tocar, como los cables, circuitos, pantallas, teclados, etc.
* Software: El software es el conjunto de programas, instrucciones y datos que le dicen al hardware qué hacer. Es la parte lógica de un sistema informático, y no se puede tocar físicamente.
* Interfaz: Una interfaz es un punto de conexión o comunicación entre dos sistemas, dispositivos, programas o componentes, que permite el intercambio de información o interacción entre ellos.
* Label: Elemento que se utiliza para asociar un texto descriptivo a un campo de formulario, como una caja de texto, un botón, etc.
* Button: Etiqueta que crea un botón interactivo en una aplicación.
* Picker: Etiqueta que se utiliza para mostrar una lista desplegable donde el usuario puede seleccionar un elemento de una lista de opciones.
* Image: Etiqueta que se utiliza para mostrar imágenes en la interfaz de usuario.
* DatePicker: Etiqueta que se utiliza para que el usuario seleccione una fecha dentro de un calendario o seleccionador de fechas.
* Entry: Etiqueta que se utiliza para entrada de texto de una sola línea.

REFERENCIAS

Anandmeg. (s. f.). ¿Qué es el IDE de Visual Studio? Microsoft Learn. <https://learn.microsoft.com/es-es/visualstudio/get-started/visual-studio-ide?view=vs-2022>

Android Developers. (s. f.). Novedades de Android 7.0 para desarrolladores. Google Developers. <https://developer.android.com/about/versions/nougat/android-7.0?hl=es-419>

AppMaster. (s. f.). Sistema operativo móvil (por ejemplo, Android, iOS). AppMaster - Ultimate All-in No-code Platform. <https://appmaster.io/es/glossary/sistema-operativo-movil-por-ejemplo-android-ios>

Arsys. (s/f). TypeScript: qué es y para qué sirve. Recuperado el 29 de mayo de 2025, de <https://www.arsys.es/blog/typescript-javascript>

Baltazar, M. (2022, 27 abril). Aprende TODO sobre Slogans: ¡cómo crear uno para tu empresa y conoce los más creativos! Rock Content - ES. <https://rockcontent.com/es/blog/slogan/#:~:text=Un%20slogan%20es%20una%20frase,y%20servicios%20por%20los%20consumidores>

Bambu Mobile. (2023, 26 mayo). ¿Qué es Xamarin? <https://bambu-mobile.com/que-es-xamarin/>

Bustos, J. L. (2024, 7 junio). Características de SQLite | KeepCoding Bootcamps. KeepCoding Bootcamps. <https://keepcoding.io/blog/caracteristicas-de-sqlite/>

Clarcat. (s. f.). Xamarin - Clarcat. <https://www.clarcat.com/herramientas/xamarin/>

El diagrama de casos de uso en UML. (s/f). IONOS Digital Guide. Recuperado el 29 de mayo de 2025, de <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/diagrama-de-casos-de-uso/>

GitHub. (s. f.). About GitHub. GitHub. <https://docs.github.com/en/get-started/quickstart/about-github>

Gomez, C. (2020, junio 21). Requerimientos de software. Diario de QA. <https://www.diariodeqa.com/post/quiero-ser-qa-que-debo-aprender-requerimientos-de-software>

Google. (2016, 22 agosto). Android 7.0 Nougat: Un sistema operativo más potente, hecho a tu medida. Google Latinoamérica. <https://latam.googleblog.com/2016/08/android-70-nougat-un-sistema-operativo.html>

Google. (s. f.). New features on Android. Android. <https://www.android.com/new-features-on-android/>

IBM. (2023, 12 mayo). ¿Qué es MongoDB? | IBM. <https://www.ibm.com/mx-es/topics/mongodb>

IBM. (2024, 23 octubre). Recolección de basura (Garbage Collection) de Java. IBM. <https://www.ibm.com/mx-es/topics/garbage-collection-java>

IBM. (2025, 16 enero). SQL. IBM. <https://www.ibm.com/mx-es/think/topics/structured-query-language>

ifeder. (2022, 3 febrero). Recursos tecnológicos. Lifeder. <https://www.lifeder.com/recursos-tecnologicos/>

Jesús. (2024, 5 mayo). *¿Qué es SQLite y sus Características?* Tutoriales Dongee. <https://www.dongee.com/tutoriales/que-es-sqlite-y-sus-caracteristicas/>

Microsoft. (s. f.). Especificación del lenguaje C#. Microsoft Learn. <https://learn.microsoft.com/es-es/dotnet/csharp/language-reference/language-specification/introduction>

Microsoft. (s. f.). Guía de C#. Microsoft Learn. <https://learn.microsoft.com/es-es/dotnet/csharp/>

Microsoft. (s. f.). Introducción a XAML en Xamarin.Forms. Microsoft Learn. <https://learn.microsoft.com/es-es/xamarin/xamarin-forms/xaml/>

Microsoft. (s. f.). MVVM en Xamarin.Forms. Microsoft Learn. <https://learn.microsoft.com/es-es/xamarin/xamarin-forms/enterprise-application-patterns/mvvm>

Microsoft. (s. f.). Tour por el lenguaje C#. Microsoft Learn. <https://learn.microsoft.com/es-es/dotnet/csharp/tour-of-csharp/overview>

Microsoft Learn. (s.f.). Hardware requirements. Recuperado de: <https://learn.microsoft.com/>

MongoDB Inc. (2022). Realm Database. <https://www.mongodb.com/realm>

Moreno, P., & Moreno, P. (2024, 13 febrero). ¿Qué es el lenguaje C++ y para qué sirve? Blog Centro de e-Learning. <https://blog.centrodeelearning.com/lenguaje-c/>

Narvaez, M. (2020, noviembre 10). Tipos de necesidades del cliente y cómo solucionarlas. QuestionPro. <https://www.questionpro.com/blog/es/tipos-de-necesidades-del-cliente/>

OpenWebinars. (2019, 28 octubre). Qué es MongoDB. OpenWebinars. <https://openwebinars.net/blog/que-es-mongodb/>

Ortega, J. M. M. (2019, 6 febrero). ¿Qué es XAML en .NET? OpenWebinars.net. <https://openwebinars.net/blog/que-es-xaml-en-net/>

Pmoinformatica.com. (s/f). Requerimientos funcionales: Ejemplos. Recuperado el 29 de mayo de 2025, de <https://www.pmoinformatica.com/2017/02/requerimientos-funcionales-ejemplos.html>

Quiocho, C. (2024, 14 noviembre). ¿Qué es el polimorfismo? - NinjaOne. <https://www.ninjaone.com/es/it-hub/it-service-management/polimorfismo/>

Tailor Brands. (2025, 21 febrero). ¿Qué es un logo? | Tailor Brands. <https://www.tailorbrands.com/es/logo-maker/que-es-un-logo>

Universidad Europea. (2024, 22 enero). Qué son las aplicaciones nativas. <https://universidadeuropea.com/blog/apps-nativas/>

Urrutia, D. (2024, 20 octubre). Qué es Android - Definición, significado y para qué sirve. Arimetrics. <https://www.arimetrics.com/glosario-digital/android>

Visure Solutions. (2022, octubre 21). Qué son los requisitos no funcionales: ejemplos, definición, guía completa. Visure Solutions. <https://visuresolutions.com/es/blog/non-functional-requirements/>

Westreicher, G. (2022, 24 noviembre). Optimización Qué es, definición y concepto. Economipedia. <https://economipedia.com/definiciones/optimizacion.html>

¿Qué es JavaScript? - Explicación de JavaScript (JS). (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/javascript/>

¿Qué es la IA? - Explicación de la inteligencia artificial. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/artificial-intelligence/>

¿Qué es open source? [GUÍA BÁSICA CON PROS Y CONTRAS]. (2024, 4 enero). Zendesk. <https://www.zendesk.com.mx/blog/que-es-open-source/>

¿Qué es Python? - Explicación del lenguaje Python. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/python/>

Nunez, L. (2023, enero 4). *Tipos de aplicaciones, características, ejemplos y comparativa*. | EMMA. <https://emma.io/blog/tipos-aplicaciones-caracteristicas-ejemplos/>

¿Qué es una base de datos relacional (RDBMS)? | Google Cloud. (s. f.). Google Cloud. <https://cloud.google.com/learn/what-is-a-relational-database?hl=es-419>

Content Studio. (2024, 18 marzo). *¿Qué es un diccionario de datos?* Pure Storage. <https://www.purestorage.com/es/knowledge/what-is-a-data-dictionary.html>

What is an App Interface? Definition & Examples. (2024, 14 febrero). Appetiser: Web And Mobile App Development Melbourne, Australia. <https://appetiser-com-au.translate.goog/glossary/appinterface/?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc>