Proyecto formativo HIDDENPASS

Aprendices

Juan Andrés Menéndez Villarraga

David Fernando Gomez Aristizabal

Nicky Alexander Florez Bustamante

Ficha 2828523

Área Tecnología de la Información y la Comunicación

Programa análisis y desarrollo de software

Servicio Nacional de Aprendizaje Sena

2025

Resumen

Este proyecto nace de la necesidad actual de optimizar y mejorar la seguridad de los usuarios. HIDDENPASS será una aplicación de escritorio y móvil para la gestión de contraseñas, y notas, en el que prima la seguridad y facilidad de uso. Para cumplir esta finalidad, la aplicación tendrá capacidad de crear y gestionar contraseñas y notas de usuario; además, tendrá login seguro, mediante la utilización de contraseña maestra. Con factores diferenciadores tales como carpetas, generación de contraseñas, búsqueda y eliminación de las mismas, enlace y sincronización a dispositivos móviles.

Palabras claves

Seguridad, login, optimizar, gestión, aplicación.

Abstract

This project was born from the need to optimize and improve user security nowadays. HIDDENPASS will be a desktop and mobile application for managing passwords and notes, with a focus on security and ease of use. To achieve this goal, the application will be able to create and manage user passwords and notes; in addition, you will be able to log in securely using a master password. Differentiating factors include folders, password generation, password search and deletion, and mobile device linking and synchronization.

Keywords

Security, login, optimization, management, application.

Índice

Contenido

Tablas

Figuras

Glosario

Frontend

Es la parte visible de una aplicación web, la interfaz de usuario con la que los usuarios interactúan directamente. Incluye todo lo que los usuarios ven y con lo que pueden interactuar, como menús, botones, imágenes, texto y diseño (HubSpot, 2024).

Backend

La parte oculta o la infraestructura de un sistema, aplicación o sitio web que se encarga de la lógica, el procesamiento de datos y la comunicación con el servidor (HubSpot, 2024).

POO

La Programación Orientada a Objetos (POO) es un paradigma de programación, esto es, un modelo o un estilo de programación que proporciona unas guías acerca de cómo trabajar con él y que está basado en el concepto de clases y objetos (Edteam, 2022)

Linux

Linux es un sistema operativo de código abierto, lo que significa que su código fuente es público y puede ser modificado y distribuido libremente por cualquier persona (Humanidades, 2025)

Framework

Es una estructura o conjunto de herramientas y componentes predefinidos que se utilizan para desarrollar aplicaciones de software de manera organizada, eficiente y estándar (EBAC, 2023)

Spring

Es un framework de código abierto para Java que facilita el desarrollo de aplicaciones empresariales. Simplifica la creación, configuración y despliegue de aplicaciones Java, especialmente aplicaciones web y microservicios (OpenWebinars, 2018)

API

Las APIs permiten que una aplicación "llame" a otra aplicación para realizar una tarea o acceder a información, sin necesidad de conocer los detalles internos de la implementación de esa otra aplicación (Amazon AWS, 2018)

ORM

Es una técnica que permite mapear objetos de un lenguaje de programación orientado a objetos a filas de una base de datos relacional. Esto facilita la interacción con la base de datos utilizando una estructura de objetos en lugar de consultas SQL (Edteam, 2023)

Biblioteca

Es una colección de código reutilizable que facilita el desarrollo de software. Estas bibliotecas suelen contener funciones, clases, módulos y rutinas predefinidas que resuelven tareas comunes (educaopen, 2025).

Java anotaciones

Las anotaciones en Java comienzan con '@'. No cambian la actividad de un programa ordenado. Ayudan a relacionar metadatos (datos) con los componentes del programa, es decir, constructores, estrategias, clases, entre otros (keepcoding, 2024)

Interface

Una interfaz define un contrato o especificación de cómo un objeto debe comportarse sin especificar cómo ese comportamiento se implementa internamente. Es una forma de definir un conjunto de métodos que una clase o tipo debe implementar (LEGSA, s. f.).

Pruebas unitarias

Son un tipo de prueba de software que se enfoca en verificar el funcionamiento de componentes individuales de código, como funciones, métodos o clases, de forma aislada (AWS, s. f.).

Método

Un método es una subrutina (o función) que está asociada a un objeto o clase y que realiza una acción específica. Los métodos permiten a los objetos "actuar" o "hacer cosas" (Dart, documentación oficial, s. f.).

Petición Get

Es un tipo de solicitud HTTP que se utiliza para recuperar información de un servidor web. Es un método de petición común que se usa para obtener datos, como HTML, JSON o imágenes, del servidor (MDN Web Docs, 2025).

Peticion Put

Es un tipo de solicitud HTTP utilizada para actualizar o reemplazar un recurso existente en un servidor. Es idempotente, lo que significa que realizar la misma petición varias veces tendrá el mismo efecto (MDN Web Docs, 2025).

Hive

Hive es una base de datos clave-valor ligera, ultrarrápida y escrita en Dart puro. (Dart API docs, s. f.).

GitHub

GitHub es una plataforma basada en la nube donde se puede almacenar y trabajar junto con otros usuarios para escribir codigo (GitHub Docs, s.f.).

Introducción

HIDDENPASS es un sistema de gestión de contraseñas, el cual fue desarrollado con el propósito de almacenar notas, generar contraseñas y/o almacenar contraseñas de forma segura. Actualmente nos encontramos en una era digital, donde gran parte de la vida de una persona se encuentra digitalizada, por lo que este proyecto nace por la necesidad de proteger o hacer más segura esa información y evitar filtraciones o accesos no deseados a dicha información.

Esta iniciativa se lleva a cabo en la ciudad de Medellín y es desarrollado por un grupo de aprendices del Programa de Análisis y Desarrollo de Software del SENA, con la colaboración de instructores técnicos en el área, los cuales hicieron parte del proceso.

Este proyecto se inicia en el año 2024 y se proyecta su lanzamiento oficial para julio del 2025.

Su ejecución incluye varias etapas, las cuales son: Investigación de campo, desarrollo de software y su testeo por parte de comunidades involucradas, todo esto se lleva a cabo con la implementación de metodología ágiles, permitiendo llevar a cabo el desarrollo del proyecto y adaptarlo de la mejor manera a la necesidad real existente.

Tabla 1

Cronograma de actividades

Cronograma de actividades

	Nombre actividad	Fecha de inicio	Fecha de fin
01	Creación funcionalidad gestión de usuarios	01-10-2024	31-10-2024
03	Creación funcionalidad gestión de contraseñas	01-11-2024	31-11-2024
04	Creación funcionalidad gestión de notas	01-12-2025	31-12-2024
05	Implementación gestión de usuarios (UI)	01-01-2025	31-01-2025
06	Implementación gestión de contraseñas (UI)	01-04-2025	31-04-2025
07	Implementación de gestión de notas (UI)	01-03-2025	31-03-2025
08	Creación funcionalidad gestión de carpetas	01-04-2025	31-05-2025
09	Implementación de gestión de carpetas (UI)	01-05-2025	31-05-2025

Nota. Cronograma de actividades realizadas durante el desarrollo de la aplicación. Autoría propia (2025).

1. Planteamiento del problema

La gran mayoría de los usuarios enfrentan dificultades para gestionar sus credenciales digitales de forma segura, lo que conlleva al uso de contraseñas débiles o repetidas, aumentando el riesgo de ciberataques. Los gestores de contraseñas existentes pueden ser complejos para personas con bajo nivel tecnológico, lo que limita su adopción.

2. Formulación del proyecto

¿Cómo el desarrollo de un software gestor de contraseñas puede mejorar la seguridad digital y facilitar la gestión de credenciales en usuarios con diversos niveles de experiencia tecnológica?

3. Estado del arte

HIDDENPASS nace gracias a la seguridad del usuario, debido a que hace años se evidencia que la seguridad con las contraseñas de los usuarios ha sido vulnerable, esto se debe a que no tiene un buen manejo de estas mismas contraseñas o incluso tienen una contraseña poco segura o repetitiva.

Actualmente esto se ha ido solventado gracias a aplicaciones como KeePass2 o NordPass, las cuales han sido creadas especialmente para fortalecer esta seguridad. Estos programas cuentan con sus beneficios o sus propios problemas, pero en caso general este tipo de programas han ido solucionando la seguridad de estos usuarios.

Con HIDDENPASS se espera que los usuarios aumenten su propia seguridad, guardando sus datos privados o su información sensible de una manera robusta de seguridad con cifrado integrado, para que el usuario se sienta seguro al momento de guardar esta información.

4. Descripción del proyecto

HIDDENPASS es un sistema de gestión de contraseñas que permite almacenar notas, y a su vez permite generar y almacenar contraseñas de forma segura. HIDDENPASS nace por la

necesidad actual de mantener salvaguardada la información de las personas, porque hoy en día gran parte de la vida de una persona se encuentra digitalizada y por tanto expuesta a riesgo de filtraciones y accesos no autorizados.

5. Justificación

HIDDENPASS busca solucionar posibles amenazas a la privacidad de las cuentas de las personas. Mejorando la seguridad, gracias a parámetros tales como la utilización de contraseñas seguras y diferentes para cada cuenta, cifrado de las mismas evitando ataques como Keyloggers u otros. Este proyecto busca desarrollar un software gestor de contraseñas que sea accesible, intuitivo y seguro. Los beneficios esperados son la reducción general de posibles problemas de seguridad y la prevención de filtración de datos personales.

6. Objetivos

a. 6.1 General

Desarrollar una aplicación segura para escritorio y móvil nativo que permita el almacenamiento y gestión de contraseñas, mediante un sistema de cifrado de datos, la sincronización segura entre varios dispositivos y la generación de contraseñas únicas.

6.2 Específicos

Analizar los requerimientos para generar una solución para la recopilación de la información necesaria.

Diseñar la base de datos, codificar los módulos establecidos del sistema e implementar cifrados fuertes para asegurar que las contraseñas estén protegidas contra accesos no autorizados.

Proporcionar una herramienta para la creación de contraseñas fuertes y únicas, favoreciendo la sincronización segura de contraseñas entre múltiples dispositivos (escritorio y móvil).

Realizar pruebas a la solución.

7. Resultados

Limitaciones

Financiación económica para poder desplegar este proyecto a producción, haciendo uso de servidores para realizar copias de datos cifradas a través de los servidores.

Incompatibilidades y/o vulnerabilidades con el sistema.

Alcances

Tras finalizar la etapa de implementación se esperan tres tipos de resultados basándose en ámbitos de negocio y producto:

Resultados de negocio

Por parte de los usuarios les facilitará la gestión y les agregará seguridad al momento de guardar sus contraseñas y al momento de guardar algunas notas.

Resultados de producto

HIDDENPASS será segura, fácil de utilizar con interfaz amigable para todos los públicos objetivos, tendrá las funcionalidades esperadas en el ambiente de seguridad laboral. Este aplicativo también proporcionará estabilidad para garantizar que el producto sea lo más competitivo posible.

Para lograr este resultado HIDDEN PASS contará con:

Módulo de inicio de sesión para garantizar el acceso únicamente a personas autorizadas.

Módulo de registro de contraseñas.

Módulo de registros para facilitar la edición y adición de registros de una manera accesible.

Módulo de generación de contraseñas

Módulo de gestión de notas.

Resultados de proceso

Se deben de cumplir con los cronogramas previamente establecidos para el aplicativo, este mismo será depurado para prevenir cualquier tipo de error o problema.

8. Procedimiento

Descripción general del proceso de desarrollo Modelo de desarrollo

En esta ocasión se usó la metodología SCRUM, para iniciar y seguir el avance del proyecto, como la misma metodología se hará la planeación del Sprint, donde todos los involucrados en el equipo se reúnen para planificar el Sprint o el desarrollo, donde se decidirá qué requerimientos o tareas se asignarán a cada integrante del equipo.

Cada integrante deberá asignar el tiempo que crea prudente para llevar a cabo sus requerimientos; además, habrá reuniones de equipo de Scrum. A estas reuniones se les deben dedicar máximo 15 minutos diarios, y deberán ser siempre en herramientas como pueden ser Microsoft Teams, Google Meet entre otros. En ellas, cada integrante del equipo deberá responder tres simples preguntas. Esta metodología fue seleccionada porque permite saber más a profundidad el progreso actual del proyecto y en caso de haber complicaciones saber resolverlas a tiempo

Fases del proyecto

Concepción

HIDDENPASS surgió a partir de los mismos miembros del proyecto debido a que se ha detectado que el uso de múltiples contraseñas es una tarea bastante tediosa y muchas veces se

opta por medidas o alternativas que hablando de seguridad no son buenas, como llegar a usar repetidamente una misma credencial y/o usar credenciales “fáciles de recordar”.

Tras denotar de esta problemática llegó la idea de HIDDENPASS un gestor de contraseñas para hacer esta gestión más fácil y de una forma sencilla, a partir de esto se utilizó la técnica de recolección de requisitos, las encuestas, la cual fue compartida con múltiples personas que usan medios tecnológicos constantemente, gracias a esto se puede dar cuenta que esta problemática detectada si es una problemática real y pasa repetidamente.

Flujo de trabajo

El flujo de trabajo en el desarrollo de la aplicación HIDDENPASS se organiza en varias fases interrelacionadas entre sí, manteniendo una correlación lógica que abarca desde la planificación hasta el despliegue de la aplicación, a continuación, se describen las etapas principales y su interacción:

2. Actividades específicas por fase

Análisis de requisitos, Recolección

Gracias al uso de encuestas permitió hacer el levantamiento de requerimientos, lo que posteriormente llevó a crear el documento de requerimientos que contiene requerimientos funcionales como no funcionales.

A continuación, se encuentra la documentación de requisitos concertada por el grupo.

Tabla 2

Requerimientos funcionales

ID de requerimiento	Nombre	Característica	Descripción	Requerimiento no funcional	Prioridad

RF. 01	Crear cuenta	El sistema permitirá que un usuario pueda crear una cuenta.	El sistema permite escribir un correo y una contraseña para crear una cuenta.	RNF. 01 RNF. 02	Alta
RF. 02	Iniciar sesión	El sistema permitirá que un usuario inicie sesión	El sistema permite escribir su usuario y contraseña maestra.	RNF. 01 RNF. 02	Alta
RF. 03	Crear contraseña	El sistema permitirá que se creen y almacenen contraseñas.	El sistema permite la creación y almacenamiento de contraseñas.	RNF. 07 RNF. 06 RNF. 08	Alta
RF. 04	Crear nota	El sistema permitirá que se creen notas.	El sistema permite la creación y almacenamiento de notas.	RNF. 14 RNF. 15 RNF. 16	Alta
RF. 05	Generar contraseña	El sistema permite guardar los registros en una base de datos	Permite crear aleatoriamente una contraseña segura	RNF. 06 RNF. 07	Media
RF. 06	Gestión de contraseñas	El sistema permite realizar toda la gestión de todas las contraseñas.	Permitirá editar y eliminar contraseñas.	RNF. 07 RNF. 06 RNF. 08	Alta
RF. 07	Gestión de notas	El sistema permite realizar la gestión de notas	Permite agregar, editar y eliminar notas.	RNF. 14 RNF. 15 RNF. 16	Media
RF. 08	Crear carpetas	El sistema permite crear carpetas o divisiones.	El sistema permite la creación de carpetas para una mejor organización de sus	RNF. 18	Alta

			registros y/o notas.		
RF. 09	Gestión de carpetas	El sistema permite realizar la gestión de carpetas.	Permitirá editar y eliminar carpetas.	RNF. 18	Media
RF. 10	Gestión de usuarios	El sistema permite realizar la gestión de usuarios	Permitirá editar información del usuario.	RNF. 01 RNF. 02 RNF. 04	Alta
RF. 11	Recuperar contraseña maestra	El sistema permite la recuperación de la contraseña maestra.	Permitirá recuperar la contraseña maestra mediante un código personalizado.	RNF.01 RNF.02 RNF.03 RNF. O4	Media
RF. 12	Cerrar sesión	El sistema permite cerrar sesión activa.	Permitiá cerrar la sesión activa del usuario.	RNF.04 RNF.05	

Nota. Listado de requisitos funcionales de la aplicación datos tomados fuente propia (2025)

Tabla 3

Requerimientos no funcionales

ID requerimiento no funcional	Descripción
RNF. 01	La contraseña maestra de los usuarios no debe tener menos de 5 caracteres, deben tener un carácter especial y un número
RNF. 02	La contraseña maestra debe estar encriptada por el algoritmo SHA-256
RNF. 03	El programa debe estar accesible en todo momento que se necesite
RNF. 04	El sistema debe tener un fácil manejo
RNF. 05	El sistema tendrá una apariencia con los colores principales de la aplicación
RNF. 06	Las contraseñas en los registros deben estar cifradas por el algoritmo AES
RNF. 07	Al guardar una contraseña se debe guardar en la base de datos o en el dispositivo.
RNF. 08	Las contraseñas no pueden tener un nombre vacío
RNF. 09	El sistema debe responder rápido a las solicitudes.
RNF. 10	El sistema debe ser fácil de mantener y actualizar.

RNF. 11	El sistema debe cumplir con todas las leyes y reglamentos aplicables.
RNF. 12	El sistema debe escalar hacia arriba o abajo según sea necesario.
RNF. 13	El sistema debe estar protegido contra el acceso no autorizado.
RNF. 14	Las notas estarán cifradas
RNF. 15	Las notas no pueden tener un nombre vacío
RNF. 16	Las notas no pueden tener una prioridad vacía
RNF. 17	Las carpetas no pueden tener un nombre vacío

Nota. Listado requisitos no funcionales datos tomados fuente propia (2025).

Diseño

Diseño de la Arquitectura

HIDDENPASS se desarrolló en conjunto de varias tecnologías, las cuales permitieron hacer una aplicación segura, eficiente, escalable y con una buena experiencia de usuario.

Las principales herramientas y leguajes usados son los siguientes:

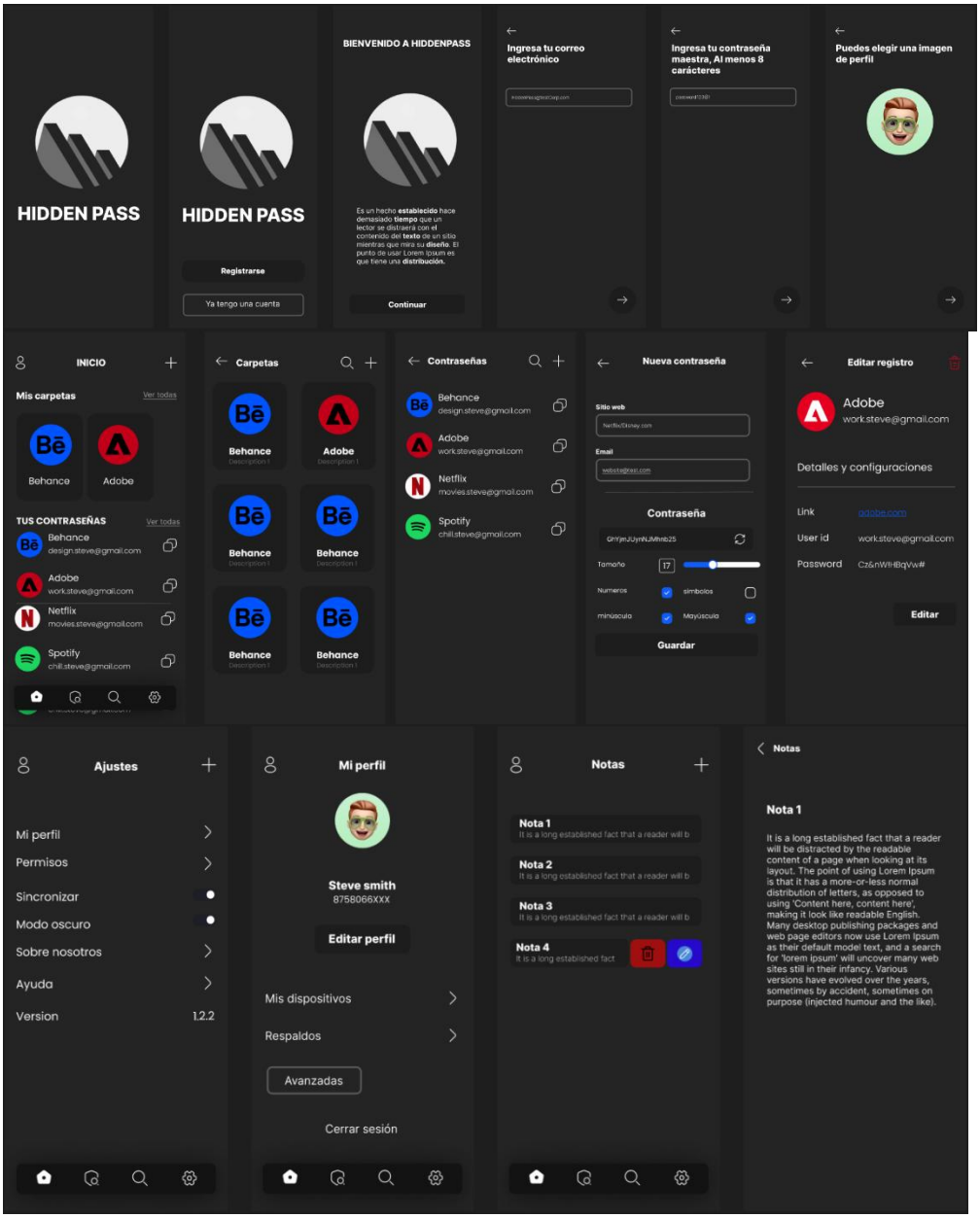
Diseño de Interfaz de Usuario (UI) y Experiencia de Usuario (UX)

Luego de contar con los requerimientos bien desarrollados y establecidos se procedió a definir los componentes, tecnologías y arquitecturas que se usarán en el desarrollo del proyecto. Además de esto se especifica y se desarrollan los mockups los cuales servirán para tener un diseño establecido para la interfaz de usuario, para este caso se usó Figma que cuenta con un excelente diseño y funcionalidades específicas para esta función.

Estos prototipos fueron un boceto principal con los cuales se puede basar para la aplicación real.

Figura 1

Mockups del aplicativo HIDDENPASS



Nota. Las imágenes muestran una visualización preliminar de la aplicación en móvil. Fuente propia (2025).

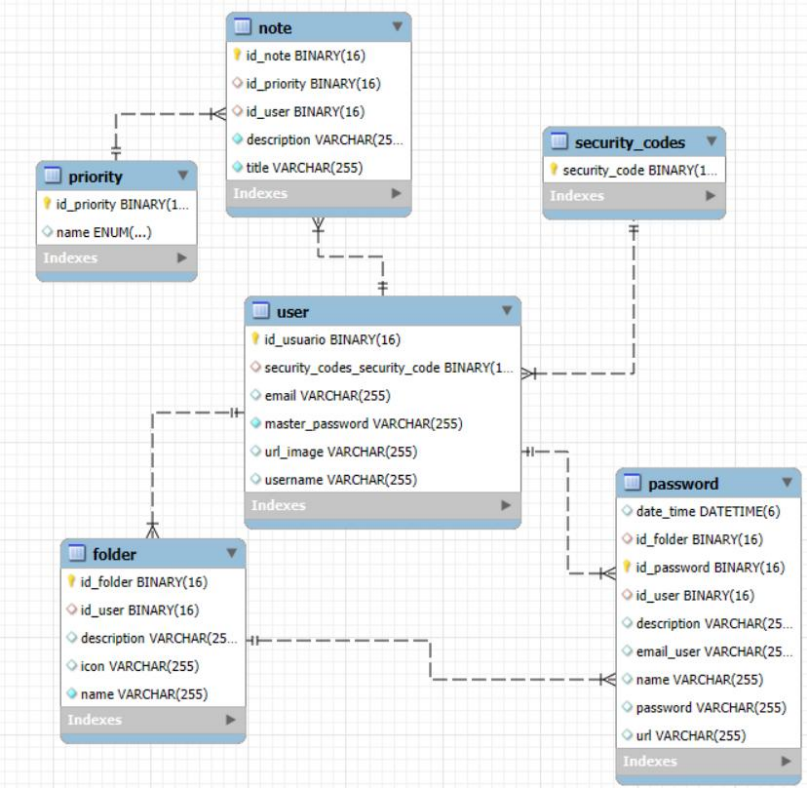
Diseño de la Base de Datos

Aparte de los Mockups se diseñó y se planteó el diseño de la base de datos relacional, la cual se basa en 6 tablas que están relacionadas entre sí con todos sus atributos, llaves primarias y

llaves foráneas.

Figura 2

Diagrama relacional de la base de datos.



Nota. Diagrama entidad relación. Autoría propia (2025)

Diseño de componentes

Interfaz de usuario e implementación

Flutter, framework de desarrollo multiplataforma para construir interfaces de usuario.

Dart, lenguaje de programación principal utilizado en Flutter.

Lógica del servidor

Springboot (Java), framework para el desarrollo de servicios web, facilita la creación de APIs seguras.

JWT, utilizado para la autenticación de usuarios.

Base de datos

MySQL, sistema de gestión de bases de datos relacional.

API REST

Arquitectura implementada para la comunicación entre la interfaz de usuario y la lógica.

Estándares de Codificación

Para los estándares se está usando una nomenclatura de Pascal Case para el nombre de las clases, además snake_case para nombrar las pantallas de la interfaz de usuario y por último se usa camel Case para el nombre de las funciones o métodos, además de esto para variables constantes todo está en mayúscula y usando snake_case.

Control de Versiones

Para el control de versiones se optó por usar Git con GitHub que contiene muchas funciones y permite una muy buena gestión de trabajo en equipo.

Revisiones de codigo

La utilización de GitHub nos permite analizar cada cambio realizado en el codigo, analizamos cada commit en busca de:

Errores lógicos obvios en el codigo.

Implementación de requisitos, se comprueba que se implementen completamente.

Análisis de directrices de estilo, se comprueba la correcta utilización de un diseño uniforme en toda la aplicación.

Pruebas

Para las pruebas solo se usarán en el Backend y se hará uso de pruebas unitarias con una cobertura mínima de 90% usando la técnica TLD la cual consiste en realizar las pruebas luego de tener la implementación de la lógica.

Despliegue

Para lograr un buen despliegue se hizo uso de diversos servicios web, los cuales prestan alojamiento web y alojamiento para bases de datos relacionales como MySQL, En el caso de HIDDENPASS solo hizo falta de estos servicios para desplegar el Backend porque para el despliegue del Frontend no se necesita alojamiento ya que se crea nativo para la plataforma.

Los servicios de alojamiento que se usó para el despliegue son dos, Render y Railway, Render para el alojamiento de la web mediante un contenedor Docker y Railway para el despliegue de la base de datos MySQL. Estos dos servicios se pensaron principalmente por sus costos y su interfaz de usuario, además por su integración con las tecnologías usadas en el Backend.

Para hacer el despliegue primero se necesitó construir todo el Backend en un solo archivo .jar, luego de esto se hizo uso de Docker para crear una imagen compatible con Render, ya en Render se conectó el repositorio de GitHub del proyecto con la plataforma. Un paso fundamental para lograr un despliegue exitoso es la creación y configuración de variables de entorno, debido a que sin estas variables el proyecto expondría información sensible y credenciales secretas las cuales solo deben ser visibles por el equipo de desarrollo, dentro de

estás credenciales se encuentra la conexión a la base de datos alojada en Railway, claves secretas para lograr el cifrado y encriptación dentro de la aplicación y la información necesaria para que el proyecto pueda iniciar. El despliegue en Railway es una tarea sencilla, porque solo se necesita la base de datos, sin ningún esquema de tablas debido a que la aplicación lo hace automáticamente al iniciarse, ya con la base de datos creada Railway proporciona las variables necesarias para conectarse y poder usar y manipular la base de datos

Mantenimiento

Para facilitar el mantenimiento a posteriori del proyecto, además de la utilización de convenciones al momento del desarrollo y la utilización de metodologías ordenadas descritas anteriormente, también se implementa en la aplicación la sección de ayuda, donde podrá enviar un correo reportando acciones correctivas, adaptativas o perfectivas que se deberán implementar en la aplicación siguiendo el proceso descrito a continuación.

Recepción del reporte, apartado en el cual se revisan los mensajes recibidos a través del correo electrónico.

Clasificación de la incidencia, paso en el cual se determina el tipo de mantenimiento requerido (Correctivo, adaptativo y/o perfectivo) y la prioridad relativa al problema, en este paso también se desestima dado el caso de que el problema sea ajeno a nuestro control (Problemas a nivel de sistema operativo, entre otros), enviando un correo respondiendo al usuario por que se ha desestimado.

Resolución, basándose en la clasificación de la incidencia y su prioridad se procede al desarrollo de la solución que corresponda.

Verificación, antes de realizar una actualización publica se efectúan pruebas para garantizar una correcta resolución de la incidencia.

Herramientas y tecnologías

Para llevar a cabo un desarrollo optimo y eficaz de HIDDENPASS, se implementan las siguientes tecnologías y herramientas de desarrollo:

GitHub: control de versiones para llevar a cabo un desarrollo ágil y coordinado.

Azure: control de actividades, asignación de roles.

Visual Studio Code: entorno de desarrollo usado para desarrollar la parte visual.

IntelliJ: entorno de desarrollo usado para desarrollar la parte lógica.

Dart: Lenguaje implementado para llevar a cabo la parte visual.

Java: lenguaje implementado para llevar a cabo la parte lógica.

Flutter: framework usado en conjunto con Dart para llevar a cabo la parte visual.

Spring Boot: framework usado en conjunto con Java para llevar a cabo la parte lógica.

Librerías Flutter: Hive, providers, material, http. dart, jsonwebtoken, jwt.

Librerías java: JPA, spring security, spring web, MySQL, lombok, jwt, jacoco

MySQL: motor de base de datos implementado para almacenar los registros.

Roles y responsabilidades

Para el desarrollo de HIDDENPASS se distribuyen los roles de forma equitativa entre los integrantes quedando así, la distribución:

David Fernando Gomez Aristizabal, analista, diseñador, programador.

Juan Andrés Menéndez Villarraga, analista, diseñador, programador.

Nicky Alexander Florez, analista, diseñador, programador, tester.

Criterios de aceptación

Para garantizar la funcionalidad y cumplimiento de los objetivos de HIDDENPASS, se definen los siguientes criterios de aceptación, permitiendo estos validar cada fase y entregable del desarrollo, asegurando que se ajusten a los requerimientos propuestos.

Fase 1: investigación y requisitos.

Se documentaron todos los requerimientos funcionales y no funcionales.

Se revisó la viabilidad del proyecto.

Se entregó un documento de análisis validado por el grupo de desarrollo y revisado por los instructores.

Fase 2: Desarrollo del sistema.

Las funcionalidades planteadas están implementadas.

El sistema puede ejecutar operaciones CRUD sin errores.

El Backend y Frontend están conectados correctamente.

Se verifica que HIDDENPASS cumpla con las normas de seguridad básicas (cifrado de contraseñas).

Fase 3: Interfaz visual

La interfaz es intuitiva y responsive.

El usuario puede navegar de forma fluida entre secciones.

Fase 4: Testeo de errores y validaciones

No se hayan errores críticos en las funcionalidades.

Todos los bugs detectados fueron corregidos.

Fase 5: Entrega final

Se entrega el código fuente completo, bien estructurado y documentado.

El manual técnico y de usuario está completo y actualizado.

El repositorio del proyecto está organizado y contiene README funcional.

Tabla 4

Costos del proyecto

Proveedor	Producto	Cantidad	Valor unitario	Valor total
Google	Dart	1	$0.00	$0.00
Google	Flutter	1	$0.00	$0.00
Google	Android Studio Community	3	$0.00	$0.00
Microsoft	Visual Studio Code	3	$0.00	$0.00
Microsoft	GitHub	3	$0.00	$0.00
Oracle	Java	1	$0.00	$0.00
JetBrains	Intellij IDEA Community	1	$0.00	$0.00
Render	Render hosting	1	$0.00	$0.00
Técnologo software		3	$2’000.000	$6’000.000

Nota: Listado de costos datos tomados fuente propia (2025).

Conclusiones

En conclusión, es posible mejorar la seguridad actual de las personas en la web mediante la utilización de contraseñas seguras que, además, mejoren el sentimiento de seguridad frente a la utilización de las diferentes herramientas web.

Mediante la utilización del lenguaje de programación Dart y el Framework Flutter los aprendices de la ficha pueden llevar a cabo el aplicativo con la capacidad de almacenar contraseñas y notas de manera organizada mediante carpetas, accesible en varias plataformas cono Android y Windows.

Creando entonces un aplicativo multiplataforma (Android y Windows) sencillo de utilizar y que cumple con los requerimientos, tanto funcionales como no funcionales, manteniendo y mejorando la seguridad de las cuentas de los usuarios.

Referencias

¿Qué es la Programación Orientada a Objetos (POO)? (2022). EDteam - En Español Nadie Te Explica Mejor. https://ed.team/blog/que-es-la-programacion-orientada-a-objetos-poo

¿Qué es un ORM y cómo funciona? (s. f.). EDteam - En Español Nadie Te Explica Mejor. https://ed.team/blog/que-es-un-orm-y-como-funciona

¿Qué es una API? - Explicación de interfaz de programación de aplicaciones - AWS. (s. f.). Amazon Web Services, Inc. https://aws.amazon.com/es/what-is/api/

¿Qué es una interfaz? | Concepto y Ejemplos. (s. f.). https://legsa.com.mx/pyru/interfaz

¿Qué son las pruebas unitarias?: explicación de las pruebas unitarias en AWS. (s. f.). Amazon Web Services, Inc. https://aws.amazon.com/es/what-is/unit-testing/

Acerca de GitHub y Git - documentación de GitHub. (s. f.). GitHub Docs. https://docs.github.com/es/get-started/start-your-journey/about-github-and-git

De Catalunya, U. O., & Cristina, D. T. A. (2012, 18 junio). Metodología Scrum. Empresas | INCIBE. (s. f.). https://www.incibe.es/empresas/blog/gestores-de-contrasenas-queson-y-como-pueden-mejorar-la-seguridad-de-las-empresas

Equipo editorial, Etecé. (2025, 11 febrero). *Sistema operativo Linux - Qué es, características y más*. Enciclopedia Humanidades. https://humanidades.com/sistema-operativo-linux/

Frisoli, C. (2025, 1 abril). Qué es Frontend y Backend: guía con diferencias, ejemplos y herramientas. *HubSpot*. https://blog.hubspot.es/website/frontend-y-backend

Gestores de contraseñas: qué son y cómo pueden mejorar la seguridad de las empresas |

hive - Dart API docs. (s. f.). https://pub.dev/documentation/hive/latest/

https://oa.upm.es/68361/

https://openaccess.uoc.edu/handle/10609/17885

Javier, M. R. (s. f.). Gestor centralizado de credenciales | Archivo Digital UPM.

Librería o biblioteca en programación: para qué sirven y tipos. (s. f.). EDUCAOPEN. https://www.educaopen.com/digital-lab/metaterminos/l/libreria

Magaña, L. M. L. (2018, 3 diciembre). Qué es Spring framework. *OpenWebinars.net*. https://openwebinars.net/blog/que-es-spring-framework/

Maldonado, R. (2024, 17 octubre). Java annotations y su diferencia con los comentarios. *KeepCoding Bootcamps*. https://keepcoding.io/blog/que-son-las-java-annotations/

Methods. (s. f.). Dart. https://dart.dev/language/methods

Métodos de petición HTTP - HTTP | MDN. (2025, 27 marzo). MDN Web Docs. https://developer.mozilla.org/es/docs/Web/HTTP/Reference/Methods

Rodríguez, A. S. A., Gómez, D. M. H., & Sierra, G. J. G. (2022). Algoritmo internacional de cifrado de datos (IDEA) que utiliza la variante de cifrado SHA-256. Dialnet. https://dialnet.unirioja.es/servlet/articulo?codigo=9020213

Sandoval, E. (2023, 21 septiembre). *Frameworks: Marcos de trabajo para programadores*. Ebac. https://ebac.mx/blog/frameworks

Vicente, Á. B. J., & De Valladolid Escuela de Ingeniería Informática de Segovia, U. (2019). Gestor de contraseñas seguras. Universidad de Valladolid. https://uvadoc.uva.es/handle/10324/36497