



Innovando la Seguridad de Contraseñas en la Era Digital

INSTITUCIÓN

Escuela Internacional De Estudios Superiores - INTER

Técnico Profesional En Programación De Software

Estudiante

Duvan Andres Florian Salazar

Docente

Brayan Stiven Restrepo

2024

Tabla de contenido

Título Alpha Web: Innovando la Seguridad de Contraseñas en la Era Digital.....	3
Introducción.....	3
Abstract.....	4
Pregunta Problema.....	4
Objetivo general.....	5
Objetivos específicos.....	5
Metodología.....	5
Cronograma del Proyecto.....	6
Marco Teórico.....	7
Marco Conceptual.....	9
Justificación.....	11
ISO/IEC 12207.....	11
Swetbook v4.....	13
Revolucionando la Seguridad de las Contraseñas en la Era Digital.....	13
La Importancia de un Codificador de Contraseñas.....	13
Cómo Swetbook Protege Tu Información.....	14
Diseño.....	16
Fase1:.....	16
Fase 2:.....	17
.....	20
.....	20
.....	20
Bibliografías.....	26

Título

Alpha Web: Innovando la Seguridad de Contraseñas en la Era Digital

Introducción

En el mundo digital de hoy, la seguridad de las contraseñas es más importante que nunca. Con tantos servicios en línea que requieren contraseñas recordarlas todas puede ser un desafío. Entra en juego la idea de un gestor de contraseñas seguro y cifrado. Este gestor no solo permitiría a los usuarios crear y almacenar contraseñas únicas y complejas para cada servicio en línea sin la necesidad de recordarlas todas, sino que también podría generar contraseñas aleatorias y robustas para garantizar la máxima seguridad. Dubanoia Adelin, 2023)

La importancia de un codificador de contraseñas radica en su capacidad para proteger la información sensible del usuario. Cuando las contraseñas se almacenan en su forma original, son vulnerables a los ataques de los ciberdelincuentes. Sin embargo, cuando se codifican, se vuelven incomprensibles y, por lo tanto, inútiles para cualquier persona que pueda obtener acceso no autorizado a ellas, para ello se debe tener en cuenta los siguientes puntos.

- 1 Un codificador de contraseñas es esencial en la actualidad para proteger la información sensible del usuario.
- 2 Transforma las contraseñas en una representación codificada, o 'hash', que es difícil de descifrar, haciendo que las contraseñas sean inútiles para los ciberdelincuentes.
- 3 El proceso de 'salting', que implica añadir datos aleatorios a la contraseña antes de codificarla, proporciona una capa adicional de seguridad.

Un codificador de contraseñas es crucial para proteger la información sensible del usuario en la era digital basado en el principio de la protección de información, además se encarga de transformar las contraseñas en una representación codificada, dificultando su descifrado.

En consecuencia, un codificador de contraseñas es esencial para proteger la información sensible del usuario. Este programa transforma las contraseñas en una representación codificada, dificultando su descifrado y haciéndolas inútiles para los ciberdelincuentes. El codificador de contraseñas protege la información personal del usuario, ayudando a prevenir el acceso no autorizado. Esto es especialmente relevante en nuestra sociedad cada vez más conectada, donde la seguridad de los datos es vital. Un buen codificador de contraseñas incorpora un proceso llamado 'salting', que añade datos aleatorios a la contraseña antes de codificarla. Esto proporciona una capa adicional de seguridad, haciendo que el hash sea aún más difícil de descifrar.

Abstract

In today's digital world, the security of our passwords is more important than ever. With so many online services requiring passwords to remember them all can be challenging, the idea of a secure and encrypted password manager comes into play. Not only would this manager allow users to create and store unique and complex passwords for each online service without the need to remember them all, but it could also generate random and robust passwords to ensure maximum security. Innovative solution promises to revolutionize the way we handle passwords and information online. (Dubanoaia Adelin, 2023)

The importance of a password encoder lies in its ability to protect sensitive user information. When passwords are stored in their original form, they are vulnerable to attacks by cybercriminals. However, when they are encoded, they become incomprehensible and therefore useless to anyone who might gain unauthorized access to them, for this the following points should be taken into account.

- 1 A password encoder is essential in today's digital age to protect sensitive user information.
- 2 Transforms passwords into a hard-coded representation, or 'hash', that is difficult to crack, making passwords useless to cybercriminals.
- 3 The process of 'salting', which involves adding random data to the password before it is encrypted, provides an additional layer of security.

A password encoder is crucial to protect the user's sensitive information in the digital age based on the principle of information protection, it is also responsible for transforming passwords into an encrypted representation, making it difficult to decipher.

In today's digital age, a password encoder is essential for protecting sensitive user information. This program transforms passwords into a coded representation, making them difficult to crack and rendering them useless to cybercriminals. The password encoder protects the user's personal information, helping to prevent unauthorized access. This is especially relevant in our increasingly connected society, where data security is vital. A good password encoder incorporates a process called 'salting', which adds random data to the password before it is hardcoded. This provides an extra layer of security, making the hash even more difficult to crack.

Pregunta Problema

¿Cómo mejoran los gestores de contraseñas seguros y cifrados la protección de la información de datos sensible para cada uno de los usuarios frente a ciberataques en la era actual?

Objetivo general

Programar una aplicación segura y eficiente que permita codificar contraseñas, para mejorar la seguridad de la información personal a partir del lenguaje de programación Python.

Objetivos específicos

- Establecer una aplicación de fácil acceso para el usuario, donde se implemente un algoritmo de cifrado con el lenguaje de Python, en el que cada usuario tenga una excelente interacción con sus contraseñas de manera práctica y precisa.
- Implementar un diseño gráfico para la aplicación que sea atractivo para el usuario, eficiente, práctico y preciso que genere un entorno confiable y un ecosistema de seguridad diferente al modelo estándar que se maneja en la actualidad.

Metodología

En el presente proyecto se desea utilizar la metodología Scrum dado su buen manejo y rendimiento en el que se ha fijado unos pasos basados en esta metodología la cual esta direccionada bajo los principios de procesos cortos y entregas rápidas de cada una de las actividades realizadas en un proyecto, estas entregas a corto plazo se les denomina como “SPRINTS” que traducido por la real academia española nos dice que es aceleración. En este orden de idea, el proyecto sera orientado en dirección de procesos de manera ágil, precisa y eficaz dando campo a la actualización constante de este mismo.

Si se trabaja en equipos de desarrollo de software, productos o ingeniería, o cerca de ellos, es probable que ya haya escuchado el término Scrum. Scrum es una metodología para la gestión de proyectos complejos en los que se necesita obtener un resultado rápido en entornos muy cambiantes. Incluso si no eres parte de un equipo de desarrollo de software, productos o ingeniería, también puedes aprovechar las ventajas de la metodología Scrum. Se trata de una de las metodologías de gestión de proyectos más modernas junto con Agile y el modelo Canvas. En este artículo, abordaremos todo lo que necesitas saber: qué es Scrum, cómo aplicarlo y por qué funciona tan bien. (Martins Julia, 2024)

Implementar la metodología Scrum en el proyecto de seguridad de contraseñas permitirá una gestión ágil y eficaz del desarrollo. Se puede asegurar una progresión constante y organizada de las tareas, el enfoque colaborativo y adaptativo de Scrum fomenta la comunicación abierta y la rápida respuesta a cualquier cambio o problema que surja. Esto es crucial para un proyecto que requiere tanto desarrollo técnico detallado como adaptación a nuevas amenazas de ciberseguridad. En resumen, Scrum proporciona una estructura dinámica para entregar un gestor de contraseñas seguro y cifrado, asegurando que el producto final sea robusto y fiable.

Cronograma del Proyecto

Fase	Fecha	Actividad
Fase 1: Investigación y Planificación (Agosto - Septiembre 2024)	01-15 agosto	Reunión inicial del equipo y definición de objetivos.
	16-31 Agosto	Búsqueda y revisión de literatura relacionada (artículos, informes, libros).
	01-15 Septiembre	Elaboración del marco teórico (seguridad de contraseñas y ciberseguridad).
	16-30 Septiembre	Definición del alcance del proyecto y herramientas a utilizar.
Fase 2: Diseño y Desarrollo Inicial (Octubre - Diciembre 2024)	01-31 Octubre	Diseño del sistema de codificación de contraseñas.
	01-30 Noviembre	Desarrollo del prototipo del gestor de contraseñas.
	01-31 Diciembre	Implementación de técnicas de Hashing y salting.
Fase 3: Pruebas y Ajustes (Enero - Febrero 2025)	01-15 Enero	Pruebas iniciales del sistema y recogida de feedback.
	16-31 Enero	Ajustes y mejoras según feedback inicial.
	01-15 Febrero	Pruebas de seguridad y evaluación de la efectividad del sistema.
	16-28 Febrero	Revisión y optimización del sistema.
Fase 4: Documentación y Preparación de la Entrega (Marzo - Abril 2025)	01-15 Marzo	Redacción de la documentación técnica y manual de usuario.
	16-31 Marzo	Preparación de presentaciones y materiales adicionales.
	01-15 Abril	Revisión final y corrección de errores.
	16-30 Abril	Simulaciones y ensayos de la presentación final.
Fase 5: Presentación y Entrega (Mayo 2025)	01-15 Mayo	Revisión final de todos los materiales y preparación de la entrega.
	16-31 Mayo	Presentación del proyecto y entrega oficial.

Marco Teórico

La seguridad de las contraseñas es fundamental para proteger la información sensible de los usuarios. Los codificadores de contraseñas, también conocidos como gestores de contraseñas, juegan un papel crucial en esta protección al transformar las contraseñas en representaciones codificadas que son difíciles de descifrar. Las contraseñas son la primera línea de defensa contra los cibercriminales. Según un estudio de Informe de Informe de Verizon sobre las investigaciones de fugas de datos 2023 Verizon Data Breach Investigations (DBIR/ investigaciones de violación de datos), el 60% de las violaciones relacionadas con la piratería informática utilizaron contraseñas robadas o poco seguras con la metodología de ingeniería social la que consiste en hacer que la persona de tu contraseña de manera inconsciente o de indicios de ella. La reutilización de contraseñas es una práctica común en los usuarios que aumenta el riesgo de violaciones de seguridad cada día (Bedard Tim,2023).

La codificación de contraseñas implica transformar una contraseña en una representación codificada o 'hash'. Este proceso hace que las contraseñas sean incomprensibles para cualquier persona que obtenga acceso no autorizado. Un buen codificador de contraseñas también incorpora el proceso de 'salting', que añade datos aleatorios a la contraseña antes de codificarla, proporcionando una capa adicional de seguridad a la hora de poder contar con una seguridad adicional se ha podido ver que el codificador de contraseñas a prestado un buen funcionamiento generando una confiabilidad en cada uno de los usuarios de cada una de las plataformas que se usan en la actualidad.

Un estudio realizado por Keeper Security destacó que las contraseñas vulneradas causaron el 80% de todas las violaciones de datos en 2019, resultando en pérdidas financieras tanto para empresas como para consumidores en el cual se demuestra la importancia de poder realizar una adaptación a las nuevas tendencias de seguridad informática dado que día a día los ciberdelincuentes se camuflan con cada uno de los avances tecnológicos (Cutler Anne,2024).

Kaspersky (2024) ha investigado ampliamente sobre las amenazas a la seguridad de las contraseñas y ha proporcionado consejos sobre cómo crear contraseñas seguras y únicas. En su investigación, Dubanoia (2023) enfatiza la importancia de los gestores de contraseñas seguros y cifrados para crear y almacenar contraseñas únicas y complejas, mejorando así la seguridad general de los usuarios en línea.

El uso de contraseñas exclusivas, la habilitación de la autenticación multifactor (MFA) y el guardado seguro de contraseñas en un gestor de contraseñas se consideran buenas prácticas relacionadas con las contraseñas. Lo que se denomina "higiene de contraseñas" encapsula las prácticas recomendadas en torno a la protección de contraseñas y cuentas en línea. Cuando se tiene una buena higiene de contraseñas, las contraseñas seguras evitarán que el usuario en cuestión se convierta en víctima de ataques cibernéticos, infecciones por virus y malware, vulneraciones de contraseñas y mucho más. D'Andrea Ashley (2024).

Recomendaciones y Buenas Prácticas

- 1 Utilice contraseñas seguras.
- 2 Utilice un gestor de contraseñas.
- 3 Habilite las claves de acceso siempre que estén disponibles.
- 4 Utilice la autenticación MFA para todas las cuentas.
- 5 No utilice la misma contraseña o variaciones de la misma para varias cuentas.
- 6 Evite compartir contraseñas de forma no segura.
- 7 Cambie sus contraseñas solo cuando sea necesario.

En el momento de la seguridad se debe poner en prácticas estos consejos los cuales ayudaran cada vez más al usuario poder gestionar sus contraseñas y seguridad ponerse al tanto de la actualidad y contar con un respaldo ya enfatizado a cada necesidad del individuo u organización la cual este contemplando mejorar sus redes de seguridad en la actualidad y el futuro.

Impacto Económico y la Necesidad de Adaptación Continua

Las consecuencias financieras de las violaciones de datos son sustanciales, con costos que incluyen pérdida de ingresos, daños a la reputación y gastos legales. Un enfoque proactivo en ciberseguridad no solo protege los activos de una organización, sino que también refuerza la confianza de los clientes y socios comerciales. Adaptarse continuamente a las nuevas amenazas y adoptar tecnologías emergentes es crucial para mantener una postura de seguridad eficaz.

A la hora de pensar en la inversión de la seguridad se suele caer en el error de para que voy a pagar algo que es caro, pero no se piensa en la prevención que se debe tener antes de poder adquirir un servicio para resguardar nuestra seguridad es muy importante poner atención a los datos que cada uno debe cuidar así sean privados o públicos no se debe pasar por alto este parámetro dado que en la actualidad los datos tienen un valor mas alto del que se tenía anteriormente.

Marco Conceptual

- Gestor de contraseñas

Un gestor de contraseñas es una aplicación encargada de almacenar contraseñas para un usuario de alguna plataforma digital. Estas contraseñas se guardan en una base de datos protegida por una contraseña maestra. Esta aplicación tiene el beneficio de ahorrar tiempo al almacenar o recordar una contraseña creada por el usuario para ello quiero resaltar seis puntos de usos esenciales los cuales son los siguientes:

1 Almacenamiento Seguro: Es la forma en la que se guardan todas las contraseñas en una base de datos que se encuentra protegida por una contraseña maestra.

2 Generación de Contraseñas: Es el encargado de crear contraseñas fuertes y únicas para cada una de las cuentas del usuario, mejorando la seguridad.

3 Autocompletado: Este proceso rellena automáticamente los campos de inicio de sesión en sitios web y aplicaciones.

4 Sincronización: Almacena tus contraseñas en múltiples dispositivos, permitiendo acceder a las cuentas desde cualquier lugar que el usuario desee.

5 Almacenamiento de Información Adicional: Guarda no solo contraseñas, sino también notas seguras, información de tarjetas de crédito y otros datos sensibles.

6 Alertas de Seguridad: Notifica si alguna de tus contraseñas ha sido comprometida en una brecha de seguridad.

- Seguridad de Contraseñas en la Era Digital

En el mundo digital actual, la seguridad de las contraseñas es crucial debido al creciente número de servicios en línea que requieren autenticación, A la hora de Recordar múltiples contraseñas se ha vuelto complejo dado que se usan diferentes contraseñas para diferentes servicios, lo cual se puede tornar difícil para los usuarios.

- Gestor de Contraseñas Seguro y Cifrado

Esta aplicación Permite a los usuarios crear y almacenar contraseñas únicas y complejas sin necesidad de recordarlas todas. Brindando la Capacidad para generar contraseñas aleatorias y robustas, garantizando la máxima seguridad de los datos del usuario.

- Codificación de Contraseñas

La codificación transforma las contraseñas en una representación incomprensible para los ciberdelincuentes, Las contraseñas almacenadas en su forma original son susceptibles a ataques.

ALPHA WEB

- Proceso de Hashing

Transformación de contraseñas en una representación codificada o 'hash'. Los hashes son difíciles de descifrar, haciendo que las contraseñas sean inútiles para los ciberdelincuentes.

- Proceso de Salting

Añadir datos aleatorios a la contraseña antes de codificarla. El salting proporciona una protección extra, haciendo que el hash sea aún más difícil de descifrar.

- Relevancia en la Sociedad Conectada

El codificador de contraseñas ayuda a prevenir el acceso no autorizado a la información personal del usuario. En una sociedad cada vez más conectada, la seguridad de los datos es esencial.

- Lenguaje de Programación

Un lenguaje de programación es una herramienta que usamos para comunicarnos con las computadoras y darles instrucciones. Hay diversos tipos por ejemplo Python, Java y C++. Se ve asociado al aprendizaje del ser humano en diferentes idiomas, pero en vez de comunicarte con personas, te comunicas con máquinas.

- Python

Python es un lenguaje de programación de alto nivel, conocido por su simplicidad y legibilidad. Fue creado por Guido van Rossum y lanzado por primera vez en 1991. Python se utiliza en una amplia variedad de aplicaciones, desde desarrollo web hasta análisis de datos y aprendizaje automático. Es muy popular debido a su sintaxis clara y su gran comunidad de usuarios, este lenguaje se caracteriza por la fácil comprensión a la hora de poder interactuar entre programadores dado que su sintaxis trata de igualar el lenguaje humano de la manera más precisa posible.

- Programar

Programar es el arte de escribir instrucciones claras y precisas para que una computadora realice tareas específicas. Estas instrucciones se escriben en lenguajes de programación, El Programar no solo implica dar órdenes a la máquina, sino también resolver problemas y crear soluciones tecnológicas de la manera más corta y precisa a la hora de ejecutar una solución a cualquier actividad específica dada por el programador.

- Programador

Es la persona o sujeto que se encarga de dar una orden o una instrucción específica a la computadora para generar una actividad en específico y dar solución a un problema estipulado es la persona encargada de poder brindar un soporte a cada una de las aplicaciones, paginas o diversidad de procesos tecnológicos que implican actividades de actualizaciones o modificaciones de un programa.

- Programa

Un programa es un conjunto de instrucciones escritas en un lenguaje de programación que le dice a una computadora cómo realizar una tarea específica. Desde las aplicaciones en tu teléfono hasta los videojuegos y los sistemas operativos, todos son programas. Pueden ser tan simples como una calculadora básica o tan complejos como un sistema de inteligencia artificial todas estas van articuladas la una de la otra.

Justificación

En la era digital actual, la seguridad de las contraseñas es más crucial que nunca debido al aumento de los ciberataques y la reutilización de contraseñas. Los gestores de contraseñas seguros y cifrados se presentan como una solución innovadora para este problema, permitiendo a los usuarios crear y almacenar contraseñas únicas y complejas sin la necesidad de recordarlas todas se ha vuelto en un apoyo a la hora de poder gestionar la información en diferentes plataformas que requieren información explícita y específica de cada uno de los usuarios, La importancia de un gestor de contraseñas radica en su capacidad para proteger la información sensible del usuario. Al transformar las contraseñas en una representación codificada o 'hash', estos gestores hacen que las contraseñas sean incomprensibles y por lo tanto, inútiles para cualquier persona que pueda obtener acceso no autorizado a ellas. Además, el proceso de 'salting', que añade datos aleatorios a la contraseña antes de codificarla, proporciona una capa adicional de seguridad, haciendo que el hash sea aún más difícil de descifrar.

La implementación de gestores de contraseñas en la era actual no solo mejora la seguridad de la información personal, sino que también facilita la gestión de múltiples contraseñas, reduciendo el riesgo de reutilización de contraseñas y aumentando la protección contra ciberataques. En un mundo cada vez más conectado, donde la seguridad de los datos es vital, los gestores de contraseñas seguros y cifrados se convierten en una herramienta esencial y prescindible para proteger la información sensible de los usuarios en todo momento. Esta justificación se basa en la necesidad creciente de proteger nuestras contraseñas, la efectividad demostrada de los gestores de contraseñas actualmente ha proporcionado una mejora en la seguridad digital.

ISO/IEC 12207

Integración de ISO/IEC 12207 en el Proyecto de Codificación de Contraseñas	
1. Definición del Proceso	Identificar y definir los procesos del ciclo de vida del software, como desarrollo, operación, mantenimiento y aseguramiento de la calidad.
2. Planificación del Proyecto	
Hitos	- Definición de requisitos (1 semana)
	- Diseño del sistema (2 semanas)
	- Desarrollo del prototipo (3 semanas)
	- Pruebas de seguridad (2 semanas)
	- Despliegue y evaluación (1 semana)
Recursos Necesarios	- Desarrolladores de software

ALPHA WEB

	- Herramientas de codificación y Hashing
	- Herramientas de pruebas de seguridad
Riesgos Potenciales	- Vulnerabilidades en el algoritmo de hash
	- Fallos en el proceso de 'salting'
	- Requerimientos de hardware/software insuficientes
3. Análisis de Requisitos	
Seguridad	- Longitud mínima de las contraseñas (8 caracteres)
	- Tipos de caracteres (alfanuméricos, símbolos)
	- Frecuencia de cambio de contraseñas (cada 90 días)
Funcionalidad	- Compatibilidad con diferentes sistemas operativos
	- Interfaz de usuario intuitiva
4. Diseño del Sistema	
Arquitectura	- Uso de algoritmos de hash seguros (SHA-256)
	- Implementación del proceso de 'salting'
Seguridad	- Encriptación de datos en tránsito y en reposo
	- Políticas de gestión de contraseñas
5. Implementación y Codificación	
Desarrollo	- Codificar el algoritmo de hash y el proceso de 'salting'
	- Usar bibliotecas de seguridad probadas
Mejores Prácticas	- Revisión de código
	- Programación defensiva
6. Pruebas y Validación	
Pruebas Unitarias	- Verificar que cada componente funcione correctamente
Pruebas de Integración	- Asegurarse de que los componentes funcionen juntos sin problemas
Pruebas de Penetración	- Simular ataques para identificar vulnerabilidades
Análisis de Vulnerabilidades	- Usar herramientas automatizadas para escanear posibles problemas

ALPHA WEB

	de seguridad
7. Despliegue y Mantenimiento	
Implementación	- Despliegue gradual en el sistema de producción
	- Monitoreo continuo de la aplicación
Actualizaciones	- Aplicar parches de seguridad regularmente
	- Mejoras basadas en el feedback del usuario
8. Documentación y Revisión	
Documentación	- Registro detalle

Swetbook v4

Revolucionando la Seguridad de las Contraseñas en la Era Digital

En el mundo digital de hoy, la seguridad de nuestras contraseñas es más importante que nunca. Con tantos servicios en línea que requieren contraseñas, recordarlas todas puede ser un desafío. Aquí es donde entra en juego Swetbook v4, un gestor de contraseñas seguro y cifrado. Este gestor no solo permite a los usuarios crear y almacenar contraseñas únicas y complejas para cada servicio en línea sin la necesidad de recordarlas todas, sino que también puede generar contraseñas aleatorias y robustas para garantizar la máxima seguridad. Esta innovadora solución promete revolucionar la forma en que manejamos contraseñas e información en línea (Dubanoaia Adelin, 2023).

La Importancia de un Codificador de Contraseñas

La importancia de un codificador de contraseñas radica en su capacidad para proteger la información sensible del usuario. Cuando las contraseñas se almacenan en su forma original, son vulnerables a los ataques de los ciberdelincuentes. Sin embargo, cuando se codifican, se vuelven incomprensibles y, por lo tanto, inútiles para cualquier persona que pueda obtener acceso no autorizado a ellas. Los siguientes puntos destacan su relevancia:

- Un codificador de contraseñas es esencial en la era digital actual para proteger la información sensible del usuario.
- Transforma las contraseñas en una representación codificada, o 'hash', que es difícil de descifrar, haciendo que las contraseñas sean inútiles para los ciberdelincuentes.
- El proceso de 'salting', que implica añadir datos aleatorios a la contraseña antes de codificarla, proporciona una capa adicional de seguridad.

Cómo Swetbook Protege Tu Información

Swetbook v4 se basa en el principio de la protección de información, transformando las contraseñas en una representación codificada que dificulta su descifrado. Este codificador de contraseñas protege la información personal del usuario, ayudando a prevenir el acceso no autorizado. Esto es especialmente relevante en nuestra sociedad cada vez más conectada, donde la seguridad de los datos es vital. Swetbook v4 incorpora un proceso llamado 'salting', que añade datos aleatorios a la contraseña antes de codificarla. Esto proporciona una capa adicional de seguridad, haciendo que el hash sea aún más difícil de descifrar.

KA 1: Planificación del Proyecto

Procesos y prácticas: Definir objetivos, alcance, cronograma y presupuesto del proyecto.

Herramientas y técnicas: Planificación de proyectos, gestión de riesgos, control de cambios.

Entradas y salidas: Documentación del proyecto, plan de proyecto, cronograma, presupuesto.

KA 2: Requisitos del Proyecto

Procesos y prácticas: Identificar y documentar los requisitos del proyecto.

Herramientas y técnicas: Análisis de requisitos, modelado de requisitos, prueba de requisitos.

Entradas y salidas: Documentación de requisitos, modelo de requisitos, plan de prueba.

KA 3: Diseño del Proyecto

Procesos y prácticas: Diseñar la solución del proyecto.

Herramientas y técnicas: Modelado de diseño, análisis de diseño, prueba de diseño.

Entradas y salidas: Documentación de diseño, modelo de diseño, plan de prueba.

KA 4: Implementación del Proyecto

Procesos y prácticas: Implementar la solución del proyecto.

Herramientas y técnicas: Desarrollo de software, prueba de software, depuración de software.

Entradas y salidas: Código fuente, plan de prueba, informe de pruebas.

KA 5: Prueba del Proyecto

Procesos y prácticas: Probar la solución del proyecto.

Herramientas y técnicas: Prueba de software, prueba de rendimiento, prueba de seguridad.

Entradas y salidas: Informe de pruebas, plan de prueba, resultados de pruebas.

KA 6: Deploy del Proyecto

Procesos y prácticas: Desplegar la solución del proyecto.

Herramientas y técnicas: Deploy de software, configuración de entorno, prueba de despliegue.

Entradas y salidas: Informe de despliegue, plan de despliegue, resultados de despliegue.

KA 7: Operación del Proyecto

Procesos y prácticas: Operar la solución del proyecto.

Herramientas y técnicas: Gestión de operaciones, monitoreo de rendimiento, gestión de incidentes.

Entradas y salidas: Informe de operaciones, plan de operaciones, resultados de operaciones.

KA 8: Mantenimiento del Proyecto

Procesos y prácticas: Mantener la solución del proyecto.

Herramientas y técnicas: Actualización de software, mantenimiento de hardware, gestión de cambios.

Entradas y salidas: Informe de mantenimiento, plan de mantenimiento, resultados de mantenimiento.

KA 9: Gestión de la Documentación

Procesos y prácticas: Crear y mantener la documentación del proyecto.

Herramientas y técnicas: Creación de documentación, mantenimiento de documentación, gestión de versiones.

Entradas y salidas: Documentación del proyecto, plan de documentación, resultados de documentación.

KA 10: Gestión de la Calidad

Procesos y prácticas: Garantizar la calidad de la solución del proyecto.

Herramientas y técnicas: Análisis de calidad, mejora de calidad, gestión de defectos.

Entradas y salidas: Informe de calidad, plan de calidad, resultados de calidad.

Diseño

Para el presente documento, se realizaron las fases de creación del aplicativo, en las cuales se detallan los diferentes aspectos necesarios para alcanzar los objetivos del proyecto. A continuación, se describen las fases clave que se llevaron a cabo:

Fase1:

Diagrama de Casos de Uso: Se identificaron los actores principales (Usuario y Sistema de Gestión de Contraseñas) y los casos de uso esenciales, como crear, almacenar, generar, consultar, modificar y eliminar contraseñas. Esto ayuda a entender las interacciones entre los usuarios y el sistema.

Diagrama de Clases: Se definieron las clases principales (Usuario, GestorContraseñas y Contraseña) con sus respectivos atributos y métodos. Esto proporciona una visión clara de la estructura del sistema y cómo interactúan las diferentes entidades.

Prototipo de Interfaz de Usuario: Se diseñaron las pantallas principales del sistema, incluyendo la pantalla de inicio de sesión, la pantalla principal con secciones para crear y gestionar contraseñas, y la pantalla de detalles de contraseña. Esto ayuda a visualizar cómo los usuarios interactuarán con el sistema.

Diagrama ER (Entidad-Relación): Se describieron las entidades principales (Usuario y Contraseña) y sus relaciones. Esto es crucial para el diseño de la base de datos, asegurando que las contraseñas estén correctamente asociadas a los usuarios.

Tipo de Diagrama	Descripción
Diagrama de Casos de Uso	Actores: Usuario, Sistema de Gestión de Contraseñas Casos de Uso: Crear una contraseña, Almacenar una contraseña, Generar una contraseña aleatoria, Consultar una contraseña, Modificar una contraseña, Eliminar una contraseña
Diagrama de Clases	Clases Principales: Usuario - Atributos: ID, <u>nombreUsuario</u> , <u>email</u> - Métodos: <u>registrar()</u> , <u>iniciarSesión()</u> , <u>cerrarSesión()</u> GestorContraseñas - Atributos: ID, <u>listaContraseñas</u> - Métodos: <u>añadirContraseña()</u> , <u>eliminarContraseña()</u> , <u>consultarContraseña()</u> Contraseña - Atributos: ID, <u>valorContraseña</u> , <u>fechaCreación</u> , <u>fechaModificación</u> - Métodos: <u>generarHash()</u> , <u>aplicarSalting()</u>
Prototipo de Interfaz de Usuario	Pantalla de Inicio de Sesión: Campos: <u>nombreUsuario</u> , <u>contraseña</u> - Botón: Iniciar Sesión, Registro Pantalla Principal: Sección de creación de contraseña, Sección para generar contraseñas aleatorias, Lista de contraseñas almacenadas con opciones para consultar, modificar y eliminar Pantalla de Detalles de Contraseña: Mostrar detalles de la contraseña, Opciones para modificar y eliminar
Diagrama ER (Entidad-Relación)	Entidades Principales: Usuario - Atributos: ID, <u>nombreUsuario</u> , <u>email</u> , <u>contraseña</u> Contraseña - Atributos: ID, <u>valorContraseña</u> , <u>fechaCreación</u> , <u>fechaModificación</u> , <u>usuarioID</u> (clave foránea) Relaciones: Un Usuario puede tener muchas Contraseñas, Cada Contraseña pertenece a un Usuario

Imagen 1/Tabla de Diagramas.

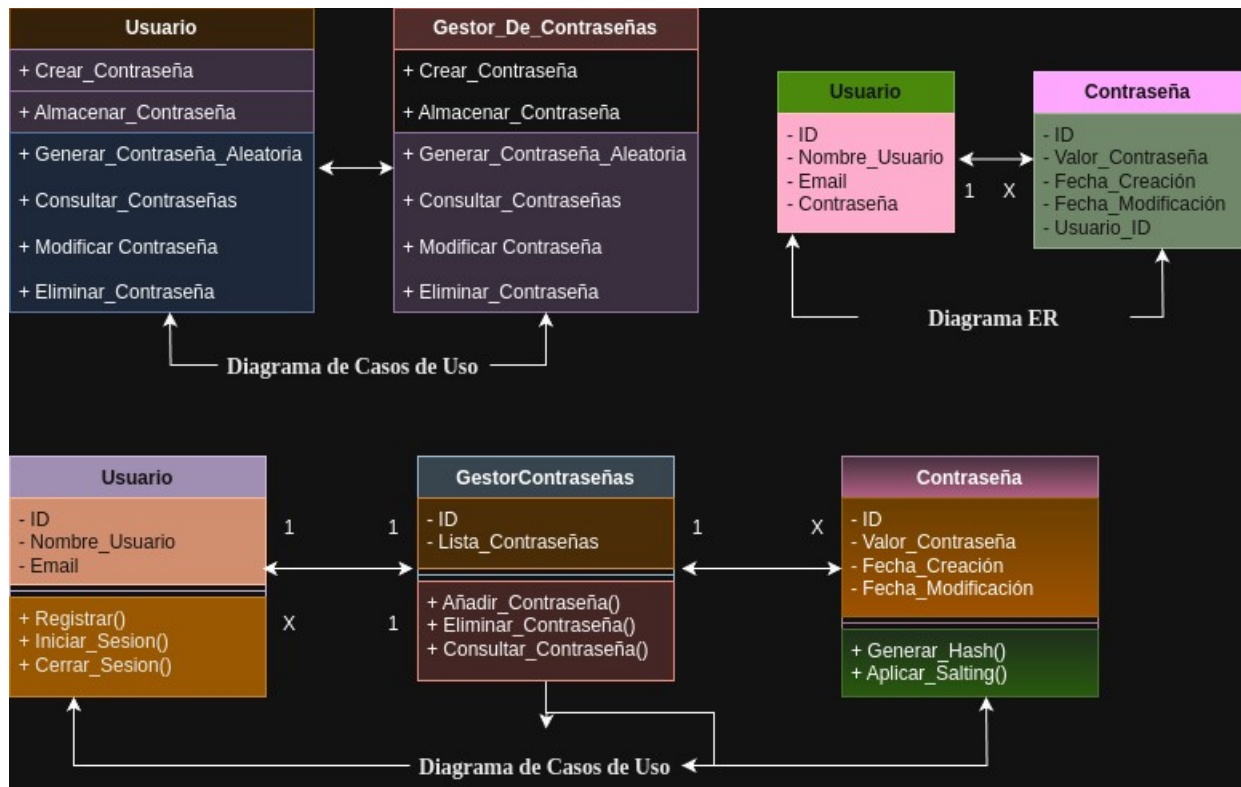


Imagen 2/Tipos de Diagramas.

Fase 2:

En esta fase, se implementó el diseño de diferentes diagramas de flujo, los cuales pasaron de ser creados en una tabla a ser organizados y categorizados. Este proceso permitió transformar un estilo inicial estipulado en un diseño concreto para cada una de las interfaces que se presentarán a continuación.

The screenshot shows a registration form with the following structure:

- INFORMACIÓN BÁSICA**
 - Nombre
 - Apellido
 - Edad
 - Género (Dropdown menu showing Masculino)
- INFORMACIÓN DE CONTACTO**
 - Email
 - Teléfono
- Enviar Ahora** (Button)

Imagen 3/Interfase de Registro.

ALPHA WEB

En esta imagen se puede observar el diseño implementado para la interfaz de registro, la cual permite al usuario ingresar por primera vez a la página. En este formulario, el usuario debe proporcionar información básica, como su nombre, apellido, edad, género, correo electrónico y número de teléfono. Estos datos serán registrados y almacenados en una base de datos para su posterior uso.

El diseño del formulario está organizado en secciones claras y bien definidas, lo que facilita su llenado. Además, todos los campos son obligatorios, lo que garantiza que la información ingresada sea completa y precisa. Esta interfaz está integrada con Django, utilizando plantillas y archivos estáticos para mantener un diseño coherente y funcional en toda la aplicación.

```
Alpha_web > Aplicacion > templates > dj Registro.html
1  {% extends "base.html"%}
2
3  {% load static %}
4
5  {% block content%}
6
7  <body>
8
9  <br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br>
10 <form action="/submit" method="post">
11     <fieldset>
12         <legend>INFORMACIÓN BÁSICA</legend>
13         <input type="text" name="nombre" placeholder="Nombre" required>
14         <input type="text" name="apellido" placeholder="Apellido" required>
15         <input type="number" name="edad" placeholder="Edad" required>
16         <label for="genero">Género</label>
17         <select id="genero" name="genero" required>
18             <option value="Masculino">Masculino</option>
19             <option value="Femenino">Femenino</option>
20         </select>
21     </fieldset>
22
23     <fieldset>
24         <legend>INFORMACIÓN DE CONTACTO</legend>
25         <input type="email" name="email" placeholder="Email" required>
26         <input type="text" name="telefono" placeholder="Teléfono" required>
27         <input type="file">
28     </fieldset>
29
30     <br><input type="submit" value="Enviar Ahora">
31 </form>
32 </center><br><br><br>
33
34 </body>
35 </html>
36 {% endblock %}
```

Imagen 4/Código de Registro.

El código comienza con `{% extends "base.html" %}`, lo que indica que esta plantilla hereda de una plantilla base (`base.html`). Esto permite reutilizar elementos comunes como el encabezado, el pie de página o estilos globales.

Bloque de contenido:

El contenido específico de esta página se define dentro del bloque {% block content %}, que será insertado en la plantilla base.

2. Formulario HTML

Campos de información básica: Se crea un formulario con campos para recopilar información básica del usuario, como:

ALPHA WEB

Nombre.

Apellido.

Edad.

Género (con un menú desplegable para seleccionar entre "Masculino" y "Femenino").

Campos de información de contacto: Se incluyen campos para recopilar datos de contacto, como:

Email.

Teléfono.

Un campo para subir archivos (input type="file").

Validación de campos: Todos los campos son obligatorios (required), lo que garantiza que el usuario no pueda enviar el formulario sin completar la información necesaria.

3. Diseño y Estilo

Espaciado: Se utilizan múltiples etiquetas
 para agregar espacios en blanco y separar visualmente el contenido. Sin embargo, esto no es una práctica recomendada para el diseño; en su lugar, deberían usarse estilos CSS.

Fieldset y leyendas: Los campos del formulario se agrupan en dos secciones (<fieldset>) con leyendas (<legend>) para mejorar la organización y la legibilidad.

4. Envío del Formulario

Método y acción: El formulario utiliza el método POST y se envía a la ruta /submit cuando el usuario hace clic en el botón "Enviar Ahora".

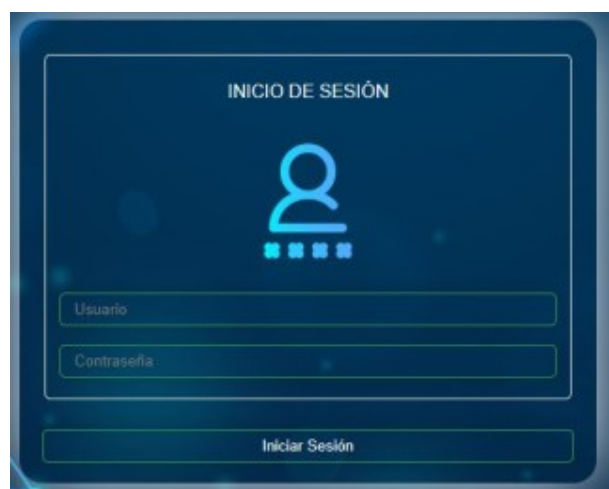


Imagen 5/Inicio de Sesión.

ALPHA WEB

En esta imagen se puede observar el diseño implementado para la interfaz de inicio de sesión, la cual permite al usuario acceder a la página ingresando sus credenciales. En este formulario, el usuario debe proporcionar su nombre de usuario y contraseña, datos que serán validados para permitir el acceso al sistema.

El diseño incluye una imagen relacionada con la seguridad de la contraseña, lo cual refuerza visualmente la importancia de proteger la información del usuario. Además, se han incorporado dos botones: uno para iniciar sesión y otro para registrarse, en caso de que el usuario no tenga una cuenta.

Todos los campos del formulario son obligatorios, lo que garantiza que el usuario complete la información necesaria antes de proceder. Este formulario está integrado con Django, utilizando plantillas y archivos estáticos para mantener un diseño coherente y funcional en toda la aplicación.

```
Alpha_web > Aplicacion > templates > dj_inicio_sesion.html
1  {% extends "base.html"%}
2  {% load static %}
3
4  {% block content%}
5
6  <!DOCTYPE html>
7  <html lang="es">
8
9  <head>
10 <meta charset="UTF-8">
11 <meta name="viewport" content="width=device-width,initial-scale=1.0">
12 <title>Inicio de Sesión</title>
13 <link rel="stylesheet" type="text/css" href="{% static 'css/Main4.css' %}">
14 </head>
15
16 <body>
17
18 <center>
19 <br><br><br><br><br> <br><br><br><br><br> <br><br><br><br><br>
20 <form action="/login" method="post">
21 <fieldset>
22 <br><center><legend>INICIO DE SESIÓN</legend></center>
23 
24 <input type="text" name="username" placeholder="Usuario" required>
25 <input type="password" name="contraseña" placeholder="Contraseña" required>
26 <input type="button" value="Inicio Sesión" onclick="window.location.href='Planes.html'">
27 <input type="button" value="Registrarse" onclick="window.location.href='Registro.html'">
28 </fieldset>
29 </form>
30 </center>
31
32 <br><br><br><br><br>
33 </body>
34 </html>
35 {% endblock %}
```

Imagen 6/Código de Inicio Sesión.

1. Formulario de Inicio de Sesión

Campos de inicio de sesión: Se crea un formulario con campos para que el usuario ingrese sus credenciales:

Usuario: Un campo de texto (input type="text") para el nombre de usuario.

Contraseña: Un campo de contraseña (input type="password") para ingresar la contraseña de manera segura.

Validación de campos: Ambos campos son obligatorios (required), lo que garantiza que el usuario no pueda enviar el formulario sin completar la información necesaria.

ALPHA WEB

Botones de acción:

Inicio Sesión: Un botón que redirige a la página Planes.html cuando se hace clic.

Registrarse: Un botón que redirige a la página Registro.html cuando se hace clic.

2. Diseño y Estilo

Espaciado: Se utilizan múltiples etiquetas `
` para agregar espacios en blanco y separar visualmente el contenido. Sin embargo, esto no es una práctica recomendada para el diseño; en su lugar, deberían usarse estilos CSS.

Fieldset y leyenda: Los campos del formulario se agrupan en una sección (`<fieldset>`) con una leyenda (`<legend>`) que indica que es un formulario de inicio de sesión.

Imagen: Se incluye una imagen (``) relacionada con la contraseña, que se carga desde la carpeta de archivos estáticos (`{% static 'Iconos/contraseña.jpg' %}`).

Estilos CSS: Se vincula una hoja de estilos externa (Main4.css) para aplicar estilos personalizados a la página.

3. Envío del Formulario

Método y acción: El formulario utiliza el método POST y se envía a la ruta `/login` cuando el usuario hace clic en el botón "Inicio Sesión". Sin embargo, en este caso, el botón no envía el formulario, sino que redirige a otra página (Planes.html). Esto debería corregirse para que el formulario se envíe correctamente.

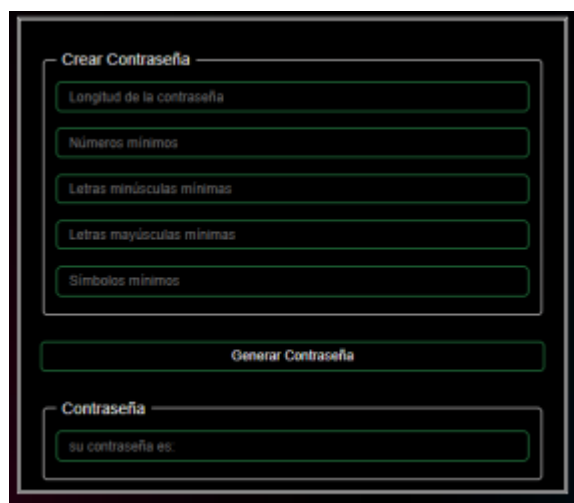
El formulario se encuentra dentro de un `<fieldset>` con el `<legend>` "Crear Contraseña". Contiene cinco campos de texto con el atributo `placeholder` para especificar requisitos: "Longitud de la contraseña", "Número mínimos", "Letras minúsculas mínimas", "Letras mayúsculas mínimas" y "Símbolos mínimos". Debajo de estos campos hay un botón "Generar Contraseña". En la parte inferior del `<fieldset>`, hay un `<div>` con el `<label>` "Contraseña" que precede a un campo de texto con el `placeholder` "su contraseña es:".

Imagen 7/Código de Inicio Sesión.

ALPHA WEB

En esta imagen se puede observar el diseño implementado para la interfaz de generación de contraseñas, la cual permite al usuario crear una contraseña segura y personalizada. En este formulario, el usuario puede definir parámetros específicos, como la longitud de la contraseña, la cantidad mínima de números, letras minúsculas, letras mayúsculas y símbolos. Estos datos son utilizados para generar una contraseña que cumpla con los requisitos de seguridad establecidos.

El diseño incluye un formulario organizado en secciones claras y bien definidas, lo que facilita su uso. Además, se ha incorporado un botón para generar la contraseña, la cual se muestra en un campo de texto una vez creada. Esto permite al usuario visualizar y copiar fácilmente la contraseña generada.

Todos los campos del formulario son obligatorios y cuentan con validaciones para garantizar que los valores ingresados sean válidos. Esta interfaz está integrada con Django, utilizando plantillas y archivos estáticos para mantener un diseño coherente y funcional en toda la aplicación. La herramienta es intuitiva y eficaz, ideal para cualquier sistema que requiera la creación de contraseñas seguras.

```
Alpha_web > Aplicacion > templates > dj_Contrasenas.html
1  {% extends "base.html"%}
2
3  {% load static %}
4
5  {% block content%}
6  <body>
7
8      <script>
9          function crearNumero() {
10             return String.fromCharCode(Math.floor(Math.random() * 10) + 48);
11         }
12
13         function crearLetraMinuscula() {
14             return String.fromCharCode(Math.floor(Math.random() * 26) + 97);
15         }
16
17         function crearLetraMayuscula() {
18             return String.fromCharCode(Math.floor(Math.random() * 26) + 65);
19         }
20
21         function crearSimbolo() {
22             const simbolos = "!@#$%^&*()_+[]{}|;:.,<=>?";
23             return simbolos[Math.floor(Math.random() * simbolos.length)];
24         }
25
26         function actualizarCampos() {
27             const charLength = parseInt(document.querySelector('input[name="char_length"]').value);
28             const maxNums = Math.floor(charLength * 0.2);
29             const maxMinus = Math.floor(charLength * 0.3);
30             const maxMayus = Math.floor(charLength * 0.3);
31             const maxSimbolos = Math.floor(charLength * 0.2);
32
33             document.querySelector('input[name="min_nums"]').max = maxNums;
34             document.querySelector('input[name="min_minus"]').max = maxMinus;
35             document.querySelector('input[name="min_mayus"]').max = maxMayus;
36             document.querySelector('input[name="min_simbolos"]').max = maxSimbolos;
37         }
38
39         function generarContraseña() {
40             const charLength = parseInt(document.querySelector('input[name="char_length"]').value);
```

Imagen 7/Código de Contraseña.


```

40     const charLength = parseInt(document.querySelector('input[name="char_length"]').value);
41     let minNums = parseInt(document.querySelector('input[name="min_nums"]').value);
42     let minMinus = parseInt(document.querySelector('input[name="min_minus"]').value);
43     let minMayus = parseInt(document.querySelector('input[name="min_mayus"]').value);
44     let minSimbolos = parseInt(document.querySelector('input[name="min_simbolos"]').value);
45
46     let password = [];
47
48     while (password.length < charLength) {
49         if (minNums > 0) {
50             password.push(crearNumero());
51             minNums--;
52         }
53         if (minMinus > 0 && password.length < charLength) {
54             password.push(crearLetraMinuscula());
55             minMinus--;
56         }
57         if (minMayus > 0 && password.length < charLength) {
58             password.push(crearLetraMayuscula());
59             minMayus--;
60         }
61         if (minSimbolos > 0 && password.length < charLength) {
62             password.push(crearSimbolo());
63             minSimbolos--;
64         }
65     }
66
67     // Completa la contraseña con caracteres aleatorios dentro del límite ya especificado
68     while (password.length < charLength) {
69         const functions = [crearNumero, crearLetraMinuscula, crearLetraMayuscula, crearSimbolo];
70         password.push(functions[Math.floor(Math.random() * functions.length)]());
71     }
72
73     document.querySelector('input[name="contraseña"]').value = password.join('');
74 }
75
76 document.addEventListener('DOMContentLoaded', () => {

```

Imagen 8/Código de Contraseña.

```

76     document.addEventListener('DOMContentLoaded', () => {
77         document.querySelector('input[name="char_length"]').addEventListener('input', actualizarCampos);
78     });
79 </script>
80 <form action="javascript:void(0);" onsubmit="generarContraseña()">
81     <fieldset>
82         <legend>Crear Contraseña</legend>
83         <input type="number" name="char_length" placeholder="Longitud de la contraseña" min="1" max="20" required>
84         <input type="number" name="min_nums" placeholder="Números mínimos" min="0" max="20" required>
85         <input type="number" name="min_minus" placeholder="Letras minúsculas mínimas" min="0" max="20" required>
86         <input type="number" name="min_mayus" placeholder="Letras mayúsculas mínimas" min="0" max="20" required>
87         <input type="number" name="min_simbolos" placeholder="Símbolos mínimos" min="0" max="20" required>
88     </fieldset>
89     <br><input type="submit" value="Generar Contraseña">
90     <fieldset>
91         <legend>Contraseña</legend>
92         <input type="text" name="contraseña" placeholder="su contraseña es: ">
93     </fieldset>
94 </form>
95 </center><br><br><br>
96 </body>
97 </html>
98 </endblock>

```

Imagen 9/Código de Contraseña.

1. Funcionalidad Principal

El objetivo principal del código es permitir al usuario generar una contraseña segura basada en parámetros específicos que él mismo define. Estos parámetros incluyen:

Longitud de la contraseña: El usuario puede elegir cuántos caracteres tendrá la contraseña (entre 1 y 20).

Cantidad mínima de números: Especifica cuántos dígitos debe contener la contraseña.

Cantidad mínima de letras minúsculas: Define el número mínimo de letras en minúscula.

Cantidad mínima de letras mayúsculas: Establece el número mínimo de letras en mayúscula.

Cantidad mínima de símbolos: Indica cuántos caracteres especiales (como !, @, #, etc.) debe incluir la contraseña.

2. Funciones JavaScript

Se implementaron varias funciones en JavaScript para lograr la generación de la contraseña:

`crearNumero()`: Genera un número aleatorio entre 0 y 9.

`crearLetraMinuscula()`: Genera una letra minúscula aleatoria.

`crearLetraMayuscula()`: Genera una letra mayúscula aleatoria.

`crearSimbolo()`: Selecciona un símbolo aleatorio de una lista predefinida.

`actualizarCampos()`: Ajusta los valores máximos de los campos de entrada según la longitud de la contraseña seleccionada.

`generarContraseña()`: Combina los caracteres generados aleatoriamente para crear una contraseña que cumpla con los requisitos especificados por el usuario.

3. Interfaz de Usuario

Formulario: Se creó un formulario con campos de entrada para que el usuario defina los parámetros de la contraseña.

Botón de generación: Al hacer clic en "Generar Contraseña", se ejecuta la función `generarContraseña()`, que muestra la contraseña generada en un campo de texto.

Validación: Todos los campos son obligatorios y tienen límites predefinidos para garantizar que los valores ingresados sean válidos.

4. Integración con Django

El código está integrado en una plantilla de Django (`{% extends "base.html" %}`), lo que permite reutilizar elementos comunes como el encabezado, el pie de página y los estilos globales. Se utilizan archivos estáticos (`{% load static %}`) para cargar recursos como imágenes o estilos CSS si fuera necesario.

5. Diseño y Usabilidad

Organización: El formulario está dividido en secciones claras (`<fieldset>`) para mejorar la legibilidad y la experiencia del usuario.

Interactividad: El formulario es dinámico, ya que los valores máximos de los campos se ajustan automáticamente según la longitud de la contraseña seleccionada.

Simplicidad: La interfaz es intuitiva y fácil de usar, lo que permite a los usuarios generar contraseñas seguras sin complicaciones.

Bibliografías

Adelin Octavian Dubanoaia, (2023). Trabajo Final de Grado, Gestor de Contraseñas, <http://hdl.handle.net/10234/205381>.

Ana Isabel Sordo, (2023). Blog, Metodología Scrum: qué es, cuáles son sus fases y cómo implementarla, <https://blog.hubspot.es/marketing/metodologia-scrum>.

Tim Bedard, (2023). Informe de Verizon sobre las investigaciones de fugas de datos 2023 Verizon Data Breach Investigations (DBIR) Las 3 conclusiones más importantes, <https://www.proofpoint.com/es/blog/takeaways-from-2023-verizon-data-breach-investigations-report>.

Anne Cutler, (2024). Informe de Enterprise Management Associates: Tendencias de 2024 y futuras en materia de seguridad de la información y conformidad, <https://www.keepersecurity.com/blog/es/2024/05/23/keeper-ema-report-infosec-and-compliance-trends/>.

Kaspersky, (2024). La mitad de las empresas latinoamericanas carece de personal calificado en ciberseguridad, revela Kaspersky, <https://latam.kaspersky.com/about/press-releases/la-mitad-de-las-empresas-latinoamericanas-carece-de-personal-calificado-en-ciberseguridad-revela-kaspersky?srsltid=AfmBOopi4Um-EV093ldeFU48gGwDin5oE71BKd53WENLQdNSrz7V3gUB>.

D'Andrea Ashley, (2024). Prácticas buenas y recomendadas en materia de contraseñas, <https://www.keepersecurity.com/blog/es/2024/07/23/password-hygiene-tips-and-best-practices/>.

Martins Julia, (2024). Scrum: conceptos clave y cómo se aplica en la gestión de proyectos, <https://asana.com/es/resources/what-is-scrum>.

<https://www.computer.org/education/bodies-of-knowledge/software-engineering>.

Innovando la Seguridad de Contraseñas en la Era Digital tiene licencia bajo CC POR 4.0 © 2 por DUVAN ANDRÉS FLORIAN SALAZAR 