



Proyecto Integrador

Nelson Israel Espinosa Rivera

Universidad internacional del Ecuador

2-ECC-1A: Lógica de programación

Jueves, 27 de marzo de 2025

El impacto de las nuevas tecnologías en la sociedad: visualización del futuro

1 Introducción

El presente proyecto integrador tiene como objetivo analizar y visualizar el impacto de las nuevas tecnologías en la sociedad, fomentando una reflexión profunda sobre su potencial futuro. Se integra contenido de las cuatro unidades de la asignatura, combinando teoría y práctica a través del desarrollo de un software orientado a la seguridad digital.

2 Descripción General del Problema

2.1 Contexto y Justificación

El avance constante de las nuevas tecnologías está transformando todos los ámbitos de la vida, desde la comunicación hasta la gestión de datos personales. Sin embargo, este desarrollo trae consigo retos importantes, como la seguridad de la información, especialmente en lo referente a la creación y manejo de contraseñas seguras.

2.2 Problema Central

La inseguridad de las contraseñas utilizadas por los usuarios representa un riesgo para la protección de la información personal y datos bancarios. Las vulnerabilidades en las contraseñas pueden ser explotadas por atacantes, lo que compromete la privacidad y seguridad digital de los usuarios.

3 Propósito del Proyecto

El proyecto tiene un doble propósito:

1. Visualizar y analizar el impacto de las nuevas tecnologías en la sociedad y sus posibles consecuencias futuras.
2. Desarrollar un software práctico: un generador de contraseñas seguro y personalizado, que sirva como ejemplo de la aplicación de conocimientos en programación funcional y de seguridad digital.

4 Alcance y Cronograma del Proyecto

El proyecto se desarrollará en un periodo de 8 semanas, integrando los 8 temas abordados en clase.

El cronograma es el siguiente:

Semana 1	Semana 2	Semana 3	Semana 4	Semana 5	Semana 6	Semana 7	Semana 8
Unidad 1		Unidad 2		Unidad 3		Unidad 4	
Tema 1	Tema 2	Tema 3	Tema 4	Tema 5	Tema 6	Tema 7	Tema 8
Proyecto integrador de 4 Unidades							
Actividad autónoma 1 Selección del programa a desarrollar / Generación de diagramas funcionales y arquitectura de Software. 1. Investigación de diagramas de funcionalidad y arquitectura. 2. Selección: diagrama de casos de uso, diagrama de componentes y diagrama de arquitectura y se escogió el generador de contraseñas seguras y en función de eso realicé los pasos de resolución de problemas aprendido, lo cual ya tuve la idea clara: que voy a resolver dentro de este problema 3. Fase de análisis anterior a la fase de diseño de las funcionalidades mostrado en los diagramas respectivamente. 4. Diseñe el diagrama de arquitectura de la aplicación a nivel macro como va a trabajar el software	Actividad autónoma 2 Inicio del Desarrollo de software/ Configuración del entorno 1. Configure el GitHub para cargar mi desarrollo 2. Desarrolle diagramas de flujo sobre las funcionalidades identificadas 3. Prepare el ambiente de desarrollo para iniciar con la etapa decodificación del sistema. 4. Entregue el avance respectivo de la codificación del desarrollo del software.			Actividad autónoma 3 Desarrollo del programa seleccionado 1. Estudiando los temas de la tercera unidad los implemente en el software que estaba desarrollando 2. Implemente funcionalidades y estructuras lógicas 3. Agregue e n la codificación manejo de estructuras repetitivas. 4. Agregue comentarios en las funcionalidades más complejas para su adecuado entendimiento.		Actividad autónoma 4 Foro para dudas, comentarios, presentaciones del proyecto, solicitud de las incorporaciones de información de los temas 7 y 8 previo a la entrega Aplicando técnicas de programación funcional 1. estudie los temas de la cuarta unidad y evalúe la codificación de otro estudiante. 2. Participe en varias ocasiones. en cada participación evalúe una opción de optimización del Código del compañero expuse mis dudas para recibir retroalimentación	Integración del proyecto, contemplando las 4 unidades. Entrega de un software funcional terminado 1. El proyecto completo está en mi repositorio de GitHub 2. El repositorio tiene un archivo radme que contiene la explicación de las principales funcionalidades del Código datos del grupo objetivo del programa, fecha 3. Introducción al proyecto. 4. El software integra los conocimientos de las 4 unidades 5. En la última clase presento el proyecto.

- 1 **Revisión de temas:** Investigación de tecnologías y análisis del impacto social.
- 2 **Diseño:** Elaboración de diagramas de funcionalidad y arquitectura (casos de uso, componentes, diagramas de flujo).
- 3 **Análisis del problema:** Identificación y comprensión de la inseguridad en contraseñas.
- 4 **Desarrollo inicial:** Investigación de información relevante a seguridad de un programa como el que voy a crear.
- 5 **Integración:** Implementación de la interfaz gráfica y módulos de seguridad, permitiendo la personalización de la contraseña, diseño y programación del generador de contraseñas (lógica de generación y validación).
- 6 **Pruebas y ajustes:** Ejecución de pruebas para asegurar el correcto funcionamiento y la seguridad del sistema.
- 7 **Documentación:** Elaboración del README, documentación técnica en el repositorio de GitHub; preparación de la presentación y el video demostrativo.
- 8 **Entrega final y presentación:** Integración completa del proyecto y exposición final en clase.

5 **Características del Proyecto Final**

Mi proyecto final incluirá:

- **Repositorio en GitHub:**
 - Código fuente completo del software.
 - Archivo README que contenga la explicación de las principales funcionalidades, objetivo del programa y fecha.
- **Software Integrado:**
 - Desarrollo de una aplicación (generador de contraseñas) que integre conceptos de las 4 unidades y técnicas de programación funcional. El cual llame Generador de contraseñas NIER

6 Desarrollo del Software: Generador de Contraseñas Seguro

6.1 Diseño y Diagramas (AA1)

Se elaboraron los siguientes diagramas para planificar la solución:

- **Diagrama de Casos de Uso:**
Representa las interacciones entre el usuario y el sistema.
- **Diagrama de Componentes:**
Detalla los módulos principales del sistema y sus interacciones.
- **Diagramas de Flujo:**
Muestran el proceso de creación, validación y generación de contraseñas.

6.2 Análisis y Resolución del Problema

1 Identificación del Problema

El desafío consiste en la inseguridad de las contraseñas actuales, que comprometen la protección de datos personales y financieros.

2 Comprensión del problema

Las contraseñas débiles son vulnerables a ataques como la fuerza bruta, permitiendo a los hackers acceder a información sensible.

3 Soluciones Alternativas

- **Solución 1:**
Generador de contraseñas simple, sin opciones de personalización (contraseñas aleatorias de longitud fija sin caracteres especiales).
- **Solución 2:**
Generador de contraseñas con interfaz gráfica, que permite al usuario personalizar la longitud y seleccionar caracteres, utilizando seguridad criptográfica.
- **Solución 3:**
Generador avanzado que incluya almacenamiento seguro de configuraciones y la opción de generar contraseñas basadas en patrones o frases.

4 Selección de la Mejor Solución

Se selecciona la **Solución 2**, ya que permite al usuario personalizar la generación de contraseñas a través de una interfaz intuitiva, garantizando a la vez la seguridad mediante algoritmos criptográficos.

5 Pasos para la Solución

1. Diseño de la Interfaz de Usuario:

- Realizar investigaciones para que el usuario seleccione la longitud de la contraseña y decida si incluir caracteres especiales.

2. Desarrollo del Generador de Contraseñas:

- Implementar la lógica de generación aleatoria combinando caracteres alfanuméricos y especiales, de acuerdo con las preferencias del usuario.

3. Integración del Módulo de Seguridad:

- Utilizar algoritmos de criptografía (la librería secrets en Python) para garantizar contraseñas seguras.

4. Pruebas del Sistema:

- Verificar el correcto funcionamiento del generador, la validación de parámetros y la robustez de las contraseñas generadas.

5. Despliegue:

- Implementar el sistema en un entorno de producción para que los usuarios puedan acceder a la herramienta sin inconvenientes.

6. Requisitos y Funcionalidades

- **Interfaz de Usuario Simple:**

Permite modificar el tema para una interfaz agradable para el usuario, permite ingresar parámetros de longitud y selección de caracteres.

- **Validación de Entrada:**

Evita errores, como una longitud o interacciones de contraseña inferior a lo establecido.

- **Opción de Copiado:**

Facilita copiar la contraseña generada al portapapeles.

- **Generación Aleatoria Segura:**

Basada en algoritmos que aseguran la creación de contraseñas fuertes.

7. Evaluación de la Solución

La solución propuesta garantiza que los usuarios puedan generar contraseñas seguras, reduciendo significativamente el riesgo de hackeos y protegiendo su información personal.

Se han implementado mecanismos de validación y seguridad que cumplen con los estándares actuales. Adicionando temas para que el usuario pueda sentirse cómodo con el software mientras crea seguridad en su entorno digital.

7 Documentación y Recursos

1. Repositorio en GitHub

El proyecto se encuentra alojado en el siguiente repositorio:

- Generador de Contraseñas - GitHub: <https://github.com/IsraelRiveraSxEc/GenDCCorePy>

El repositorio incluye:

- Código fuente completo.
- Diagramas (casos de uso, arquitectura, componentes, flujos).
- Archivo README con la descripción de funcionalidades, datos del grupo, objetivos y fecha.
- Instrucciones de instalación y uso.

2. Informe Final

El informe contiene

- Descripción detallada del desarrollo del software.
- Principios de diseño aplicados.
- Explicación de la lógica de programación y validación implementada.
- Análisis de diagramas y pruebas realizadas.
- Reflexión sobre el proceso de aprendizaje.

8 Conclusiones y Reflexiones

El desarrollo de este proyecto ha permitido:

- Integrar y aplicar conocimientos teóricos y prácticos de las 4 unidades de la asignatura.
- Desarrollar un software que aborda un problema real de seguridad digital, demostrando la importancia de contraseñas robustas en la protección de información.
- Fomentar mi aprendizaje autónomo y la colaboración entre compañeros, en la participación del foro de discusión (aprendizaje autónomo 4).
- Comprender la relevancia de la planificación y el uso correcto de diagramas según el proyecto que se va a realizar, para estructurar y ejecutar un proyecto de software funcional.

Se concluye que la combinación de metodologías, programación funcional y diseño orientado a la seguridad permite desarrollar soluciones innovadoras y eficientes, preparándonos para afrontar los desafíos que presenta el futuro en el ámbito de las nuevas tecnologías.

Este documento integra los requerimientos establecidos para el proyecto y refleja el proceso de análisis, diseño, desarrollo y evaluación de una solución orientada a mejorar la seguridad digital. Se evidencia el compromiso con el aprendizaje autónomo y la aplicación práctica de los conocimientos adquiridos en clase.