Carta de empresa

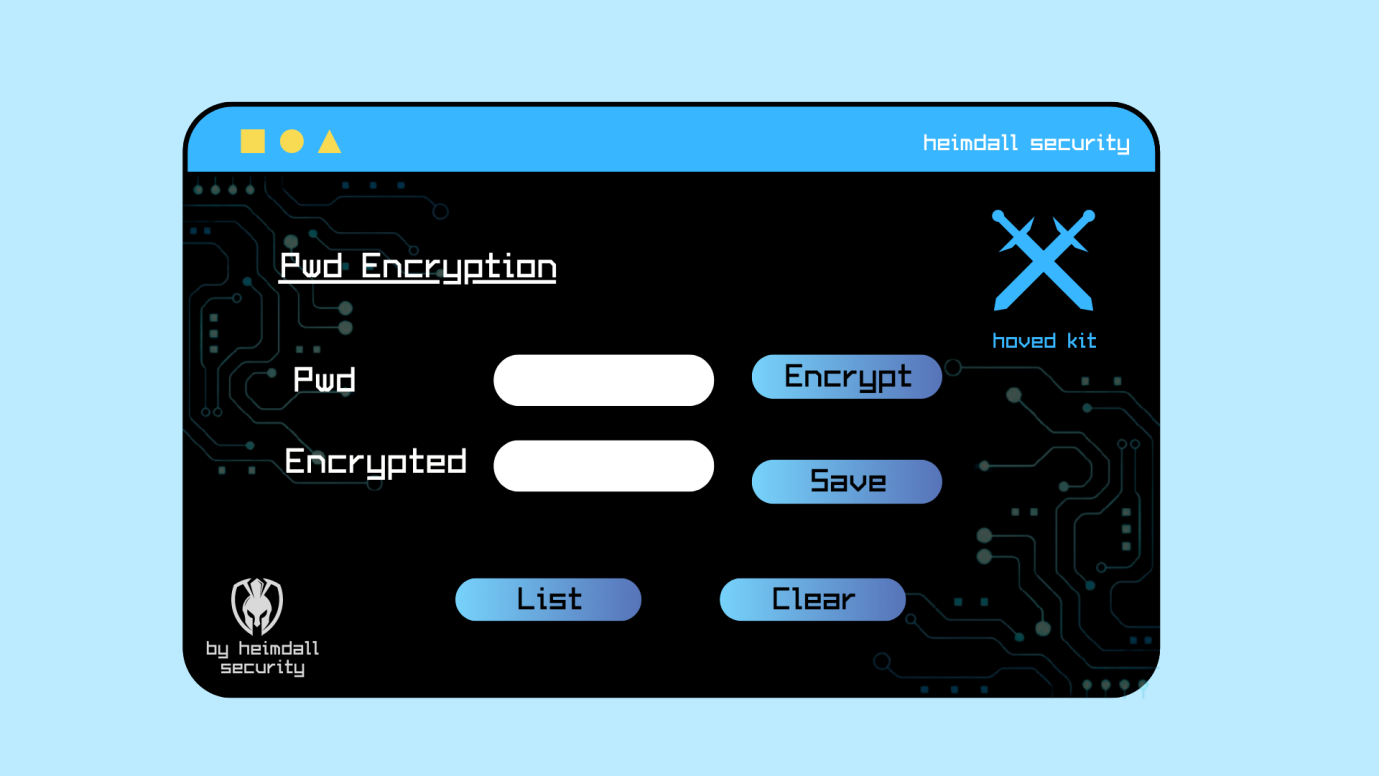
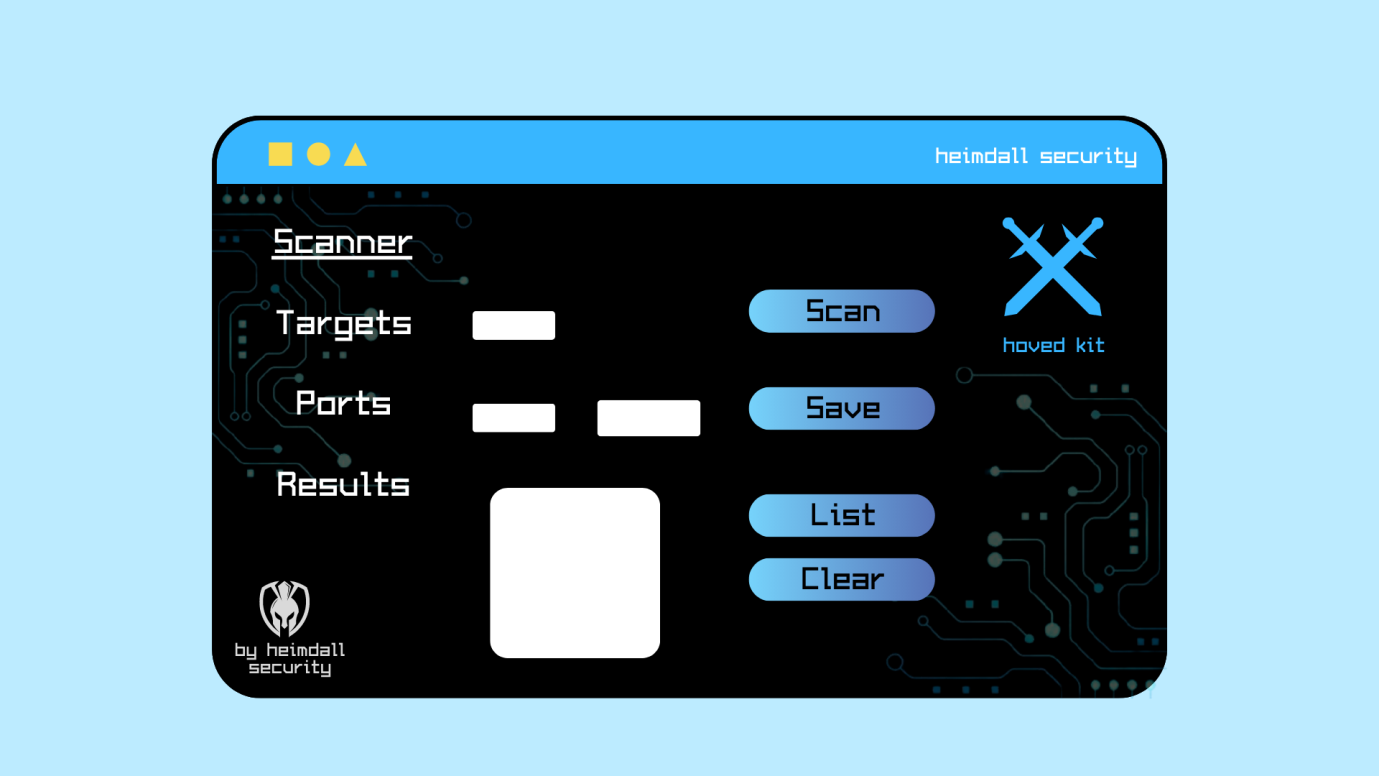
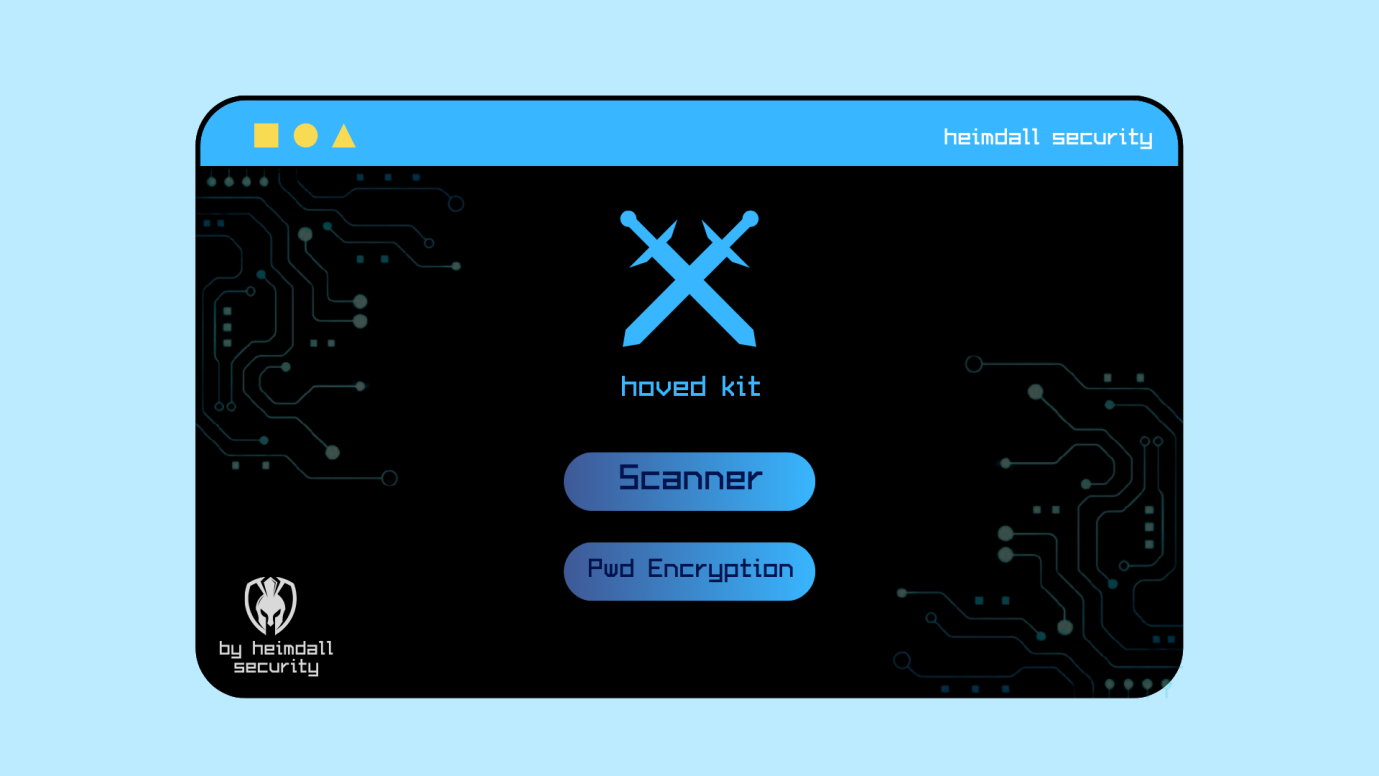
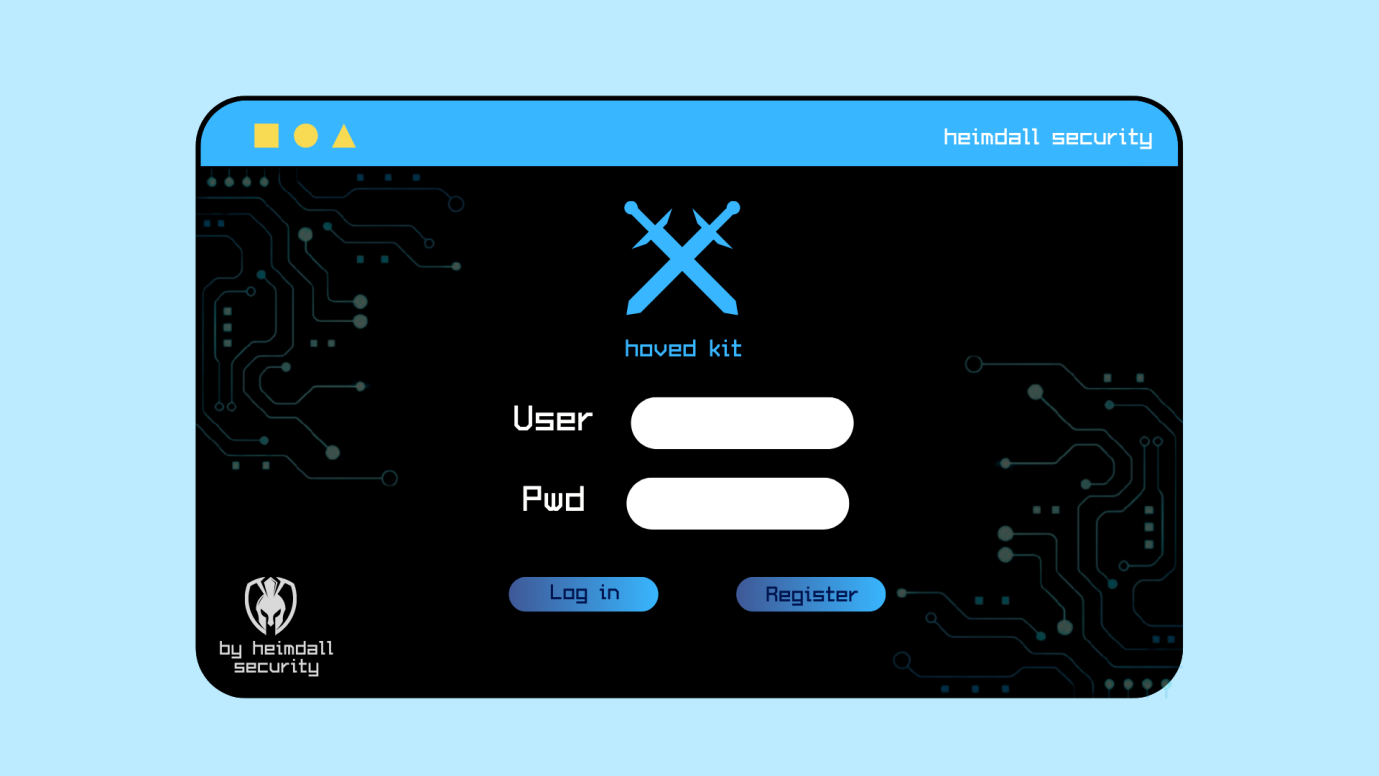
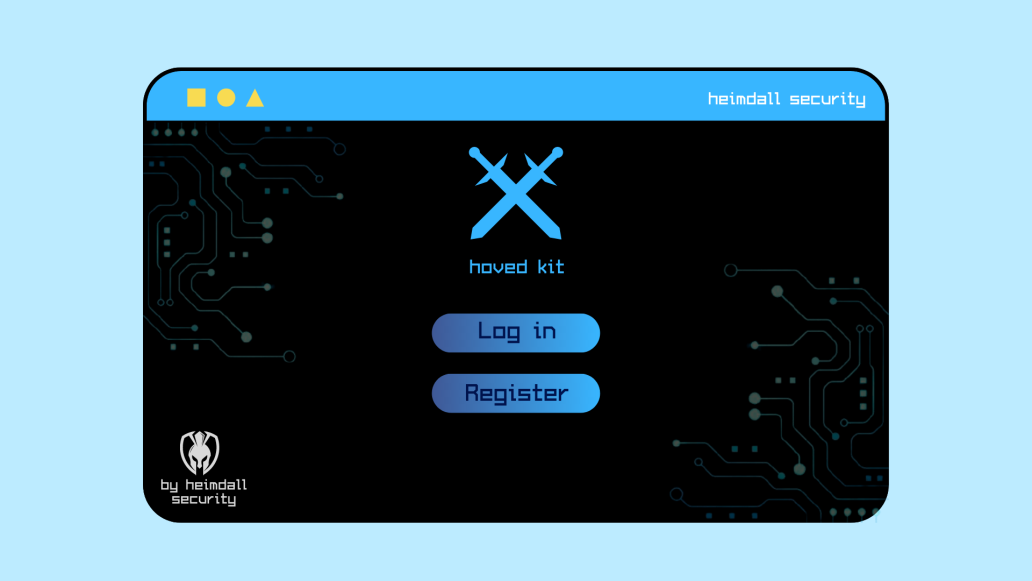
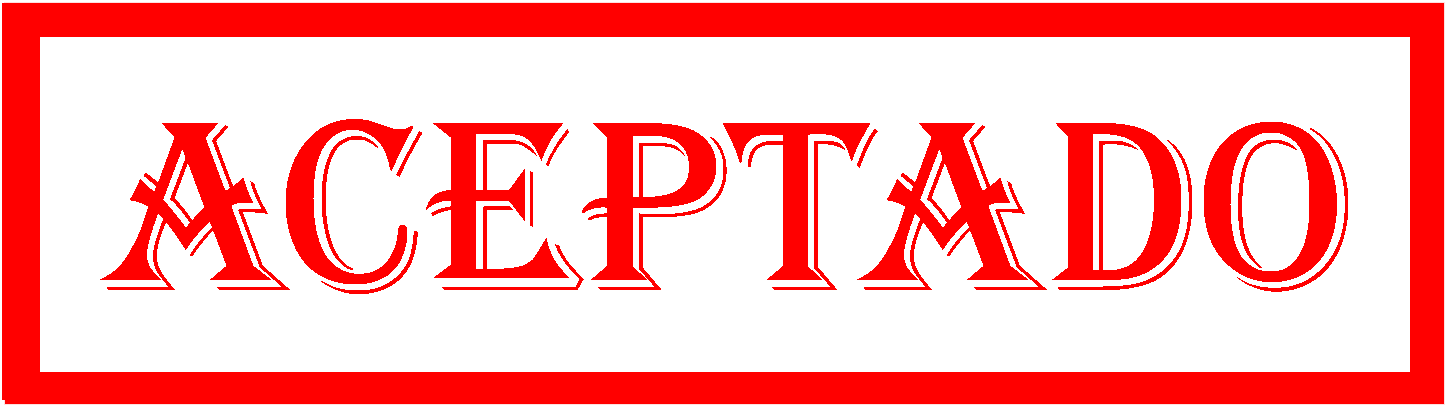
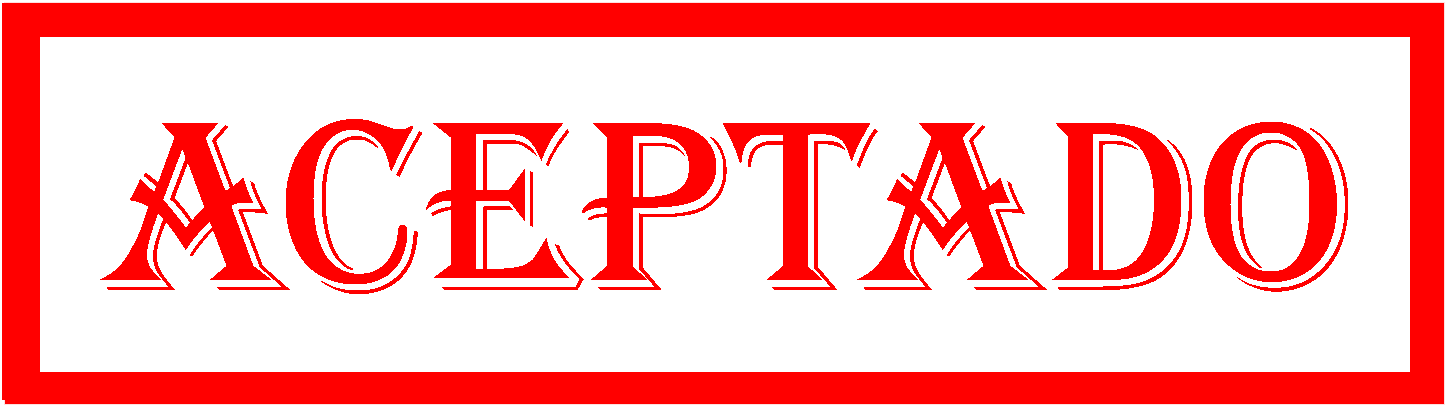
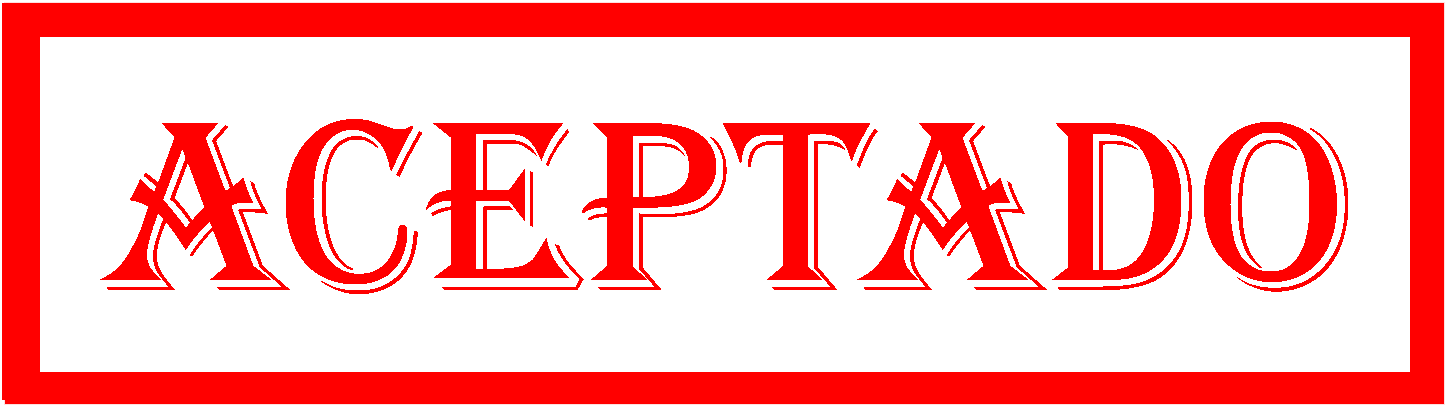
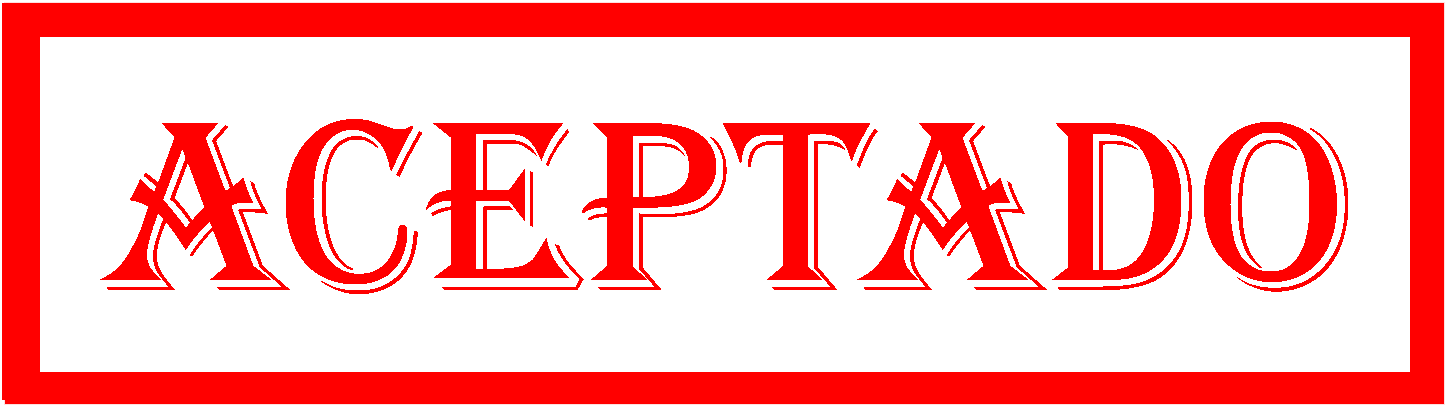
*Esta es la versión de texto, adjunto la presentación que había preparado en pdf*

* Nombre de la empresa desarrolladora: Heimdall Security
* Logotipo de la empresa desarrolladora:
* 
* Nombre del proyecto a desarrollar (el nombre debe estar en función del producto que se desea realizar el desarrollo de Interface): Hoved kit
* Logotipo del Producto a desarrollar:
* 
* Nombre de la empresa para la que se desarrolla el Interface: Pentesters Inc.
* Justificación de los colores y formas escogidos en la imagen corporativa:
  + Se trata de una empresa de ciberseguridad y el producto es un toolkit que ofrece una serie de herramientas sencillas pero útiles para agilizar el trabajo de un Pentester o de un BackendDeveloper de una empresa informática. La aplicación tiene diversas funciones, como realizar escaneosde los puertos de la máquina (guardando una lista de los escaneos realizados) o de encriptar contraseñas.
  + El nombre de la empresa, Heimdall, se refiere a la deidad nórdica del mismo nombre que representa la seguridad, por eso el logotipo es un casco de batalla. El nombre de la aplicación es el nombre de la espada de dicho dios nórdico, de ahí el logotipo correspondiente.
  + Teoría de colores:
    - Negro = Misterio ( #000000)
    - Blanco = Luz, claridad (#FFFFFF)
    - Azul = Seriedad, serenidad (#1b8de2)
    - La combinación de colores de la imagen corporativa pretende transmitir que la empresa (nombre y logo) brinda claridad ante la oscuridad del fondo. La del producto pretende evocar confianza en la seguridad del producto.

FASE 1: INICIO

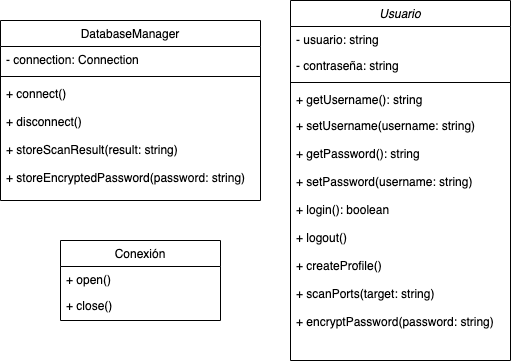
* Objetivos:
  + Generales: Agilizar el trabajo de desarrolladores con herramientas y listas
  + Específicos: Realizar una app que pueda escanear los puertos de una máquina, encriptar contraseñas.y guardar estos datos
  + Limitaciones:
    - fecha limite 09/06
    - economía: 6000
    - técnicas: mis conocimientos
  + Audiencia:
    - El usuario medio serán hombres (85%) de entre 25-47 años, mayoritariamente Europeos/Estadounidenses o Asiáticos(70%), con mucho conocimiento sobre informática, que sepan manejar la lengua inglesa y cuyos equipos tengan capacidades suficientes para utilizar este software (Windows).
  + Competencia
    - Queremos la fiabilidad y rapidez de la competencia
    - Mejoramos porque ofrecen una herramienta sola y sin logs de uso
    - Suelen ser herramientas de consola, ofrecemos una app aparte que otorga facilidad y que no es tan intimidante para el usuario medio
  + Funcionalidades
    - Instalador
    - Guías de usuario
    - Interfaz similar y amigable
    - Uso sencillo para el usuario medio
    - Control de errores sencillo y comprensible por el usuario
  + No es necesario hacer mantenimiento

FASE 2: CRECIMIENTO

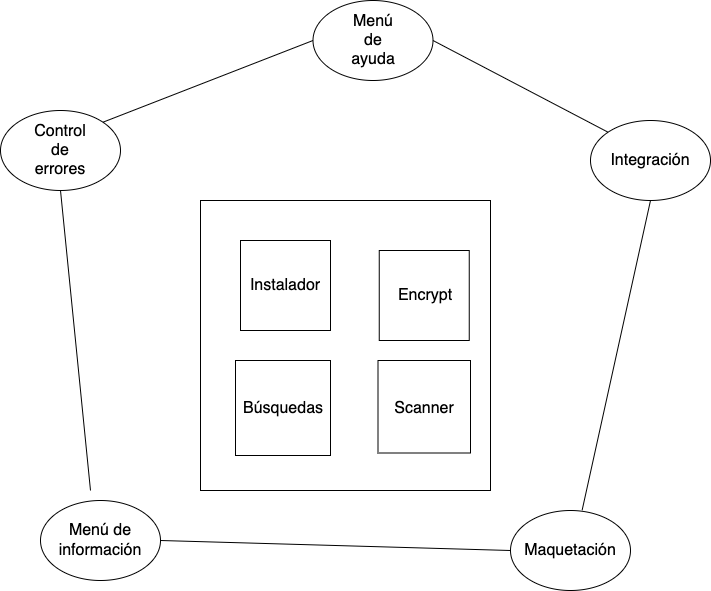
* Se trata de una aplicación CRUD con soporte en base de datos que permite guardar, editar y listar listas de contraseñas y escáneres.
* Requisitos mínimos Hardware y software:
  + Windows 7 , Mac OS X 10.11 (64 bit), Linux: RHEL 6/7 (64-bit)
  + x86 64-bit CPU (Intel / AMD architecture) no ARM
  + 4 GB RAM
  + 5 GB espacio libre de disco
* Herramientas utilizadas: Python 3 (IDE:PYCharm, Tkinter), MySQL, Canva
* Se usa Python al ser un lenguaje orientado a objetos, sencillo, moderno y maleable
* Se utiliza MacOs Ventura y Windows 11.
* Idealmente utilizara la metodología de planificación en espiral
* 6000
* Navegación por pantallas:
* Fase 0 – 1 y 2 aceptadas
* 
* 
* 

FASE 3: ANÁLISIS

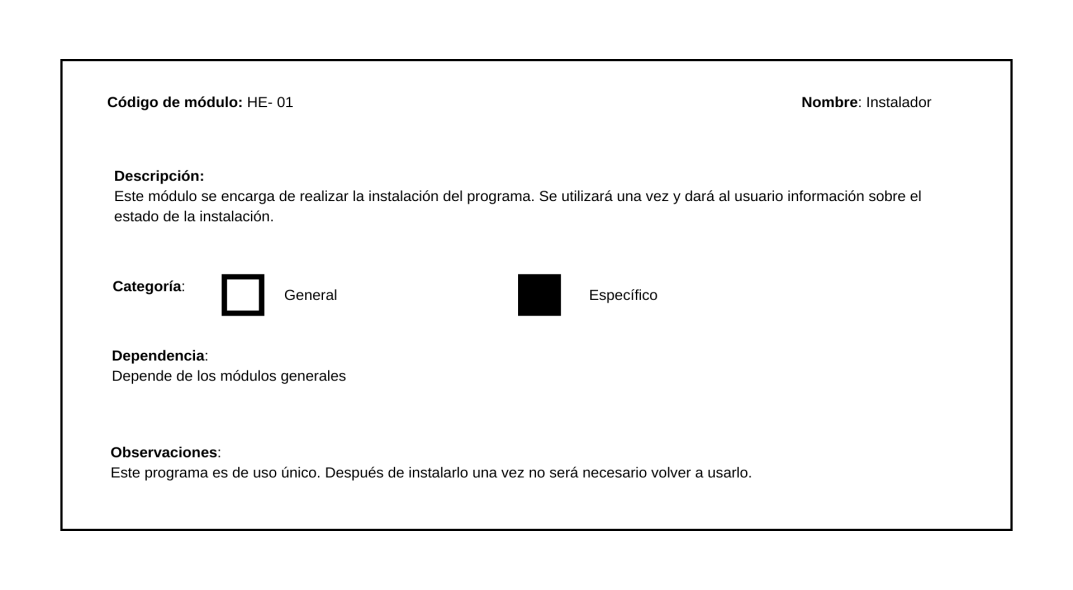
* Diagrama de clases:

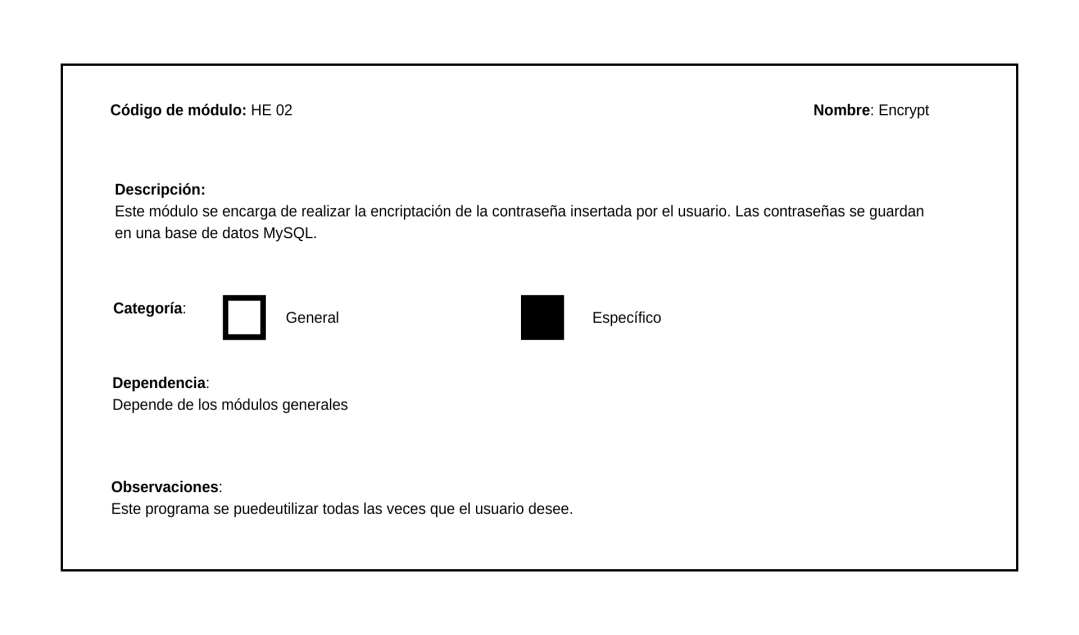


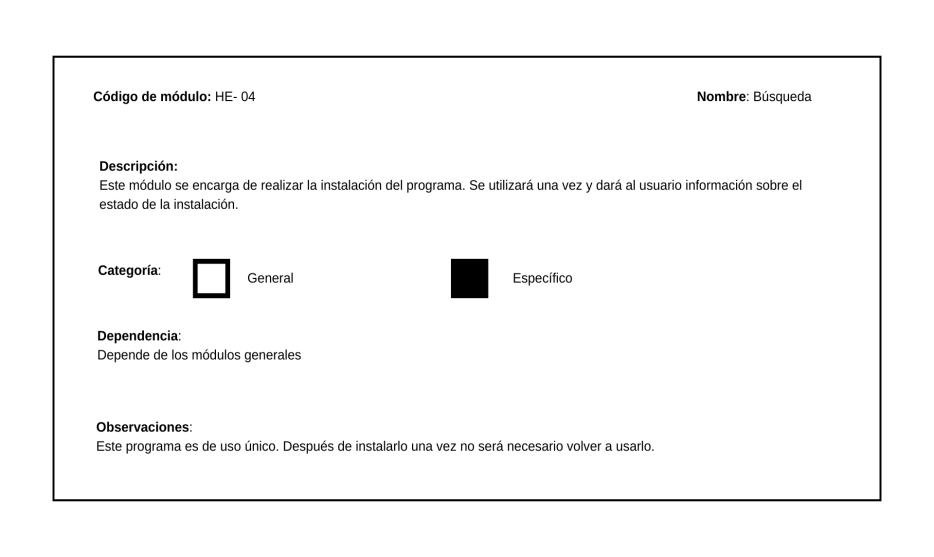
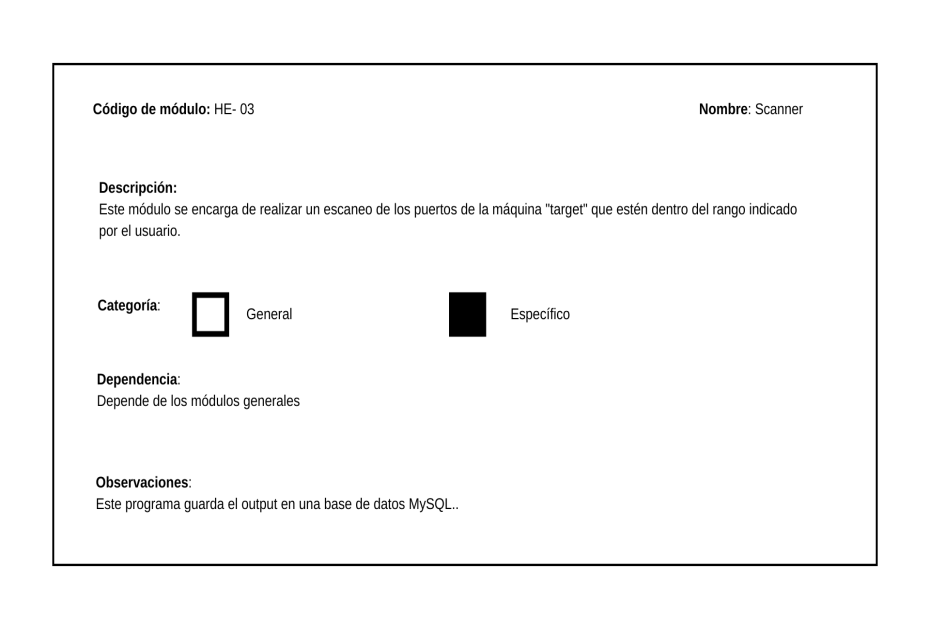
* Diagrama de funcionalidades:

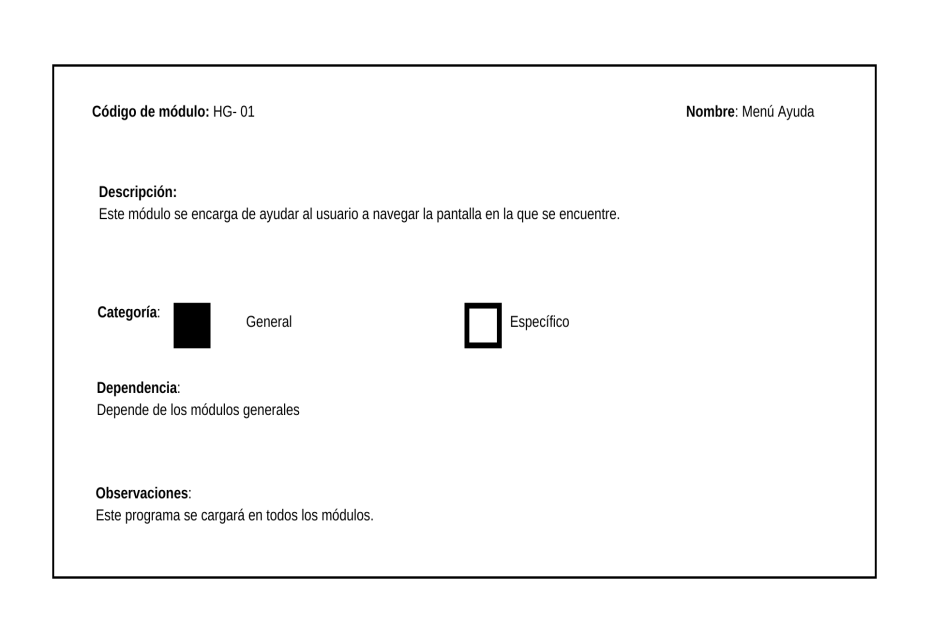


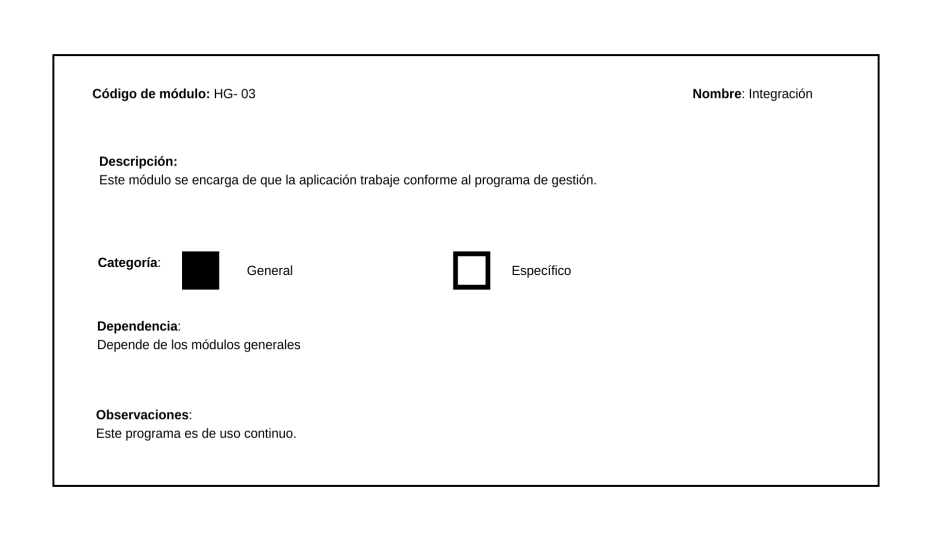
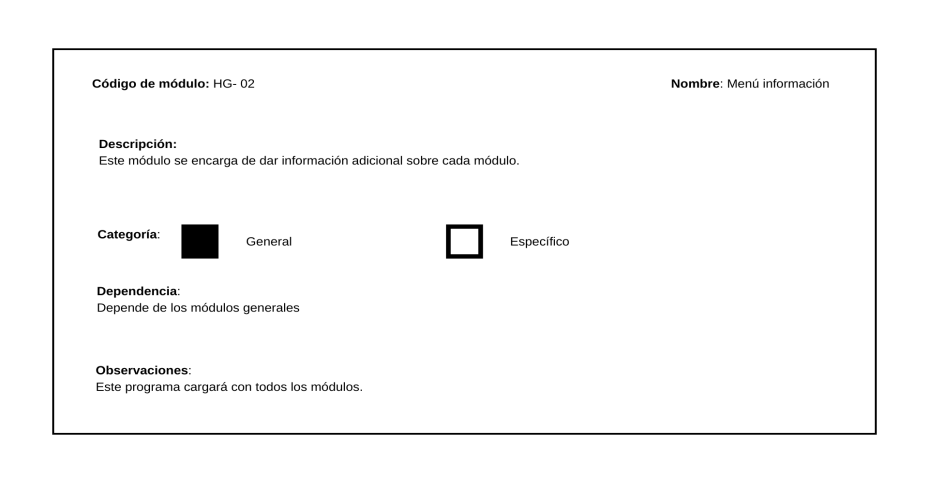
* Módulos de interfaz:

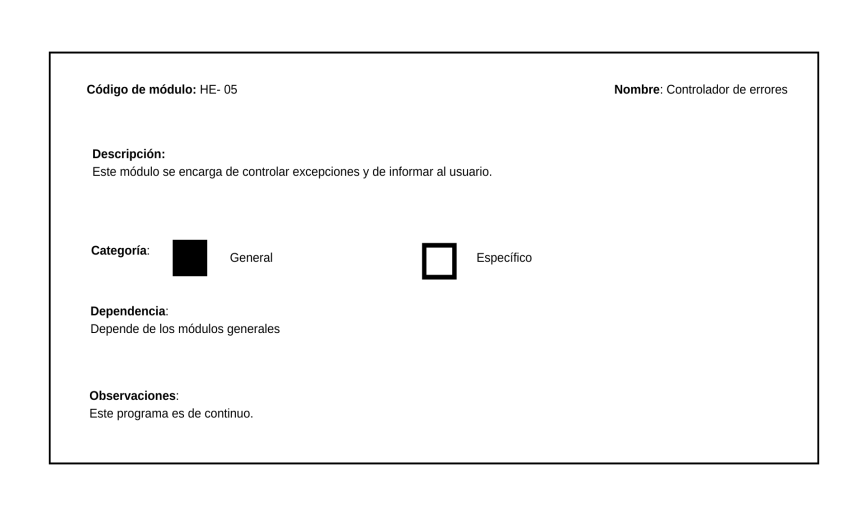
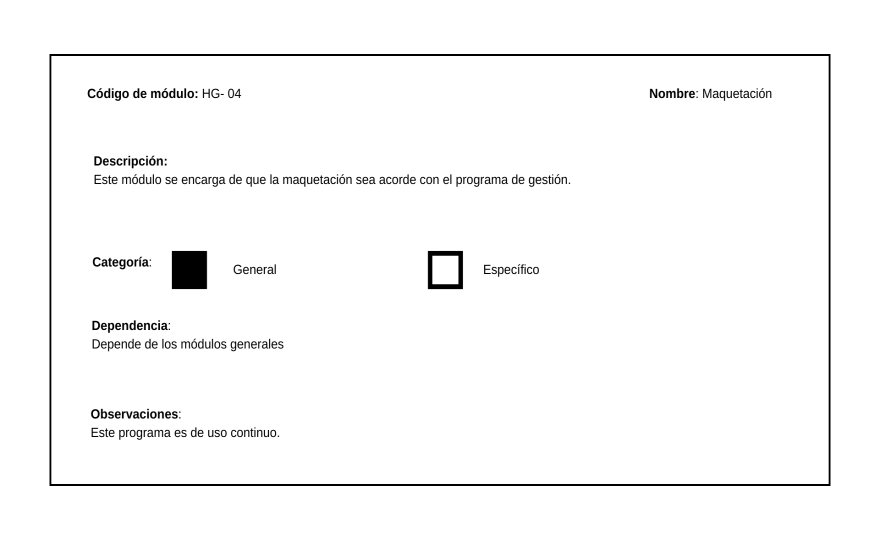




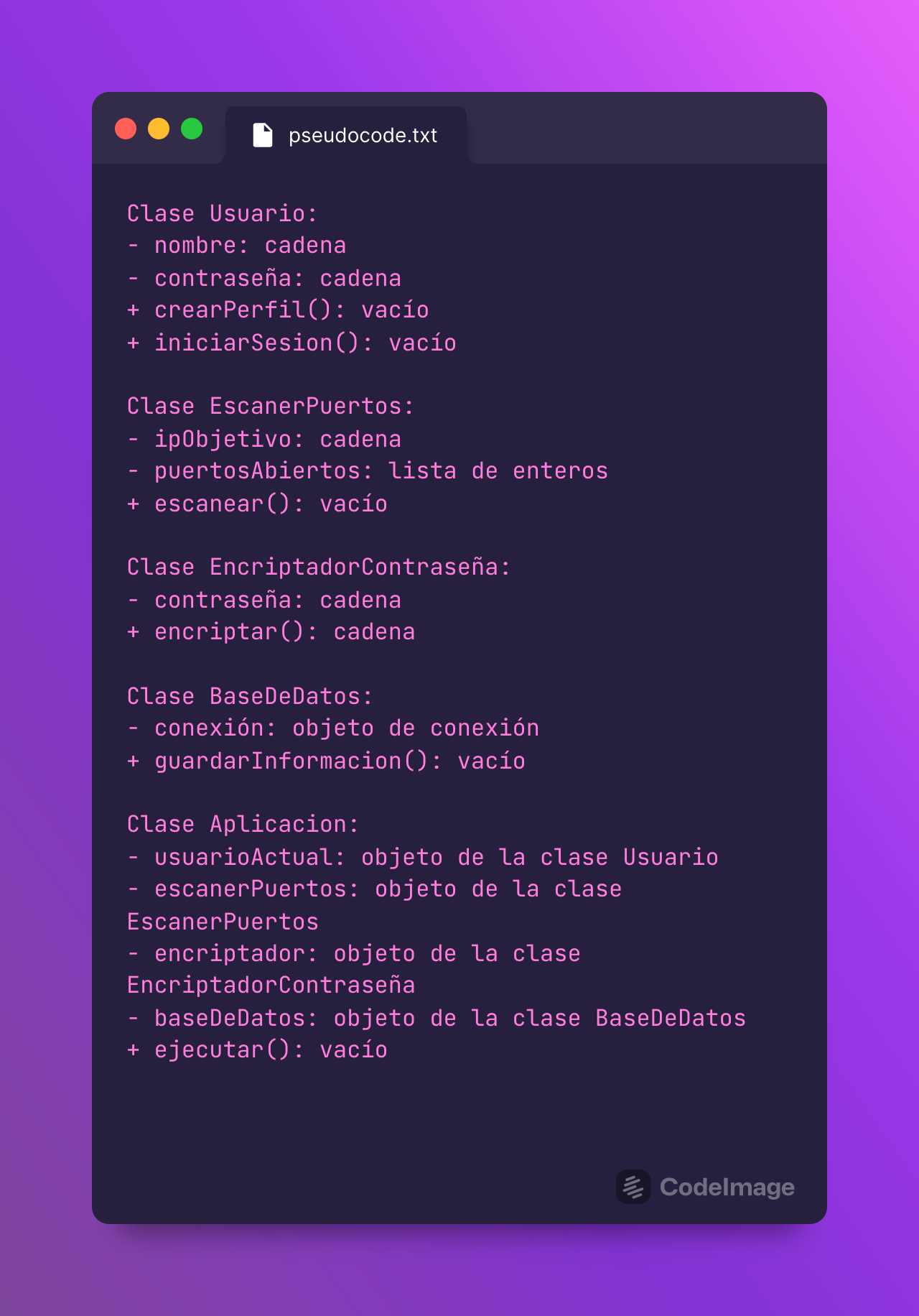


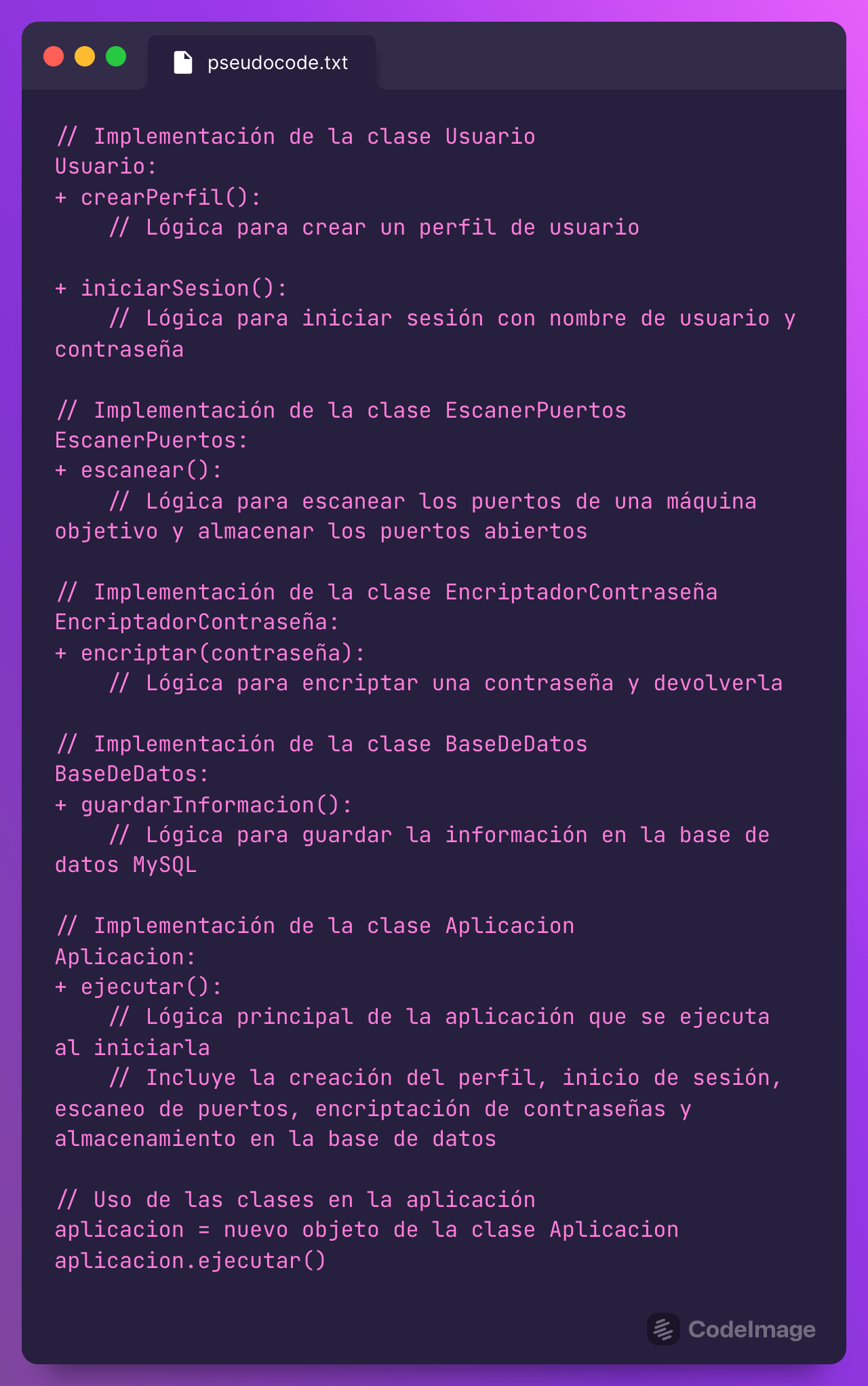


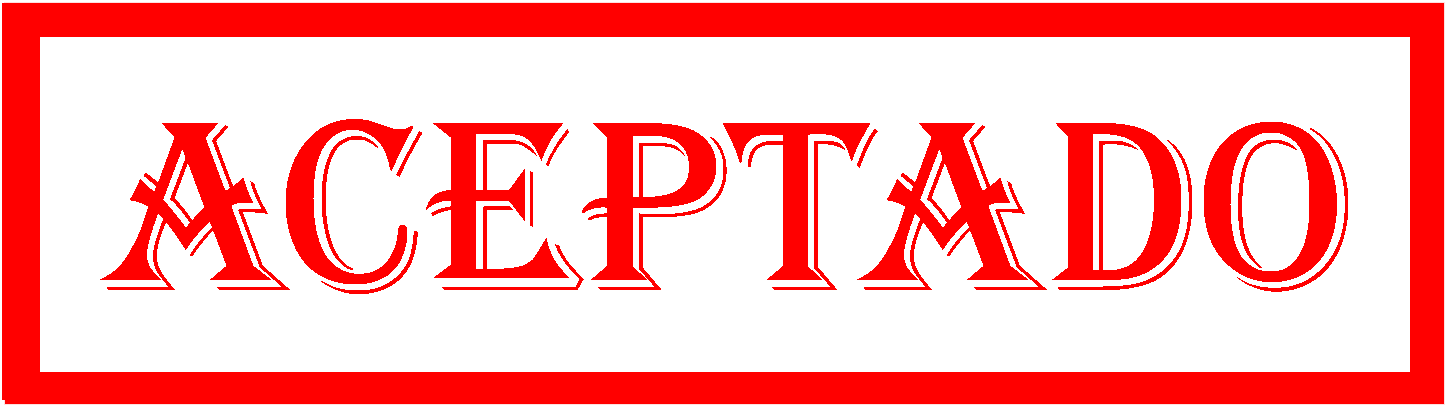




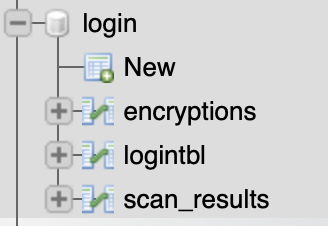
* Pseudocódigo:





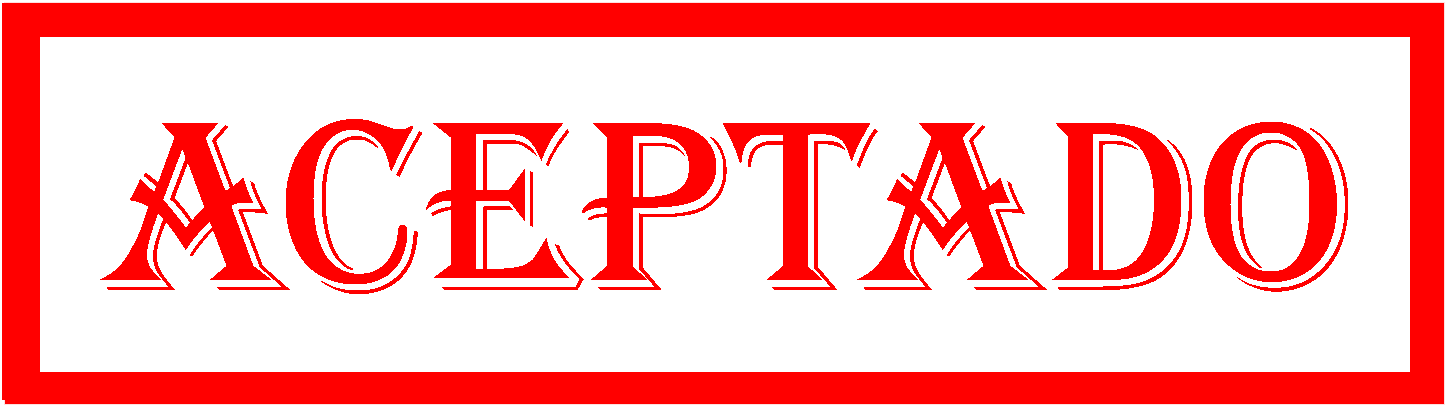


FASE 4: PROGRAMACIÓN

* Diagrama de clases:
  + Estudio de diseño y programación:
    - Fuente: Courier
    - Colores: Cyan (letras) , blanco (letras clickadas), negro (fondo)
    - Imagen corporativa y de la aplicación presente en cada pantalla
    - Base de datos: MySQL
      * 
    - Estudio de la codificación:
      * Login:
        + Importaciones: Se importan los módulos tkinter, PIL y pymysql, para la interfaz gráfica, manipulación de imágenes y acceso a la base de datos MySQL, respectivamente.
        + Clase Login: Define la interfaz de inicio de sesión y registro. Aquí se establecen las configuraciones iniciales de la ventana principal y se crea el formulario de inicio de sesión.
        + Método loginform(): Crea los elementos de la interfaz gráfica para el formulario de inicio de sesión, como etiquetas, campos de entrada y botones.
        + Método login(): Se invoca cuando se hace clic en el botón "Login". Realiza la validación de los campos de entrada y verifica las credenciales ingresadas en la base de datos MySQL.
        + Método Register(): Se invoca cuando se hace clic en el botón "Registro". Crea la interfaz para el formulario de registro y realiza la validación de los campos de entrada. Luego, se inserta la información del nuevo usuario en la base de datos.
        + Método register(): Se ejecuta cuando se hace clic en el botón "Registrar" en el formulario de registro. Realiza la validación de los campos de entrada y verifica si el usuario ya existe en la base de datos antes de insertar los datos del nuevo usuario.
        + Método appscreen(): Cierra la ventana de inicio de sesión y abre una nueva ventana para mostrar la pantalla de la aplicación principal. Aquí se definen botones para las funcionalidades de "Scanner" y "Encrypt".
        + Métodos open\_scanner() y open\_encrypt(): Importan y ejecutan otros módulos (scanner.py y encrypt.py) que parecen ser responsables de otras funcionalidades de la aplicación. Los detalles de estos módulos no están presentes en el código proporcionado.
      * Encrypt:
        + El programa permite al usuario escribir un texto en una textbox, encriptarlo con una contraseña específica (turtle) y guardar tanto el texto original como el resultado encriptado en una base de datos MySQL. También se puede desencriptar el texto encriptado utilizando la misma contraseña.

El código comienza importando las bibliotecas necesarias: pybase64 para la codificación base64, pymysql para interactuar con la base de datos MySQL y tkinter para crear la GUI. A continuación, se establece una conexión con la base de datos y se crea un cursor para ejecutar consultas.

El código incluye varias funciones importantes. La función "clear" se encarga de borrar el contenido de las cajas de texto en la GUI. La función "encrypt" obtiene el texto ingresado por el usuario, lo encripta utilizando una contraseña específica, muestra el resultado en la GUI y guarda tanto el texto original como el resultado encriptado en la base de datos. La función "decrypt" desencripta el texto encriptado utilizando la contraseña proporcionada y muestra el texto desencriptado en la GUI. La función "listEncryptions" recupera los resultados encriptados de la base de datos y los muestra en la GUI. También abre una nueva ventana que muestra una vista tabular de los resultados encriptados y ofrece opciones para eliminar entradas. La función "openUpdateWindow" se utiliza para abrir una nueva ventana que muestra una vista tabular de los resultados encriptados y proporciona opciones para eliminar entradas. La función "deleteEntry" se encarga de eliminar una entrada seleccionada de la base de datos y actualizar la vista tabular.

* + - * Scanner:
        + El programa comienza importando una serie de bibliotecas: socket, sys, threading, time, tkinter, pymysql y PIL. A continuación, se definen algunas variables y listas que se utilizarán en el programa, como ip\_s y ip\_f para especificar el rango de direcciones IP a escanear, log para almacenar los registros de escaneo y ports para almacenar los puertos abiertos encontrados durante el escaneo. También se establece la variable target con el valor "localhost" como dirección IP predeterminada a escanear.
        + El programa incluye varias funciones importantes. La función scanPort se encarga de escanear un puerto específico en la dirección IP objetivo. Si el puerto está abierto, se guarda en la lista ports y se muestra en la GUI. También se guarda el resultado del escaneo en una base de datos MySQL. La función updateResult actualiza la etiqueta en la GUI para mostrar la cantidad de puertos abiertos encontrados hasta el momento. La función startScan se ejecuta cuando se presiona el botón "Scan" en la GUI. Escanea todos los puertos dentro del rango especificado y muestra los resultados en la GUI y en el archivo de registro. La función listScans recupera los resultados almacenados en la base de datos y los muestra en la GUI. La función saveScan guarda el resultado del escaneo en un archivo de texto. La función clearScan borra los resultados mostrados en la GUI.
      * 

FASE 5: CIERRE

* Descripción: En la carpeta adjunto el manual de usuario, el ejecutable de la aplicación junto con un instalador, el diseño del CD de la aplicación, una vídeo guía y la tabla de MySQL para poder importarla a la máquina personal. Solo hay que crear una tabla en phpMyAdmin llamada login e importar mi archivo SQL.