PROYECTO FINAL DE CURSO

Desarrollo de Aplicaciones de Escritorio con Python & Qt

Descripción General

El alumno deberá desarrollar, empaquetar y entregar una aplicación de escritorio completa. El objetivo es demostrar el dominio de la lógica de programación, el diseño de interfaces gráficas y la gestión de datos persistentes.

Requerimientos Técnicos Generales

Para aprobar, cualquier proyecto (Opción A o B) debe cumplir con los siguientes estándares de calidad:

1. **Navegación Fluida:** Uso de `QStackedWidget` para gestionar las vistas en una sola ventana.
2. **Persistencia de Datos:** Uso obligatorio de archivos `.json` (o bases de datos) para guardar la información. La app no debe iniciar vacía cada vez que se abre.
3. **Robustez:** El programa no debe cerrarse inesperadamente ante errores del usuario (uso de validaciones y `try/except`).

Opciones de Proyecto

El alumno puede elegir libremente entre la **Opción A** (Estructurada) o la **Opción B** (Creativa).

1. Opción A: Punto de Venta (POS) - Guiada

Desarrollar un sistema para administrar un negocio ficticio con roles definidos.

Requerimientos Específicos

- **Estructura de Roles:** Login obligatorio con distinción entre **Gerente** y **Cajero**.
- **Funciones Gerente:** ABM (Alta, Baja, Modificación) de Inventario y Usuarios. Visualización de Reportes de ventas.
- **Funciones Cajero:** Interfaz de venta rápida. Selección de productos, cálculo de totales y **descuento automático de stock** al finalizar la venta.

2. Opción B: Proyecto Libre - Creativa

Desarrollar una aplicación de software sobre cualquier tema de interés personal (Videojuegos, Ciencia, Productividad, Hobbies, Herramientas, etc.).

Condición Única: Al tener libertad creativa, se exigirá que la complejidad del código sea equivalente a la Opción A. Tu proyecto no puede ser estático; debe gestionar información.

Requerimientos de Complejidad (Checklist)

Para que tu idea sea aceptada, debe incluir obligatoriamente:

- **Gestión de Datos:** La aplicación debe permitir Crear, Leer, Editar y Borrar información.
    - *Ejemplos válidos:* Un Pokedex (Editar datos de pokemon), Un Gestor de Tareas (Marcar como completadas/borrar), Una Agenda.
- **Lógica de Negocio:** La aplicación debe "hacer algoçon los datos, no solo mostrarlos.
    - *Ejemplos:* Calcular promedios, filtrar listas por categorías, generar una gráfica simple, convertir formatos, algoritmo de búsqueda.
- **Interactividad Avanzada:** Uso de al menos 4 tipos de Widgets diferentes (Tablas, Combos, Calendarios, Checkboxes, etc.) organizados en Layouts.
- **Sistema de Acceso o Perfiles:** Aunque no sea un negocio, debe tener una pantalla de inicio o selección de perfil (ej: "Seleccionar Usuario") que cargue configuraciones o datos guardados en JSON.

Entregables (Para ambas opciones)

Se entregará una carpeta comprimida (.zip) con los siguientes 5 o mas elementos obligatorios:

1. **Código Fuente (`.py`):** El archivo `main.py` con toda la lógica.
2. **Interfaz (`.ui`):** El archivo editable de Qt Designer.
3. **Codigo compilado de la interfaz (`ui.py`):** El archivo compilado de Qt Designer.
4. **Datos Iniciales (`.json`):** Archivos necesarios para que la app funcione.
5. **Ejecutable (`.exe`):** La aplicación compilada con PyInstaller ('–onefile –noconsole').

Rúbrica de Evaluación

Esta tabla aplica tanto para el Punto de Venta como para el Proyecto Libre.

Criterio	Descripción	Valor
Interfaz	Diseño limpio, uso de Layouts y navegación correcta con `QStackedWidget`.	20 %
Funcionalidad	**Opción A:** Roles y Ventas funcionan. **Opción B:** La lógica propuesta resuelve el problema planteado.	25 %
Manejo de Datos	Capacidad de leer y escribir archivos JSON correctamente (Persistencia).	20 %
Gestión de Datos	La app permite agregar, ver y eliminar registros de forma dinámica en tablas.	20 %
Calidad Final	El programa no tiene errores (bugs) y se entrega el ejecutable `.exe` funcional.	15 %

Importante: Si eliges la Opción B, se recomienda consultar brevemente tu idea con el profesor para asegurar que cumple con la complejidad necesaria. Mandar mensaje por classroom para retroalimentacion