

PROYECTO: Planificador de Proyectos Empresariales

Módulo: Iniciación a la Programación y Base de Datos

Índice

Situación Profesional.....	1
Objetivo Principal.....	1
Objetivos específicos.....	1
Actividades Integradoras del Módulo.....	3
Actividad Integradora N° 3.....	3
Programación I.....	3
Base de Datos.....	4
Entregables.....	4
Presentación escrita y defensa oral del ABP.....	5

Situación Profesional

El presente enunciado se acomoda al proyecto con temática a elegir en la Práctica Profesionalizante.

Objetivo Principal

Desarrollar un programa de consola con un menú interactivo que facilite la administración de usuarios según sus roles, diferenciando entre administradores y usuarios estándar. Este sistema permitirá la gestión integral de registro de usuarios e inicio de sesión, asegurando un control de acceso adecuado y una experiencia intuitiva para los usuarios.

Además, se deberá diseñar una base de datos que permita la creación, almacenamiento y manipulación eficiente de los datos de los usuarios, garantizando su correcta gestión.

Objetivos específicos

1. Diseñar la base de datos. Definir la estructura, las relaciones entre tablas y los atributos necesarios para almacenar la información de los usuarios y sus roles.
2. Implementar la gestión de usuarios con SQL. Desarrollar consultas de definición y manipulación de datos.
3. Definir el diagrama de clases que servirá como base para la implementación del programa.
4. Implementar la gestión de usuario. Desarrollar Clases, atributos, métodos y demás funciones para registrar, eliminar y visualizar usuarios según sus roles: administrador y usuario estándar.
5. Integrar el sistema de autenticación. Establecer mecanismos de inicio de sesión y control de acceso para garantizar la seguridad de la información.

6. Implementar un menú interactivo. Diseñar una interfaz basada en texto (consola) que facilite la interacción y gestión de usuarios mediante opciones claras.

Requerimientos funcionales

- El programa de consola debe permitir el registro de usuarios con sus respectivas validaciones. Contraseña. Mínimo 6 caracteres. Debe contener letras y números.
- El programa debe contar con un mecanismo de inicio de sesión que valide las credenciales almacenadas.
- En caso de un inicio de sesión satisfactorio, el programa debe definir claramente los roles de usuario estándar y administrador con permisos diferenciados. Es decir que, el programa de consola, debe permitir tanto a los usuarios estándar como administradores el acceso restringido a ciertas funciones según su rol mediante opciones en un menú de consola. Ej, el usuario estándar no debería modificar su rol o bien eliminar otros usuarios.
- El programa debe manejar correctamente los errores de entrada y proporcionar mensajes informativos.

Requerimientos no funcionales

- **Modularidad.** El código fuente debe estar estructurado en módulos, clases y funciones independientes para facilitar la escalabilidad, mantenimiento y comprensión.
- **Legibilidad.** El código fuente debe ser claro, con variables, funciones, módulos y clases descriptivas.
- **Usabilidad.** La interfaz de consola debe ser intuitiva y permitir al usuario interactuar fácilmente con el programa. (Mensajes claros)

Actividades Integradoras del Módulo

Actividad Integradora N° 3

Programación I

Basándose en los objetivos y requerimientos no funcionales, se solicita:

1. Realizar el **Diagrama de Clases**, para ello:
 - a. Identificar las clases del sistema, atributos y métodos y las relaciones entre ellas.
 - b. Nombrar las clases, atributos y métodos en base a la nomenclatura acordada con el equipo.
 - c. Realizar el diagrama de clases mediante herramientas como draw.io, lucidchart, creately, miro u otro.
2. Realizar un **programa de consola** que cumpla con los requerimientos generales del programa que son:
 - a. Registro de usuarios e inicio de sesión.
 - b. Si el usuario inició sesión satisfactoriamente, el programa debe proporcionar un menú a los usuarios según el rol como sigue:
 - i. **Usuario estándar**. Para acceder sólo a sus datos personales.
 - ii. **Admin**. Para visualizar el listado de usuarios registrados, cambiar el rol de un usuario y eliminar usuarios.



Sugerencias para la escritura del código fuente (convenciones de nomenclatura estándar - comunidad de Python):

- **Archivos y Directorios**

- Archivos Python: Utilizar el formato snake_case en minúsculas. Ejemplo: main.py, module1.py.
- Directorios: Utilizar nombres en minúsculas sin espacios ni caracteres especiales. Ejemplo: docs, src.

- **Funciones.** Utilizar el formato snake_case en minúsculas comenzando la primera palabra con un verbo en infinitivo. Ejemplo: obtener_usuario, actualizar_datos.
- **Variables.** Utilizar el formato snake_case. Ejemplo: contador, nombre_usuario.
- **Constantes.** Utilizar el formato snake_case en mayúsculas. Ejemplo: MAX_INTENTOS.

Base de Datos

1. Identificar las entidades principales del sistema.
2. Definir los atributos para cada entidad, considerando los datos necesarios para cumplir con las funcionalidades especificadas.
3. Establecer las relaciones entre las entidades identificadas.
4. Aplicar el proceso de normalización a las entidades y sus atributos para eliminar redundancias y dependencias funcionales. Se debe alcanzar la Tercera Forma Normal (3FN).
5. Crear el modelo relacional resultante de la normalización, representando las entidades como tablas con sus respectivos atributos y relaciones. Definir las claves primarias y foráneas necesarias para establecer las relaciones entre las tablas.
6. Documentar el diseño de la base de datos, incluyendo una pequeña descripción de cada tabla. Y las aclaraciones que creas necesarias en cada una de ellas. Por ejemplo si se realizó una suposición de algo que no estaba en el enunciado y eso determinó que una relación es uno a muchos, cuando se podría haber interpretado por una ambigüedad que podía ser muchos a muchos, dejar asentada tal suposición.
7. Definir las consultas necesarias en SQL para implementar el CRUD del usuario, y el script ejecutable.

Notas: Se recomienda utilizar una herramienta de modelado de bases de datos, como draw.io, para crear el diagrama del modelo relacional.

Entregables

- **Programación I.** Documento que contenga el diagrama de clases y la documentación del diseño del Diagrama de Clases. Link a repositorio en Github con el código fuente del programa.



Todos los integrantes deben tener confirmaciones (commits) con un claro incremento al producto.

Se **valorará** modularidad, código que respeta las nomenclaturas estándar de la comunidad de Python, evidencia de la implementación de fundamentos de Programación Orientada a Objetos, confirmaciones (commits) con mensajes descriptivos. Cumplimiento de requerimientos no funcionales.

- **Base de Datos.** Documento que contenga el modelo relacional resultante y la documentación del diseño de la Base de Datos (DER y Modelo Relacional).

Script ejecutable de la creación de la Base de Datos.

Script ejecutable con las instrucciones necesarias para la realización del CRUD de usuario.

- **Documento** que exprese las actividades realizadas para la resolución del proyecto por cada estudiante.

Presentación escrita y defensa oral del ABP

- Completar puntos pendientes de evidencias (si los hubiere).
- Escrito. Completar y entregar el documento “Estructura para el Proyecto ABP - 2025”
- Oral. Presentar el [póster](#).

Nota. Independientemente de en qué parte participó cada estudiante, TODOS deben tener claro todo el proyecto. Se sugiere repasar los contenidos teóricos abordados durante el cuatrimestre para la defensa oral dado que, es importante el uso correcto del vocabulario técnico.