*Nombre y apellido:* ***Gastón Di Campli***

*DNI: 31055660*

*Correo electrónico: gaston.dicampli@gmail.com*

*Nombre y apellido:* ***Valentino Lorenzati***

*DNI: 46508945*

*Correo electrónico:* [*valentinolorenzati@gmail.com*](mailto:valentinolorenzati@gmail.com)

*Nombre y apellido:* ***Alejo Nicolás Terreno***

*DNI: 42783052*

*Correo electrónico:* [*anterreno12@gmail.com*](mailto:anterreno12@gmail.com)

Fecha de entrega: 8 de septiembre de 2024

Aplicación de Gestión Hospitalaria

Innovación en Gestión de Datos (Base de Datos II y Programación I)

**Resumen del Proyecto**

El presente informe describe el desarrollo de una aplicación de escritorio para la gestión y administración de un hospital. Este proyecto intenta dar solución a la necesidad de centralizar y simplificar la gestión de información en instituciones de salud, que manejan grandes volúmenes de datos relacionados con pacientes, profesionales médicos, servicios, obras sociales y turnos. La aplicación busca proporcionar a los administrativos del hospital una herramienta eficiente, intuitiva y segura para realizar sus tareas diarias.

La solución propuesta será desarrollada en el lenguaje de programación Python, utilizando una base de datos relacional gestionada por MySQL. Este proyecto pretende dar continuidad al desarrollo propuesto para el trabajo final del módulo programador del primer semestre de la carrera.

En línea con los objetivos de la cátedra, con el avance del proyecto se implementará el uso de librerías específicas de Python tales como Pandas, Numpy y Matplotlib. Asimismo, se implementará el paradigma de Programación Orientada a Objetos (POO), lo cual permitirá una estructuración más clara y mantenible del código, facilitando su escalabilidad en el futuro.

**Objetivos principales**

Los objetivos de este proyecto son varios, pero se pueden destacar los siguientes:

1. **Crear una interfaz gráfica amigable**: Diseñar una interfaz que facilite la navegación entre las distintas funcionalidades del programa, permitiendo a los administrativos del hospital gestionar de manera efectiva los datos de los pacientes, profesionales médicos, turnos y servicios ofrecidos.
2. **Centralizar la información hospitalaria**: Permitir que toda la información relevante sobre los pacientes, los médicos y los servicios del hospital se gestione desde un único lugar, reduciendo errores humanos y mejorando la eficiencia en el proceso de atención médica.
3. **Asegurar la integridad y la consistencia de los datos**: Implementar un sistema de base de datos relacional que mantenga la integridad de los datos y permita su correcto manejo a través de operaciones CRUD (crear, leer, actualizar y eliminar).
4. **Optimizar la gestión de turnos**: Incorporar un módulo de turnero que permita agendar, consultar y cancelar turnos de manera ágil, evitando solapamientos o errores de programación, mejorando así la atención a los pacientes y la organización de las consultas.

**Metodología a utilizar**

El desarrollo de esta aplicación seguirá una metodología **iterativa e incremental**, en la que se implementarán y probarán módulos de forma secuencial y en etapas. El uso de **Programación Orientada a Objetos (POO)** será clave para la modularización del código, dividiendo las funcionalidades en clases bien definidas, lo que permitirá una fácil extensión de la aplicación en el futuro. Cada clase representará una entidad clave del sistema (por ejemplo, Paciente, Profesional Médico, Turno, etc.) y contendrá métodos que permitirán interactuar con la base de datos.

En cuanto al manejo de la base de datos, se emplearán **consultas SQL** para la manipulación de los datos, y el acceso a la base de datos se realizará a través de la librería mysql-connector-python, que permite una integración eficiente entre Python y MySQL.

Para el control de versiones, se utilizará **Git** junto con **GitHub**, permitiendo a los miembros del equipo colaborar de manera eficiente y mantener un historial de cambios documentado. Además, se implementará el uso de **GitHub Projects** para gestionar issues y milestones, lo que facilitará el seguimiento de las tareas del equipo y asegurará que el proyecto se mantenga dentro de los plazos establecidos.

**Relevancia del proyecto en el contexto académico y profesional**

El desarrollo de esta aplicación no solo representa un desafío académico, sino que también tiene una relevancia significativa en el ámbito profesional. Los sistemas de gestión hospitalaria son esenciales para el correcto funcionamiento de los centros de salud, y una solución como la que aquí se propone podría ser utilizada en la práctica para mejorar la eficiencia y calidad de los servicios hospitalarios.

Desde el punto de vista académico, este proyecto ofrece la oportunidad de aplicar y consolidar conocimientos adquiridos en las materias de **Base de Datos II** y **Programación I**, además de otras habilidades complementarias como el manejo de **librerías para el análisis de datos (Pandas y Numpy)**, el uso de **algoritmos de búsqueda y ordenamiento**, y la implementación de **mecanismos de control de excepciones** para garantizar la robustez del sistema. Adicionalmente, la integración de sistemas de control de versiones como Git y GitHub es una práctica estándar en la industria de software, lo que permitirá a los estudiantes familiarizarse con herramientas y metodologías empleadas en proyectos profesionales.

**Descripción de la Base de Datos**

La base de datos propuesta estará compuesta por varias tablas que almacenarán los datos clave del sistema hospitalario. Las principales tablas serán:

1. **Pacientes**:
   * **ID\_Paciente** (PK): Identificador único del paciente.
   * **Nombre**: Nombre del paciente.
   * **Apellido**: Apellido del paciente.
   * **DNI**: Número de identificación del paciente.
   * **Dirección**: Dirección de residencia del paciente.
   * **Teléfono**: Número de contacto.
   * **Fecha\_Nacimiento**: Fecha de nacimiento del paciente.
   * **Obra\_Social** (FK): Identificador de la obra social asociada al paciente.
2. **Profesionales Médicos**:
   * **ID\_Profesional** (PK): Identificador único del profesional.
   * **Nombre**: Nombre del profesional.
   * **Apellido**: Apellido del profesional.
   * **Matrícula**: Número de matrícula profesional.
   * **Especialidad**: Especialidad médica.
   * **Teléfono**: Número de contacto.
3. **Servicios Médicos**:
   * **ID\_Servicio** (PK): Identificador único del servicio médico.
   * **Nombre\_Servicio**: Nombre del servicio médico.
   * **Descripción**: Descripción del servicio médico.
4. **Obras Sociales**:
   * **ID\_Obra\_Social** (PK): Identificador único de la obra social.
   * **Nombre\_Obra\_Social**: Nombre de la obra social.
   * **Descripción**: Descripción de la obra social.
5. **Turnos**:
   * **ID\_Turno** (PK): Identificador único del turno.
   * **Fecha**: Fecha del turno.
   * **Hora**: Hora del turno.
   * **ID\_Paciente** (FK): Referencia al paciente que tomó el turno.
   * **ID\_Profesional** (FK): Referencia al profesional que atenderá el turno.

**Relaciones entre las tablas**

Las relaciones entre las tablas estarán claramente definidas mediante **claves foráneas (FK)** que garantizarán la integridad referencial. Por ejemplo, la tabla de "Turnos" tendrá referencias a las tablas de "Pacientes" y "Profesionales Médicos", ya que un turno será asignado a un paciente y un profesional específico. De igual manera, la tabla de "Pacientes" incluirá una referencia a la tabla de "Obras Sociales", de modo que se pueda consultar qué obra social cubre al paciente.

Estas relaciones permiten que el sistema mantenga datos consistentes y optimicen las consultas SQL que se realicen dentro de la aplicación. Además, facilitarán la implementación de las diferentes funcionalidades del sistema, como la gestión de turnos y la visualización de servicios médicos ofrecidos.