

**CODIFICACION DE MODULOS DE SOFTWARE SEGÚN REQUERIMIENTOS
DEL PROYECTO**

GA7-220501096-AA2-EV01

PRESENTADO POR:

Sergio Andrés Palomino Silva.

PRESENTADO A:

Milton Iván Barbosa Gaona.

Tecnólogo en Análisis y Desarrollo de Software.

No. De ficha: 2977481

Centro de la Tecnología del Diseño y la Productividad Empresarial.

16 de septiembre de 2025

TABLA DE CONTENIDO

Introducción.....	3
Objetivos.....	4
Informe técnico del plan de trabajo para construcción del software.....	6

TABLA DE IMÁGENES

Diagrama de casos de uso del proyecto de software.....	5
Diagrama de clases del proyecto de software.....	5

INTRODUCCION

Realizaremos la codificación a partir del programa o lenguaje seleccionado que incluya nombramiento de variable, clases, método y funciones... además, la instalación de paquetes (pip), conexión a la base de datos, realización CRUD (INSERCIÓN, ACTUALIZAR, ELIMINAR Y CONSULTAR) la base de datos.

Herramientas para utilizar:

- MySQL Workbench (DB)
- Visual Studio Code
- Extensión Python

➤ Variables

Las variables son datos con nombre cuyos valores cambian durante la ejecución del programa, los nombres de las variables deben seguir las convenciones de PEP8.

➤ Clases

Las clases proporcionan una manera de agrupar datos y funcionalidades. Al crear una nueva clase crea un objeto de un nuevo tipo y le permite crear nuevas instancias de ese tipo también se puede adjuntar atributos a cada instancia de clase para mantener su estado.

➤ Métodos

Los métodos pueden hacer referencia a nombres globales de la misma manera que las funciones generales. El alcance global asociado con un método es el módulo que contiene su definición.

➤ Funciones

Las funciones te permiten definir bloques de código reutilizables y estas pueden ejecutarse varias veces dentro de un programa.

OBJETIVOS

Objetivo general:

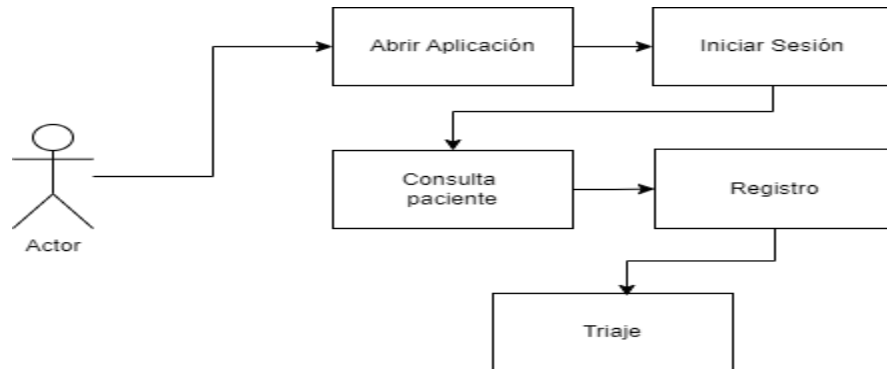
Elaborar por medio de diversos aplicativos la codificación de los módulos del proyecto de software.

Objetivos específicos:

- Instalar y configurar el paquete de mysql installer 8.0.
- Elaborar las tablas (módulos) en la BD de Workbench.
- Codificar los módulos en netbeans.
- Realizar el código que permita la conexión entre Workbench y netbeans.

1. Diagrama de casos de uso del proyecto de software:

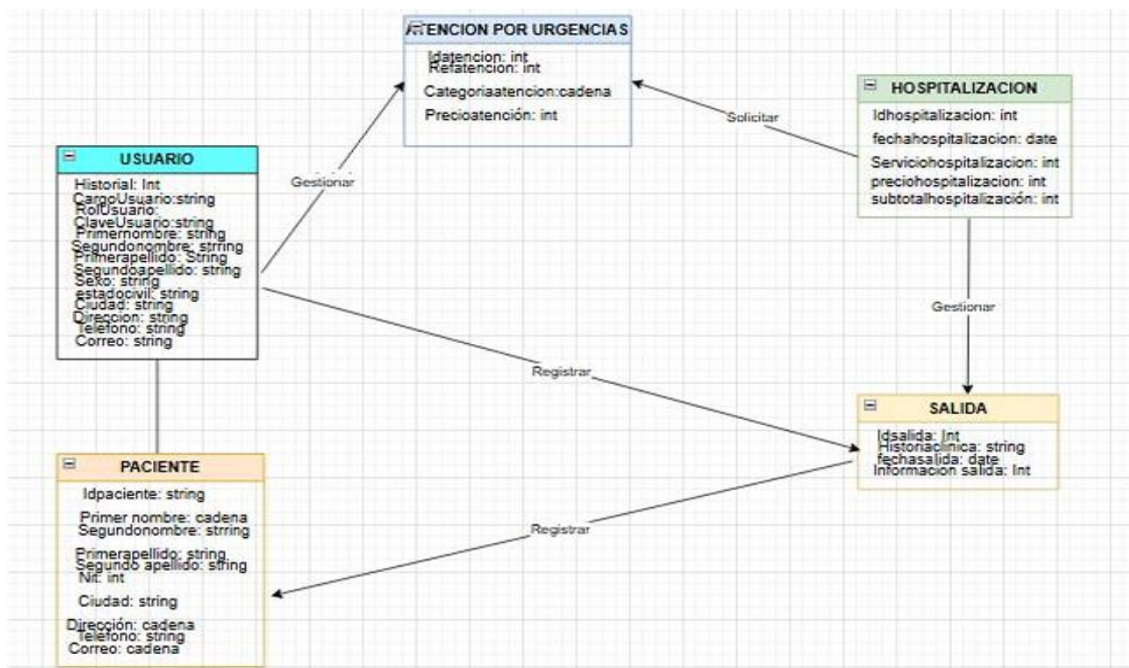
Diagrama 1 Casos de uso Triage



Nota: Diagrama creado por Sergio Andrés Palomino Silva.

2. Diagrama de clases del proyecto de software:

Diagrama 2 Diagrama de clases sistema Triage



Nota: Diagrama creado por Sergio Andrés Palomino Silva.

3. INFORME TECNICO DE PLAN DE TRABAJO PARA CONSTRUCCION DE SOFTWARE

3.1 Control de Versiones Centralizado

3.1.1 Subversión (SVN)

Descripción: Subversión es un sistema de control de versiones centralizado

ampliamente utilizado. Permite rastrear cambios en archivos y directorios a través de un servidor central, lo que facilita la colaboración.

Ventajas:

- Soporte para versionamiento de directorios completos.
- Seguimiento de cambios en binarios.
- Acceso a revisiones anteriores.

Desventajas:

- Requiere conexión constante al servidor central.
- Menos eficiente en la gestión de ramas.

3.2 Control de Versiones Distribuido

3.2.1 Git

Descripción: Git es un sistema de control de versiones distribuido ampliamente utilizado.

Cada desarrollador tiene una copia completa del repositorio, lo que

permite un trabajo sin conexión y una gestión eficiente de ramas.

Ventajas:

- Alta velocidad y eficiencia.
- Facilita la colaboración y la gestión de ramas.
- Ampliamente adoptado en la industria.

Desventajas:

- Curva de aprendizaje pronunciada para principiantes.
- No es ideal para gestionar grandes archivos binarios.

4. Plataformas de Hospedaje de Repositorios**4.1 GitHub**

Descripción: GitHub es una plataforma de desarrollo colaborativo que utiliza Git como sistema de control de versiones. Ofrece características como seguimiento de problemas, solicitudes de extracción y una amplia comunidad de desarrolladores.

Ventajas:

- Facilita la colaboración entre equipos.
- Integración con numerosas herramientas y servicios.
- Amplia comunidad y recursos de aprendizaje.

Desventajas:

- Algunas funciones avanzadas requieren una suscripción de pago.

4.2 GitLab

Descripción: GitLab es una plataforma similar a GitHub que proporciona una gestión completa de ciclo de vida del desarrollo de software. Puede ejecutarse en servidores propios o en la nube.

Ventajas:

- Ofrece opciones tanto en la nube como en instalaciones locales.
- Gestión de CI/CD integrada.
- Herramientas de gestión de proyectos y seguridad.

Desventajas:

- Menos usuarios y proyectos en comparación con GitHub.

5. Control de Versiones para Datos**5.1 DVC (Data Version Control)**

Descripción: DVC es una herramienta de control de versiones diseñada

específicamente para datos y modelos de aprendizaje automático. Permite rastrear cambios en archivos de datos y modelos de manera eficiente.

Ventajas:

- Gestión de datos y modelos de manera eficiente.
- Integración con sistemas de almacenamiento en la nube.
- Control de versiones sin duplicidad de datos.

Desventajas:

- Enfoque específico para proyectos de datos y ML.

CONCLUSIONES

SVN y Git tienen sus propias ventajas y desventajas en función de las necesidades del proyecto. GitHub y GitLab facilitan la colaboración y la gestión de desarrollo de software, pero la elección depende de las necesidades específicas del equipo.

La codificación de los módulos es fundamental para el desarrollo del proyecto de software y poder culminar el proyecto de la mejor manera.