

**Codificación de módulos del software Stand alone, web y móvil de acuerdo al proyecto a desarrollar**

Presentado por Karen Julieth Sosa Espitia

Análisis y desarrollo de software

2977481

Presentado a Instructor Milton Ivan Barbosa Gaona

Centro de la Tecnología del Diseño y la Productividad Empresarial Girardot –

Cundinamarca – SENA (Servicio Nacional de Aprendizaje)

22 de septiembre 2025

## Tabla de contenido

Introducción .....	4
Justificación .....	5
Objetivos.....	6
Objetivo General .....	6
Desarrollar e implementar los módulos de software basados en los artefactos previos.....	6
Objetivos Específicos .....	6
1.    Artefactos del Ciclo de Software.....	7
1.1    Diagrama de Clases .....	7
1.2 Diagramas de Casos de uso .....	8
1.3 Historias de Usuario .....	10
1.4 Prototipos .....	11
2 Informe Técnico de Plan de Trabajo: .....	13
2.1 Herramientas de versionamiento .....	13
Tipos .....	13
Ventajas de las herramientas de Versionamiento .....	13
2.2 GIT .....	13
Características de GIT .....	13
Comandos de Git .....	14
Ventajas de GIT .....	14
Productos.....	15
Conclusiones .....	16
Bibliografía.....	17

### **Contenido de Imágenes**

Imagen 1 Prototipo Modulo Autenticación .....	11
Imagen 2 Prototipo Modulo Gestión de citas 1 .....	11
Imagen 3 Prototipo Modulo Gestión de citas 2 .....	12
Imagen 4 Prototipo Modulo Gestión de usuarios.....	12

## **Introducción**

El presente trabajo tiene como finalidad la codificación del módulo de software considerando los artefactos desarrollados en las fases previas del ciclo de vida del software. Para ello, se han tenido en cuenta elementos fundamentales como el diagrama de clases, los diagramas de casos de uso, las historias de usuario, los prototipos, así como el informe técnico del plan de trabajo.

### **Justificación**

La codificación del módulo se fundamenta en la necesidad de transformar los diseños y planificaciones previas en un sistema funcional y operativo. Al considerar los artefactos generados en etapas anteriores, se reduce la probabilidad de errores, se optimiza el tiempo de desarrollo y se asegura la alineación del producto final con los requerimientos del cliente.

## **Objetivos**

### **Objetivo General**

Desarrollar e implementar los módulos de software basados en los artefactos previos

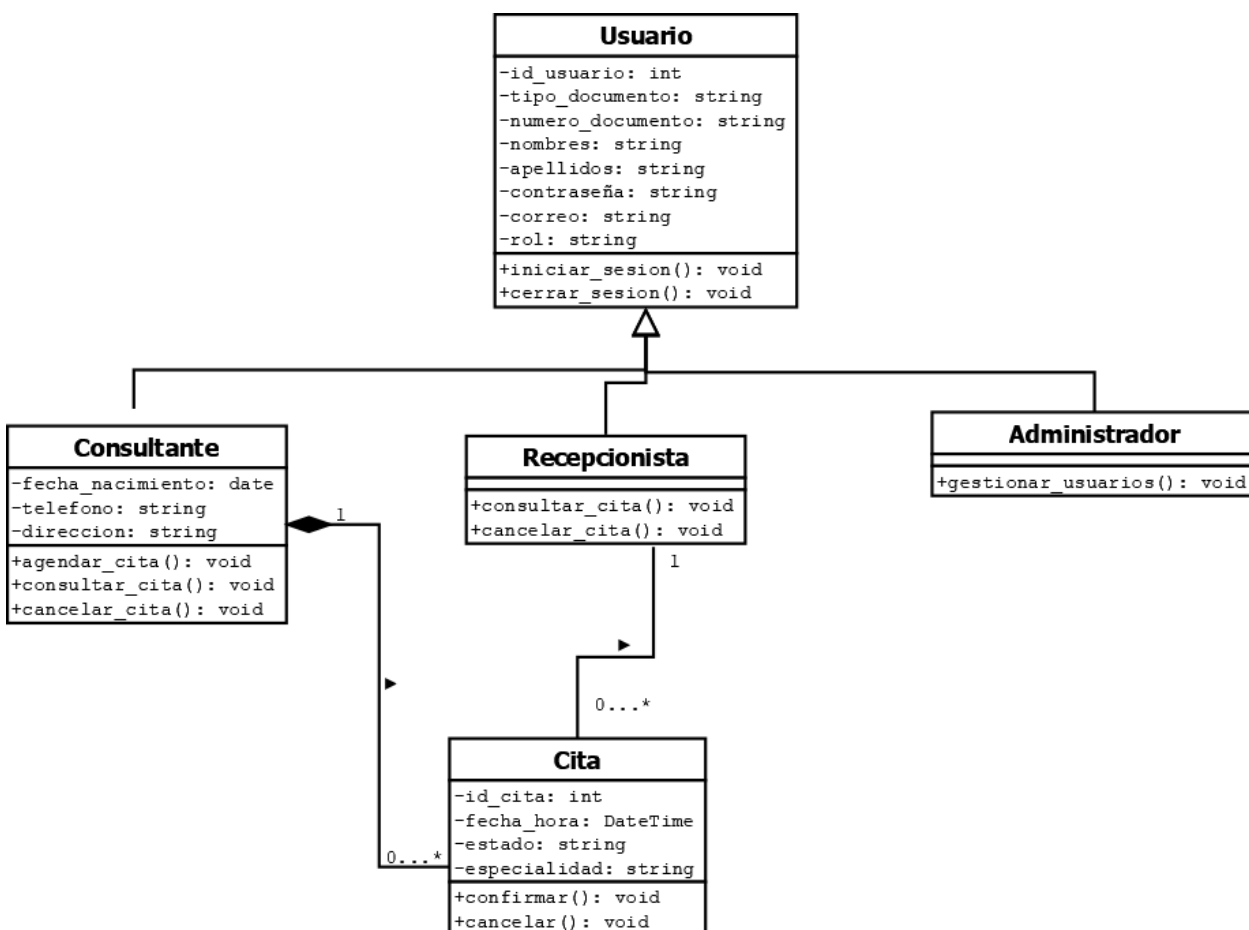
### **Objetivos Específicos**

- Transformar los artefactos en código funcional
- Garantizar que exista coherencia
- Verificar que se cumplan los requerimientos del proyecto.

## 1. Artefactos del Ciclo de Software

### 1.1 Diagrama de Clases

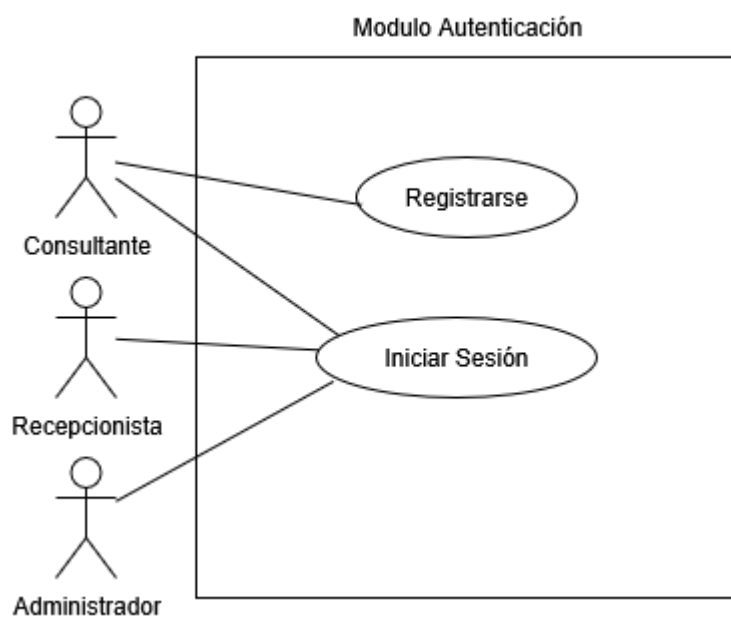
Diagrama 1 Diagrama de clases



Nota: Diagrama elaborado por Julieth Sosa

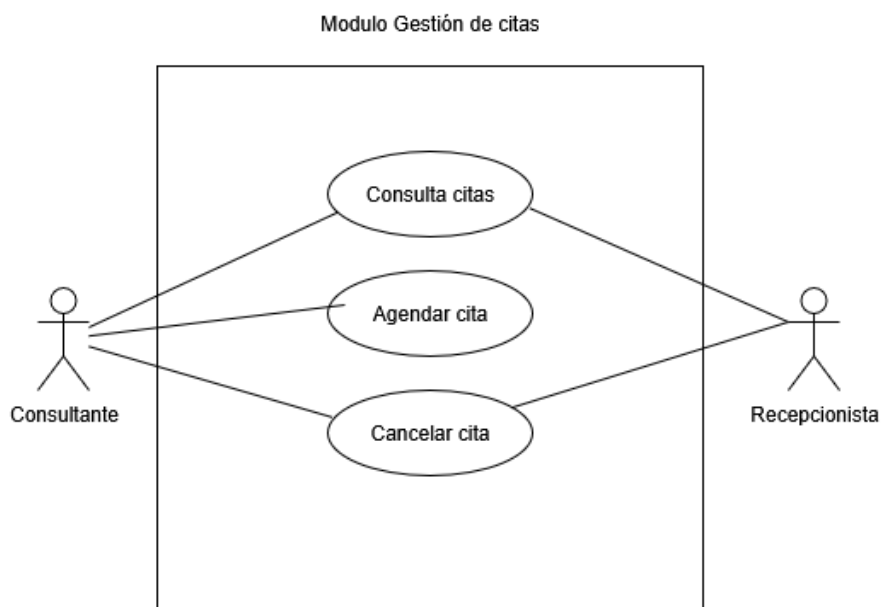
## 1.2 Diagramas de Casos de uso

Diagrama 2 Modulo de autenticación



*Nota: Diagrama elaborado por Julieth Sosa*

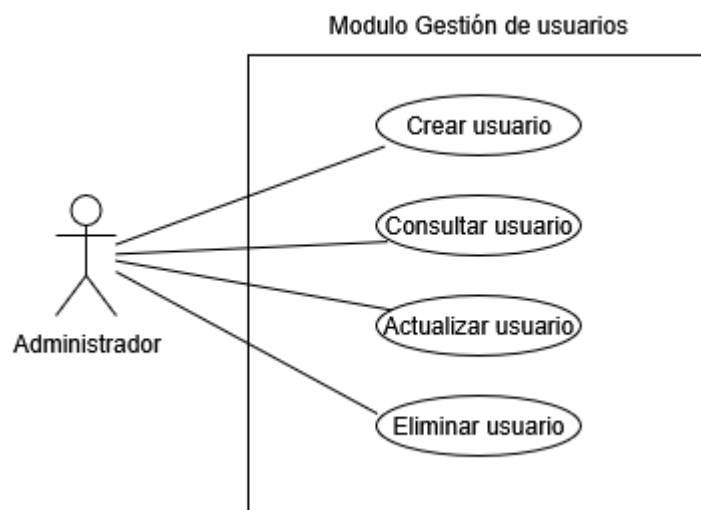
Diagrama 3 Modulo Gestión de citas



*Nota: Diagrama elaborado por Julieth Sosa*



Diagrama 4 Modulo Gestión de usuarios



Nota: Diagrama elaborado por Julieth Sosa

### 1.3 Historias de Usuario

HISTORIAS DE USUARIO				
ID	Rol	Funcionalidad	Razón/ Resultado	Criterio de aceptación
<b>Módulo autenticación</b>				
HU01	Como consultante	Quiero registrarme en el sistema	Para poder gestionar mis citas	-El sistema solicita los datos y el registro se completa si se diligencia todos los campos obligatorios -El sistema confirma que fue exitoso el registro
HU02	Como usuario	Quiero iniciar sesión	Para acceder al sistema según mi rol	-El sistema valida usuario y contraseña, si los datos son correctos otorga el acceso, de lo contrario muestra un mensaje de error
<b>Módulo de Gestión de Citas</b>				
HU03	Como consultante	Quiero agendar citas	Para recibir atención de un profesional	-El sistema muestra fechas, horas, y profesionales disponibles - El consultante hace la selección y el sistema confirma la cita
HU04	Como consultante	Quiero consultar mis citas	Para recordar los datos	-El sistema muestra un listado de las citas con su detalle
HU05	Como consultante	Quiero cancelar mis citas	Para cuando no pueda asistir	-El consultante puede seleccionar una cita y el sistema le da la opción de cancelarla, y le confirma.
HU06	Como recepcionista	Quiero consultar las citas	Para supervisar la agenda.	- El sistema muestra las citas con los detalles.
HU07	Como recepcionista	Quiero cancelar citas	Para cuando el consultante lo requiera o cuando sea necesario	-El recepcionista puede seleccionar una cita y el sistema le da la opción de cancelarla, y le confirma.
<b>Módulo de Gestión de usuarios</b>				
HU08	Como administrador del sistema	Quiero gestionar los usuarios	Para mantener el acceso al sistema actualizado y controlado	-El sistema le permitirá gestionar los usuarios del sistema (crear, actualizar, eliminar o consultar), el sistema confirma cada acción realizada

## 1.4 Prototipos

Imagen 1 Prototipo Modulo Autenticación



*Nota: Imagen elaborada por Julieth Sosa*

Imagen 2 Prototipo Modulo Gestión de citas 1

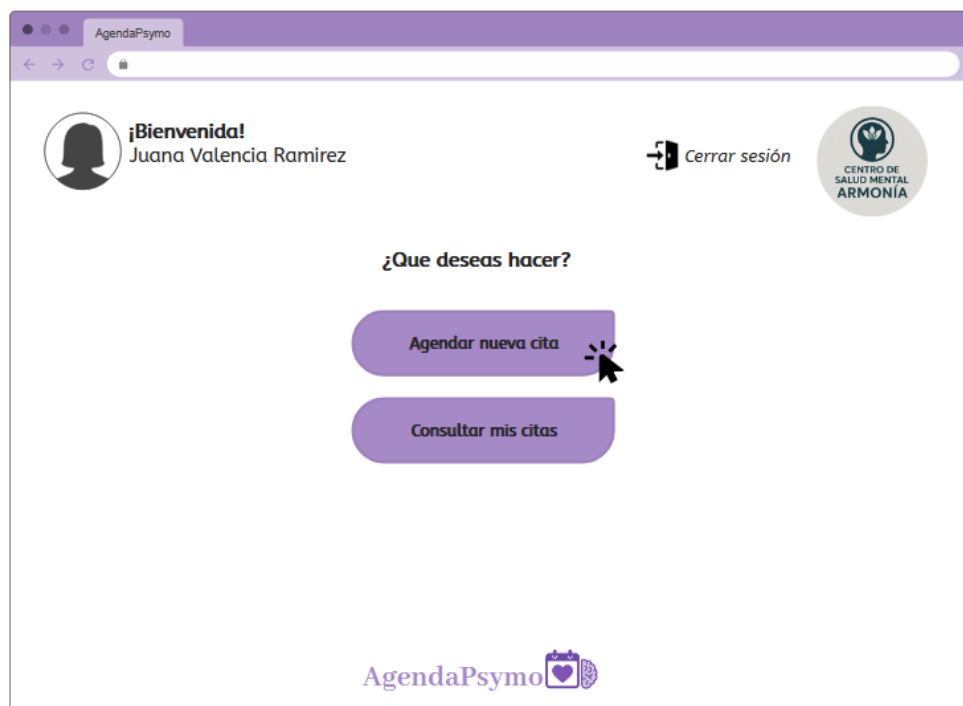


Imagen 3 Prototipo Modulo Gestión de citas 2

AgendaPsymo

¡Bienvenida!  
Juana Valencia Ramirez

Regresar

Selecciona la especialidad  
Psicología

Selecciona la fecha  
May 24, 2025

Fecha mas cercana ☒

Consultar

Citas disponibles

Nombre profesional	Fecha	Hora
Ana Hernandez Jimenez	24- Mayo-2025	7:00am
Juan Carlos Figueroa Lopez	24 -Mayo -2025	8:00am
Luz Marina Puentes Gil	24 -Mayo 2025	8:00am
Cristian David Suarez Ponte	24-Mayo-2025	10:00am

AgendaPsymo

Nota: Imagen elaborada por Julieth Sosa

Imagen 4 Prototipo Modulo Gestión de usuarios

AgendaPsymo

¡Bienvenido!  
Administrador

Cerrar sesión

Gestión de cuentas de Profesionales

Agregar profesional

Modificar/Eliminar profesional

Nombre completo

Especialidad

Correo Electrónico

Tipo de documento

Número

Contraseña

Guardar

AgendaPsymo

Nota: Imagen elaborada por Julieth Sosa

## **2 Informe Técnico de Plan de Trabajo:**

### **2.1 Herramientas de versionamiento**

Son herramientas (software) que pueden gestionar cambios dentro de archivos, de forma tal que cada versión sea una fotografía del momento, pero a su vez puedas tener múltiples fotos y estas fotos puedan ser comparadas, unificadas y consolidadas, siempre sabiendo que y quien realizó cada cambio. (Lacruz, 2022)

#### **Tipos**

Existen 3 tipos de sistemas de control de versiones. Estos son:

- Locales: Que se caracterizan por la sincronización manual.
- Centralizados: Que permite el trabajo en grupo desde diferentes equipos, pero dependiente de uno central.
- Distribuidas: En donde cada equipo tiene una réplica completa y permite varios flujos de trabajo.

#### **Ventajas de las herramientas de Versionamiento**

- Historial completo de cambios
- Creación de ramas y fusiones
- Trazabilidad

### **2.2 GIT**

Es la herramienta de versionamiento más popular y utilizada actualmente, diseñado para gestionar desde pequeños hasta grandes proyectos. El uso de git principalmente es a través de comandos. Usando sólo git tenemos ya disponible el sistema de versionamiento de forma local. (Lacruz, 2022).

#### **Características de GIT**

- Es multiplataforma, por lo que puede ser utilizado en Windows, Linux y Mac
- No se necesita pago, es de código abierto

- Tiene 3 estados principales: Confirmado, modificado y preparado.
- Tiene 3 secciones principales: Git directory, working directory y staging area.

### **Comandos de Git**

A continuación, se mencionan los principales comandos en Git.

- Para inicializar el repositorio desde un directorio existente: `$ git init`
- Para obtener copia desde un repositorio: `$ git clone https://url_del_repositorio`
- Para visualizar estado actual de los archivos: `$ git status`
- Para rastrear archivos nuevos: `$ git add Nombre_archivo`
- Para rastrear cada archivo de un directorio: `$ git add Directorio`
- Para confirmar cambios: `$ git commit`
- Para ver el histórico de confirmaciones: `$ git log`

### **Ventajas de GIT**

Git es una herramienta que permite gestionar los cambios en el código de manera eficiente y segura, especialmente cuando se trabaja en equipo. A continuación, se presentan algunas de sus ventajas principales:

- Control de versiones distribuido: Cada desarrollador tiene una copia completa del proyecto, lo que permite trabajar sin conexión.
- Tiene un historial completo de cambios, permitiendo ver quién hizo cambios, en dónde y cómo.
- Se puede trabajar en diferentes funciones o pruebas sin afectar el proyecto principal.
- Git integra bien los cambios de varias personas, facilitando el trabajo en equipo, especialmente cuando es remoto.
- Seguridad: Usa hashes (SHA-1) para verificar la integridad del contenido y evitar manipulaciones.
- Permite la colaboración con otras herramientas como GitHub o GitLab

- Permite volver a versiones anteriores fácilmente si algo falla, lo que lo vuelve excelente para la recuperación ante errores.

### **Productos**

Algunos de los productos que ofrecen alojamiento Git son:

- GitHub
- GitLab
- BitBucket
- Gitea
- Radicle
- Codeberg

### **Conclusiones**

- Considero que la codificación de los módulos se ve fortalecida al utilizar los artefactos previos
- La integración de historias de usuario, diagramas y prototipos en el proceso de codificación permite que el software final cumpla con las expectativas y necesidades del usuario.
- La documentación y planificación previas son herramientas esenciales para asegurar la calidad del software desarrollado.



## **Bibliografía**

Kinsta. (21 de Agosto de 2023). *Git vs GitHub: ¿Cuál es la Diferencia y cómo Empezar?*

Obtenido de <https://kinsta.com/es/base-de-conocimiento/git-vs-github/>