**Informe Proyecto APT**

Sistema de gestión CAPSI

Asignatura: Capstone

Sigla: PTY4614-005V

Integrantes:

Nicolas Ávalos

José Morales

Maximiliano Tapia

Escuela de Informática y Telecomunicaciones

Mes 2025

Integrantes del Equipo de Trabajo

| **N°** | **Rut** | **Apellidos** | **Nombres** |
| --- | --- | --- | --- |
| 1 | 20.467.315-2 | Tapia | Maximiliano |
| 2 | 20.421.314-3 | Morales | José |
| 3 | 20.108.601-9 | Ávalos | Nicolas |

**ÍNDICE DE CONTENIDO**

[ABSTRACT (RESUMEN) 4](#_rxg74q4fb5x1)

[DESCRIPCIÓN Y RELEVANCIA DEL PROYECTO 5](#_6aontsrob6hf)

[RELACIÓN COMPETENCIAS/PERFIL DE EGRESO 6](#_oqpn1s862d1u)

[RELACIÓN CON INTERESES PROFESIONALES 7](#_381g3jufx2eh)

[FACTIBILIDAD DEL PROYECTO 8](#_lea1jypsvsaj)

[OBJETIVOS DEL PROYECTO 9](#_84nlpi4thm1d)

[METODOLOGÍA DEL PROYECTO 10](#_hym0khkqw10c)

[ARQUITECTURA TÉCNICA Y DISEÑO DEL SISTEMA 11](#_rvtj5kpij452)

[Arquitectura lógica: Modelo-Vista-Controlador (MVC) 11](#_ouum4uifxqzb)

[Arquitectura de infraestructura y despliegue 11](#_kdq0bztqh10a)

[STACK TECNOLÓGICO DETALLADO 12](#_3avvzebrkkil)

[Tabla de stack con sus tecnologías 12](#_7hhdukfrogsm)

[Modelo de datos 13](#_idurhzt85jil)

[PLAN DE TRABAJO PROYECTO 16](#_4p3qrhpfo282)

[CARTA GANTT DEL PROYECTO CAPSTONE 18](#_vz6o49m7h9tj)

[CONCLUSIÓN 20](#_lsg1kf28hqm7)

[FUENTE BIBLIOGRÁFICA 21](#_spr2ynwbjuja)

# ABSTRACT (RESUMEN)

El informe desarrollado a continuación detalla la definición de nuestro proyecto de Capstone (Proyecto APT) “Sistema de gestión CAPSI” una solución de software integral que busca optimizar los procesos clínicos y administrativos del Centro de Atención Psicológica y Estudios (CAPSI) de la Universidad de la Serena. El sistema, desarrollado como aplicación web responsiva, satisface la necesidad de digitalizar la gestión de pacientes, planes de intervención, sesiones terapéuticas y roles de usuario, facilitando a su vez la recopilación de datos para la investigación académica. La arquitectura del proyecto se basa en el patrón MVC. La infraestructura se despliega en la nube a través de Railway con contenedores Docker, garantizando escalabilidad y un flujo de trabajo CI/CD eficiente. El objetivo principal es entregar una herramienta robusta, segura y de alta calidad que mejore la eficiencia clínica y apoye la formación de futuros profesionales de la psicología.

# DESCRIPCIÓN Y RELEVANCIA DEL PROYECTO

El objetivo de este proyecto es desarrollar una solución web responsiva, accesible tanto desde computadores como desde dispositivos móviles, que permita, en términos generales, lo siguiente:

* Gestión de roles (psicólogos, estudiantes, secretarios, administradores e investigadores).
* Creación y gestión de pacientes.
* Creación y gestión de planes de intervención.
* Gestión de sesiones por paciente.
* Envío de instrumentos CORE.
* Descarga de datos en distintos formatos.

En síntesis, se trata de un sistema de gestión integral que busca cubrir diversas necesidades:

* Por un lado, facilitar que los estudiantes de psicología realicen sus prácticas profesionales llevando un control de las sesiones de manera supervisada.
* Por otro, optimizar el trabajo de los psicólogos, reduciendo el tiempo dedicado a anotaciones y al envío de formularios.
* Finalmente, al almacenar la información de los pacientes en tablas y con etiquetas específicas, se habilita la posibilidad de realizar análisis de datos, lo que resulta de gran utilidad para los investigadores de doctorado.

Este proyecto contempla numerosas funcionalidades y detalles que fueron definidos de manera específica por la universidad, y que se encuentran documentados en el documento de requerimientos del proyecto.

Escogimos este proyecto porque integra distintas áreas de la informática en un caso real de renovación tecnológica para la Universidad de La Serena. El proyecto, denominado CAPSI, consiste en una solución tecnológica para el servicio de psicología de la universidad, el cual brinda terapia psicológica breve (con un máximo de 12 sesiones).

Al tratarse de una implementación real, el desarrollo abarca diversas disciplinas de la informática, tales como:

* Modelado de datos
* Infraestructura en la nube
* Arquitectura de proyectos
* Desarrollo frontend y backend
* Análisis de requerimientos

La Universidad de La Serena, específicamente el departamento de CAPSI, requiere un sistema que permita calendarizar sesiones y gestionar pacientes, psicólogos e investigadores. Esto facilitará el análisis de datos y apoyará a los alumnos de doctorado en sus investigaciones.

Además, CAPSI es el espacio donde los estudiantes de psicología de la institución realizan sus prácticas profesionales, por lo que la calidad de este sistema también impacta directamente en la calidad de su formación académica.

# RELACIÓN COMPETENCIAS/PERFIL DE EGRESO

El Proyecto APT CAPSI se relaciona directamente con el perfil de egreso de la carrera, ya que constituye una solución de software innovadora aplicada a un caso real dentro de la Universidad de La Serena. Su propósito es optimizar la gestión de sesiones, pacientes, psicólogos e investigadores en el área de psicología, a través del diseño e implementación de un sistema que integra diversas áreas de la informática: modelado de datos, desarrollo frontend y backend, arquitectura en la nube, y análisis de requerimientos.Estas características permiten vincular de manera práctica los conocimientos adquiridos en la carrera con una problemática real, impactando tanto en la calidad de la atención brindada por CAPSI como en la formación de los estudiantes de psicología que realizan sus prácticas profesionales en este espacio.

# RELACIÓN CON INTERESES PROFESIONALES

El interés principal que tiene nuestro equipo de trabajo es el desarrollo de soluciones tecnológicas que generen un impacto real y positivo. Y este proyecto es ideal para nosotros. Permitiéndonos desde un desarrollo desde 0 full-stack con la implementación de Laravel, JavaScript, MySQL, Git, Railways y Docker usandolas integralmente para ofrecer una solución tecnológica. Trabajando con tecnologías en la nube y enfrentando desafíos de seguridad y cifrado de datos de los pacientes.

Adicionalmente, como profesionales buscamos hacer un impacto social positivo para los pacientes y la formación académica e investigativa de los psicólogos.

# FACTIBILIDAD DEL PROYECTO

Como equipo consideramos que es factible desarrollar nuestro proyecto APT, debido a que este proyecto empezó en julio y fue entregado para marcha blanca en agosto, hasta el dia de hoy se sigue trabajando en el proyecto corrigiendo errores y manteniéndolo para los clientes, esto a finales de este semestre ya estaría terminado y documentado al 100%.

Las horas asignadas son correctas para poder organizarnos como grupo y delegar tareas entre todos. Contamos con equipos portátiles para trabajar y desarrollar el trabajo, como trabajaremos en la nube no se necesita cosa mayor. El acercamiento al cliente es un factor externo importante que nos facilita el desarrollo, debido a que siempre está atento para solicitarnos un cambio o explicar nuevamente los requerimientos. Un factor externo que puede dificultar nuestro desarrollo es la privacidad y cifrado de datos de los pacientes, también teniendo que siempre tener un respaldo para presentar a organizaciones del Estado.

Adicionalmente, la factibilidad técnica se sustenta en la elección de un stack tecnológico maduro y ampliamente documentado (Laravel, MySQL, Bootstrap), lo que minimiza la curva de aprendizaje en componentes críticos y acelera la resolución de problemas. La modularidad inherente del framework Laravel también ha permitido una división de tareas eficiente entre los miembros del equipo, asegurando un desarrollo paralelo y un avance constante, validado por la entrega exitosa del producto final.

Teniendo en cuenta estos factores, es posible desarrollar este proyecto correctamente en el plazo del semestre

# 

# OBJETIVOS DEL PROYECTO

Desarrollar un sistema web responsivo e integral para el servicio de psicología CAPSI de la Universidad de La Serena, que permita la gestión de pacientes, sesiones, planes de intervención, roles de usuarios y análisis de datos, con el fin de mejorar la eficiencia del trabajo clínico, apoyar la formación práctica de estudiantes de psicología y facilitar la investigación académica.

**Objetivos específicos:**

* Diseñar e implementar un sistema de gestión de roles que contemple a psicólogos, estudiantes, secretarios, administradores e investigadores, garantizando la adecuada distribución de funciones y niveles de acceso dentro de la plataforma.
* Desarrollar un módulo de registro y administración de pacientes que permita almacenar información personal, historial clínico y seguimiento de casos, asegurando la trazabilidad de los procesos terapéuticos.
* Implementar un sistema para la planificación y gestión de sesiones, que incorpore la calendarización, el registro de actividades clínicas y la supervisión de las intervenciones realizadas por estudiantes y profesionales.
* Integrar la aplicación y gestión de instrumentos psicológicos estandarizados, como el CORE, mediante un mecanismo de envío, recepción y almacenamiento que facilite su análisis posterior.
* Habilitar funcionalidades de exportación de datos en diversos formatos, de manera que los usuarios puedan disponer de la información recopilada para fines académicos, administrativos o de investigación.
* Optimizar los procesos administrativos del equipo clínico, reduciendo la carga asociada a tareas manuales como anotaciones, generación de reportes y envío de formularios, con el fin de mejorar la eficiencia en la gestión del servicio.
* Diseñar una base de datos estructurada y categorizada, orientada a facilitar el análisis de la información por parte de investigadores y estudiantes de posgrado, contribuyendo al desarrollo de estudios y publicaciones científicas.
* Asegurar la accesibilidad y usabilidad del sistema en múltiples dispositivos, a través de un diseño responsivo que permita su uso tanto en computadores como en teléfonos móviles.

# METODOLOGÍA DEL PROYECTO

Para abordar el problema identificado, se utilizará una metodología tradicional en el desarrollo de software, siguiendo las etapas clásicas del ciclo de vida: análisis, diseño, desarrollo, pruebas, implementación y mantenimiento. Este enfoque permite mantener un control estructurado y secuencial de las fases del proyecto, asegurando que cada una se complete antes de avanzar a la siguiente, lo que se ajusta a las características y objetivos del sistema CAPSI.

El proceso comenzará con el levantamiento y análisis de requerimientos, considerando las necesidades del servicio de psicología y los usuarios finales (psicólogos, estudiantes en práctica e investigadores). Posteriormente, se elaborará el diseño de la solución, que incluye el modelado de datos, definición de la arquitectura del sistema, diagramas de flujo y prototipos de interfaz. Una vez validado este diseño, se procederá a la etapa de desarrollo, donde se implementará la solución tecnológica tanto en frontend como en backend, junto con la infraestructura en la nube.

Tras la construcción, se realizará la fase de pruebas, destinadas a verificar la calidad, seguridad y correcto funcionamiento del sistema, considerando tanto pruebas unitarias como de integración. Finalmente, se llevará a cabo la implementación en el entorno productivo y la documentación del sistema, dejando también previsto un plan básico de mantenimiento correctivo y evolutivo.

Actualmente, el proyecto se encuentra en la fase de pruebas (y verificación). Esta etapa como dijimos es importante para asegurar la máxima calidad del producto y obtener un feedback del mismo para corregir los errores que se encuentren en el sistema.

Una vez superada esta prueba se procederá con la implementación final y documentación final para el mantenimiento futuro del sistema. Este enfoque nos ha ayudado a tener un producto sólido y que se espera satisfacer al máximo las necesidades del cliente. Dándoles un software confiable y de alta calidad.

Respecto a la organización del equipo, las funciones y responsabilidades se distribuyen de la siguiente forma:

Nicolás Ávalos (Jefe de Proyecto y Líder Técnico): Encargado de la gestión global del proyecto, coordinación del equipo, definición de la arquitectura técnica y toma de decisiones estratégicas. Supervisa el cumplimiento de plazos y asegura la calidad de la solución.

José Morales (Desarrollador Fullstack): Responsable del desarrollo de funcionalidades en frontend y backend, implementación de la lógica de negocio, pruebas unitarias y apoyo en la integración de la base de datos.

Maximiliano Tapia (Desarrollador Fullstack): Encargado del desarrollo en frontend y backend, integración de servicios en la nube, pruebas de sistema y documentación técnica del software

De esta manera, el equipo trabajará de forma coordinada bajo un marco tradicional, asegurando que el proyecto avance de manera ordenada, cumpliendo con los requerimientos planteados y con los objetivos establecidos para el sistema CAPSI

# ARQUITECTURA TÉCNICA Y DISEÑO DEL SISTEMA

La aplicación del sistema CAPSI ha sido diseñada con un stack que puede garantizar robustez, seguridad, escalabilidad y facilidad de mantenimiento. En el siguiente párrafo, se detallarán las decisiones claves del diseño.

## Arquitectura lógica: Modelo-Vista-Controlador (MVC)

La aplicación está hecha bajo un patrón arquitectónico muy conocido, el modelo vista controlador, implementado en Laravel. Esta elección no fue decidida porque sí, sino que también se pensó en los beneficios directos al proyecto.

El **modelo** representa la lógica de negocio y la interacción con la base de datos. Centraliza todas las operaciones de datos (creación, lectura, actualización, eliminación), asegurando consistencia y permitiendo que la lógica de negocio no se acople a la presentación (vista).

La **vista** compuesta por archivos Blade, se encarga de la presentación de la interfaz del usuario. Las vistas no contienen la lógica del negocio, reciben los datos directamente desde los controladores, lo que permite una partición entre vista/negocio, permitiendo a los de trabajo de front-end no interferir en el trabajo de los de backend.

El **controlador** en este proyecto actuará como intermediario, gestionando las peticiones del usuario, interactuando con los modelos para obtener o manipular datos. Esta separación es clara de las responsabilidades clave que son para tener un código organizado y que cumpla estas reglas.

## Arquitectura de infraestructura y despliegue

Se ha decidido utilizar una arquitectura moderna basada en contenedores y un flujo de integración y despliegue continuo (CI/CD) que minimiza errores manuales y agiliza la entrega del producto a actualizaciones.

* Entorno de desarrollo local: Para garantizar que haya uso correcto de los entornos de desarrollo y producción se ha utilizado Docker y Docker Compose. Cada desarrollador debe ejecutar la aplicación en contenedores aislados replicando fielmente el entorno de producción.
* Plataforma de despliegue: Se eligió **Railway** como plataforma como servicio (PaaS) Esta decisión nos ayuda con la complicación de la gestión de servidores, permitiendo al equipo enfocarse en el desarrollo continuo. Railway se conecta directamente al repositorio de GitHub y despliega automáticamente cualquier cambio en la Branch principal (main) al entorno de Staging
* Almacenamiento de archivos (Storage): Para gestionar archivos subidos por los usuarios (fotos de perfil, documentos adjuntos), se utiliza **BackBlaze B2**, un servicio de almacenamiento de objetos. Esta práctica ofrece **mejor rendimiento, escalabilidad independiente y seguridad**.

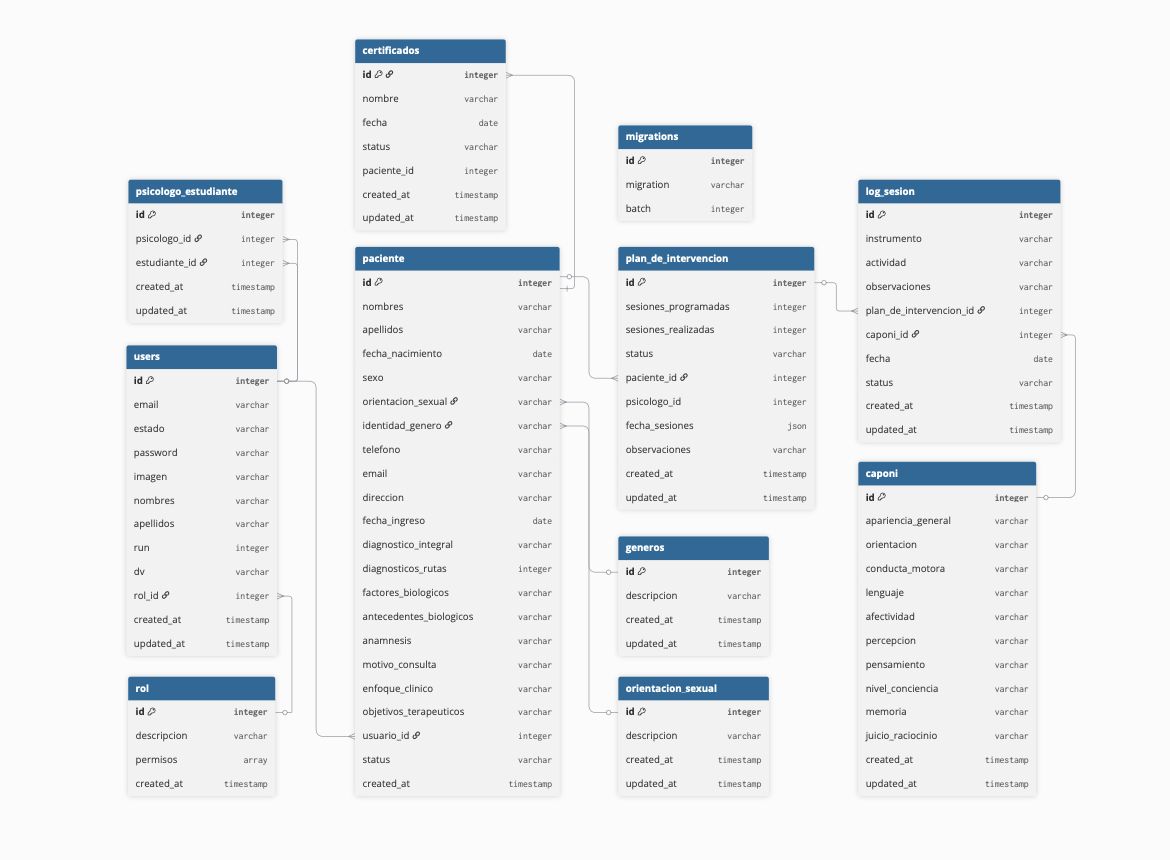
# STACK TECNOLÓGICO DETALLADO

## Tabla de stack con sus tecnologías

| Componente | Tecnología | Versión | Justificación |
| --- | --- | --- | --- |
| Backend | Laravel | 11.x | Framework PHP que provee MVC, ORM (Eloquent), sistema de rutas, seguridad y un ecosistema que acelera el desarrollo. |
|  | PHP | 8.2+ | Stack de backend cómodo, actual y funcional escogido por comprensión del equipo |
|  | MySQL (producción) | 8.0+ | Sistema de gestión de bases de datos relacional líder en la industria. |
|  | SQLite (desarrollo) | 3.50.4 | Con el fin de usar una herramienta más rápida y de menos costo para desarrollo. En producción nos preocupamos de utilizar una de mayor calidad en la industria. |
| Frontend | Bootstrap | 5.3.3 | Permite la construcción rápida de una interfaz de usuario limpia, profesional y responsiva. |
|  | Vanilla JS | ES6+ | Se utiliza para la interactividad de la interfaz, validaciones de formularios sin la sobrecarga de un framework pesado. |
|  | Vite | 5.3.5 | Herramienta que se utiliza para compilar y ofrece un desarrollo muy rápido de los archivos. |
| Infraestructura | Docker |  | Garantiza entornos de desarrollo consistentes y portables. |
|  | Railway |  | Automatiza la creación y gestión de la infraestructura en la nube, manejando el despliegue, escalado y entorno. |
|  | BackBlaze B2 |  | Servicio S3 compatible para almacenar y disponer de archivos estáticos de forma eficiente. |
| Herramientas | Laravel DomPDF |  | Permite crear documentos PDF dinámicamente desde el código, esencial para reportes clínicos y certificados |
|  | PHPUnit | 10.5 | Framework para escribir y ejecutar pruebas unitarias y de integración, asegurando calidad del backend. |

## Modelo de datos

La persistencia de los datos se gestiona a través de MySQL, cuyo diseño ha sido modelado para asegurar la integridad referencial, consistencia y eficiencia en las consultas.



A continuación se hará una descripción del modelo a mayor rasgo:

* Gestión de acceso: La tabla *users* almacena la información de autenticación de todos los actores del sistema. La relación *users.rol\_id* con la tabla *rol* implementa un sistema de control basado en roles la tabla *rol* contiene un campo de *permisos* que define las acciones permitidas para cada rol, proporcionando una capa de seguridad centralizada.
* Paciente, Plan Intervención y Log Session: Esta es la estructura principal de la cual se recogerán los datos para investigación académica. Un *paciente* puede tener múltiples *plan\_intervencion* a lo largo del tiempo (aunque generalmente uno activo). Cada *plan\_intervencion* agrupa un conjunto de *Log\_Session,* creando una distinción clara para la trazabilidad completa del proceso terapéutico de un paciente. La tabla *plan\_de\_intervencion* utiliza un campo JSON (*fecha\_sesiones)* para registrar de manera flexible las fechas planificadas y realizadas.
* Las tablas como *certificado* y *caponi* están diseñadas como una extensión de la información del paciente, almacenando información específica de evaluaciones o documentos generados (como un certificado). Estas relaciones permiten que el historial del paciente crezca de forma modular.
* Supervisión académica (*Psicologo\_Estudiante)*: Esta tabla es fundamental para el ambiente formativo de CAPSI. Vincula a un psicólogo supervisor con un estudiante en práctica, permitiendo al sistema implementar funcionalidades donde el supervisor puede revisar y validar las anotaciones o planes creados por el estudiante (incluso, editarlos).

# PLAN DE TRABAJO PROYECTO

| **Plan de Trabajo Proyecto APT** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Competencia o unidades de competencias | Nombre de Actividades/Tareas | Descripción Actividades/Tareas | Recursos | Duración de la actividad | Responsable[[1]](#footnote-0) | Observaciones |
| **Análisis y diseño de requerimientos** | Fase previa 1: Planificación y diseño | Reuniones con Stakeholders, levantamiento de requerimientos funcionales y no funcionales, definición roles de usuario, modelo Entidad-Relacion de la base de datos, creación de mockups | Documentos de Google Docs, herramientas de diagramación (Figma) | 3 Semanas (YA COMPLETADO | Nicolas Ávalos (Líder) | Inicio del proyecto, definición de ‘’qué’’ y ‘’cómo’’ en profundidad |
| **Desarrollo de Software / Infraestructura en la nube** | Fase previa 2: Desarrollo del MVP | Configuración del entorno de desarrollo, desarrollo del backend, lógica de negocio, APIs, gestión de roles. Desarrollo del Front End, integración de base de datos y despliegue inicial en Railways. Implementación de funcionalidades clave | Laravel, JS, MySQL, Git, Github, Docker, Railways, VS Code | 4 semanas (YA COMPLETADO) | Nicolas Ávalos (Líder) | Construcción principal del sistema basado en los requerimientos iniciales del cliente |
| **Gestión de proyectos/ Análisis de requerimientos** | Fase 3: Capacitación y Marcha blanca | Sesión de capacitación a usuarios clave. Despliegue del MVP para uso real y supervisado. Recopilación de feedback inicial. | Sistema desplegado, herramientas de comunicación (Discord, Microsoft Teams) | 2 semanas (Actual) | Nicolas Ávalos (coordina), Equipo completo (analiza) | Estamos en esta fase del proyecto APT. Fase crucial para validar el desarrollo y planificar los siguientes pasos. |
| **Desarrollo de software/ Optimización** | Fase 4: Refinamiento, optimización. | Corrección de bugs reportados durante la marcha blanca. Optimización de consultas a la base de datos.  Refuerzo de la seguridad (cifrado, validación de permisos)  Implementación de mejoras menores de usabilidad (UX). | Laravel, JS, My SQL, Git, Docker, VS Code | 2 semanas | José Morales y Maximiliano Tapia (Desarrollo) Nicolas Ávalos (Supervisión) | Elevar calidad del MVP un producto final. |
| **Aseguramiento de la calidad** | Fase 5: Pruebas de regresión y aceptación final | Ejecución de un plan de pruebas integral. Verificación de que las correcciones no crearon nuevos errores. Presentación de la versión final estable al cliente para su aprobación formal. | Plan de pruebas, entorno de staging. | 2 semanas | Maximiliano Tapia (Lidera) Equipo completo (Apoyo) | Validación final del proyecto |
| **Documentación y comunicación** | Fase 6: Creación de documentación final | Consolidación y redacción formal del Manual de usuario y Manual técnico del sistema en su versión final.. | Google Docs, Captura de pantallas, Git | 4 semanas | Jose Morales (Documentación) Equipo completo (Corrección y aprobación | Esencial para mantenibilidad del sistema y correcta comunicación con el cliente. |
| **Gestión de proyectos** | Fase 7: Entrega, capacitación y cierre | Empaquetado del proyecto (código, BBDD, documentos). Sesión de capacitación con usuarios finales. Entrega formal del proyecto y credenciales a la U. de La Serena. Presentación final para la asignatura Capstone. | Repositorio, Git, Presentación | 1 semanas | Nicolas Ávalos (Coordina) Equipo completo (Presenta) | Cierre exitoso del ciclo de vida del proyecto |

# CARTA GANTT DEL PROYECTO CAPSTONE

# 

# CONCLUSIÓN

El desarrollo del proyecto "Sistema de Gestión CAPSI" ha representado un desafío tecnológico de ingeniería de software, abordando con éxito el desafío de digitalizar y optimizar los procesos de un centro de atención psicológica real. Siguiendo una metodología en cascada, el proyecto es robusto, el cual se encuentra actualmente en la fase crítica de Verificación y Pruebas.

La implementación de un stack tecnológico moderno, centrado en Laravel y una infraestructura en la nube con Docker y Railway, ha permitido no solo cumplir con los requerimientos funcionales, sino también establecer una base sólida para la escalabilidad, seguridad y mantenibilidad futuras del sistema. El diseño del modelo de datos, la arquitectura MVC y el flujo de trabajo CI/CD son testimonio de la aplicación de buenas prácticas de la industria.

Finalmente, el proyecto CAPSI será un producto de software de alta calidad, alto rendimiento y además cumplirá un valor importante para los psicólogos académicos que busquen investigar para tesis o investigaciones que pueden ayudar a los pacientes.

# FUENTE BIBLIOGRÁFICA

Considerar: Norma APA 7ma Edición.

1. En caso de que el Proyecto APT sea grupal, en esta columna deben indicar el nombre de los responsables de cada tarea o actividad. Esto posteriormente permitirá diferenciar la evaluación por cada integrante. [↑](#footnote-ref-0)