}

1. **PARTE I**

| **1. Antecedentes Personales** |
| --- |

| Nombre estudiante | Fernando cavada;Lucas Cisternas; Joaquín Madariaga |
| --- | --- |
| Rut | **21.483.547-8;20.761.504-8;21.265.231-8** |
| Carrera | Ingeniería en Informática |
| Sede | Maipú |

| **2. Descripción Proyecto APT** |
| --- |

| Nombre del proyecto | Psicolink |
| --- | --- |
| Área (s) de desempeño(s) | **Análisis y planificación de requerimientos informáticos:** El proyecto parte con un levantamiento de necesidades (usuarios, psicólogos, administradores) y definición de backlog en Scrum.  **Gestión de proyectos informáticos:** Se usa Scrum como metodología ágil, incluyendo roles, sprints, backlog y roadmap.  **Programación de software:** Desarrollo de aplicación móvil (Ionic/Angular), backend (Django), APIs y plataforma web.  **Inteligencia de negocios:** Incorporación de métricas de uso, reportes, estadísticas en el panel de administración y soporte a la toma de decisiones.  **Análisis y desarrollo de modelos de datos:** Modelado de la base de datos en PostgreSQL para soportar perfiles, citas, pagos y transacciones.  **Arquitectura de software**: Se utiliza una arquitectura en capas (presentación, lógica de negocio y datos), desplegada en la nube con servicios externos. El backend en Django será alojado en Render, y la base de datos se gestionará en Supabase como instancia PostgreSQL en la nube.  **Calidad de software:** Implementación de pruebas unitarias, de integración y de carga; además del cumplimiento de normas de calidad (ISO 25010, ISO 9001). |
| Competencias | **Competencias de especialidad**   * **Ofrecer propuestas de solución informática analizando de forma integral los procesos:** El proyecto responde a una problemática social con una solución digital integral. * **Desarrollar una solución de software utilizando técnicas sistematizadas**: Se sigue Scrum y buenas prácticas de desarrollo. * **Construir modelos de datos para soportar los requerimientos de la organización**: Modelado de Base de datos en PostgreSQL. * **Programar consultas o rutinas para manipular información de una base de datos**: Implementación de lógica para agendamiento, perfiles, pagos, etc. * **Construir el modelo arquitectónico de una solución sistémica**: Infraestructura en la nube, integración de múltiples servicios como pasarelas de pago, videollamadas, etc. * **Resolver vulnerabilidades sistémicas para asegurar cumplimiento de normas de seguridad**: Aplicación de ISO 27001, GDPR (ley de protección de datos) y encriptación. * **Gestionar proyectos informáticos**: Planificación, control de riesgos, backlog, roadmap.   **Competencias genéricas**   * **Desarrollar la habilidad emprendedora y elaborar proyectos innovadores**: PsicoLink es un proyecto innovador en el país que responde a una necesidad social en auge. * **Reconocer un desempeño correcto en situaciones de la profesión**: Aplicación de buenas prácticas de codificación, seguridad y calidad. |

| **3. Fundamentación Proyecto APT** |
| --- |

| Relevancia del proyecto APT | El proyecto busca dar respuesta a la dificultad de acceso a atención psicológica, la cual se ve limitada por la escasez de plataformas digitales confiables que permitan verificar las credenciales de los profesionales, gestionar citas de manera sencilla y ofrecer un espacio seguro para la comunicación. A esto se suma el estigma social que muchas veces impide que las personas busquen ayuda oportuna, generando desinformación y desconfianza en los servicios disponibles.  Esta problemática se sitúa en el contexto nacional, afectando principalmente a jóvenes y adultos que buscan atención en salud mental en Chile, donde existe una creciente necesidad de servicios accesibles, rápidos y confiables. El impacto se dirige tanto a los usuarios que requieren apoyo psicológico como a los profesionales del área, quienes necesitan herramientas digitales que respalden su labor y den confianza a sus pacientes.  Desde la perspectiva de la Ingeniería en Informática, la problemática es altamente relevante, ya que el desarrollo de soluciones tecnológicas puede contribuir directamente a mejorar la calidad de vida de las personas mediante plataformas seguras, accesibles y transparentes. El diseño e implementación de sistemas de verificación de identidad, procesos de agendamiento automatizados y espacios digitales protegidos son campos propios de la profesión, vinculados a la innovación en servicios digitales y la seguridad informática. El aporte de este proyecto es ofrecer un valor real al contexto social y laboral, simulando un sistema que, de ser implementado, permitiría mejorar la confianza en la atención psicológica, optimizar el uso de los recursos y facilitar el acceso oportuno a profesionales certificados. |
| --- | --- |
| Descripción del Proyecto APT | El proyecto tiene como objetivo desarrollar una plataforma digital segura y accesible que permita a las personas acceder a atención psicológica de manera confiable, mediante la verificación de credenciales de los profesionales, la automatización del agendamiento de citas y la creación de un espacio protegido para la comunicación entre usuarios y especialistas. Con ello se busca disminuir las barreras de acceso, reducir la desinformación y fomentar la confianza en la búsqueda de apoyo en salud mental.  La propuesta consiste en diseñar e implementar una aplicación web que integre funcionalidades clave como:   * **Sistema de verificación de credenciales** para garantizar que los profesionales de la salud mental estén debidamente acreditados. * **Módulo de agendamiento automatizado**, que permita a los usuarios seleccionar horarios disponibles y gestionar sus citas de manera sencilla. * **Entorno de comunicación seguro**, con protocolos de protección de datos que resguarden la privacidad de la información sensible. * **Interfaz amigable**, pensada para que cualquier usuario pueda navegar y acceder sin necesidad de altos conocimientos tecnológicos. * **Integración de Zoom para videollamadas**, ofreciendo un canal de atención en línea que combine accesibilidad y seguridad en la comunicación entre profesional y paciente. * **Integración con pasarela de pago Stripe,** que permita a los usuarios pagar por los servicios de los profesionales de la salud mental. |
| Pertinencia del proyecto con el perfil de egreso | El Proyecto APT se relaciona directamente con el perfil de egreso de Ingeniería en Informática, ya que plantea el diseño y desarrollo de una plataforma digital segura que facilite el acceso a atención psicológica mediante verificación de profesionales, agendamiento automatizado e integración de videollamadas. Este tipo de iniciativa requiere levantar y analizar requerimientos de usuarios, diseñar arquitecturas tecnológicas robustas, integrar sistemas externos (como Zoom o Stripe) y aplicar medidas de ciberseguridad, todos ellos ámbitos contemplados en el perfil de egreso.  En particular, las **competencias profesionales seleccionadas** resultan esenciales para abordar la problemática:   * **Ofrecer propuestas de solución informática**: el proyecto surge como respuesta a una necesidad social concreta, transformándola en un sistema tecnológico viable que aporta valor real. * **Desarrollar una solución de software sistematizada y con aseguramiento de calidad**: la plataforma debe garantizar un correcto funcionamiento en la gestión de citas, validación de credenciales y en la comunicación segura entre usuarios y profesionales. * **Construir modelos de datos de acuerdo a un diseño definido y escalable en el tiempo**: la solución requiere un almacenamiento confiable y escalable de información sensible (usuarios, horarios, historial de atenciones) * **Programar consultas o rutinas para manipular información de una base de datos:** Las bases de datos que manejan la información de la plataforma deben tener consultas eficientes para su gestión. * **Construir el modelo arquitectónico de una solución sistémica**: integrar módulos como agenda, verificación, comunicación y Zoom requiere un diseño arquitectónico sólido que soporte la escalabilidad y la seguridad. * **Resolver vulnerabilidades sistémicas y asegurar cumplimiento de normas de seguridad**: la protección de datos de salud es crítica, por lo que el proyecto debe implementar buenas prácticas de seguridad exigidas por la industria. * **Gestionar proyectos informáticos**: la planificación, priorización de funcionalidades y la gestión del ciclo de vida del software son competencias claves para llevar este proyecto desde la idea hasta la implementación. |
| Relación con los intereses profesionales | Nuestros intereses profesionales se orientan hacia el desarrollo de soluciones tecnológicas innovadoras que aporten valor a la sociedad, especialmente en ámbitos donde la informática puede mejorar la calidad de vida de las personas mediante sistemas accesibles, seguros y confiables. Nos interesa particularmente el diseño de software, la integración de tecnologías y la aplicación de buenas prácticas de ciberseguridad en proyectos que tengan impacto real en el entorno social.  El Proyecto APT refleja directamente estos intereses, ya que busca implementar una plataforma digital para mejorar el acceso a la atención psicológica, integrando herramientas como verificación de profesionales, agendamiento automatizado y videollamadas seguras mediante Zoom. Este enfoque combina innovación tecnológica con un alto impacto social, alineándose con nuestra motivación de desarrollar soluciones informáticas con propósito. |
| Factibilidad de desarrollo del Proyecto APT | El desarrollo del Proyecto APT es posible dentro del semestre, ya que se cuenta con el tiempo y las horas asignadas a la asignatura para avanzar en cada etapa: levantamiento de requerimientos, diseño, desarrollo, pruebas y documentación. Además, ya fue discutido con varios docentes sobre su factibilidad, los cuales, validaron la factibilidad de la propuesta.  En cuanto a los materiales y herramientas, el equipo dispone de los recursos necesarios: computadores personales, acceso a internet y un conjunto de tecnologías probadas como *HTML, CSS, Angular, Django, Supabase, PostgreSQL, Git,* entre otras. Estas herramientas permiten construir un sistema seguro, escalable y adaptable a las necesidades del proyecto.  Los factores externos que facilitan su desarrollo incluyen la amplia disponibilidad de documentación, tutoriales y librerías asociadas a las tecnologías seleccionadas, además del soporte de la comunidad de desarrolladores y la experiencia previa del equipo en su uso.  Por otro lado, los factores externos que podrían dificultar el desarrollo son principalmente la integración entre múltiples servicios (por ejemplo, conexión entre módulos de agenda, pagos en línea, verificación y videollamadas), además se tiene en consideración el factor interno de la gestión de tiempo de cada integrante del equipo dentro del semestre. Para mitigar estas dificultades, se plantea trabajar con metodologías ágiles, utilizando aplicaciones para trabajar remotamente durante todo el tiempo de desarrollo del proyecto y ajustando el alcance si fuera necesario. |

1. **PARTE II**

| **4. Objetivos** |
| --- |

| Objetivo general | Desarrollar una plataforma digital segura y accesible que facilite el acceso a atención psicológica, integrando verificación de credenciales de profesionales, agendamiento automatizado y videollamadas mediante Zoom, con el fin de reducir barreras de acceso y aumentar la confianza en los servicios de salud mental. |
| --- | --- |
| Objetivos específicos | * Implementar un sistema de registro y autenticación segura que garantice la protección de los datos personales y clínicos de los usuarios. * Desarrollar un módulo para la verificación de credenciales de los profesionales de la salud mental, asegurando la validez de su formación y certificaciones. * Diseñar e implementar un sistema de agendamiento automatizado que permita a los usuarios reservar, modificar o cancelar citas de manera eficiente. * Integrar la funcionalidad de videollamadas mediante Zoom o Teams, garantizando calidad en la comunicación entre paciente y profesional. * Crear un entorno intuitivo y accesible, priorizando la usabilidad de la plataforma para personas con distintos niveles de alfabetización digital. * Establecer mecanismos de retroalimentación y evaluación de la atención recibida, con el fin de mejorar continuamente la calidad del servicio. * Desarrollar políticas y protocolos de seguridad informática y confidencialidad, alineados con estándares de protección de datos en salud. |

| **5. Metodología** |
| --- |

| Descripción de la Metodología |
| --- |
| **Metodología de Trabajo**  Para el desarrollo del proyecto PsicoLink se utilizará la metodología ágil Scrum, dado que permite una gestión flexible, colaborativa y orientada a resultados incrementales. Esta metodología facilita la adaptación continua a los cambios y asegura la entrega de valor de forma temprana, lo que resulta clave para un proyecto digital enfocado en la atención psicológica en línea.  El trabajo se organizará en sprints de 1 a 3 semanas, en los cuales se planificaran, desarrollarán y revisarán entregables funcionales de la plataforma. Al final de cada sprint se realizará una revisión y retrospectiva con el objetivo de identificar mejoras y mantener la alineación con los objetivos definidos.  El enfoque se centrará en:   * **Definición del Product Backlog:** identificación de funcionalidades principales (registro seguro, verificación de credenciales, agenda automática, comunicación segura, retroalimentación de usuarios). * **Planificación de Sprints:** priorización de tareas y asignación de actividades alcanzables en cada periodo. * **Desarrollo Iterativo e Incremental:** construcción progresiva de funcionalidades probadas y validadas al finalizar cada sprint. * **Revisión y Retrospectiva:** evaluación del avance, levantamiento de observaciones y propuestas de mejora continua en el producto y en la dinámica del equipo.   Con este enfoque, PsicoLink podrá desarrollarse de manera ordenada, colaborativa y adaptable, asegurando que la solución cumpla con el objetivo de facilitar el acceso seguro y confiable a la atención psicológica en línea.  **Responsabilidades del Equipo**  El equipo está conformado por tres integrantes, quienes asumirán roles principales pero colaborarán en actividades transversales:   * **Joaquín Madariaga – Backend y Coordinación Técnica:** se encargará de la programación de la lógica del negocio en Django y la creación de las APIs para las funcionalidades principales (registro, agendamiento, seguridad de usuarios). También apoyará en la definición técnica de los requerimientos y en la documentación del código. * **Fernando Cavada – Base de Datos y Backend de Apoyo:** será responsable del diseño e implementación de la base de datos en PostgreSQL, desarrollando consultas y rutinas necesarias para el funcionamiento del sistema. Colaborará en el backend con la integración de módulos y validaciones. * **Lucas Cisternas – Frontend y Calidad:** se ocupará de la programación de la aplicación móvil en Ionic/Angular y del portal web, asegurando la correcta interacción con las APIs. Además, liderará las pruebas de calidad (unitarias, de integración y usabilidad) para garantizar el correcto funcionamiento de la plataforma.   De forma colaborativa, los tres integrantes participarán en la planificación de sprints, revisiones, retrospectivas, gestionar la comunicación con los stakeholders y documentar avances asegurando la toma de decisiones en conjunto y una distribución equitativa de las tareas. |

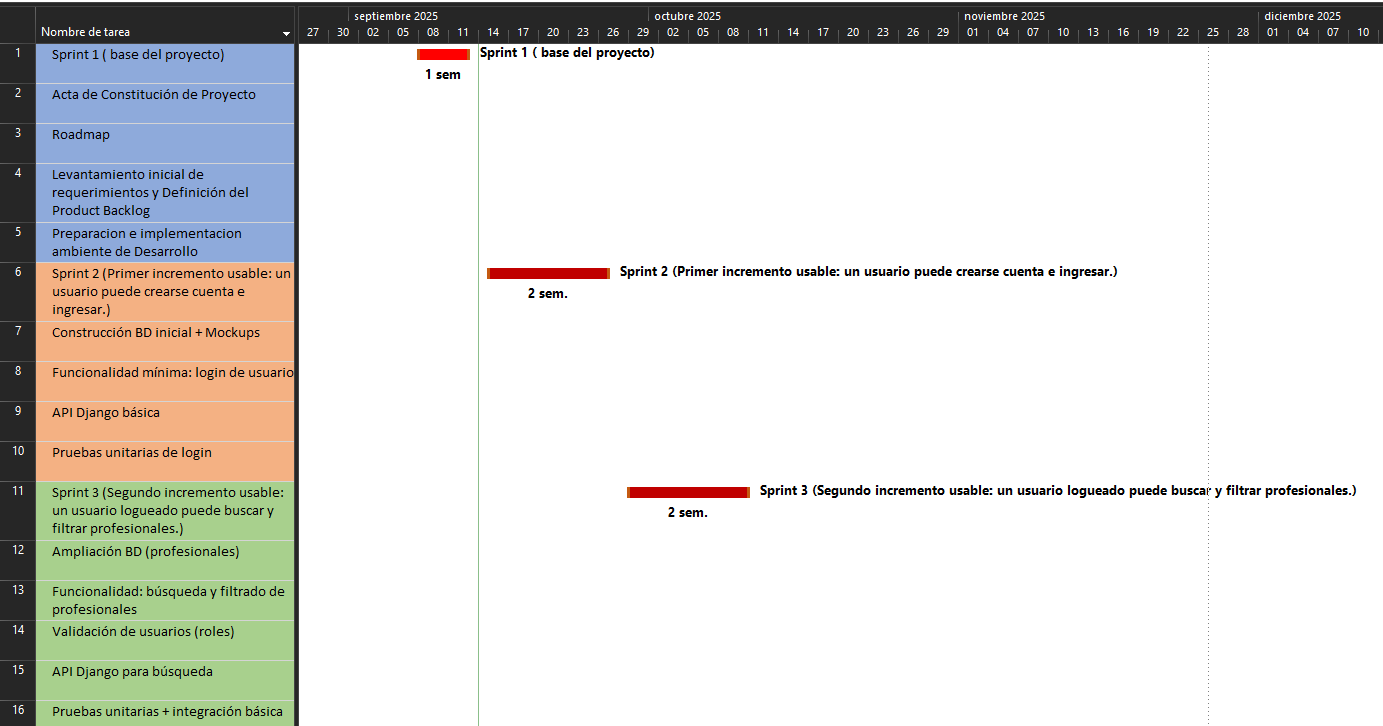
| **6. Evidencias** |
| --- |

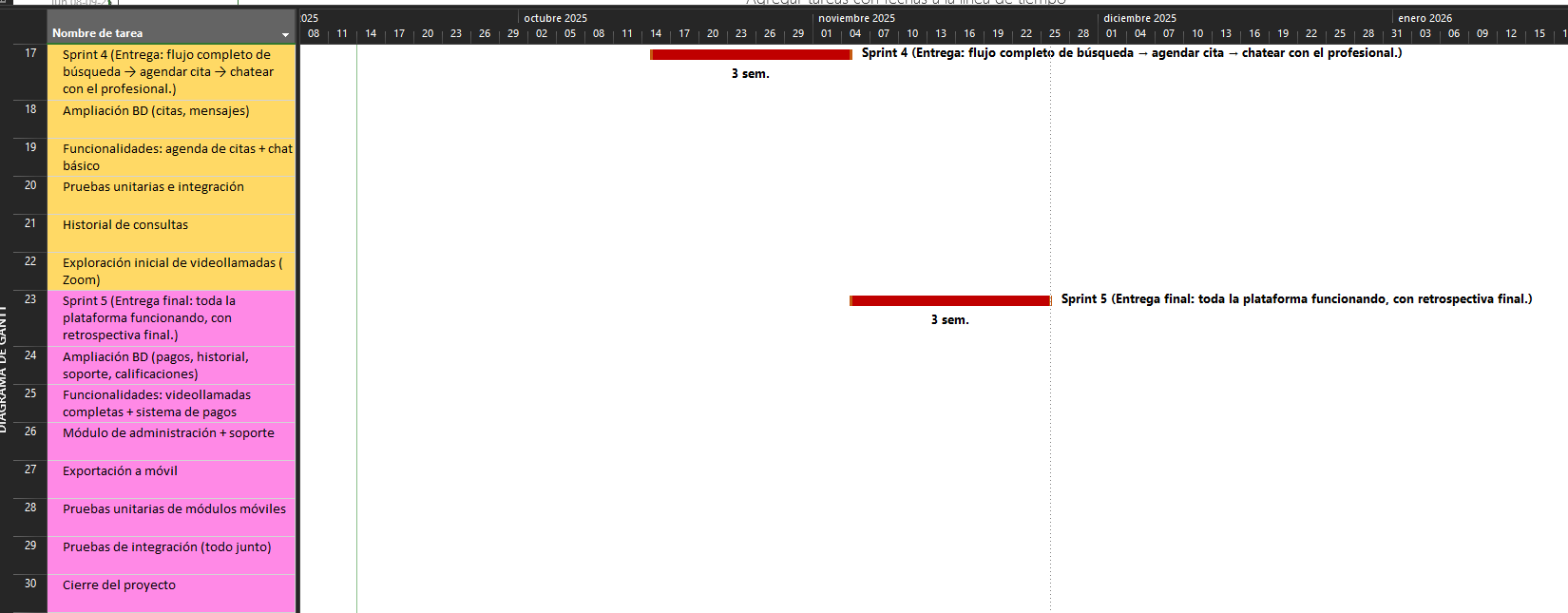
| **Tipo de evidencia**  **(avance o final)** | **Nombre de la evidencia** | **Descripción** | **Justificación** |
| --- | --- | --- | --- |
| **Avance e incremental (se actualiza en cada sprint).** | **Historias de Usuario y Product Backlog** | **Lista priorizada de funcionalidades y requerimientos expresados en historias de usuario.** | **Permite gestionar y dar visibilidad al alcance del proyecto, alineado con las necesidades de los usuarios.** |
| **Entregable (entregas por sprint)** | **Sprint Backlog** | **Conjunto de historias de usuario y tareas seleccionadas para cada sprint.** | **Da claridad sobre qué se va a desarrollar en cada iteración y permite hacer seguimiento del avance.** |
| **Avance (entregas incrementales por sprint) y final (versión integrada).** | **Incremento de producto** | **Versión funcional de la plataforma generada al final de cada sprint** | **Evidencia el progreso real y permite validar continuamente la solución con los stakeholders.** |
| **Entregable** | **Modelo de datos y diagrama de componentes** | **Incluye el modelo de datos de la base PostgreSQL y el diagrama de componentes para visualizar las piezas de software que conforman el sistema.** | **Estos modelos son necesarios para estructurar la solución de manera clara, asegurando consistencia en el diseño. Además, permiten validar la escalabilidad y viabilidad técnica de la solución.** |
| **Avance (entregas incrementales por sprint) y final (versión integrada).** | **Código fuente** | **Repositorio en Git que contiene el desarrollo del backend en Django, la base de datos en PostgreSQL, el portal web y la aplicación móvil híbrida. Cada sprint aporta un incremento funcional que se consolida hacia la versión final.** | **El código es la evidencia tangible del desarrollo. Al mantenerlo en un repositorio, se asegura trazabilidad, control de versiones y colaboración entre los miembros del equipo.** |
| **Avance (sprints) y final (versión integrada).** | **Pruebas automatizadas** | **Conjunto de pruebas unitarias, de integración y funcionales que validan la calidad del sistema. Se implementan gradualmente en cada sprint y se consolidan al final para asegurar la estabilidad de la aplicación antes de la entrega final.** | **Garantizan la calidad del producto y reducen el riesgo de errores al implementar cambios. Se alinean con las buenas prácticas de Scrum al entregar incrementos funcionales probados.** |

| **7. Plan de Trabajo** |
| --- |

| **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 |
| Gestionar proyectos informáticos, ofreciendo alternativas para la toma de decisiones de acuerdo a los requerimientos de la organización. | Acta de Constitución de Proyecto | Formalizar la existencia del proyecto, definir objetivos de alto nivel, requisitos de alto nivel, riesgos iniciales y asignar al Gerente de Proyecto. | -Plantilla de Acta de Constitución  -Reunión de kick-off | Sprint 1 | Joaquín Madariaga | Dificultades: Posible falta de alineación inicial sobre el alcance. Facilitadores: La plantilla estandarizada guía la discusión y asegura que se cubran todos los puntos críticos. |
| Gestionar proyectos informáticos, ofreciendo alternativas para la toma de decisiones de acuerdo a los requerimientos de la organización. | Roadmap | Crear un plan visual de alto nivel que describa la visión, los objetivos estratégicos y las características principales (epics) a lo largo del tiempo (trimestres/sprints). | -Herramientas (Jira, Confluence, Miro)  -Visión del Producto | Sprint 1 | Joaquín Madariaga | Dificultades: definir los tiempos de cada tarea para cumplir en el tiempo y al mismo tiempo dando un margen de error |
| Ofrecer propuestas de solución informática analizando de forma integral los procesos de acuerdo a los requerimientos de la organización. | Levantamiento inicial de requerimientos y definición del Product Backlog | Realizar workshops y usuarios para recopilar y priorizar todas las funcionalidades (historias de usuario). Crear el Backlog inicial priorizado. | -Herramientas (Jira, Trello, Excel)  -Técnicas de entrevista | Sprint 1 | Joaquín Madariaga | Dificultades: Requerimientos ambiguos o contradictorios. |
| Desarrollar una solución de software utilizando técnicas que permitan sistematizar el proceso de desarrollo y mantenimiento, asegurando el logro de los objetivos. | Preparacion e Implementacion ambiente de Desarrollo | Configurar los entornos de desarrollo (dev), staging y producción. Esto incluye servidores, bases de datos, APIs de backend y herramientas de CI/CD (Integración Continua). Los desarrolladores configuran sus estaciones de trabajo locales: IDE, SDKs, emuladores, repositorio de código y acceso a los entornos. | -IDEs (Android Studio)  -Repositorio de código (GitHub) | Sprint 1 | Fernando Cavada, Joaquín Madariaga | Dificultades: Conflictos de configuración entre entornos o dependencias. |
| "Construir modelos de datos y el modelo arquitectónico de una solución sistémica que soporte los procesos de negocio de la organización, de acuerdo a un diseño definido, escalable en el tiempo y cumpliendo los estándares de la industria." | Construcción BD inicial + Mockups | Diseñar, crear y normalizar la base de datos siguiendo las reglas de integridad y consistencia. Se incluye modelado relacional y revisión con el equipo Scrum.  Crear el diseño visual final de la aplicación: paleta de colores, tipografía, iconografía y mockups interactivos que sirvan como guía para el desarrollo. | -DBMS PostgreSQL, Supabase  -Diagramas: modelo de datos  -Herramientas: Figma | Sprint 2 | Fernando Cavada  Lucas Cisternas | Riesgo: posibles cambios en requerimientos que obligan a reestructurar tablas. Se recomienda validación temprana en Sprint Review.  Facilitadores: Figma te permite diseñar mockups interactivos. |
| Construir programas y rutinas de variada complejidad para dar solución a requerimientos de la organización, acordes a tecnologías de mercado y utilizando buenas prácticas de codificación. | Funcionalidad mínima: login de usuario | Este módulo se enfoca en el registro y autenticación de usuarios y profesionales a través de formularios seguros y sencillos. | -Proyecto de la aplicación creado (templates de inicio)  -Modelo de datos de la base de datos de usuarios.  -Base de datos construida | Sprint 2 | Joaquín Madariaga, Lucas Cisternas | Dificultades: Riesgo de bloqueo de cuentas por malas prácticas y es complejo implementar medidas de seguridad robustas (ej. encriptación y prevención de ataques)  Facilitadores: La implementación de un ambiente de pruebas sólido y la metodología Scrum para gestionar la complejidad. |
| Construir programas y rutinas de variada complejidad para dar solución a requerimientos de la organización, acordes a tecnologías de mercado y utilizando buenas prácticas de codificación. | API Django básica | El API Rest es el backend de la aplicación. Consiste en un conjunto de servicios que gestionan la lógica de negocio central, la seguridad, la autenticación y la validación de datos. Realiza operaciones como la creación, lectura, actualización y eliminación de información. | -Base de datos PostgreSQL, Supabase.  -Frontend desarrollado (con al menos templates de inicio)  -Herramientas de pruebas: Jasmine y karma. | Sprint 2 | Joaquín Madariaga, Fernando Cavada | Dificultades: Diseñar endpoints consistentes y seguros, especialmente al manejar datos sensibles.  Facilitadores: La documentación y Django REST Framework ya provee herramientas que aceleran la construcción de APIs confiables. |
| Programar consultas o rutinas para manipular información de una base de datos de acuerdo a los requerimientos de la organización. | Pruebas unitarias de login | Diseñar pruebas automatizadas para validar la seguridad y experiencia de usuario correcta en el inicio de sesión | -Infraestructura basica completa (frontend básico + base de datos para usuarios)  -Pytest | Sprint 2 | Fernando Cavada | Riesgo: tiempo adicional si surgen errores de normalización. |
| Programar consultas o rutinas para manipular información de una base de datos de acuerdo a los requerimientos de la organización. | Ampliación BD (profesionales) | Ampliar la base de datos agregando una nueva entidad para almacenar a los profesionales | -Base de datos construida  -Modelo de datos | Sprint 3 | Fernando Cavada | Puede requerir ajustar la entidad anteriormente creada para usuarios generales. |
| Construir programas y rutinas de variada complejidad para dar solución a requerimientos de la organización, acordes a tecnologías de mercado y utilizando buenas prácticas de codificación. | Funcionalidad: búsqueda y filtrado de profesionales | Este módulo es una funcionalidad de búsqueda que permite a los usuarios encontrar profesionales de la salud mental. Incluye filtros personalizados que posibilitan refinar los resultados por criterios como especialidad, experiencia, ubicación, disponibilidad y calificación, facilitando al usuario encontrar el profesional que mejor se ajuste a sus necesidades. | -Frontend en Angular/Ionic.  -Consultas optimizadas para la Base de datos PostgreSQL. | Sprint 3 | Joaquín Madariaga, Fernando Cavada | Dificultades: Riesgo de tener consultas mal optimizadas que empeoren gravemente la experiencia de usuario  Facilitadores: Una buena definición de los filtros y el uso de Django ORM para facilitar la implementación de estos. |
| Construir programas y rutinas de variada complejidad para dar solución a requerimientos de la organización, acordes a tecnologías de mercado y utilizando buenas prácticas de codificación. | Validación de usuarios (Roles) | Validar a los usuarios por roles permite diferenciarlos de las distintas funciones que tendrán disponibles | -• Django REST Framework (SimpleJWT, permissions)  • Base de datos PostgreSQL (tablas de Users, Groups, Permissions) | Sprint 3 | Joaquín Madariaga, Lucas Cisternas | Dificultades: Garantizar que la jerarquía de permisos se aplique consistentemente en todos los endpoints y vistas. Riesgo de fugas de datos si un rol accede a información restringida de otro. Facilitadores: Los grupos y permisos integrados de Django ofrecen una base sólida para la gestión de accesos. |
| Desarrollar e implementar servicios web que permitan la integración y consulta de información, asegurando eficiencia y escalabilidad. | API Django para búsqueda | Diseñar y desarrollar una API en Django que permita realizar búsquedas de información dentro del sistema | Framework Django  - Django Rest Framework (DRF)  - Documentación de la API REST | Sprint 3 | Joaquin Madariaga | Dificultades: Posible complejidad en la definición de filtros avanzados y optimización de consultas.  Facilitadores: Uso de Django Rest Framework acelera la creación de endpoints y manejo de búsquedas. |
| Construir programas y rutinas de variada complejidad para dar solución a requerimientos de la organización, acordes a tecnologías de mercado y utilizando buenas prácticas de codificación. | Pruebas unitarias + integración básica | Crear y ejecutar un conjunto de pruebas automatizadas para verificar el correcto funcionamiento de unidades individuales de código (pruebas unitarias), como funciones o métodos específicos. Además, se realizarán pruebas de integración básicas para asegurar que estos módulos individuales interactúen correctamente entre sí (ej: que el servicio de autenticación funcione con los modelos de usuario y la base de datos) | • Framework pytest + pytest-django  • Biblioteca unittest de Django  • Factory Boy para datos de prueba | Sprint 3 | Lucas Cisternas | Dificultades: Puede consumir tiempo de desarrollo inicial significativo. Facilitadores: A largo plazo, las pruebas automatizadas ahorran tiempo, previenen regresiones y facilitan los despliegues confiables. Herramientas como Pytest y la TestSuite de Django son de gran ayuda. |
| Construir programas y rutinas de variada complejidad para dar solución a requerimientos de la organización, acordes a tecnologías de mercado y utilizando buenas prácticas de codificación. | Ampliación BD (citas y mensajes) | Diseñar, migrar e implementar las nuevas estructuras de datos en la base de datos para soportar las funcionalidades de agendamiento de citas y un sistema de mensajería. Esto implica crear las tablas para gestionar citas (fecha, hora, profesional, paciente, estado), así como las tablas para almacenar mensajes (remitente, destinatario, contenido, timestamp), asegurando las relaciones foreign key y las reglas de integridad referencial. | • PostgreSQL DBMS  • Django ORM (models, migrations) | Sprint 4 | Fernando Cavada, Joaquín Madariaga. | Dificultades: Mantener la integridad referencial y consistencia de los datos al modelar relaciones complejas (ej: estado de una cita, historial de mensajes). Facilitadores: El ORM de Django simplifica la creación de estos modelos y sus relaciones, automatizando constraints a nivel de base de datos. |
| Construir programas y rutinas de variada complejidad para dar solución a requerimientos de la organización, acordes a tecnologías de mercado y utilizando buenas prácticas de codificación | Funcionalidades: Agendamiento de citas + chat basico | El sistema permite a los usuarios agendar, modificar y cancelar citas con profesionales. Estos, a su vez, pueden gestionar su disponibilidad. Para las sesiones virtuales, el sistema se integra con una API de videollamadas. Además, un módulo de chat, implementado en Supabase con PostgreSQL y Realtime, facilita la comunicación directa entre usuarios y profesionales para resolver dudas o hacer seguimiento a las consultas.. | -Backend en Django  -Base de datos (tablas de agenda)  -Calendario en frontend (Angular/Ionic),  -Librerías de gestión de tiempo/fechas. | Sprint 4 | Joaquín Madariaga, Lucas Cisternas | Dificultades: se deben evitar conflictos de horarios y asegurar disponibilidad en tiempo real, el chat puede consumir más recursos del equipo del esperado.  Facilitadores: existen librerías de calendario y validación de fechas que pueden integrarse fácilmente. Ademas Supabase ofrece soporte nativo implementar un chat en tiempo real, con buena documentación y ejemplos de integración. |
| Asegurar la calidad del software mediante la validación del código, pruebas automatizadas y verificación de la integración entre componentes. | Pruebas unitarias e integración | Diseñar y ejecutar pruebas unitarias para validar la funcionalidad de los módulos individuales, además de pruebas de integración para verificar la correcta interacción entre componentes del sistema. | - Framework de pruebas (PyTest / unittest)  - Cobertura de código  - GitHub  - Repositorio en GitHub/GitLab | Sprint 4 | Lucas Cisternas | Dificultades: Posible falta de cobertura en escenarios críticos y errores en la integración entre módulos.  Facilitadores: Automatización de pruebas y pipeline CI/CD permiten detectar fallas rápidamente. |
| Construir programas y rutinas de variada complejidad para dar solución a requerimientos de la organización, acordes a tecnologías de mercado y utilizando buenas prácticas de codificación. | Historial de consultas | El historial de consultas es un módulo que almacena y muestra los detalles de las sesiones pasadas tanto para los usuarios como para los profesionales. Permite a los pacientes revisar sus consultas previas, mientras que los profesionales pueden registrar y acceder a notas privadas relacionadas con cada sesión. | -PostgreSQL (tablas para registrar consultas).  -Django ORM.  -frontend para visualización. | Sprint 4 | Fernando Cavada, Joaquín Madariaga | Dificultades: Riesgo de vulnerabilidades en el almacenamiento de datos sensibles.  Facilitadores: El control de accesos en Django permite limitar quién puede consultar o modificar la información. |
| Implementar soluciones sistémicas integrales para automatizar u optimizar procesos de negocio de acuerdo a las necesidades de la organización. | Exploración inicial de Videollamada | Se realizará una exploración inicial de videollamadas, evaluando la integración con la API de Zoom y las funcionalidades necesarias para permitir que los usuarios reciban asistencia de manera remota. | -Integración con Zoom  -Frontend para las vistas | Sprint 4 | Joaquin Madariaga,  Fernando Cavada. | Riesgo: Dependencia de un servicio externo, lo que implica estar sujetos a cambios en políticas, precios o limitaciones de la API.  Facilitadores: La API de Zoom cuenta con documentación extensa, ejemplos de integración y un entorno de pruebas gratuito para desarrolladores. |
| Construir programas y rutinas de variada complejidad para dar solución a requerimientos de la organización, acordes a tecnologías de mercado y utilizando buenas prácticas de codificación. | Ampliación BD (pagos, historial, soporte y calificaciones) | Diseñar y construir en la base de datos, nuevas entidades para almacenar los pagos, historiales de consulta, tickets de soporte y calificaciones de los profesionales. Además se deben integrar los nuevos modelos creados en la aplicación. | -Proyecto con el frontend y backend avanzados (vistas para el pago, historial y soporte en un estado básico) | Sprint 5 | Fernando Cavada, Joaquín Madariaga | Riesgo: Problemas de obtención de datos al usar servicios externos (el diseño de la base de datos no coincide con los datos entregados por las APIs). |
| *Implementar soluciones sistémicas integrales para automatizar u optimizar procesos de negocio de acuerdo a las necesidades de la organización.* | Funcionalidades: videollamadas completas y sistema de pago | Se finaliza la integración con la API de zoom y se empieza a trabajar en la API de pago para procesar pagos de manera segura, generando comprobantes y gestionando reembolsos. | -Infraestructura general del proyecto, incluyendo el backend en Django y la base de datos PostgreSQL.  -Pasarela de pagos Stripe  -API Key de zoom  -Certificados SSL | Sprint 5 | Joaquín Madariaga, Lucas Cisternas | Dificultades: La principal dificultad es cumplir con normas de seguridad financiera (PCI-DSS) y manejar errores en transacciones.  Facilitadores: Stripe ofrece SDKs y APIs bien documentadas. |
| Implementar soluciones sistémicas integrales para automatizar u optimizar procesos de negocio de acuerdo a las necesidades de la organización. | Modulo de administracion + soporte | El panel de control para administradores permite supervisar y gestionar las operaciones de la plataforma, como la validación de credenciales de profesionales, el manejo de registros de usuarios y la revisión de estadísticas. Adicionalmente, el módulo de soporte opera como un sistema de tickets para que los usuarios puedan reportar problemas y comunicarse con el equipo técnico, asegurando un seguimiento y solución eficientes. | -Infraestructura tecnológica general del proyecto (backend y frontend). | Sprint 5 | Joaquín Madariaga, Lucas Cisternas | Dificultades: Riesgo de dar seguimiento a tickets de ayuda duplicados y riesgo de dar seguimiento a los tickets de forma ineficiente  Facilitadores: Django Admin permite configurar un panel básico rápidamente y ampliarlo según las necesidades. Existen librerías y plantillas open source para sistemas de soporte. |
| Implementar soluciones sistémicas integrales para automatizar u optimizar procesos de negocio de acuerdo a las necesidades de la organización. | Exportación a móvil | Adaptar la aplicación desarrollada para su despliegue en dispositivos móviles, asegurando compatibilidad con Android e iOS. Se incluyen pruebas de compilación, empaquetado y ajustes de interfaz. | -Android Studio  -Infraestructura tecnológica general del proyecto (backend y frontend) | Sprint 5 | Lucas Cisternas | Riesgo: diferencias en versiones de SO. Se recomienda pruebas continuas en emuladores y dispositivos físicos. |
| Asegurar la calidad del software mediante la validación del código, pruebas automatizadas y verificación de la integración entre componentes. | Pruebas unitarias a los módulos móviles | Diseñar y ejecutar pruebas unitarias para cada módulo del sistema (frontend, backend ) asegurando la funcionalidad independiente de cada componente. | -Frameworks de testing: JUnit  -Plataforma con todos los módulos desarrollados. | Sprint 5 | Lucas Cisternas,  Joaquín Madariaga,  Fernando Cavada | Riesgo: cambios en requisitos pueden afectar casos de prueba. Se recomienda mantener pruebas automatizadas. |
| Implementar soluciones sistémicas integrales para automatizar u optimizar procesos de negocio de acuerdo a las necesidades de la organización. | Pruebas de integración final | Validar la solución completa en producción, asegurando que todos los módulos funcionen de manera estable. | -Usuarios finales.  -Plataforma con todas sus funciones integradas. | Sprint 5 | Lucas Cisternas,  Joaquín Madariaga,  Fernando Cavada | Importante incluir Marcha Blanca como validación con usuarios reales. |
| Gestionar proyectos informáticos, ofreciendo alternativas para la toma de decisiones de acuerdo a los requerimientos de la organización. | Cierre del proyecto | Realizar la evaluación final del proyecto, documentar los entregables, recopilar lecciones aprendidas y formalizar la aceptación del producto. | -Documentación final  -Acta de cierre  -Retrospectiva Scrum  -Repositorio Git | Sprint 5 | Joaquin Madariaga,  Fernando Cavada,  Lucas Cisternas | Importante incluir retrospectiva final y validación con stakeholders. |

| **8. Roadmap de semanas** |
| --- |

****

****

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)