



PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

Unidad Académica de Formación Técnica y Tecnológica – PUCE TEC

**DESARROLLO DE UN SISTEMA INFORMÁTICO CENTRALIZADO PARA LA
GESTIÓN DE PACIENTES EN EL CENTRO PSICOLÓGICO ATRÉVETE.**

**Proyecto de titulación previo a la obtención del título de: Tecnología Superior En
Desarrollo De Software**

Autor: David Israel Cañizares Rivera

Tutor: Carlos Miguel Cárdenas Riofrio

Quito, Ecuador

2025

Dedicatoria

Este logro académico en la PUCE lo dedico a mi padre y a mi madre, quienes han sido mi soporte inquebrantable, especialmente desde mi transición de la etapa colegial a la vida universitaria.

De igual forma, reconozco a cada persona que integró este proceso; la suma de vivencias, tanto las favorables como las adversas, definió el trayecto que hoy recorro. Es precisamente ese cúmulo de experiencias el que hoy motiva mi compromiso de mejora continua, capitalizando las lecciones del pasado para evitar la reincidencia en los errores de ayer.

Tabla de contenidos

Dedicatoria	2
Lista de figuras	4
Agradecimientos	6
Introducción	7
Capítulo I	9
Levantamiento de Requisitos y Diseño del Sistema	9
Capítulo II	16
Construcción del Sistema.....	16
Capítulo III	23
Pruebas y Estabilización.....	23
Conclusiones	28
Referencias bibliográficas	30

Lista de figuras

1. Diagrama Arquitectura
2. Diagrama entidad-relación de la base de datos
3. Ventana de Inicio de sesión
4. Ventana Restablecer Contraseña
5. Ventana Principal
6. Ventana Principal Notificaciones
7. Ventana Principal Rol
8. Ventana Calendario
9. Ventana SRI y OlimpushAPI
10. Olimpush Servicios
11. Crear facturas electrónicas
12. Historial de facturas electrónicas
13. Validar RUC
14. Validar Establecimiento
15. Pacientes
16. Agendamiento de citas

DECLARACIÓN y AUTORIZACIÓN

Yo, David Israel Cañizares Rivera con C.I. 1750115485 autor(a) del trabajo de titulación intitulado: "Desarrollo de un sistema informático centralizado para la gestión de pacientes en el "Centro psicológico Atrévete.", previa a la obtención del título de Tecnología Superior En Desarrollo De Software en la Unidad Académica de Formación Técnica y Tecnológica PUCE TEC:

1.- Declaro tener pleno conocimiento de la obligación que tiene la Pontificia Universidad Católica del Ecuador, de conformidad con el artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la Pontificia Universidad Católica del Ecuador a difundir a través de sitio web de la Biblioteca de la PUCE el referido trabajo de titulación, respetando las políticas de propiedad intelectual de Universidad.

Quito, 27 de enero 2025



David Israel Cañizares Rivera

C.I. 1750115485

Agradecimientos

Extiendo mi gratitud a la Pontificia Universidad Católica del Ecuador – PUCE TEC, unidad académica que no solo me facilitó las herramientas tecnológicas necesarias, sino que propició un entorno de excelencia para mi formación profesional en el área de software.

Hago extensivo este reconocimiento al cuerpo docente de la carrera, cuya exigencia metodológica y transferencia de conocimientos técnicos fueron determinantes para madurar las competencias aplicadas en este proyecto. Sus directrices fueron la base sobre la cual se desarrolló esta solución de software.

Finalmente, el agradecimiento más profundo es para mi padre Luis Cañizares, y mi madre Irma Rivera. Su respaldo y sustento incondicional durante estos años constituyeron el motor que me permitió persistir ante cada desafío académico y llegar a esta meta.

Introducción

La optimización de los flujos de trabajo en el sector de la salud mental representa un desafío crítico para garantizar una atención de calidad. En el caso particular del Centro Psicológico "Atrévete", se detectó que la administración de recursos mediante métodos manuales y herramientas dispersas estaba provocando inconsistencias en la agenda, duplicidad en los registros clínicos y un desgaste operativo significativo en la emisión de comprobantes fiscales. Estas carencias tecnológicas no solo ralentizan la operatividad diaria, sino que introducen riesgos innecesarios en el manejo de la información sensible de los pacientes.

Como respuesta a esta problemática, esta propuesta de titulación se centra en la ingeniería y despliegue de una plataforma web integral. La arquitectura de software seleccionada desacopla el frontend del backend para maximizar el rendimiento: se emplea Next.js para construir una experiencia de usuario fluida, mientras que la lógica de negocio reside en Django. Para el entorno de desarrollo y prototipado, se optó por SQLite como motor de base de datos. Un rasgo distintivo de esta solución es la implementación de un servidor proxy en el backend, diseñado específicamente para dialogar con la API de Olimpush. Este mecanismo permite simular el ciclo de facturación electrónica, abarcando desde la validación del RUC hasta la firma digital en un entorno de pruebas, superando así las barreras técnicas de una integración directa con el SRI.

En cuanto a la gestión del proyecto, la ejecución se rigió por los principios del marco de trabajo Scrum. Esta metodología ágil facilitó una construcción incremental del software, permitiendo validar funcionalidades clave con la psicóloga responsable en ciclos cortos y ajustar los requerimientos en tiempo real. El producto final no es solo un repositorio de datos, sino una herramienta viva que centraliza historias clínicas, automatiza el calendario de citas e integra servicios del SRI de manera transparente para el usuario final.

A lo largo de este documento se detalla el ciclo de vida del software, partiendo desde el levantamiento de requisitos y el diseño del sistema, continuando con la construcción técnica, hasta llegar a las fases de pruebas y estabilización. Finalmente, se presentan las conclusiones derivadas de la puesta en marcha del prototipo, evidenciando cómo la solución tecnológica logra mitigar la carga administrativa y proyectando futuras líneas de escalabilidad para el sistema.

Capítulo I

Levantamiento de Requisitos y Diseño del Sistema

1.1 Análisis de Requisitos

La especificación de los requerimientos técnicos y operativos surge tras examinar las deficiencias presentes en la gestión administrativa del consultorio. El propósito central de esta intervención tecnológica es la digitalización de los flujos de trabajo manuales, con el fin de maximizar la eficiencia operativa y asegurar la integridad de la información clínica de los pacientes.

1.1.1 Requisitos Funcionales

Estas funcionalidades responden a las necesidades operativas inmediatas identificadas por la psicóloga y el personal administrativo del Centro “Atrévete”. Conforman el alcance funcional del sistema en su versión inicial (MVP), desarrollada bajo parámetros académicos y de validación técnica:

RF-01: Agendamiento y Citas Médicas

Despliegue de un calendario interactivo destinado a la administración de turnos y horarios.

Habilitación del registro de citas vinculadas a un paciente y a un servicio específico, incorporando una validación automática de disponibilidad para prevenir conflictos de agenda.

Facultad para modificar o anular citas existentes, con actualización inmediata del estado del turno en tiempo real.

RF-02: Integración SRI

El sistema opera en calidad de intermediario seguro (proxy) para la transmisión de datos de facturación, la solicitud de firma electrónica y la validación del RUC del contribuyente.

Emisión de comprobantes electrónicos a través de la integración técnica con la API del proveedor Olimpush.

RF-03: Control de Acceso y Seguridad

Implementación de un mecanismo de autenticación basado en el estándar JWT, asegurando las credenciales mediante algoritmos de hashing robustos y el uso de tokens firmados para la gestión de sesiones.

Segregación de privilegios mediante un esquema de roles: Administrador (control total) y Psicóloga/Asistente (acceso restringido a agenda y facturación básica).

Protocolo seguro para la recuperación de acceso en caso de pérdida de credenciales.

RF-04: Reportería y Métricas

Proyección gráfica del historial de citas, categorizadas por su estado: programadas, confirmadas, completadas y canceladas, facilitando el análisis estadístico de la atención.

1.1.2 Requisitos No Funcionales

Estos atributos de calidad, aunque no fueron solicitados explícitamente como funciones por el cliente, constituyen imperativos técnicos definidos por el desarrollador para garantizar la viabilidad, seguridad y robustez del software:

RNF-01: Privacidad y Cumplimiento (LOPDP)

Implementación de criptografía para datos sensibles en la base de datos y ofuscación de contraseñas, en estricto apego a la Ley de Protección de Datos Personales.

Aplicación de principios de confidencialidad, minimización de datos y acceso restringido conforme a la normativa vigente.

RNF-02: Interoperabilidad vía Proxy

El *backend* centraliza las integraciones externas, desacoplando al cliente web de dependencias directas y resolviendo restricciones de seguridad del navegador como CORS (Cross-Origin Resource Sharing).

RNF-03: Rendimiento y Escalabilidad

La arquitectura está dimensionada para soportar la gestión simultánea del calendario y la emisión de documentos con tiempos de respuesta optimizados, sin latencia perceptible para el usuario final.

Diseño preparado para una eventual migración del motor de base de datos (de SQLite a PostgreSQL) si el volumen transaccional lo demandase a futuro.

RNF-04: Ergonomía de Interfaz

Interfaz con diseño *responsive* y minimalista, optimizada para reducir la carga cognitiva, permitiendo que usuarios sin formación técnica avanzada operen el sistema tras una capacitación breve.

RNF-05: Disponibilidad

Acceso continuo a la plataforma web (24/7), facultando la gestión administrativa fuera del horario de consultorio si fuese necesario.

RNF-06: Auditoría

Todas las operaciones que impliquen creación o modificación de registros en la base de datos son auditadas mediante campos de control que se actualizan automáticamente al ejecutarse dichas transacciones.

1.1.3 Usuarios

Se han definido perfiles de usuario jerárquicos para preservar el orden y la seguridad en la manipulación de los datos:

- **Administrador (Soporte Técnico):**

Posee privilegios absolutos sobre la plataforma. Está facultado para gestionar usuarios del sistema, configurar el catálogo de servicios y acceder a la auditoría completa de las citas.

- **Psicóloga / Asistente Administrativo:**

Su alcance se circunscribe a la operatividad diaria: agendar citas, registrar nuevos pacientes y utilizar los servicios de facturación integrados. No posee permisos para alterar configuraciones globales ni para eliminar historiales clínicos de forma permanente.

1.1.4 Limitaciones

Entorno de Pruebas: El módulo de facturación electrónica se ejecuta sobre el ambiente de pruebas del SRI y Olimpush. Los documentos generados poseen validez técnica académica, pero no tributaria; la emisión en producción requiere un sustento legal y homologación oficial para evitar sanciones por parte del ente regulador.

Motor de Base de Datos: Se utiliza SQLite dada su agilidad para el prototipado y el entorno académico del proyecto; para un despliegue masivo en producción real se recomendaría transicionar a un motor más robusto.

Dependencia Externa: La disponibilidad del servicio de facturación está condicionada al tiempo de actividad de la API de Olimpush; una caída en el servicio externo detiene la facturación, aunque el módulo de agendamiento permanece operativo.

1.2 Diseño del Sistema

Las decisiones de diseño detalladas a continuación responden directamente a la necesidad de mitigar errores operativos y reducir los tiempos administrativos detectados durante el análisis de requisitos.

1.2.1 Interfaz de Usuario

Para la construcción visual se establecieron criterios orientados a la fluidez y la claridad:

- **Minimalismo Funcional:**

Se prescinde de elementos decorativos superfluos. Se emplea una paleta de colores sobria para focalizar la atención en la información del paciente y el calendario. Cabe destacar que, en esta versión, el paciente no interactúa directamente con el sistema; toda la gestión es interna.

- **Feedback Interactivo:**

Implementación de notificaciones visuales inmediatas para confirmar acciones críticas, otorgando certeza al usuario sobre operaciones como "Cita Agendada" o "Error de Validación".

- **Componentes Reutilizables:**

Gracias a la arquitectura de Next.js, la interfaz se construye mediante componentes modulares, asegurando la consistencia visual en todas las vistas (botones, formularios y tablas estandarizadas).

1.2.2 Arquitectura del Sistema

Se optó por una arquitectura desacoplada (*Headless*) que separa la lógica de negocio de la capa de presentación, favoreciendo la escalabilidad y el mantenimiento:

Frontend (Cliente): Desarrollado en Next.js. Se encarga de la renderización de la interfaz y la experiencia interactiva. Consume los datos a través de *endpoints* REST, garantizando velocidad de carga y reactividad.

Backend (API & Proxy): Construido sobre Django. Este núcleo centraliza la lógica de negocio, gestiona la seguridad y autenticación, y actúa como Servidor *Proxy* para comunicar el *frontend* con la API de Olimpush.

Base de Datos: Se emplea SQLite como sistema de almacenamiento relacional para la persistencia de usuarios, citas, pacientes y registros de transacciones, siendo ideal para la portabilidad del proyecto académico.

1.2.3 Características Clave

- **Agenda Inteligente:**

Mecanismo de prevención automática de cruce de horarios, impidiendo agendar dos pacientes con el mismo profesional en el mismo bloque temporal.

- **Historia Clínica Digital:**

Centralización de la información del paciente en un repositorio digital seguro, eliminando la dependencia de archivos físicos vulnerables al deterioro o pérdida.

Capítulo II

Construcción del Sistema

En este capítulo se detalla el proceso técnico seguido para la construcción del sistema del Centro Psicológico "Atrévete", describiendo desde la selección de herramientas hasta la estructuración lógica y de datos del proyecto.

2.1 Herramientas y tecnologías empleadas

Las decisiones técnicas adoptadas responden directamente a los requisitos funcionales y no funcionales definidos en el Capítulo I. Para el desarrollo de esta plataforma, se seleccionaron tecnologías modernas y robustas que garantizan escalabilidad y seguridad, aprovechando la idoneidad técnica de estos entornos para el caso de estudio:

- **Frontend:**

Next.js, elegido por su capacidad de renderizado híbrido y optimización de componentes, lo que mejora la experiencia de usuario.

- **Backend:**

Django (Python) utilizando Django REST Framework, debido a su seguridad integrada y rapidez para construir APIs.

- **Base de datos:**

SQLite, por su portabilidad y facilidad de implementación en entornos de desarrollo y prototipado académico.

- **ORM:**

Django ORM, incluido nativamente en el framework, lo que permitió interactuar con la base de datos usando objetos de Python en lugar de SQL puro.

- **Control de versiones:**

Git, gestionando repositorios remotos para el control de versiones y el respaldo del código fuente.

- **Gestión de dependencias:**

Pip para las librerías de Python y npm para los paquetes de JavaScript.

2.2 Organización del proyecto

Se estructuraron dos repositorios remotos independientes: uno para el frontend (cliente) y otro para el backend (servidor). A nivel de código, se organizaron dos directorios principales para desacoplar las responsabilidades y facilitar el mantenimiento:

- **Frontend (Next.js):**

Dentro de la carpeta `src/`, el código se distribuyó en `pages/` para las rutas, `components/` para elementos reutilizables (botones, formularios) y `api/` para las funciones que consumen la API, siguiendo la estructura estándar de Next.js utilizada en el proyecto.

- **Backend (Django):**

Aquí se alojan las aplicaciones modulares, donde cada carpeta representa una funcionalidad (usuarios, citas, pacientes). También incluye el archivo `settings.py` para configuraciones globales y `urls.py` para el enrutamiento.

Esta división modular asegura que los cambios en la interfaz no afecten la lógica del servidor y viceversa.

2.3 Interfaz del sistema

La capa visual se desarrolló con Next.js, enfocándose en la usabilidad para el personal médico y administrativo. Se implementó una arquitectura de componentes reutilizables para acelerar el desarrollo. Los módulos principales integrados fueron:

- **Inicio:**

Vista principal para usuarios autenticados, acciones rápidas y resumen de citas médicas en una gráfica.

- **Calendario y Citas:**

Un calendario interactivo que permite visualizar turnos disponibles y cambiar estados (confirmada/cancelada).

- **SRI:**

Interfaz dedicada al uso de distintos servicios del SRI, facturación electrónica y validación de RUC.

- **Autenticación:**

Pantallas de inicio de sesión seguro y recuperación de credenciales.

Se incorporaron validaciones en tiempo real en los formularios y notificaciones visuales para confirmar acciones exitosas o alertar errores al usuario. La interfaz fue diseñada siguiendo

criterios de simplicidad y retroalimentación inmediata, definidos previamente en el diseño del sistema.

2.4 Lógica del sistema

El backend en Django se encargó de procesar las reglas de negocio, la seguridad y la integración con servicios externos. Con base en la documentación de la API generada, se definieron las siguientes rutas principales:

`/auth/jwt/create/`: Endpoint para la generación de tokens de acceso seguro (Login).

`/api/users/`: Gestión de usuarios y roles (Administrador, Psicóloga).

`/api/patients/`: CRUD completo para la información de los pacientes.

`/api/appointments/`: Lógica de agendamiento, validación de horarios y prevención de cruces de citas.

`/api/doctors/`: Administración de los profesionales de la salud vinculados al centro.

`/api/olimpush-sri/`: Endpoint que actúa como puente (Proxy) para comunicar los datos de facturación con la API externa de Olimpish.

Toda esta lógica se estructuró utilizando *ViewSets* y *Serializers*, centralizando las consultas y validaciones antes de impactar en la base de datos.

2.5 Base de datos

Se optó por SQLite para esta etapa del proyecto debido a su eficiencia en el manejo de archivos locales y configuración cero. El modelado de datos se realizó a través de las clases

(Modelos) de Django, generando las tablas automáticamente. Las entidades principales del sistema son:

- **Users:**

Usuarios del sistema con roles personalizados.

- **Patients:**

Información demográfica e historial.

- **Psicóloga:**

Datos de los especialistas.

- **Appointments:**

Tabla transaccional que vincula pacientes, psicóloga y horarios.

- **BusinessInfo:**

- Registro del RUC con el cual está vinculado el negocio.

Se definieron relaciones de integridad referencial (Claves Foráneas) entre citas y pacientes, así como restricciones de unicidad para evitar duplicidad en cédulas y correos electrónicos.

2.6 Diagramas del sistema

El sistema opera bajo una arquitectura cliente-servidor desacoplada. El frontend (Next.js) envía solicitudes HTTP/JSON al backend (Django), el cual procesa la información, interactúa con la base de datos (SQLite) y, en el caso de la facturación, se comunica con la API externa de Olimpush.

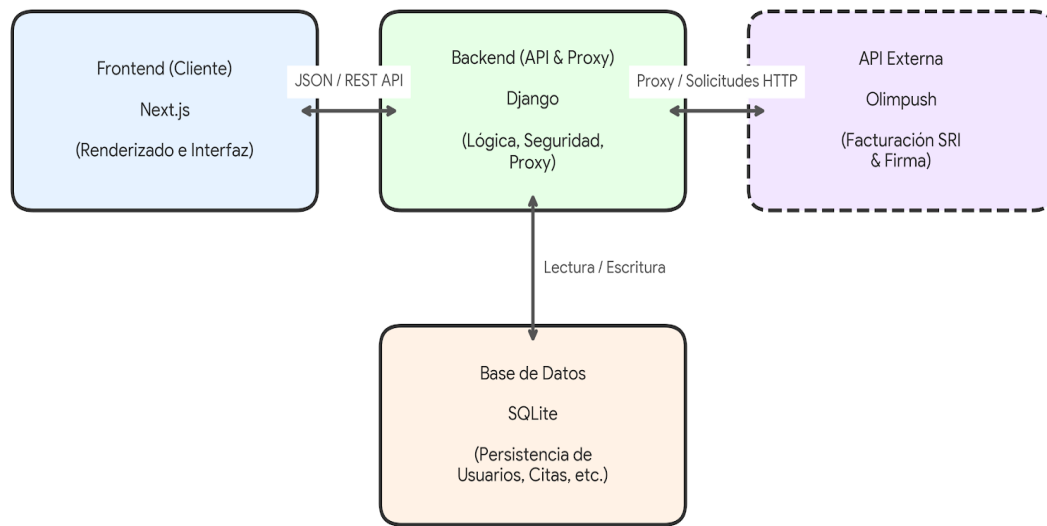


Fig. 1 - Diagrama del Sistema

- **Esquema de la interfaz principal de usuario**

La navegación se estructura mediante una barra lateral fija que da acceso a los módulos de Agenda, Pacientes y SRI. El área central de trabajo es dinámica, mostrando la información contextual según la opción seleccionada.

- **Flujo de autenticación y gestión de roles**

El acceso inicia con el envío de credenciales al servidor. Si son válidas, el sistema retorna un Token JWT que el frontend almacena para autorizar las peticiones subsiguientes, restringiendo el acceso a ciertas vistas según si el usuario es Administrador o Psicólogo.

- **Modelo entidad-relación de la base de datos**

La base de datos organiza la información en tablas normalizadas. La tabla **Appointments** actúa como eje central, relacionándose con **Patients** y **Doctors**, mientras que la tabla **Users** gestiona el acceso.

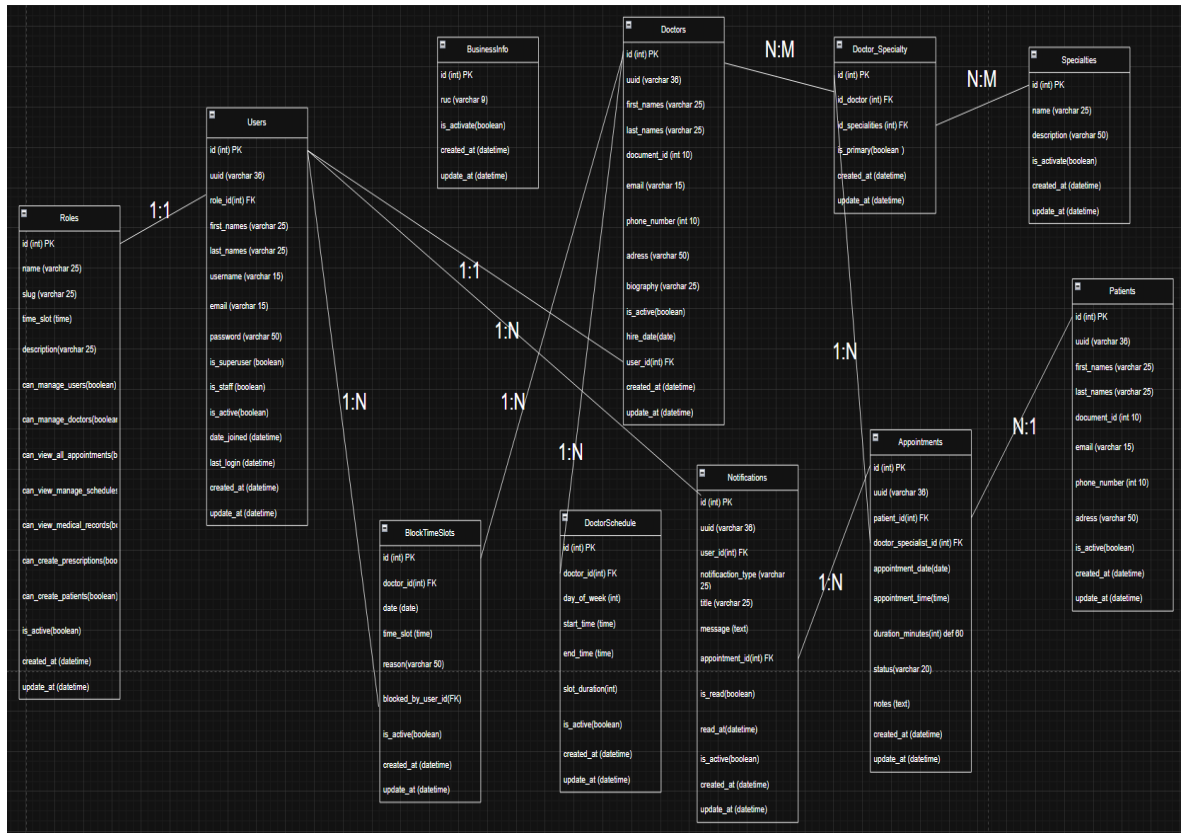


Fig. 2 - Diagrama entidad-relación de la base de datos

Capítulo III

Pruebas y Estabilización

En este capítulo se documenta el proceso de validación técnica ejecutado para certificar que el sistema cumple con los estándares de calidad antes de su entrega final. El objetivo primordial fue asegurar la integridad de los datos médicos y la correcta implementación de los servicios de facturación electrónica en el entorno de pruebas.

3.1 Herramientas de prueba

Para la ejecución y monitoreo de las pruebas, se utilizaron herramientas específicas que permitieron auditar tanto el comportamiento del servidor como la respuesta de la interfaz:

- **Postman:**

Empleado para la validación aislada de los endpoints del API REST, verificando los códigos de respuesta HTTP (200, 400, 401, 500) y la estructura de los datos JSON retornados por el backend.

- **Django Test Framework:**

Utilizado para la ejecución de pruebas unitarias automatizadas sobre los modelos y serializadores del sistema.

- **Chrome DevTools:**

Herramienta esencial para la depuración del frontend en Next.js, permitiendo inspeccionar la consola de errores, el almacenamiento local (tokens) y el tráfico de red.

- **SQLite Browser:**

Usado para verificar la integridad física de los datos almacenados y confirmar que las relaciones entre tablas se mantuvieran consistentes tras las operaciones de escritura.

3.2 Alcance de las pruebas

Se delimitó un perímetro de pruebas basado en los requerimientos funcionales críticos para la operación del consultorio.

- **Inclusiones:**

1. El flujo completo de autenticación y autorización (Login/Logout y protección de rutas por roles).
2. El ciclo de vida de una cita médica:
3. Creación, validación de cruce de horarios, modificación y cancelación.
4. La gestión de expedientes de pacientes (CRUD completo).
5. La comunicación entre el Proxy del sistema y la API de Olimpush en el entorno Sandbox (emisión de comprobantes de prueba).

- **Exclusiones:**

1. Pruebas de carga masiva: no se realizaron pruebas de estrés con miles de usuarios concurrentes, dado que el alcance es para un consultorio privado con tráfico moderado.

2. Validez tributaria real: No se emitieron facturas en el ambiente de producción del SRI, limitándose estrictamente al ambiente de pruebas académico.

17. Pagos en línea: La pasarela de pagos real no formó parte del alcance de pruebas, ya que el sistema gestiona la facturación, pero no el cobro digital bancario en esta versión.

3.3 Tipos de pruebas aplicadas

Durante el ciclo de construcción, se aplicó una estrategia de verificación escalonada, adaptando el tipo de prueba a cada componente del sistema:

1. Funcionales: Se verificó que los flujos principales, como el agendamiento de una cita o la creación de un historial clínico, cumplieran estrictamente con los requisitos funcionales levantados en el Capítulo I.
2. De interfaz: Se evaluó la respuesta de los componentes visuales en Next.js, asegurando que los formularios, botones y el calendario interactivo se renderizaran correctamente en distintos tamaños de pantalla (diseño responsive).
3. De integración: Se confirmó la interoperabilidad entre el cliente y el servidor, poniendo especial énfasis en la comunicación entre el Proxy interno y la API externa de Olimpush, validando el intercambio de archivos XML y respuestas de autorización.
4. De aceptación (Usuario): Se realizaron simulaciones de uso con los roles definidos (Administrador y Psicóloga) para validar que el sistema fuera comprensible, navegable y útil para las tareas diarias del personal.

3.4 Casos de prueba destacados

A continuación, se describen los escenarios críticos que fueron sometidos a validación para garantizar la robustez del aplicativo ante errores comunes:

1. Seguridad en el acceso: Se intentó vulnerar el inicio de sesión utilizando credenciales falsas y tokens manipulados o caducados. El sistema respondió denegando el acceso y redirigiendo al usuario a la vista de identificación, cumpliendo con los protocolos de seguridad.
2. Prevención de duplicidad: Al intentar registrar un paciente con un número de cédula ya existente en la base de datos, el sistema bloqueó la transacción y desplegó una alerta informativa, evitando la redundancia de expedientes.
3. Conflictos de agenda: Se simuló el agendamiento de dos citas distintas para el mismo profesional en el mismo bloque horario. El backend detectó la colisión temporal e impidió el guardado del registro, garantizando la integridad de la agenda.

3.5 Resultados y ajustes finales

El balance general de la fase de pruebas fue positivo, lográndose la estabilización de las funciones críticas del negocio. No obstante, durante las iteraciones iniciales se detectaron incidencias que requirieron corrección:

Errores de CORS: Se presentaron problemas de conexión iniciales entre el frontend y la API externa, situación que se solucionó mediante la correcta configuración de las cabeceras en el servicio Proxy del backend.

Validaciones de formulario: Se reforzaron los mensajes de error en los campos obligatorios (como el formato de correo electrónico) para ofrecer una mejor guía al usuario final.

Conclusiones

La implementación del sistema web centralizado logró mitigar significativamente los cuellos de botella operativos identificados en el Centro Psicológico "Atrévete". Al sustituir los registros manuales por una base de datos relacional, se eliminó la redundancia en los expedientes de los pacientes y se optimizó el tiempo dedicado al agendamiento, permitiendo al personal enfocar sus recursos en la atención clínica más que en la gestión administrativa.

Desde una perspectiva técnica, la arquitectura desacoplada (Frontend en Next.js y Backend en Django) demostró ser una estrategia eficiente. El desarrollo de un servidor Proxy intermedio solucionó exitosamente las barreras de comunicación (CORS y seguridad) con la API de Olimpush, validando la factibilidad de emitir comprobantes electrónicos sin exponer credenciales sensibles en el navegador del cliente.

El uso de la metodología Scrum facilitó una adaptación continua a los requerimientos reales del consultorio. Las entregas incrementales permitieron ajustar el módulo de facturación y el calendario interactivo basándose en el feedback directo de la psicóloga, resultando en una interfaz intuitiva que reduce la curva de aprendizaje para usuarios no técnicos.

Si bien el sistema opera actualmente en un entorno de pruebas (sandbox) para efectos académicos, la estructura del software cumple con los lineamientos técnicos del SRI y las normativas de la Ley Orgánica de Protección de Datos Personales (LOPDP), dejando una base sólida para una eventual homologación y puesta en producción legal.

Recomendaciones

Para el despliegue en un entorno productivo real, se recomienda migrar el motor de base de datos de SQLite a un sistema más robusto como PostgreSQL. Esto garantizará una mejor concurrencia y estabilidad de los datos cuando el volumen de pacientes y transacciones históricas aumente considerablemente.

Es aconsejable implementar un sistema de notificaciones automatizadas vía WhatsApp o SMS, integrando proveedores externos (como Twilio o la API de Meta) al backend en Django. Esto reduciría el índice de ausentismo de los pacientes al enviar recordatorios directos a sus dispositivos móviles 24 horas antes de la cita.

Se sugiere contenerizar la aplicación utilizando Docker para simplificar el proceso de despliegue en la nube (AWS o DigitalOcean). Esto asegurará que el entorno de ejecución sea idéntico tanto en desarrollo como en producción, minimizando errores de configuración.

Antes de pasar a la emisión real de facturas, es imperativo realizar el proceso de homologación formal con el SRI y adquirir una firma electrónica válida en archivo (p12) bajo el RUC del consultorio, sustituyendo las credenciales de prueba utilizadas en este prototipo.

Referencias bibliográficas

Django Software Foundation. (2024). Django documentation: Security and authentication. <https://docs.djangoproject.com/>

Next.js Team. (2024). Next.js documentation: Data fetching and proxying. Vercel. <https://nextjs.org/docs>

Olimpush. (2025). Introducción a OlimPush facturación API. <https://developers.olimpush.com/>

Schwaber, K., & Sutherland, J. (2020). La guía de Scrum: Las reglas del juego. Scrum.org. <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf>

Servicio de Rentas Internas. (2025). Ficha técnica de comprobantes electrónicos: Esquema offline. <https://www.sri.gob.ec/facturacion-electronica>

Anexos

Manual de Usuarios Psicólogo/Administrativos

1. Inicio de sesión y recuperación de contraseña

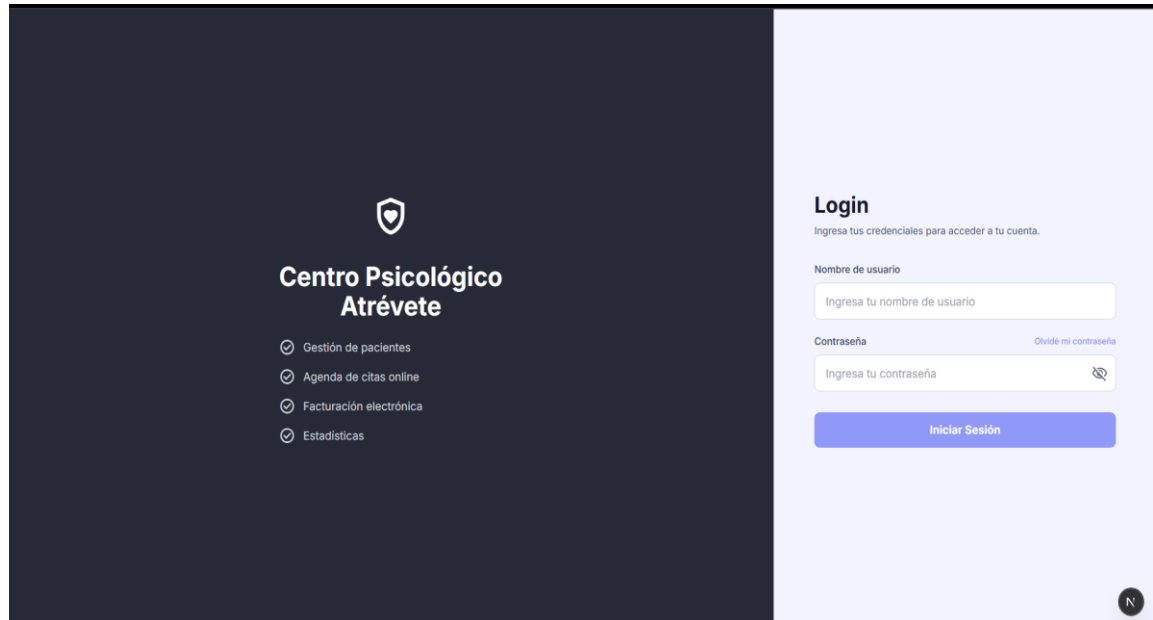


Fig 3 – Ventana de Inicio de sesión

Esta es la ventana principal donde el usuario administrativo o rol de psicólogo se le presentara para poder acceder al sistema. Deberás colocar las credenciales que fueron compartidas por el administrador del sitio. En caso de pérdida de contraseña podrías ingresar a esta vista donde podrás recuperar tu contraseña colocando el correo electrónico con el que ingresas al sistema. Una vez hecho esto recibirás un correo electrónico donde podrás restablecer la contraseña.

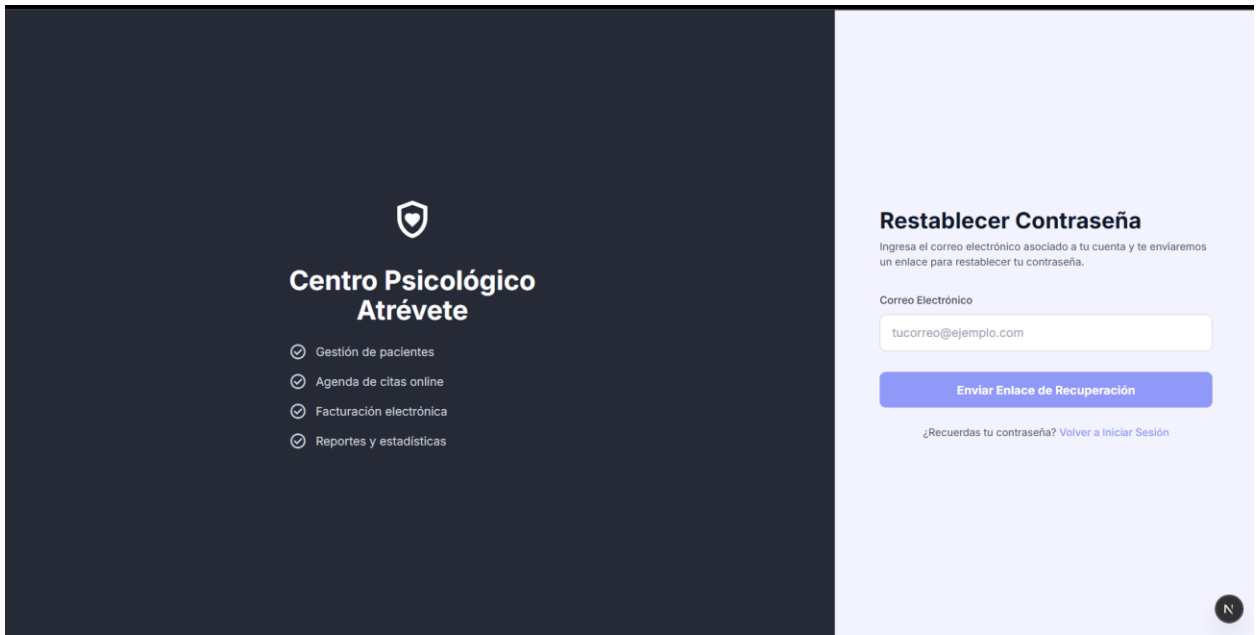


Fig 4 – Ventana Restablecer Contraseña

2. Ventana principal del sistema

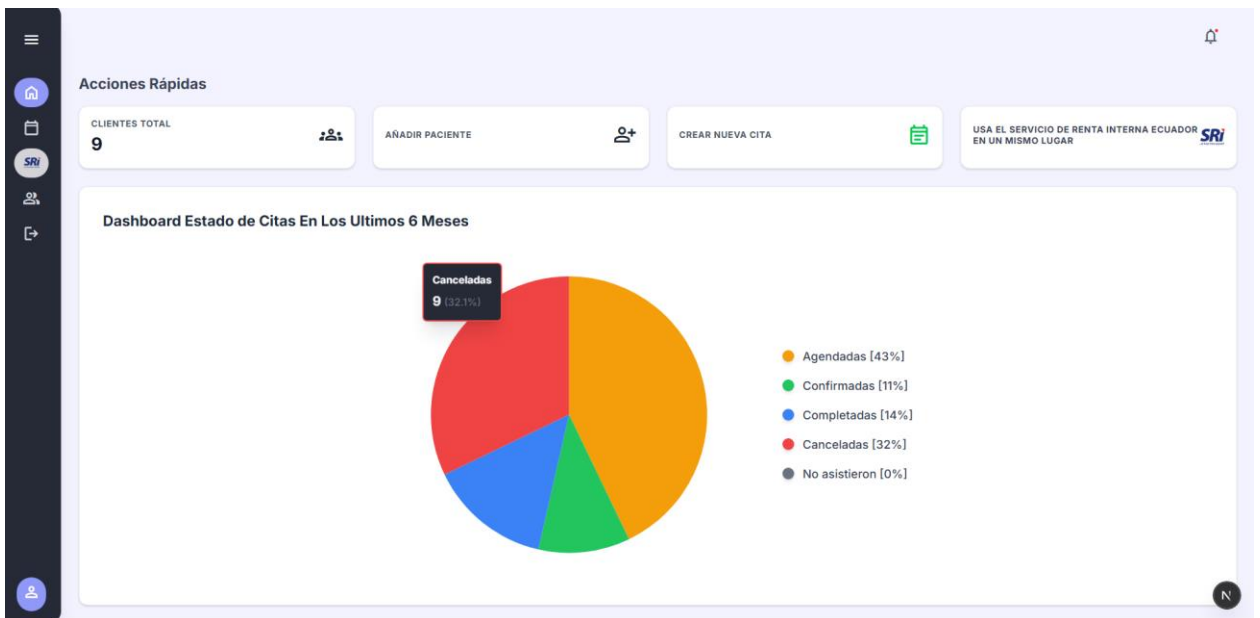


Fig 5 – Ventana Principal

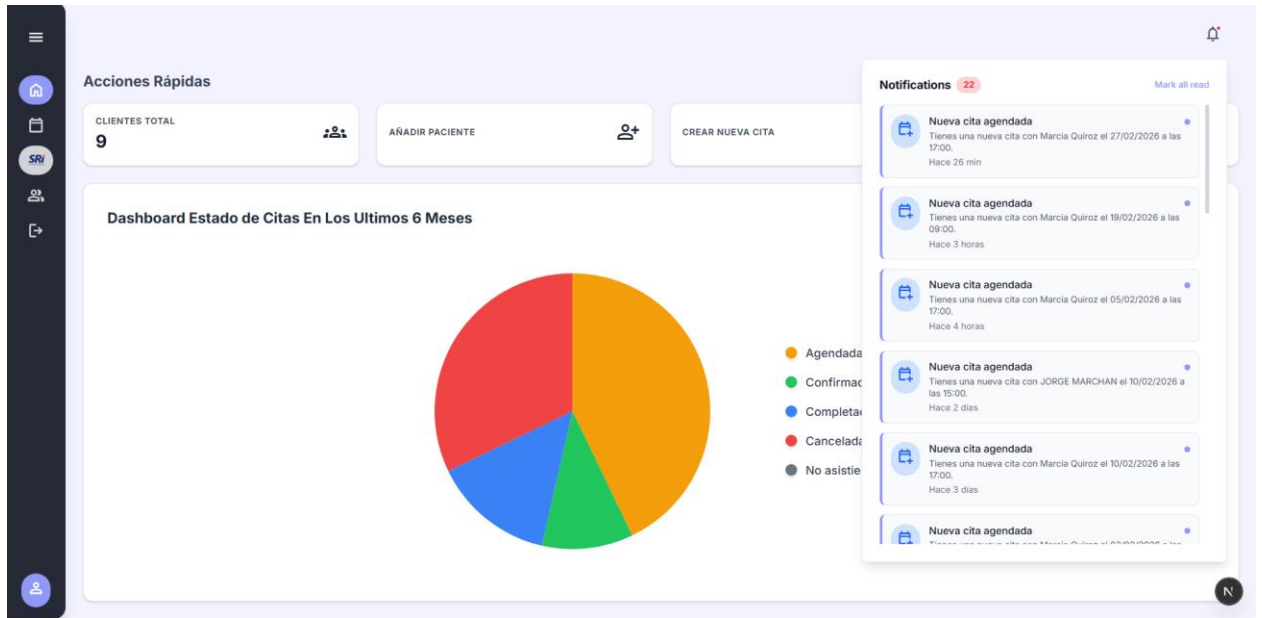


Fig 6 – Ventana Principal Notificaciones

En esta ventana se visualizarán acciones rápidas para el uso del sistema, además de un resumen sobre el estado en el que se encuentran las citas agendadas dentro de un periodo de tiempo establecido por defecto. Al dar click en cualquiera de las opciones rápidas serás redirigido a otra vista del sistema

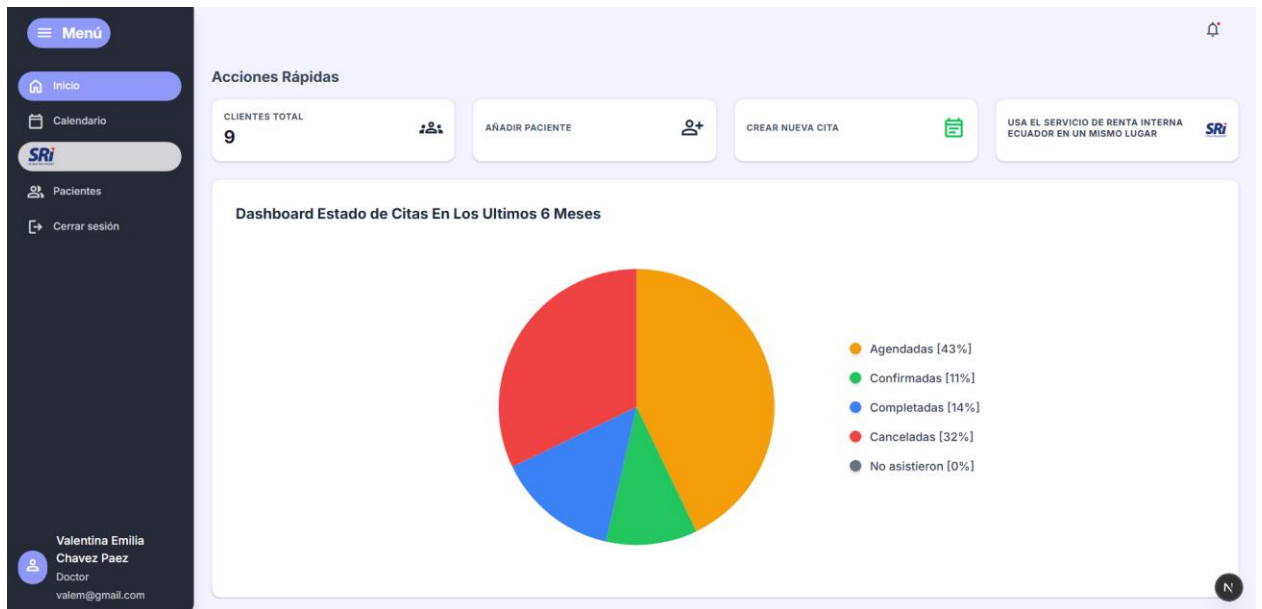


Fig 7 – Ventana Principal Rol

Al expandir el menú de la izquierda podrías visualizar tanto tu nombre como rol propio en el sistema.

3. Calendario y agendamientos

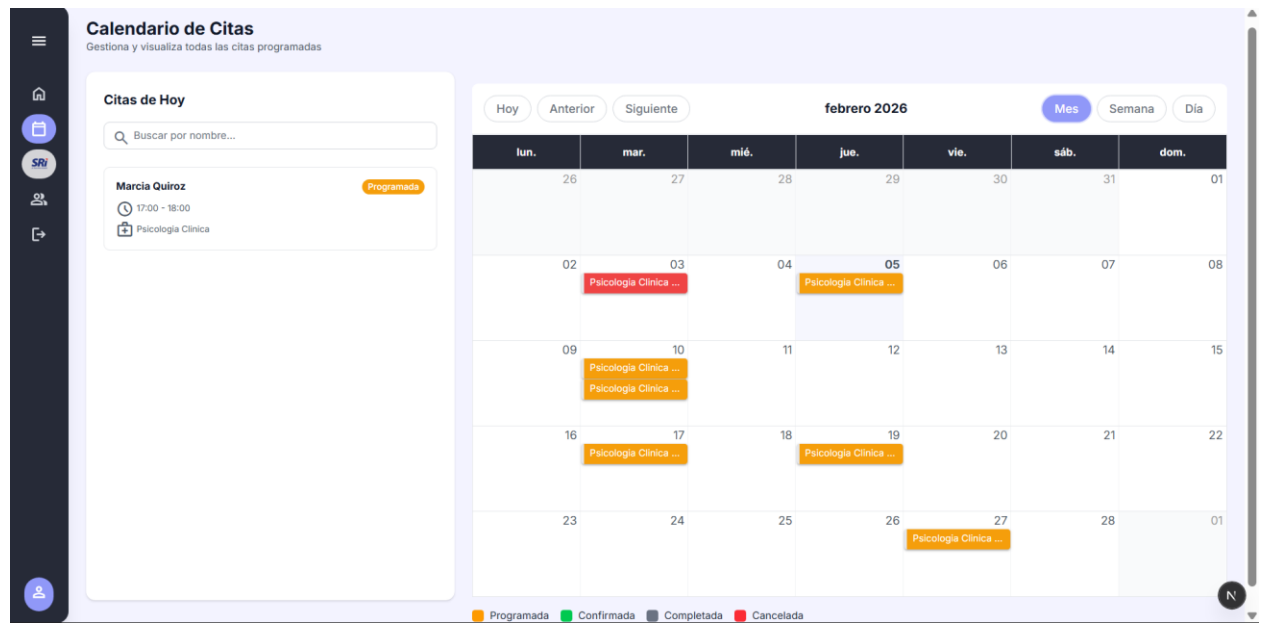


Fig 8 – Ventana Calendario

En esta sección del sistema encontraras un calendario que te ayudara a organizar, recordar y simplificar el tiempo de vida de una cita agendada en el sistema. Podrás organizar las citas dentro de las vistas Mes, Semana y Dia con la posibilidad de avanzar y retroceder dentro de un periodo determinado.

Una cita al ser agendada se encontrará en el estado Programada, esta al dar click en cualquiera de estas citas del calendario se desplegará más información como número de teléfono, notas agregadas. Además, cuando seleccionas una cita podrás cambiar de estado entre

Confirmada, Completada y Cancelada. El día y hora de la cita podrá ser cambiado solo si se encuentra disponible algún horario de la Psicóloga.

4. Paneles del SRI y OlimpishAPI

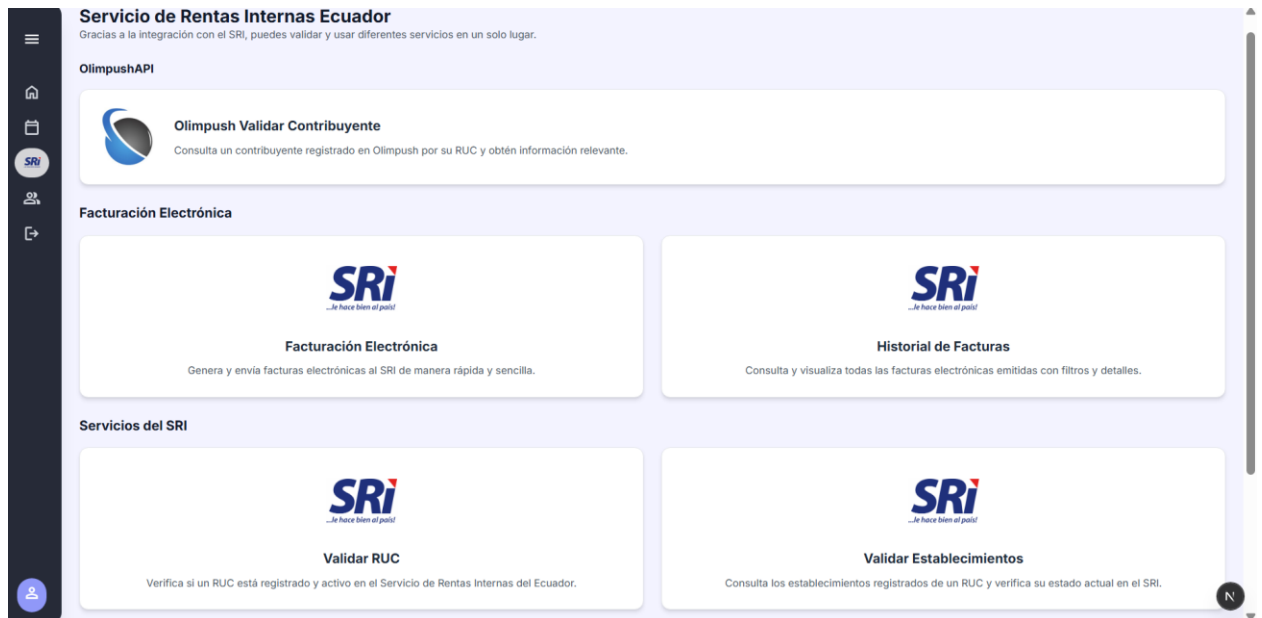


Fig 9 – Ventana SRI y OlimpishAPI

En esta sección encontraras los servicios integrados del SRI, también podras obtener información rápida sobre el proveedor de la API de facturación.

a. Olimpish Servicios

En esta encontraras detalles de la suscripción en la API de facturación Olimpish también podrás validar que el RUC del negocio concuerda con el del centro psicológico en la API de facturación, adicionalmente registrar tu firma electrónica y eliminarla definitivamente.

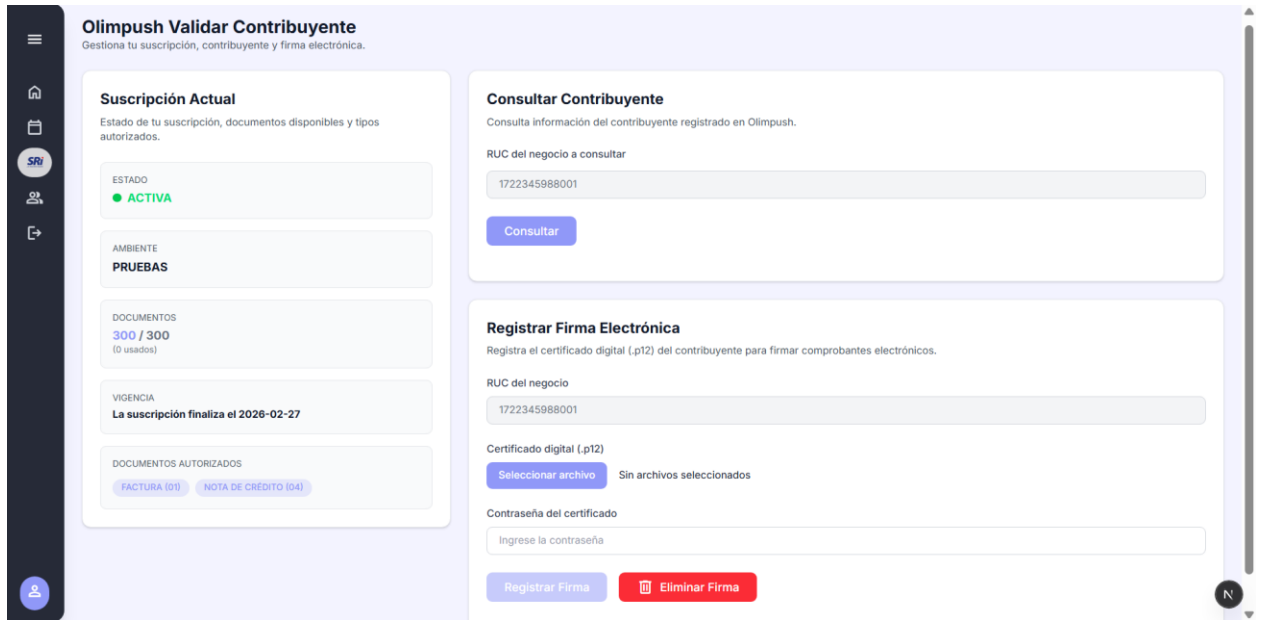


Fig 10 – Olimpush Servicios

b. Factura Electrónica

En esta vista podrás empezar a emitir facturas electrónicas a tus pacientes

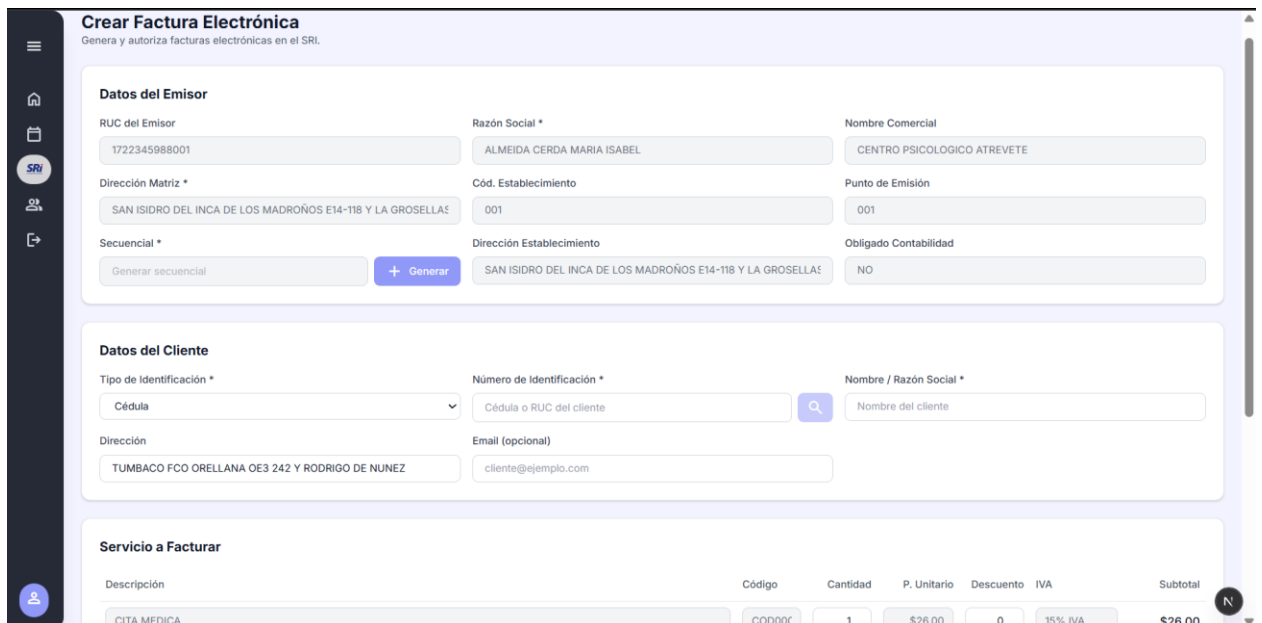


Fig 11 – Crear facturas electrónicas

c. Historial de Facturas

Podrás consultar el historial de facturas emitidas por la API de facturación Olimpush autorizadas y no autorizadas por el SRI

The screenshot displays the 'Historial de Facturas' (Invoice History) interface. At the top, there is a search bar for 'Identificación del Cliente' (Client Identification) and a dropdown for 'Estado de Autorización' (Authorization Status). Below this is a table with columns: 'Clave de Acceso' (Access Key), 'Secuencial' (Sequential), 'Cliente' (Client), 'Total', and 'Estado' (Status). The table lists five invoices, all with a status of 'PENDIENTE' (Pending). To the right, a 'Detalle de Factura' (Invoice Detail) panel shows the specific details for the first invoice, including the sequential number, date of issue, establishment, and client information.

Clave de Acceso	Secuencial	Cliente	Total	Estado
03022026011722345988...	00000006	Marcia Quiroz 1715975932	\$29.90	PENDIENTE
03022026011722345988...	00000005	Marcia Quiroz 1715975932	\$29.90	PENDIENTE
03022026011722345988...	00000004	Marcia Quiroz 1715975932	\$30.00	PENDIENTE
03022026011722345988...	00000003	Marcia Quiroz 1715975932	\$30.00	PENDIENTE
03022026011722345988...	00000001	Marcia Quiroz 1715975932	\$30.00	PENDIENTE

Detalle de Factura

030220260117223459880011001001000000003316122311

Secuencial: 00000006 | Fecha Emisión: 03/02/2026
Establecimiento: 001-001 | Ambiente: Pruebas

Cliente
Nombre / Razón Social: Marcia Quiroz
Identificación: 1715975932 | Email: davidsp1en1@gmail.com
Dirección: TUMBACO FCO ORELLANA OE3 242 Y RODRIGO DE NUNEZ

Productos/Servicios

CITA MEDICA	\$29.90
1 x \$26.00 Desc: \$0.00 IVA: \$3.90	

Totales
Subtotal: \$26.00

Fig 12 – Historial de facturas electrónicas

d. Validar RUC

En esta sección podrás validar si tu RUC se encuentra registrado en el SRI, también puedes validarlo con cualquier otro RUC.

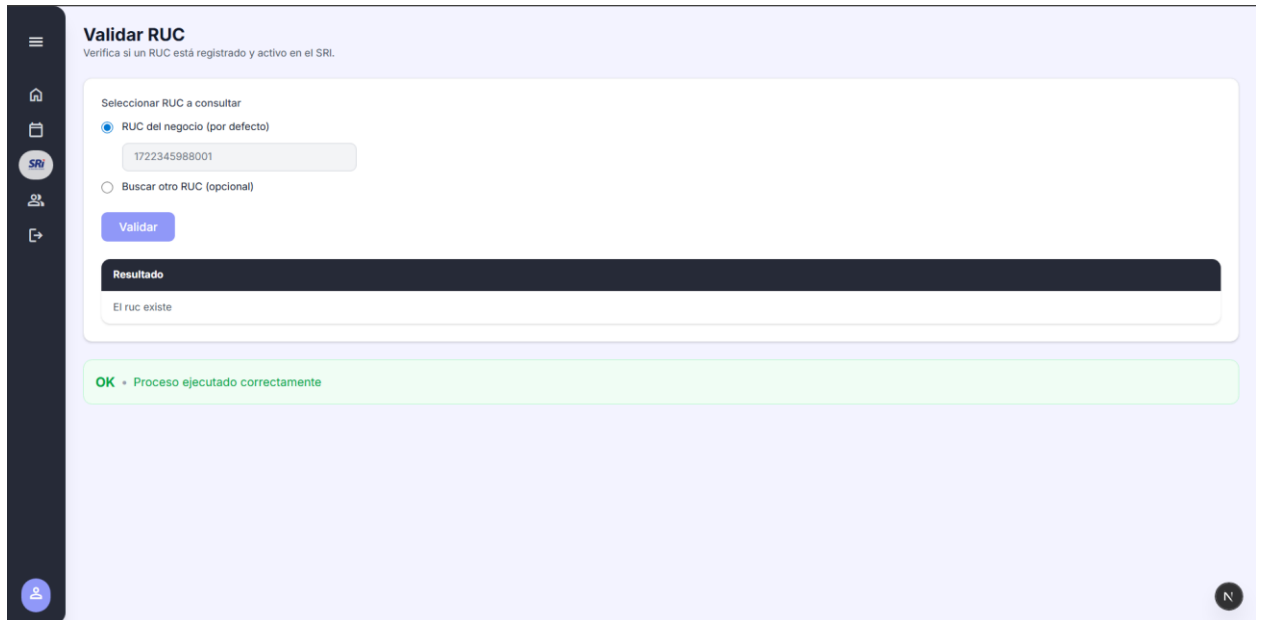


Fig 13 – Validar RUC

e. Validar Establecimientos

En esta sección podrás validar los establecimientos con tu RUC si se encuentra registrado en el SRI, también puedes validarlo con cualquier otro RUC.

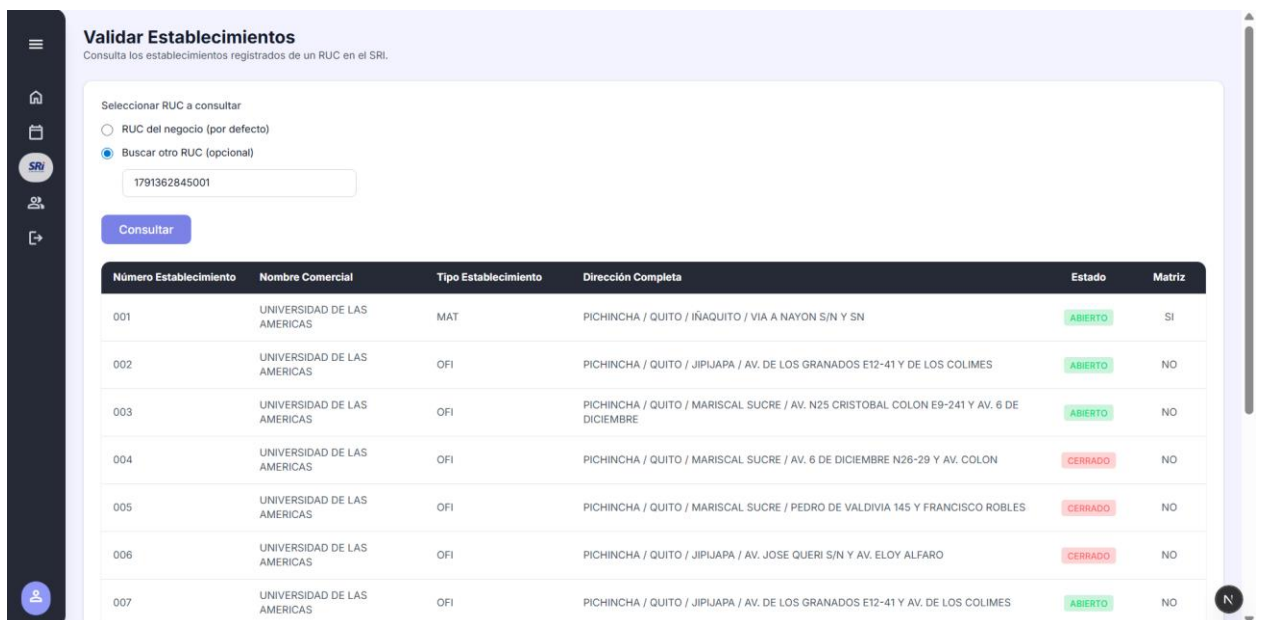
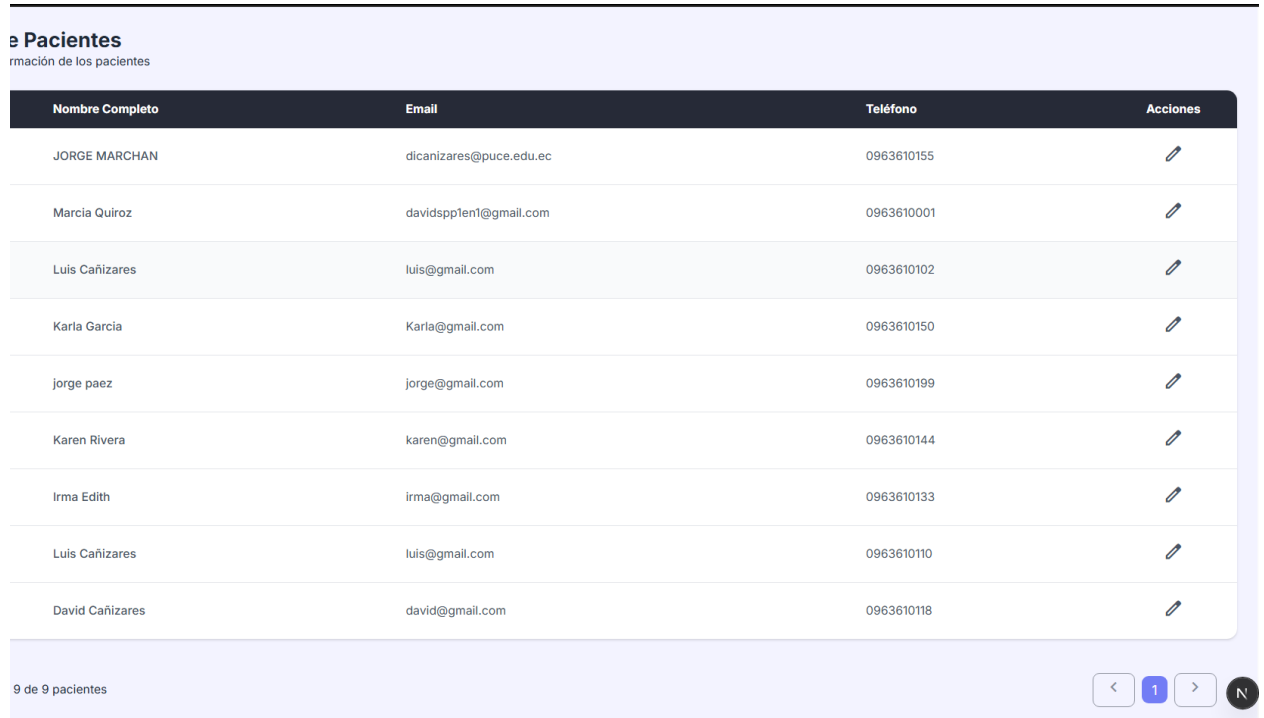











Fig 14 – Validar Establecimiento

f. Pacientes

En esta vista encontraras los pacientes registrados en el sistema.



Nombre Completo	Email	Teléfono	Acciones
JORGE MARCHAN	dicanizares@puce.edu.ec	0963610155	
Marcia Quiroz	davidsp1en1@gmail.com	0963610001	
Luis Cañizares	luis@gmail.com	0963610102	
Karla Garcia	Karla@gmail.com	0963610150	
jorge paez	jorge@gmail.com	0963610199	
Karen Rivera	karen@gmail.com	0963610144	
Irma Edith	irma@gmail.com	0963610133	
Luis Cañizares	luis@gmail.com	0963610110	
David Cañizares	david@gmail.com	0963610118	

9 de 9 pacientes < 1 > N

Fig 15 – Pacientes

Manual de Usuarios General Agendamiento de citas

Para agendar una cita te encontraras con esta vista la cual debes llenar los datos en pantalla, una vez hecho esto podrás elegir el horario y finalizar con tu cita ya agendada

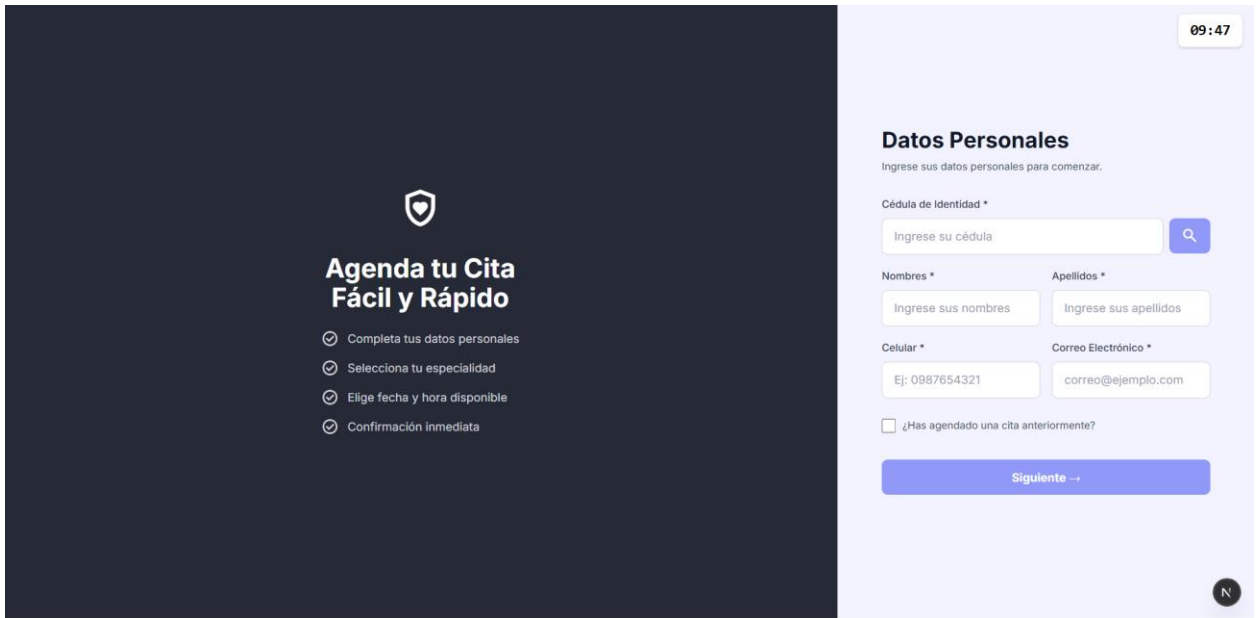


Fig 16 – Agendamiento de citas