PARQUE CIENTÍFICO Y TECNOLÓGICO UTPL

Portal Web CCL

Versión.0.0.1

Ficha Técnica

Autores:

Líder de grupo	Denis Leandro Ruiz Lopez
Estudiante	Pablo Andrés Criollo Álvarez
Estudiante	Augusto Valentino Davila Robles

Docente Tutor: Elizalde Solano Rene Rolando

Índice de contenidos

Contenido

Índice de figuras

1. Descripción del prototipo

El prototipo desarrollado corresponde a una plataforma web integral y funcional, para la Cámara de Comercio de Loja. Su propósito principal es la digitalización y optimización de los procesos centralizando en la gestión de socios, la oferta de servicios, los convenios y la comunicaciones en un único ecosistema digital.

El sistema está diseñado para servir tanto al personal administrativo, a través de un panel de gestión robusto, como a los socios y al público general, mediante una interfaz de usuario moderna e intuitiva.

Para su construcción se utilizó el framework Django, sobre el cual se implementó una arquitectura modular basada en aplicaciones especializadas. Esta técnica de desarrollo permite una clara separación de responsabilidades, facilitando el mantenimiento futuro y la escalabilidad del sistema. Cada módulo (gestión de usuarios, membresías, servicios, contenido, etc.) opera de forma independiente pero interconectada, garantizando un funcionamiento coherente y robusto. El diseño visual se apoya en Bootstrap combinado con css para asegurar una experiencia de usuario responsiva y adaptable a cualquier dispositivo, mientras que para la interactividad se han empleado herramientas como FullCalendar para la gestión de calendarios y reservas, y JavaScript con Intersection Observer para implementar animaciones dinámicas que mejoran la experiencia de navegación.

Las funcionalidades principales del prototipo se pueden segmentar según el tipo de usuario:

1. **Portal Público:** Incluye una página de inicio (landing page) que presenta la información institucional de la Cámara (misión, visión, noticias destacadas) y sirve como puerta de entrada para nuevos socios.

2. **Panel de Socios:** Tras un proceso de registro y autenticación, los socios acceden a un panel de control personalizado. Desde aquí pueden consultar el estado de su membresía, visualizar sus reservas activas, gestionar sus datos de perfil, revisar notificaciones y acceder al catálogo de servicios y convenios exclusivos.

3. **Panel de Administración (Staff):** Es un área de gestión interna y privada que otorga al personal de la Cámara el control total sobre la plataforma. Permite administrar las solicitudes de afiliación y servicios (aprobación/rechazo), gestionar los recursos y horarios disponibles, publicar y moderar contenido (noticias y comentarios) y supervisar la actividad general del sistema a través de métricas clave.

Los flujos de trabajo más importantes, como la solicitud de afiliación y la reserva de servicios, están completamente digitalizados. El usuario es guiado a través de formularios dinámicos, y el sistema gestiona automáticamente los estados de cada solicitud, notificando tanto al usuario como al administrador sobre cualquier cambio relevante. Esta automatización reduce la carga administrativa y agiliza los tiempos de respuesta, mejorando significativamente la eficiencia operativa de la Cámara.

2. Objetivos

Desarrollar e implementar un portal web funcional y accesible que presencie la digitalización de la Cámara de Comercio de Loja, optimizando la comunicación y la interacción con sus socios. Este portal servirá como una plataforma centralizada para proporcionar información relevante, agilizar consultas y mejorar la experiencia del usuario, garantizando una navegación intuitiva y segura.

Objetivos Específicos

1. Diseñar e implementar un portal web público con una interfaz de usuario intuitiva que centralice la información institucional, noticias y comunicados relevantes, mejorando la visibilidad y el acceso a los recursos de la Cámara.
2. Desarrollar un sistema de gestión de usuarios que permita el registro y la autenticación de los socios, ofreciéndoles un panel personalizado para consultar información y gestionar sus solicitudes de manera eficiente.
3. Implementar un panel de administración interno que faculte al personal de la Cámara para gestionar de forma autónoma el contenido del portal, administrar la base de datos de usuarios y supervisar la actividad de la plataforma.

3. Plataforma y tecnologías utilizadas

El prototipo es una plataforma web centralizada para ser completamente accesible desde navegadores web modernos en cualquier tipo de dispositivo, ya sea de escritorio o móvil, por su enfoque en el diseño responsivo.

Para su construcción, se seleccionó un conjunto de tecnologías orientadas a garantizar la escalabilidad, seguridad y mantenibilidad del sistema.

- **Framework de Backend: Django** Se utilizó **Python** como lenguaje de programación principal, en conjunto con el framework **Django**.
- **Tecnologías de Frontend** La interfaz de usuario fue construida utilizando los estándares web: HTML5 para la estructura semántica, CSS para la estilización y JavaScript para la interactividad. Para asegurar una experiencia de usuario consistente y adaptable, se implementó el framework de UI Bootstrap, que proporciona una base sólida para el diseño responsivo. Adicionalmente, se integraron bibliotecas específicas de JavaScript como FullCalendar.js para la creación de calendarios interactivos en el módulo de reservas de servicios.
- **Sistema de Gestión de Base de Datos** La persistencia de los datos se gestiona a través de una base de datos relacional, lo que asegura la integridad y consistencia de la información. El Mapeador Objeto-Relacional (ORM) de Django abstrae la comunicación con la base de datos. Se optó por esta arquitectura para garantizar un rendimiento eficiente y la capacidad de manejar un volumen creciente de datos.

4. Requisitos del sistema

Dado que el prototipo es una plataforma web, los requisitos se dividen en dos componentes principales: las especificaciones para el cliente (el usuario final que accede al sistema) y las especificaciones para el servidor (donde se aloja la aplicación).

Requisitos para el Cliente (Usuario Final)

La plataforma está diseñada para ser altamente accesible, por lo que no requiere la instalación de ningún software especializado por parte del usuario. Los requisitos mínimos son:

- **Sistema Operativo:** Compatible con cualquier sistema operativo moderno, incluyendo Windows 10 o superior, macOS 10.15 o superior, cualquier distribución de Linux reciente, Android 8.0 o superior, e iOS 14 o superior.
- **Navegador Web:** Una versión actualizada de cualquier navegador web moderno. Se recomienda utilizar las últimas versiones de Google Chrome, Mozilla Firefox, Microsoft Edge o Safari para garantizar la compatibilidad total con las funcionalidades y la seguridad del sitio.
- **Conexión a Internet:** Se requiere una conexión a internet estable (banda ancha o conexión móvil 3G/4G/5G) para una experiencia de usuario fluida y sin interrupciones.

Requisitos para el Servidor (Alojamiento)

Para el despliegue y correcto funcionamiento del prototipo en un entorno de producción, se requiere un servidor con la siguiente configuración de software:

- **Sistema Operativo:** Se recomienda un sistema operativo basado en Linux, como Ubuntu 20.04 LTS o superior, por su estabilidad y amplio soporte para las tecnologías utilizadas.
- **Entorno de Ejecución:**
    - **Python:** Versión 3.8 o superior.
    - **Servidor de Aplicaciones WSGI:** Se necesita un servidor como Gunicorn o uWSGI para gestionar las peticiones entre el servidor web y la aplicación Django.

- **Servidor Web:** Un servidor web como Nginx o Apache, configurado como proxy inverso para gestionar el tráfico HTTP/HTTPS y servir los archivos estáticos de manera eficiente.
- **Base de Datos:** Un sistema gestor de base de datos relacional, preferiblemente PostgreSQL (versión 12 o superior) o MySQL (versión 8.0 o superior).

5. Arquitectura y estructura

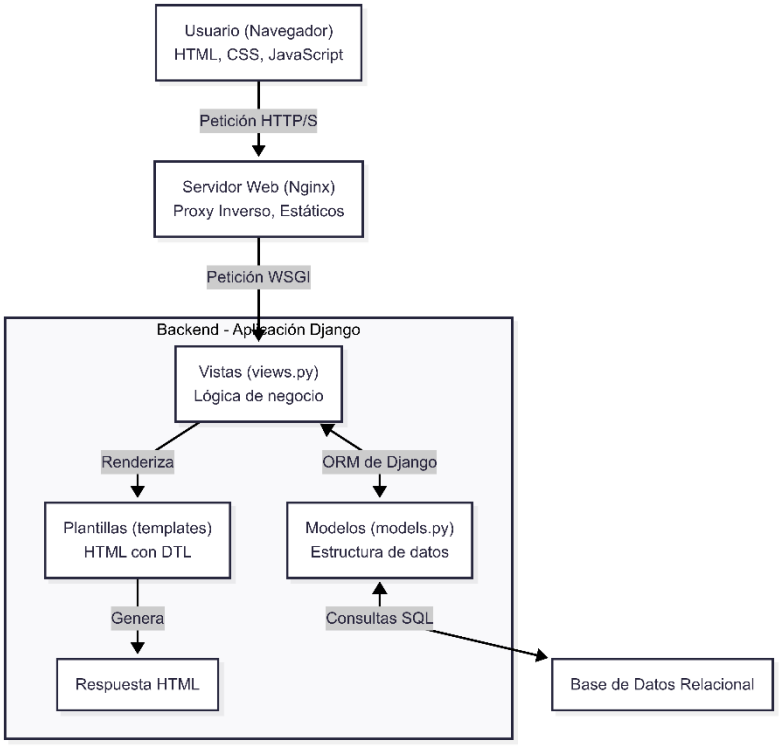
La plataforma se ha desarrollado siguiendo una arquitectura de tres capas y el patrón de diseño Modelo-Vista-Plantilla (MVT), característico del framework Django. Este enfoque garantiza una separación clara entre la lógica de negocio, la representación de los datos y la interfaz de usuario, lo que resulta en un sistema organizado, escalable y fácil de mantener.

La estructura del proyecto es eminentemente modular, organizada en un conjunto de aplicaciones Django independientes, donde cada una encapsula una funcionalidad específica del negocio. Las aplicaciones principales son:

- **core**: Gestiona los elementos globales y la página de inicio.
- **users**: Controla la autenticación, los perfiles de usuario y las membresías.
- **memberships**: Maneja el proceso de solicitudes de afiliación.
- **services**: Administra el catálogo de servicios, recursos y reservas.
- **content**: Se encarga del sistema de noticias y comentarios.
- **staff_panel**: Contiene toda la lógica del panel de administración.

Esta modularidad permite que el desarrollo y las futuras mejoras se puedan realizar de manera aislada en cada componente sin afectar al resto del sistema.

Diagrama de Arquitectura General



Descripción de los Componentes:

1. **Lógica del Servidor (Backend):** El núcleo del sistema reside en la aplicación Django.

- **Modelos (models.py):** Definen la estructura de los datos y las relaciones. A través del ORM de Django, estos modelos se traducen en tablas en la base de datos, permitiendo interactuar con ella usando código Python en lugar de SQL directo.
- **Vistas (views.py):** Contienen la lógica de negocio. Procesan las peticiones del usuario, interactúan con los modelos para acceder a los datos y seleccionan la plantilla adecuada para renderizar la respuesta.
- **Plantillas (templates):** Son archivos HTML que definen la estructura de la interfaz de usuario. Utilizan el lenguaje de plantillas de Django (DTL) para mostrar dinámicamente los datos proporcionados por las vistas.

2. **Base de Datos:** Se utiliza un sistema de gestión de base de datos relacional que actúa como el almacén persistente para toda la información de la plataforma, incluyendo datos de usuarios, socios, servicios, noticias y solicitudes.
3. **Cliente (Frontend):** Es la capa de presentación que se ejecuta en el navegador del usuario. Está compuesta por el HTML, CSS y JavaScript que el servidor envía como respuesta, permitiendo al usuario interactuar con la plataforma.

6. Funcionalidades principales

El prototipo implementa un conjunto completo de funcionalidades diseñadas para la Cámara de Comercio. Estas se pueden agrupar así:

Portal Público y Gestión de Contenido

- **Página de Inicio (Landing Page):** Presenta información institucional clave como la misión, visión y noticias destacadas, sirviendo como el principal punto de acceso público.
- **Sistema de Noticias y Blog:** Permite al personal administrativo publicar artículos, eventos y comunicados para mantener informados a los socios y al público.
- **Gestión de Convenios:** Muestra un listado de las alianzas estratégicas con otras empresas, detallando los beneficios y la información de contacto para los socios.
- **Sistema de Comentarios:** Los usuarios autenticados pueden interactuar dejando comentarios en las noticias, los cuales son sometidos a un proceso de moderación por parte del staff antes de ser publicados.

Gestión de Socios y Membresías

- **Registro y Autenticación de Usuarios:** Un sistema seguro que permite a los usuarios crear una cuenta y acceder a la plataforma.
- **Proceso de Solicitud de Afiliación:** Un flujo guiado para que personas naturales o jurídicas puedan solicitar su membresía, completando formularios dinámicos y adjuntando la documentación requerida.
- **Panel de Control del Socio (Dashboard):** Un área privada y personalizada donde cada socio puede consultar el estado de su membresía, ver sus notificaciones, gestionar sus datos de perfil y acceder a sus reservas y solicitudes.

Gestión de Servicios y Reservas

- **Catálogo de Servicios:** Un listado detallado de todos los servicios que ofrece la Cámara, como alquiler de salones o asesorías.

- **Sistema de Reservas:** Los socios pueden visualizar la disponibilidad de los recursos (ej. salones) en un calendario interactivo y solicitar horarios específicos.
- **Seguimiento de Solicitudes:** Tanto el socio como el personal pueden dar seguimiento al estado de cada solicitud de servicio (pendiente, aprobada, rechazada).

Panel de Administración (Staff)

- **Dashboard Administrativo:** Ofrece una vista general con métricas clave, como el número de solicitudes pendientes.
- **Gestión Centralizada de Solicitudes:** Permite al personal revisar, aprobar o rechazar las solicitudes de afiliación y de reserva de servicios, dejando constancia de cada acción.
- **Administración de Contenido y Servicios:** Herramientas para crear, editar y publicar noticias, convenios, servicios y los recursos asociados a ellos.
- **Gestión de Horarios:** El personal puede definir y modificar los bloques de disponibilidad para cada recurso directamente en el calendario.

7. APIs y servicios externos

En su fase actual, el prototipo se ha desarrollado para ser mayormente autocontenido, minimizando la dependencia de servicios de terceros para sus funcionalidades básicas. Sin embargo, está diseñado con una arquitectura modular que facilita la integración futura de diversas APIs y servicios externos para ampliar sus capacidades.

Las integraciones actuales y consideraciones futuras son las siguientes:

Servicios Utilizados en el Prototipo

- **Redes de Distribución de Contenidos (CDN):** Para optimizar los tiempos de carga y mejorar el rendimiento, el prototipo utiliza CDNs para la entrega de bibliotecas de frontend.
    - **Bootstrap:** Los archivos de CSS y JavaScript se enlazan a través de su CDN oficial.
    - **FullCalendar.js:** La biblioteca para la gestión de calendarios interactivos también se carga desde un servicio de CDN.
    - **Requisitos:** Para el correcto funcionamiento de la interfaz, se requiere una conexión a internet que permita el acceso a estos servicios. No se necesita ninguna clave de API.

Consideraciones para Futuras Integraciones

La arquitectura del sistema está preparada para incorporar los siguientes servicios, que son cruciales para una implementación completa y en producción:

- **Pasarela de Pagos (API de Pagos):**
    - **Descripción:** Para habilitar funcionalidades como el pago en línea de las cuotas de membresía o la reserva de servicios, será necesario integrar una pasarela de pagos como Stripe, PayPal o un proveedor local.
    - **Requisitos:** Se requerirá la obtención de claves de API (públicas y secretas) del proveedor seleccionado y la implementación de webhooks para recibir notificaciones de pago de forma segura. Se deberá configurar un entorno de pruebas (sandbox) antes del despliegue en producción.

8. Equipo de desarrollo

- Pablo Andrés Criollo Álvarez - Líder de Proyecto y Desarrollador Backend

Responsabilidades: Encargado de la coordinación general del equipo, la planificación de las fases del proyecto y la toma de decisiones arquitectónicas.

- Denis Leandro Ruiz Lopez - Desarrollador Frontend y Diseñador UI/UX

Responsabilidades: Responsable de toda la capa de presentación de la plataforma. Se encargó del código utilizando HTML, CSS y JavaScript, así como de la implementación del framework Bootstrap.

- Augusto Valentino Davila Robles - Desarrollador Full-Stack y Gestor de Base de Datos

Responsabilidades: Trabajó tanto en el frontend como en el backend, sirviendo de puente entre ambas áreas para asegurar una correcta integración.

9. Versiones y evolución del prototipo

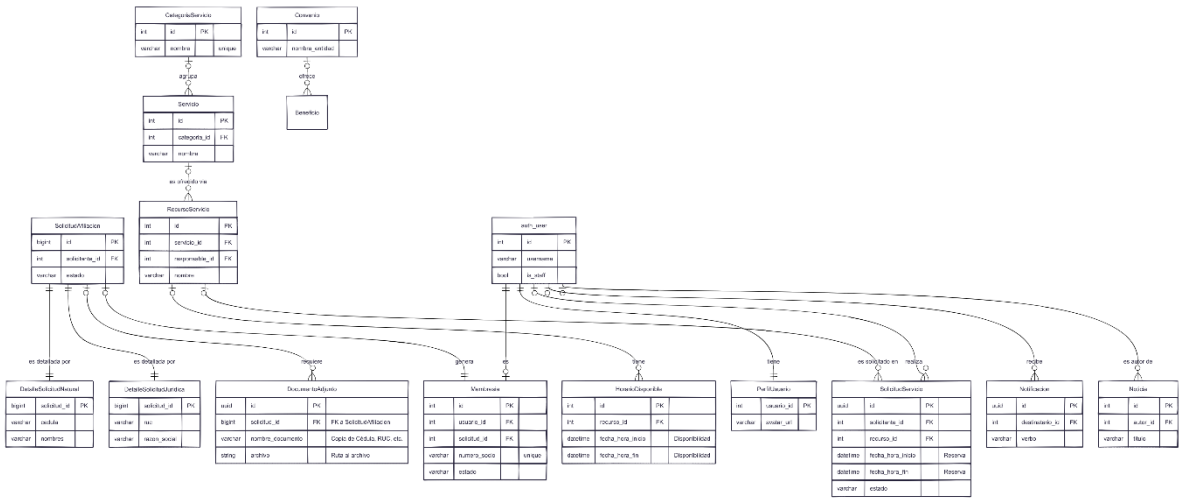
El desarrollo del prototipo se ha llevado a cabo de manera incremental, permitiendo validar las funcionalidades clave en cada etapa y asegurar que el producto final cumpla con los objetivos establecidos.

Versión 0.0.1 - Producto Mínimo Viable (MVP)

Esta primera versión se centró en establecer las bases de la plataforma y las funcionalidades esenciales para su operación. Los cambios y características principales incluyeron:

- **Estructura del Proyecto:** Creación de la arquitectura modular en Django y configuración inicial de la base de datos.
- **Autenticación y Usuarios:** Implementación del sistema de registro, inicio de sesión y perfiles de usuario básicos.
- **Gestión de Contenido:** Desarrollo del módulo de noticias, permitiendo al personal administrativo crear y publicar artículos desde un panel de gestión simple.
- **Interfaz Pública:** Diseño de la página de inicio y el listado de noticias.
- **Módulo de Socios y Solicitudes:** Se implementó el flujo completo para la solicitud de afiliación, incluyendo los formularios dinámicos y el panel de gestión para su aprobación o rechazo.
- **Módulo de Servicios y Reservas:** Se añadió el catálogo de servicios y el sistema de reservas con un calendario interactivo (FullCalendar) para la gestión de disponibilidad.
- **Dashboard del Socio:** Creación del panel de control personalizado para que los socios puedan dar seguimiento a sus solicitudes y membresía.
- **Mejoras en UI/UX:** Se refinó el diseño visual con Bootstrap y se añadieron animaciones para mejorar la experiencia de usuario.

10. Anexos


CategoriaServicio
Convenio
Servicio
Beneficio
ResourceService
SolicitudAfiliacion
auth_user
DetalleSolicitudNatural
DetalleSolicitudJuridica
DocumentoAdjunto
MiembroSocio
HorarioDisponible
PerfilUsuario
SolicitudServicio
Notificacion
Noticia