

***“MyDeptos”***

**Schallmoser Barrios, Facundo**

**Mora, Maia Lujan**

**I.P.E.T. 1308**

INSTITUTO PRIVADO DE ESTUDIOS SUPERIORES 1308

Índice

[Introducción 4](#_Toc204101490)

[Propósito del Proyecto 4](#_Toc204101491)

[Alcance 4](#_Toc204101492)

[Requisitos del Sistema 5](#_Toc204101493)

[Para Usuarios Finales (Clientes Web) 5](#_Toc204101494)

[Hardware Mínimo: 5](#_Toc204101495)

[Hardware Recomendado: 5](#_Toc204101496)

[Software: 5](#_Toc204101497)

[Dependencias y Herramientas de Desarrollo: 6](#_Toc204101498)

[Para Servidor de Producción 7](#_Toc204101499)

[Instalación y Configuración 9](#_Toc204101500)

[Pasos para Instalar 9](#_Toc204101501)

[Descarga del Repositorio 9](#_Toc204101502)

[Configuración del Entorno de Desarrollo 9](#_Toc204101503)

[Instalación de Dependencias 9](#_Toc204101504)

[Importación de la Base de Datos 9](#_Toc204101505)

[Configuración Inicial 10](#_Toc204101506)

[Variables de Entorno 10](#_Toc204101507)

[Archivos de configuración 10](#_Toc204101508)

[Estructura del Proyecto 11](#_Toc204101509)

[Directorios y Archivos Clave: 11](#_Toc204101510)

[Diagrama de Árbol de Carpetas 11](#_Toc204101511)

[Explicación de Módulos Principales 14](#_Toc204101512)

[Diagrama de Arquitectura: 15](#_Toc204101513)

[Diagrama de componentes 15](#_Toc204101514)

[Mapa canónico de la Base de Datos 16](#_Toc204101515)

[Uso del Software 22](#_Toc204101516)

[Funcionalidades Básicas: 22](#_Toc204101517)

[Flujo de trabajo típico 34](#_Toc204101518)

[Solución de Problemas 35](#_Toc204101519)

[Errores Comunes 35](#_Toc204101520)

[Preguntas Frecuentes (FAQ) 36](#_Toc204101521)

[Pruebas y Validación 37](#_Toc204101522)

[Ejecución de Tests 37](#_Toc204101523)

[Reporte de Bugs 37](#_Toc204101524)

[Referencias 37](#_Toc204101525)

[Enlaces a documentación oficial de herramientas/librerías usadas 37](#_Toc204101526)

[Apendice 38](#_Toc204101527)

[Glosario de Términos Técnicos y del proyecto 38](#_Toc204101528)

[Correo Autores 39](#_Toc204101529)

[Enlace a Repositorio de GitHub 39](#_Toc204101530)

[Conclusión final 40](#_Toc204101531)

[Agradecimientos 40](#_Toc204101532)

# Introducción

## Propósito del Proyecto

MyDeptos es una aplicación web desarrollada como plataforma inmobiliaria que facilita la búsqueda, publicación y gestión de departamentos y propiedades en la región de Oberá, Misiones. El proyecto surgió con el objetivo de resolver una necesidad común entre estudiantes y trabajadores que buscan alojamiento en una nueva ciudad: ¿Dónde puedo encontrar un departamento disponible cerca de la zona?

**La aplicación permite a los usuarios:**

* Buscar propiedades mediante filtros avanzados (precio, ubicación, cantidad de ambientes, tipo de operación)
* Publicar departamentos para venta o alquiler con información detallada y fotografías
* Gestionar perfiles de usuario con historial de publicaciones y favoritos
* Sistema de reseñas para calificar propiedades y propietarios
* Notificaciones para mantener informados a los usuarios sobre actividades relevantes
* Integración con mapas para visualizar ubicaciones precisas
* Contacto directo con propietarios vía WhatsApp

## Alcance

**Qué cubre el software:**

Funcionalidades principales:

* Gestión de usuarios: Registro, autenticación, perfiles de usuario y recuperación de contraseñas.
* Publicación de propiedades: Sistema completo para subir departamentos con múltiples fotografías, descripción detallada, ubicación en mapa interactivo y datos técnicos.
* Sistema de búsqueda avanzada: Filtros por tipo de operación (venta/alquiler), rango de precios, localidad, cantidad de ambientes.
* Sistema de favoritos: Los usuarios pueden guardar propiedades de interés.
* Sistema de reseñas y calificaciones: Permite a los usuarios calificar propiedades y dejar comentarios.
* Notificaciones: Sistema automático de alertas para nuevas reseñas y actividades relevantes.
* Integración con mapas: Visualización de ubicaciones usando Leaflet/OpenStreetMap.
* Contacto directo: Integración con WhatsApp para comunicación inmediata con propietarios.
* Panel de administración de usuario: Gestión completa de publicaciones propias, edición y eliminación.
* Responsive design: Interfaz adaptable a dispositivos móviles y desktop.

Tecnologías utilizadas:

* Backend: Flask (Python) con MySQL como base de datos
* Frontend: HTML5, CSS, JavaScript, Bootstrap
* Mapas: Leaflet.js con OpenStreetMap
* Seguridad: Autenticación con hash de contraseñas, protección CSRF, validación de formularios
* Notificaciones por email: Sistema SMTP integrado

Qué NO cubre el software:

* Procesamiento de pagos: No incluye pasarelas de pago o transacciones monetarias
* Sistema de mensajería interna: La comunicación se realiza exclusivamente vía WhatsApp externo
* Verificación de identidad: No incluye validación de documentos
* Sistema de contratos: No gestiona contratos de alquiler o venta
* Múltiples idiomas: Disponible únicamente en español
* Integración con redes sociales: No permite login con Facebook, Google, etc.
* Sistema de comisiones: No maneja aspectos financieros entre partes
* Chat en tiempo real: No incluye sistema de mensajería instantánea interna
* Aplicación móvil nativa: Es una aplicación web responsiva, no una app móvil dedicada

# Requisitos del Sistema

## Para Usuarios Finales (Clientes Web)

### Hardware Mínimo:

* RAM: 2 GB
* Procesador: Dual-core 1.5 GHz o superior
* Almacenamiento: 100 MB de espacio libre (para caché del navegador)
* Conectividad: Conexión a Internet estable (mínimo 1 Mbps)
* Pantalla: Resolución mínima 1024x768 píxeles

### Hardware Recomendado:

* RAM: 4 GB o superior
* Procesador: Quad-core 2.0 GHz o superior
* Almacenamiento: 500 MB de espacio libre
* Conectividad: Conexión a Internet de banda ancha (5 Mbps o superior)
* Pantalla: Resolución 1920x1080 píxeles o superior

### Software:

* Navegadores Web Compatibles:
* Google Chrome 90 o superior (recomendado)
* Mozilla Firefox 88 o superior
* Microsoft Edge 90 o superior
* Safari 14 o superior (macOS/iOS)
* Sistemas Operativos Compatibles:
* Windows 10/11
* macOS 10.15 (Catalina) o superior
* Linux (Ubuntu 18.04+, CentOS 7+, Debian 10+)
* Android 8.0+ (navegadores móviles)
* iOS 13+ (Safari móvil)
* JavaScript: Habilitado (obligatorio)

### Dependencias y Herramientas de Desarrollo:

**Lenguaje de Programación Principal:**

* Python: Versión 3.8 o superior (recomendado 3.9+)

**Framework Principal:**

* Flask: 3.1.0 o superior

**Base de Datos:**

* MySQL: 8.0 o superior (obligatorio)
* Alternativa: MariaDB 10.5+ (compatible)

**Servidor Web (Producción):**

* Opción 1: Apache HTTP Server 2.4+ con mod\_wsgi
* Opción 2: Nginx 1.18+ con Gunicorn
* Opción 3: Servidor integrado de Flask (solo desarrollo)

**Dependencias Python (requirements.txt):**

Flask==3.1.0

Flask-MySQLdb==2.0.0

Flask-WTF==1.2.2

Flask-Login==0.6.3

bcrypt==4.2.1

mysql-connector-python==9.2.0

mysqlclient==2.2.7

WTForms==3.2.1

Werkzeug==3.1.3

Jinja2==3.1.5

itsdangerous==2.2.0

email\_validator==2.2.0

dnspython==2.7.0

**Herramientas de Desarrollo Requeridas:**

* Git: 2.30+ (control de versiones)
* pip: 21.0+ (gestor de paquetes Python)
* XAMPP o WAMP (entorno de desarrollo local con MySQL)
* Editor de código: VS Code, PyCharm, Sublime Text, o similar

**APIs y Servicios Externos:**

* OpenStreetMap/Leaflet: Para mapas interactivos
* SMTP Gmail: Para envío de correos electrónicos
* WhatsApp Web API: Para integración de contacto

**Bibliotecas Frontend:**

* Bootstrap: 5.3.0+ (framework CSS)
* jQuery: 3.6.0+ (manipulación DOM)
* Leaflet.js: 1.9.4+ (mapas interactivos)
* Font Awesome: 6.0.0+ (iconografía)
* Slick Carousel: 1.8.1+ (carruseles de imágenes)

## Para Servidor de Producción

**Hardware Mínimo:**

* RAM: 4 GB
* Procesador: 2 vCPUs
* Almacenamiento: 50 GB SSD
* Ancho de banda: 100 GB/mes

**Hardware Recomendado:**

* RAM: 8 GB o superior
* Procesador: 4+ vCPUs
* Almacenamiento: 100 GB SSD con respaldos
* Ancho de banda: 1 TB/mes

**Software del Servidor:**

* Sistema Operativo:
* Ubuntu Server 20.04 LTS (recomendado)
* CentOS 8 Stream
* Debian 11
* Servidor Web: Nginx 1.18+ o Apache 2.4+
* Servidor de Aplicaciones: Gunicorn 20.0+
* Base de Datos: MySQL 8.0+ con InnoDB
* Certificado SSL: Let's Encrypt o certificado comercial
* Firewall: UFW (Ubuntu) o firewalld (CentOS)
* Monitoreo: Opcional (htop, netdata, etc.)

**Variables de Entorno Requeridas:**

FLASK\_APP=app.py

FLASK\_ENV=production

SECRET\_KEY=clave\_secreta\_production

MYSQL\_HOST=localhost

MYSQL\_USER=mydeptos\_user

MYSQL\_PASSWORD=contraseña\_segura

MYSQL\_DB=bdmydeptos

SMTP\_SERVER=smtp.gmail.com

SMTP\_PORT=587

SMTP\_USER=mydeptos@gmail.com

SMTP\_PASSWORD=contraseña\_aplicacion

**Consideraciones de Seguridad:**

* HTTPS: Obligatorio en producción
* Firewall: Puertos 22 (SSH), 80 (HTTP), 443 (HTTPS), 3306 (MySQL - restringido)
* Backups: Respaldos automáticos de la base de datos
* Actualizaciones: Sistema operativo y dependencias actualizadas
* Logs: Sistema de logging configurado para monitoreo

**Escalabilidad:**

* Carga ligera: Hasta 100 usuarios concurrentes
* Carga media: 100-500 usuarios concurrentes (requiere optimización)
* Carga alta: 500+ usuarios concurrentes (requiere arquitectura distribuida)

# Instalación y Configuración

## Pasos para Instalar

### Descarga del Repositorio

**Clonación desde GitHub:**

git clone https://github.com/Toukhay/MyDeptosFinal.git

cd MyDeptosFinal

### Configuración del Entorno de Desarrollo

**Crear y activar entorno virtual:**

En Windows:

python -m venv venv

.\venv\Scripts\activate

En macOS/Linux:

python3 -m venv venv

source venv/bin/activate

### Instalación de Dependencias

pip install -r requirements.txt

### Importación de la Base de Datos

XAMPP (Recomendado):

1. Descargar e instalar XAMPP
2. Ejecutar XAMPP e iniciar servicios Apache y MySQL
3. Abrir phpMyAdmin: <http://localhost/phpmyadmin>
4. Importar base de datos:

* Hacer clic en "Importar"
* Seleccionar archivo: bdmydeptos.sql
* Hacer clic en "Continuar"
* ¡Listo! La base de datos está configurada con estructura y datos iniciales

## Configuración Inicial

### Variables de Entorno

El proyecto utiliza un archivo [.env](vscode-file://vscode-app/c:/Users/facu-/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) para manejar las variables de entorno. Según el código en [app.py](vscode-file://vscode-app/c:/Users/facu-/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), las variables necesarias son:

MYSQL\_HOST=localhost

MYSQL\_USER=root

MYSQL\_PASSWORD=tu\_password\_mysql

MYSQL\_DB=bdmydeptos

SECRET\_KEY=tu\_clave\_secreta\_super\_segura

# Variables para envío de correos (función send\_email)

SMTP\_SERVER=smtp.gmail.com

SMTP\_PORT=587

EMAIL\_USER=tu\_email@gmail.com

EMAIL\_PASSWORD=tu\_password\_email

### Archivos de configuración

1. Archivo [.env.example](vscode-file://vscode-app/c:/Users/facu-/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)

Ya existe un [.env.example](vscode-file://vscode-app/c:/Users/facu-/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) que sirve como plantilla. Debes copiarlo y renombrarlo a [.env](vscode-file://vscode-app/c:/Users/facu-/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html):

cp .env.example .env

1. Configuración en app.py

La configuración se carga automáticamente:

from dotenv import load\_dotenv

load\_dotenv()

app.config['MYSQL\_HOST'] = os.getenv('MYSQL\_HOST', 'localhost')

app.config['MYSQL\_USER'] = os.getenv('MYSQL\_USER', 'root')

app.config['MYSQL\_PASSWORD'] = os.getenv('MYSQL\_PASSWORD', '')

app.config['MYSQL\_DB'] = os.getenv('MYSQL\_DB', 'bdmydeptos')

app.config['SECRET\_KEY'] = os.getenv('SECRET\_KEY', 'clave\_secreta')

Ejecución Inicial

1. Ejecutar la Aplicación

# Asegurarse de que el entorno virtual esté activo

python app.py

1. Verificar Instalación

Abrir navegador web

Navegar a: <http://localhost:5000>

# Estructura del Proyecto

## Directorios y Archivos Clave:

### Diagrama de Árbol de Carpetas

MyDeptosFlask/

├── 📁 database/

│ └── bdmydeptos.sql # Exportación completa de la base de datos

├── 📁 logs/

│ ├── mydeptos.log # Logs principales de la aplicación

│ ├── mydeptos.log.1 # Logs rotados (respaldo 1)

│ └── mydeptos.log.9 # Logs rotados (respaldo 9)

├── 📁 static/ # Archivos estáticos del frontend

│ ├── 📁 css/ # Hojas de estilo CSS

│ │ ├── about.css # Estilos para página "Sobre Nosotros"

│ │ ├── admin\_edit\_depto.css # Estilos para página "admin\_edit\_depto”

│ │ ├── admin\_edit\_user.css # Estilos para página "admin\_edit\_user”

│ │ ├── admin\_panel.css # Estilos para página "admin\_panel”

│ │ ├── cerrandosesion.css # Estilos para cierre de sesión

│ │ ├── delete\_confirmation.css # Estilos para confirmación de eliminación

│ │ ├── edit\_profile.css # Estilos para edición de perfil

│ │ ├── favorites.css # Estilos para página de favoritos

│ │ ├── forgotpassword.css # Estilos para recuperación de contraseña

│ │ ├── generate\_password.css # Estilos para generación de contraseñas

│ │ ├── home.css # Estilos para página principal

│ │ ├── iniciosesion.css # Estilos para inicio de sesión

│ │ ├── listings.css # Estilos para listado de departamentos

│ │ ├── login.css # Estilos adicionales de login

│ │ ├── modify\_depto.css # Estilos para modificar departamentos

│ │ ├── notifications\_page.css # Estilos para página de notificaciones

│ │ ├── publication\_success.css # Estilos para éxito en publicación

│ │ ├── publish\_depto.css # Estilos para publicar departamentos

│ │ ├── register.css # Estilos para registro de usuarios

│ │ ├── register\_success.css # Estilos para éxito en registro

│ │ ├── resetpassword.css # Estilos para reseteo de contraseña

│ │ ├── search\_results.css # Estilos para resultados de búsqueda

│ │ ├── user\_panel.css # Estilos para panel de usuario

│ │ └── view\_property.css # Estilos para vista de propiedades

│ └── 📁 image/ # Imágenes y archivos multimedia

│ ├── [archivos de imágenes] # Fotos de departamentos subidas

├── 📁 templates/ # Plantillas HTML (Jinja2)

│ ├── 404.html # Página de error 404

│ ├── about.html # Página "Sobre Nosotros"

│ ├── admin\_edit\_depto.html # Página "admin\_edit\_depto"

│ ├── admin\_edit\_user.html # Página "admin\_edit\_user"

│ ├── admin\_panel.html # Página "admin\_panel"

│ ├── delete\_confirmation.html # Confirmación de eliminación

│ ├── edit\_profile.html # Edición de perfil de usuario

│ ├── favorites.html # Página de favoritos

│ ├── forgotpassword.html # Recuperación de contraseña

│ ├── generate\_password.html # Generación de contraseñas

│ ├── home.html # Página principal/inicio

│ ├── listings.html # Listado completo de departamentos

│ ├── login.html # Formulario de inicio de sesión

│ ├── login\_redirect.html # Redirección post-login

│ ├── logout\_redirect.html # Redirección post-logout

│ ├── modify\_depto.html # Formulario de modificación de departamentos

│ ├── notifications.html # Página de notificaciones

│ ├── publication\_success.html # Confirmación de publicación exitosa

│ ├── publish\_depto.html # Formulario de publicación de departamentos

│ ├── register.html # Formulario de registro

│ ├── register\_success.html # Confirmación de registro exitoso

│ ├── resetpassword.html # Formulario de reseteo de contraseña

│ ├── search\_results.html # Resultados de búsqueda

│ ├── user\_panel.html # Panel principal del usuario

│ └── viewProperty.html # Vista detallada de propiedad

├── 📁 venv/ # Entorno virtual de Python

├── 📁 \_\_pycache\_\_/ # Cache de Python compilado

│ ├── app.cpython-313.pyc # Archivo principal compilado

│ └── config.cpython-313.pyc # Configuración compilada

├── 📄 .env #Variables de entorno

├── 📄 .env.example #Ejemplo para crear .env

├── 📄 .gitignore #Ignora los archivos que no tenemos que subir al repositorio

├── 📄 app.py # \*\*ARCHIVO PRINCIPAL\*\* - Aplicación Flask

├── 📄 requirements.txt # Dependencias de Python

├── 📄 README.md # Documentación principal del proyecto

├── 📄 LICENSE # Licencia del proyecto (MIT)

├── 📄 SECURITY.md # Políticas de seguridad

├── 📄 PythonWord.py # Utilidad para generación de documentos

├── 📄 Guia\_MyDeptos.docx # Guía de usuario en formato Word

├── 📄 instalacion config proyecto.txt # Instrucciones de instalación

└── 📄 # Code Citations.md # Referencias y citas del código

### Explicación de Módulos Principales

1. Archivo Principal (app.py)

* Núcleo de la aplicación Flask
* Funciones principales:
* Configuración de la aplicación y base de datos
* Definición de todas las rutas (endpoints)
* Formularios WTF para validación
* Sistema de autenticación y autorización
* Manejo de sesiones de usuario
* Sistema de logging
* Envío de correos electrónicos
* Gestión de archivos (fotos)

1. Módulo de Templates (templates/)

* Plantillas HTML con Jinja2
* Componentes principales:
* Autenticación: **login.html, register.html, forgotpassword.html**
* Gestión de usuario: **user\_panel.html, edit\_profile.html**
* Departamentos: **publish\_depto.html, modify\_depto.html, viewProperty.html**
* Navegación: **home.html, listings.html, search\_results.html**
* Interacción: **favorites.html, notifications.html**

1. Módulo de Estilos (static/css/)

* Características:
* Diseño responsivo para móviles y desktop
* Estilos específicos por funcionalidad
* Integración con Bootstrap y Font Awesome
* Temas consistentes de colores y tipografías

1. Módulo de Base de Datos (database/)

* Bdmydeptos.sql - Exportación completa
* Contiene:
* 9 tablas principales con datos reales

1. Módulo de Archivos Estáticos (static/image/)

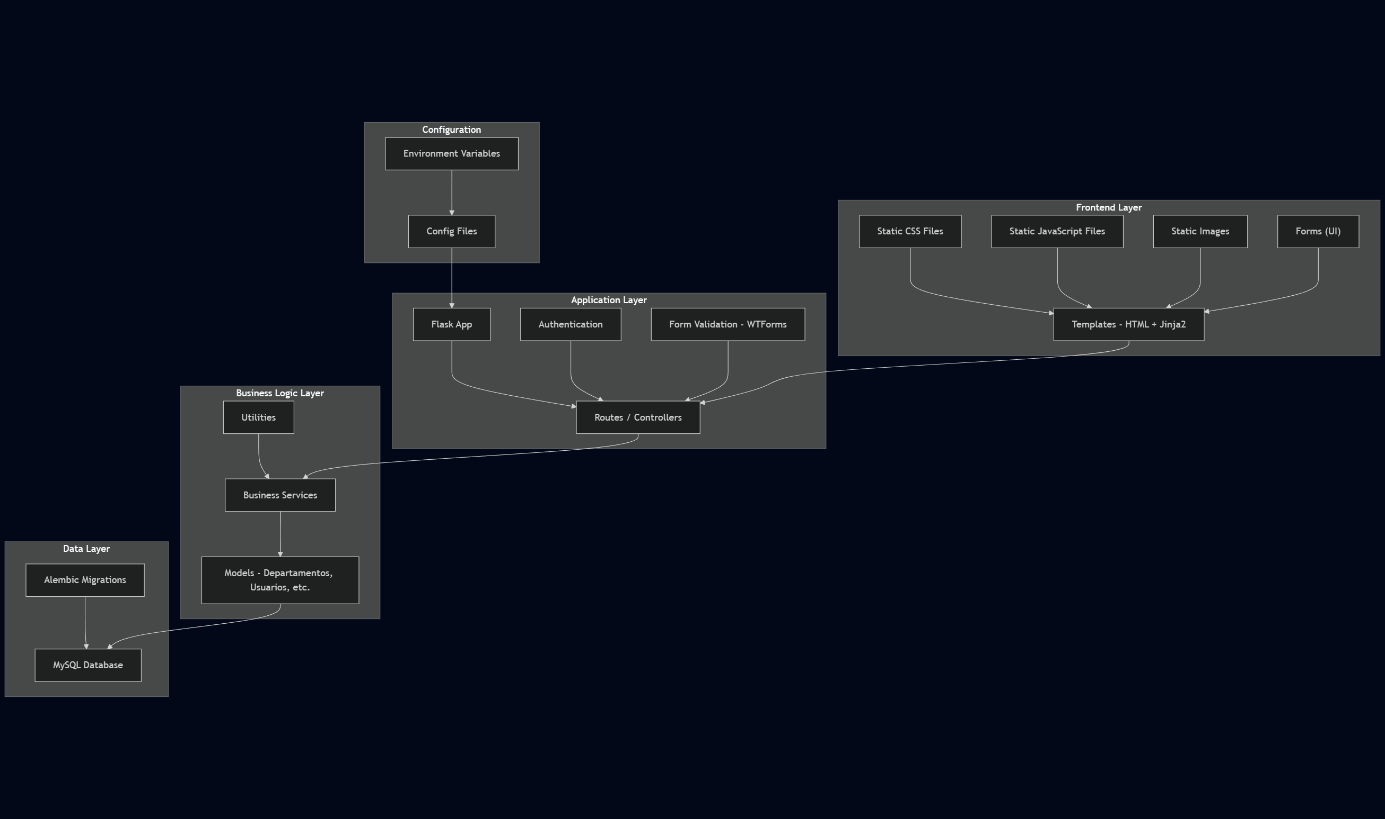
* Tipos de contenido:
* Fotos de departamentos subidas por usuarios

1. Sistema de Logs (logs/)

* Logging rotativo automático
* Información registrada:
* Accesos de usuarios
* Errores de aplicación
* Operaciones CRUD en la base de datos
* Actividad de autenticación

## Diagrama de Arquitectura:

### Diagrama de componentes



Componentes Principales:

Frontend Layer

* Templates: Archivos HTML con Jinja2 para renderizado dinámico
* Static Files: CSS, JavaScript e imágenes
* Forms: Interfaz de usuario para captura de datos

Application Layer

* Flask App: Núcleo de la aplicación web
* Routes: Controladores que manejan las peticiones HTTP
* Authentication: Sistema de login/logout de usuarios
* Forms Validation: Validación de formularios con WTForms

Business Logic Layer

* Models: Definición de entidades (Departamentos, Usuarios, etc.)
* Services: Lógica de negocio y operaciones CRUD
* Utilities: Funciones auxiliares y helpers

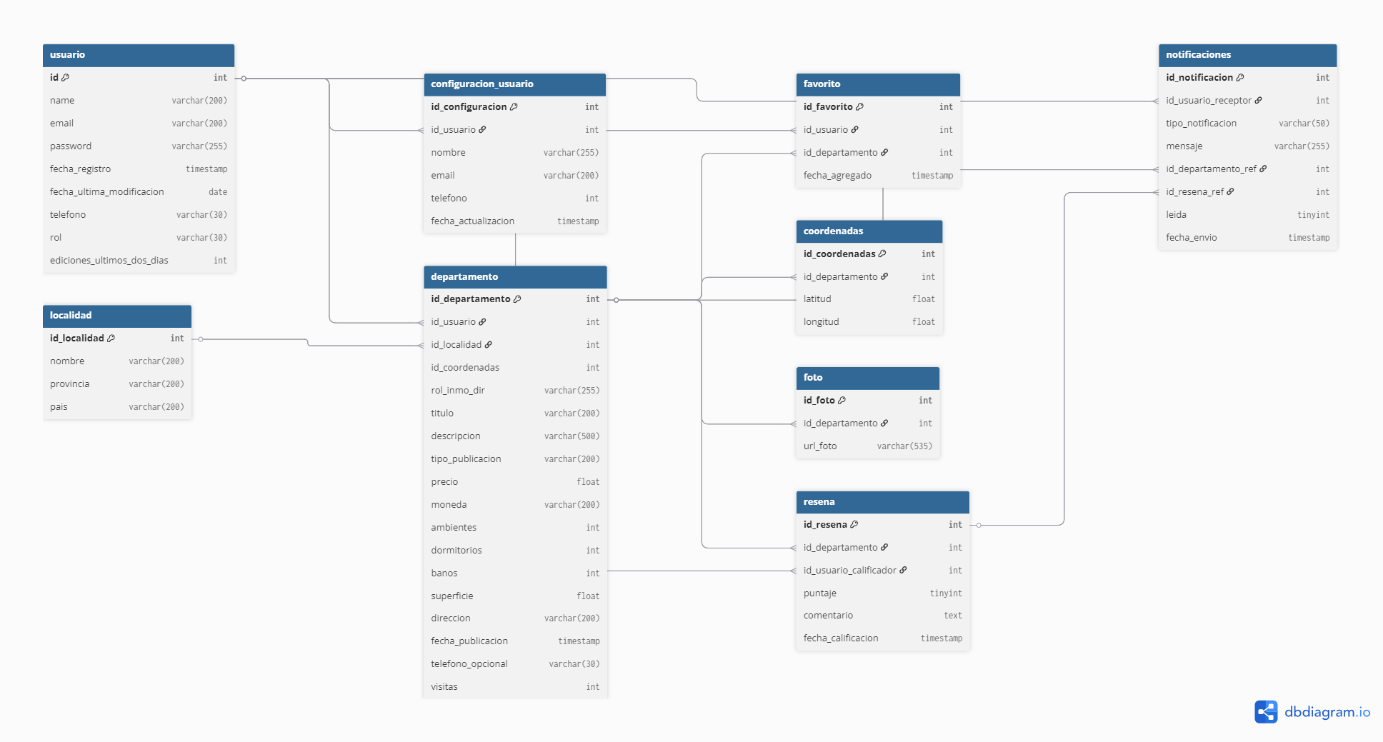
Data Layer

* Database: Base de datos MySQL para persistencia
* Migrations: Control de versiones del esquema de BD

Configuration

* Config Files: Configuración de la aplicación
* Environment: Variables de entorno para diferentes ambientes

### Mapa canónico de la Base de Datos



**DESCRIPCIÓN DE ENTIDADES Y CLAVES**

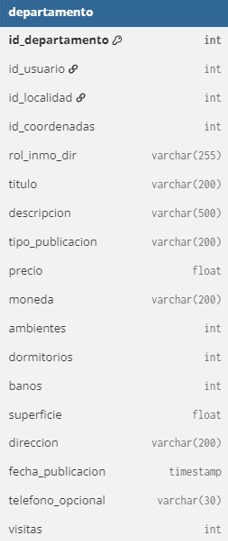
**Usuario**

* id (PK)
* name, email, telefono, password, fecha\_registro, fecha\_ultima\_modificacion, rol



**Departamento**

* id\_departamento (PK)
* FK: id\_usuario → Usuario
* FK: id\_localidad → Localidad
* FK: id\_coordenadas → Coordenadas
* Campos: titulo, descripcion, precio, moneda, ambientes, dormitorios, banos, superficie, direccion, fecha\_publicacion, visitas



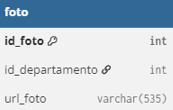
**Coordenadas**

* id\_coordenadas (PK)
* FK: id\_departamento → Departamento
* Campos: latitud, longitud



**Foto**

* id\_foto (PK)
* FK: id\_departamento → Departamento
* Campos: url\_foto



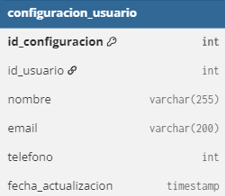
**Localidad**

* id\_localidad (PK)
* Campos: nombre, provincia, país



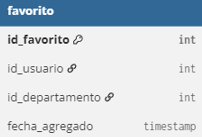
**Configuración de Usuario**

* id\_configuracion (PK)
* FK: id\_usuario → Usuario
* Campos: nombre, email, telefono, fecha\_actualizacion



**Favorito**

* id\_favorito (PK)
* FK: id\_usuario → Usuario
* FK: id\_departamento → Departamento
* Campo: fecha\_agregado



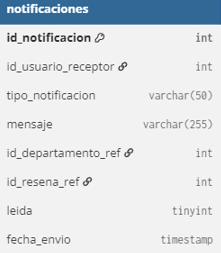
**Reseña**

* id\_resena (PK)
* FK: id\_departamento → Departamento
* FK: id\_usuario\_calificador → Usuario
* Campos: puntaje, comentario, fecha\_calificacion



**Notificaciones**

* id\_notificacion (PK)
* FK: id\_usuario\_receptor → Usuario
* FK opcional: id\_departamento\_ref → Departamento
* FK opcional: id\_resena\_ref → Reseña
* Campos: tipo\_notificacion, mensaje, leida, fecha\_envio



# Uso del Software

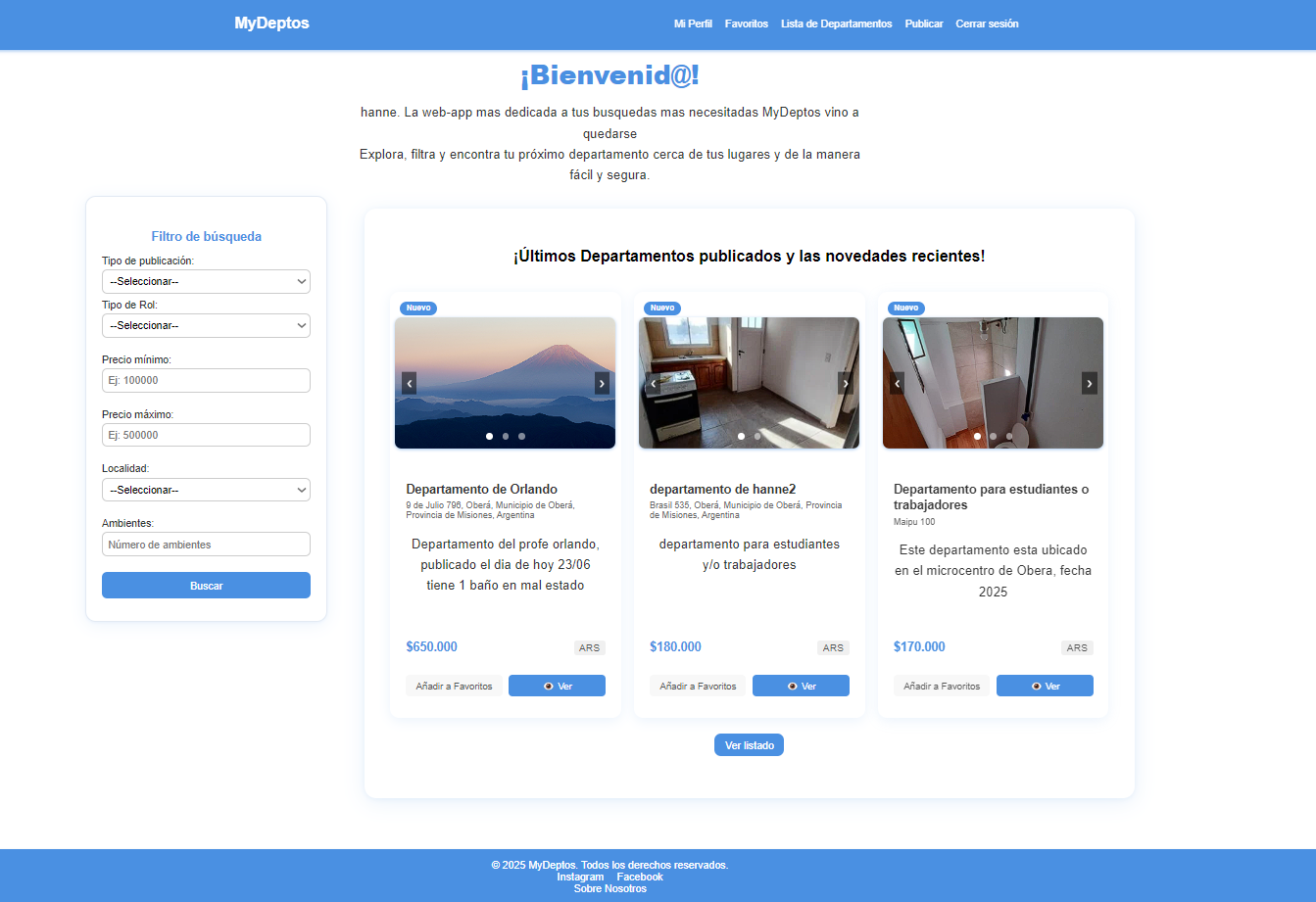
## Funcionalidades Básicas:

**Uso del Software – Manual de Usuario**

Esta sección describe las funcionalidades clave del sistema MyDeptos, explicando cómo navegar por la plataforma, realizar acciones comunes y aprovechar al máximo sus herramientas.

**Pantalla de Inicio (Home)**

Es la pantalla principal. Se accede al abrir la aplicación. También puedes volver desde cualquier pantalla utilizando el botón de inicio en la barra de navegación. Allí se muestran los **últimos departamentos publicados**, destacando sus fotos y datos principales, también se encuentra el filtro de búsqueda. Desde la Home se puede navegar a todas las funcionalidades clave.

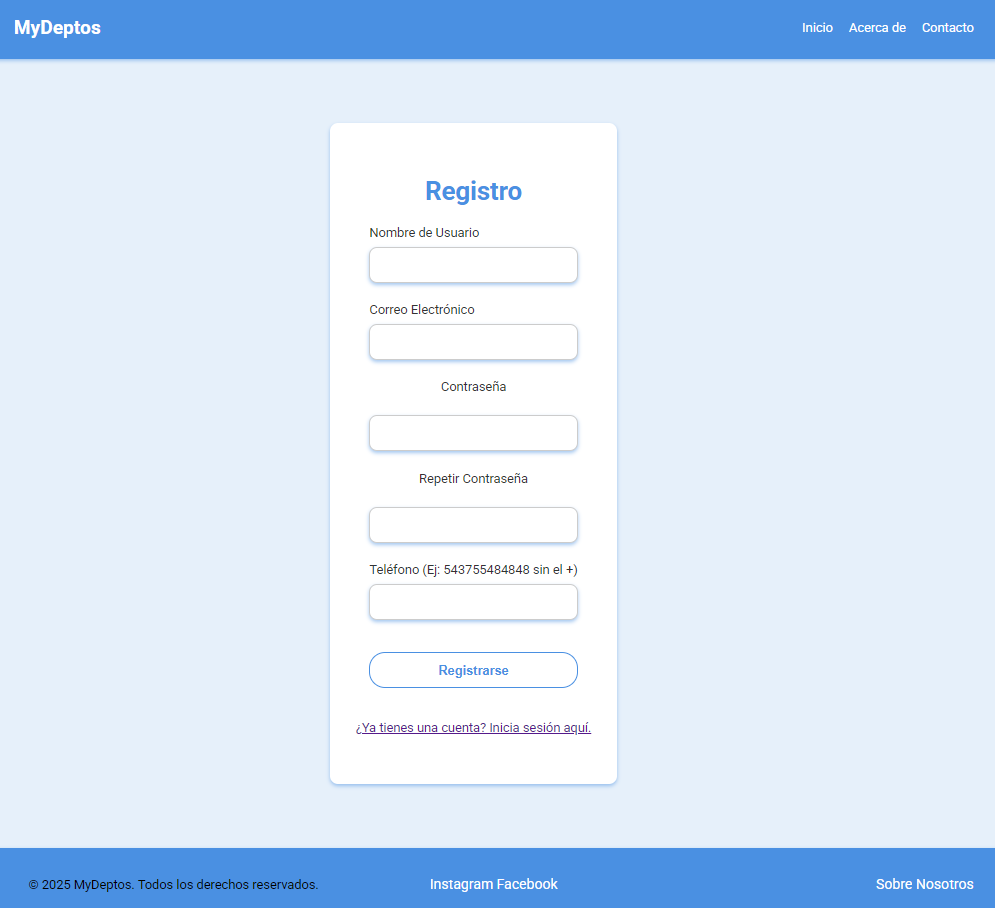


**Registro e Inicio de Sesión**

Se accede desde la Pantalla de Inicio (Home) seleccionando "Registrarse" o "Iniciar Sesión".

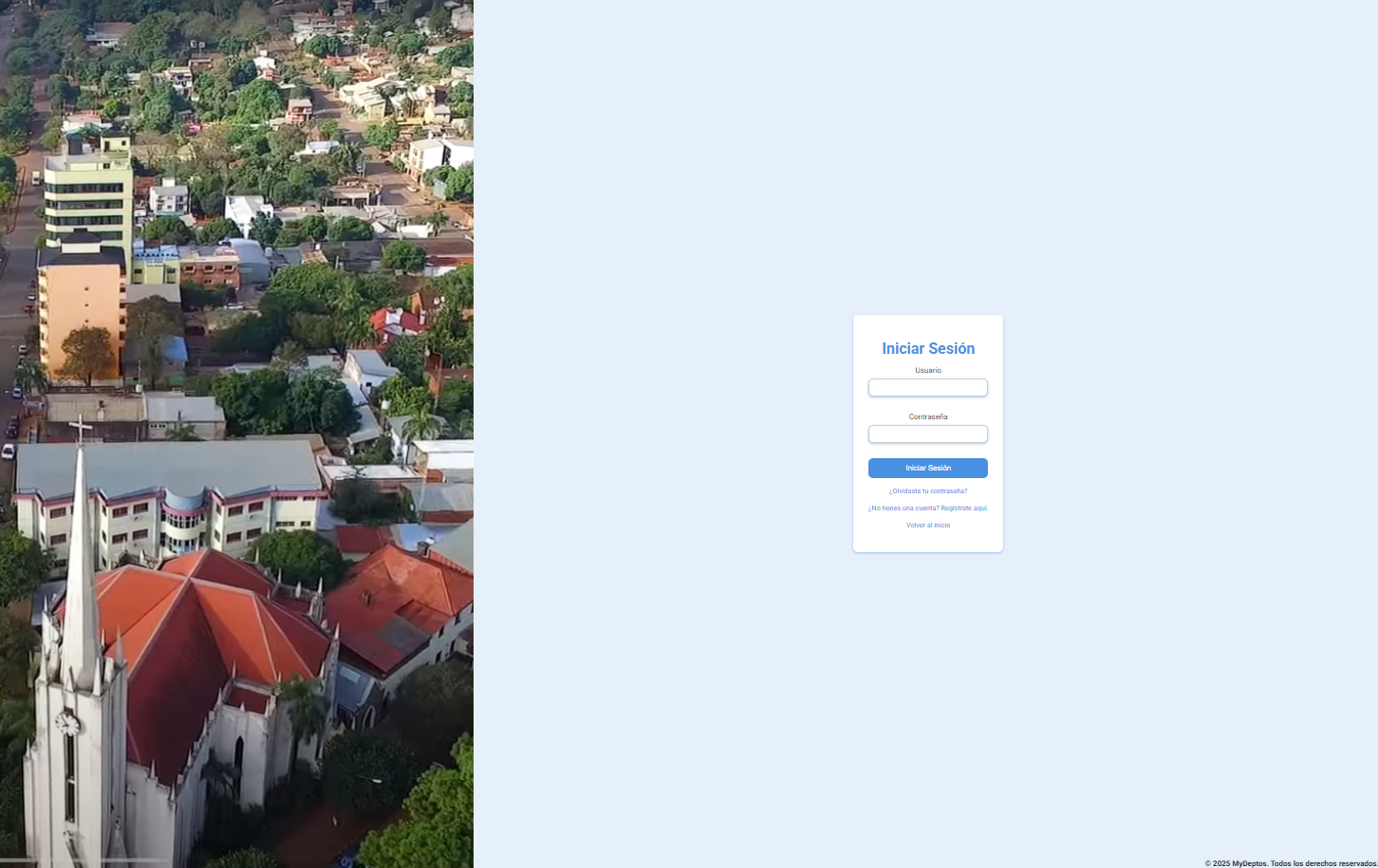
**Registrarse**

* Haz clic en "Registrarse" en el menú superior.
* Completa el formulario con nombre, email, contraseña y teléfono.
* Recibirás una confirmación del registro exitoso.



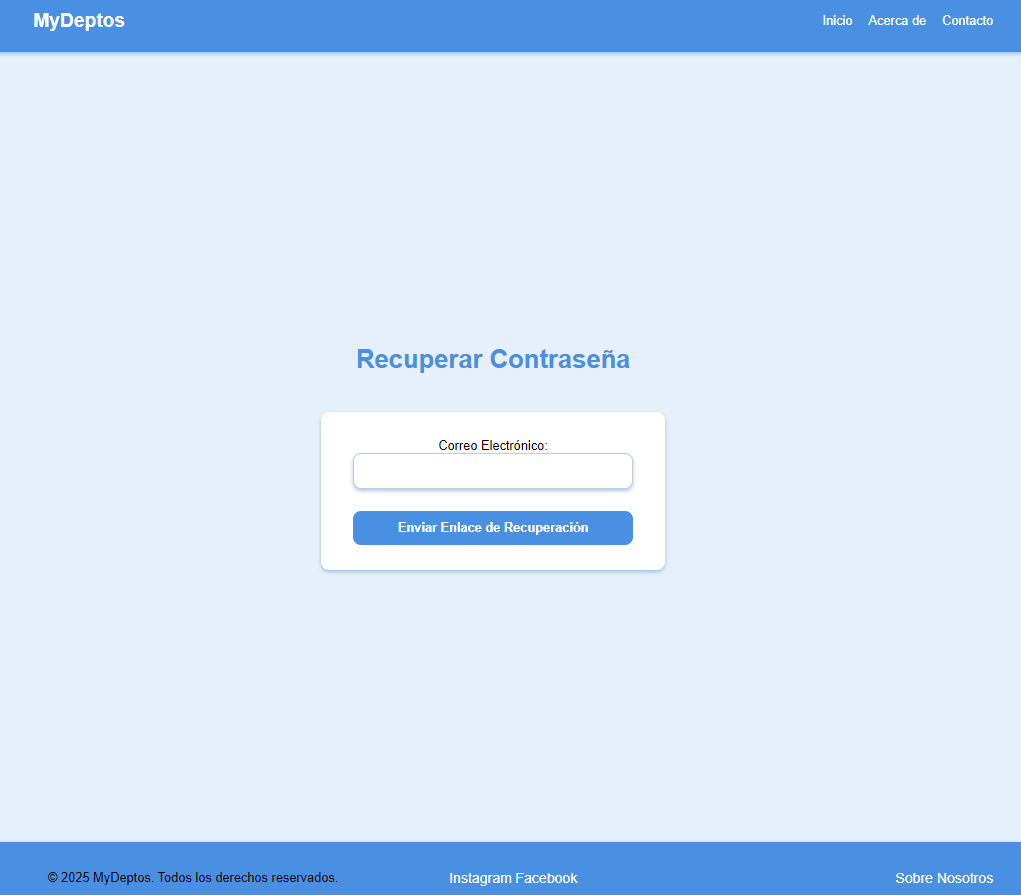
**Iniciar Sesión**

* Haz clic en "Iniciar Sesión" o accede a /login.
* Ingresá tu usuario/email y contraseña.
* Al iniciar sesión, se te redirige al Panel de Usuario.



**Recuperar Contraseña**

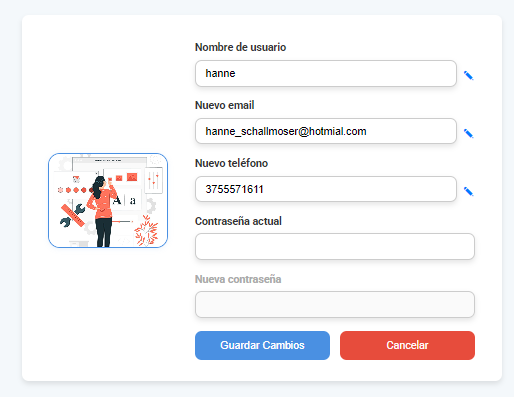
* Desde "Iniciar Sesión", haz clic en "¿Olvidaste tu contraseña?".
* Ingresá tu correo registrado.
* Recibirás un email con un enlace para restablecer la contraseña.



**Panel de Usuario y Gestión de Perfil**

**Editar Perfil**

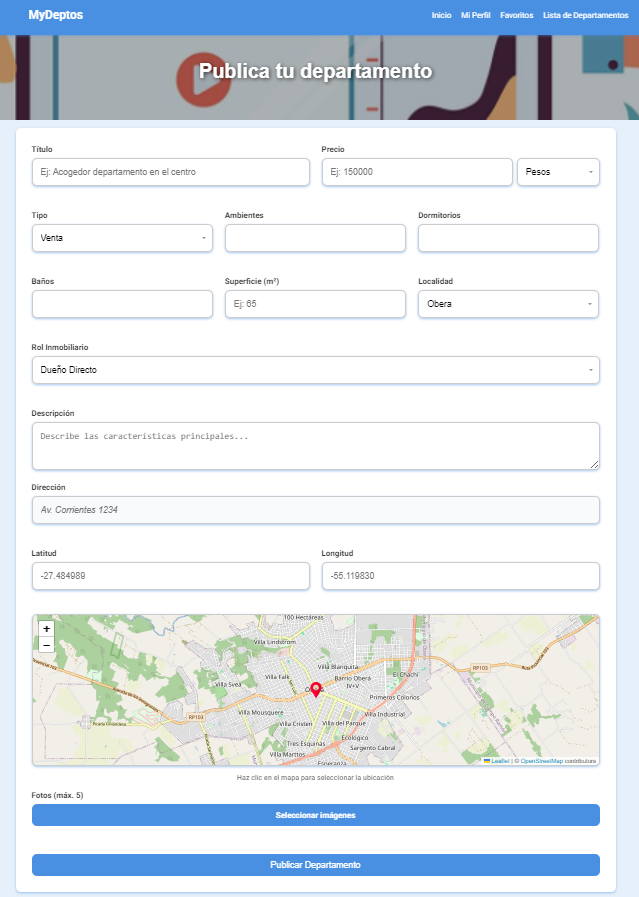
* Haz clic en “Mi perfil” desde la barra de navegación y selecciona "Editar Perfil".
* Podés modificar tu nombre, email, teléfono o contraseña.
* Guardá los cambios para actualizarlos.



**Gestión de Propiedades**

**Publicar Departamento**

* Para publicarlo debes ir al inicio de la pág. y en la barra de arriba buscar y hacer clic en "Publicar".
* Completá el formulario.
* Clic en "Publicar Departamento" una vez terminado de rellenar el formulario.



**Editar / Eliminar Departamento**

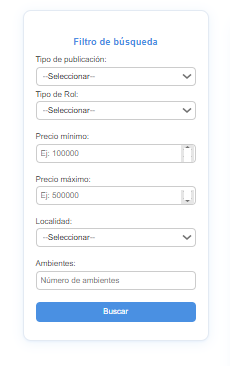
* Ve a “Mi perfil” desde la barra de navegación.
* Veras el listado de tus propiedades.
* Para editar, haz clic en "Editar", modificá y guardá.
* Para eliminar, haz clic en "Eliminar" y confirmá la acción.



**Búsqueda y Visualización**

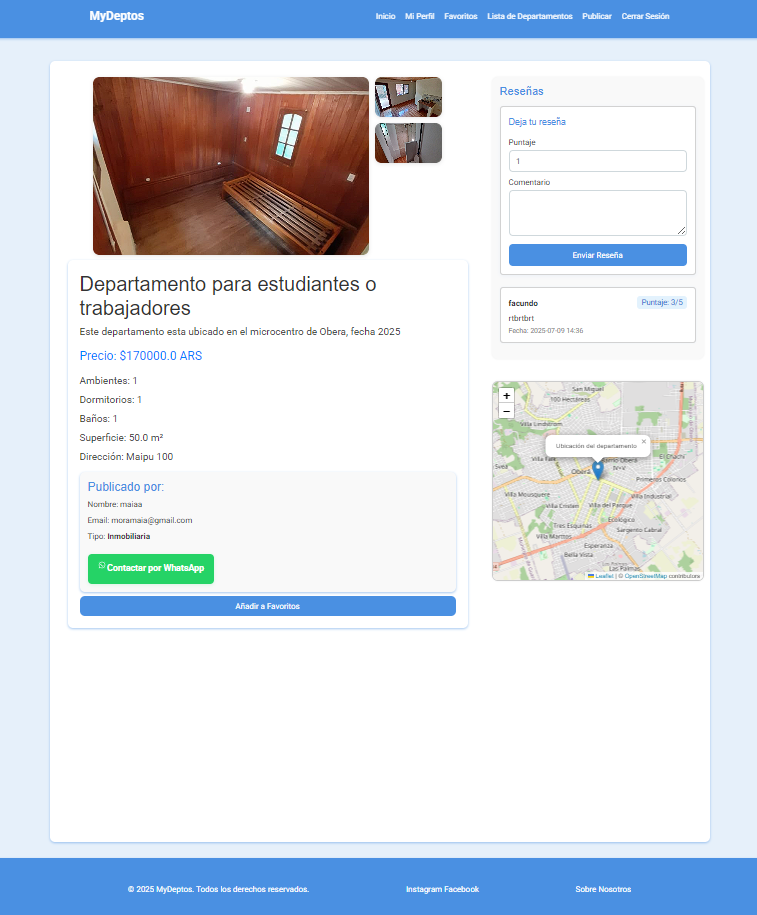
**Buscar Propiedades**

* Accedé desde el Home/Inicio
* Aplicá filtros: tipo de publicacion, precio, ambientes, localidad, etc.
* Una vez llenado el filtro de búsqueda y no tener algún error ya que si lo hubiese este arrojaría un fondo vacio. Hacer click en el botón “Buscar”.



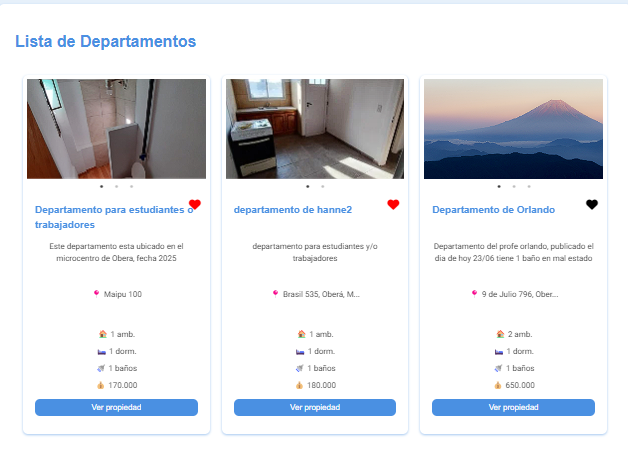
**Ver Detalles**

* Al hacer clic en “Ver propiedades” en la sección de “Lista de departamentos” o en “ver” en la sección de últimos departamentos de home podras ver los detalles del departamento.
* Podés ver todas las fotos, descripción, ubicación en mapa, datos de contacto y dejar una reseña si se quiere.



**Lista de Departamentos**

* Hacer click en “Lista de Departamentos de la barra de navegación o en “Ver listado” en la sección de últimos departamentos del home
* Podés ver todos los departamentos publicados.



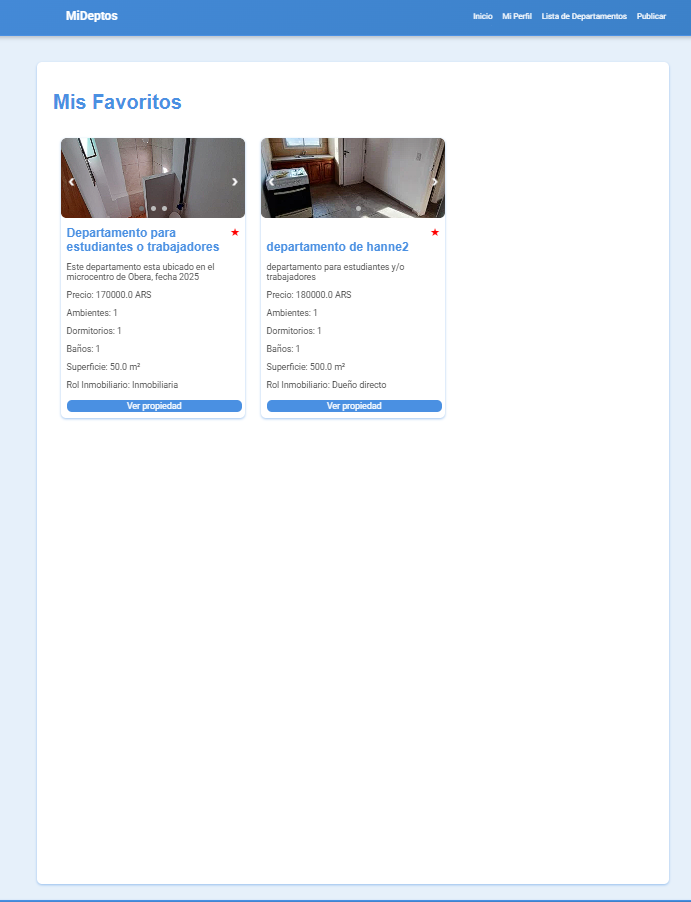
**Favoritos**

**Agregar a Favoritos**

* Si estás autenticado, hacé clic en el corazón o “Añadir a Favoritos” para marcar una propiedad como favorita.

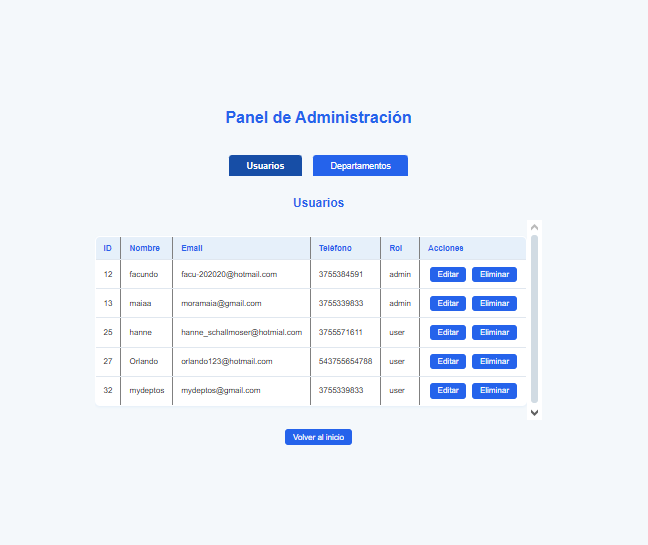
**Ver Favoritos**

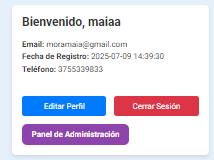
* Accedé desde la barra de navegación a "Favoritos".
* Allí verás tu lista personalizada de propiedades guardadas.



**Panel de Administradores**

* Inicia sesión con tu cuenta de administrador.
* Serás redirigido automáticamente al panel de administración. También puedes acceder desde “Mi perfil”, haciendo clic en el botón “Panel de admin”.
* Desde el panel puedes gestionar usuarios y departamentos: editar, eliminar y cambiar roles completando un ABM en su totalidad teniendo la gestión optima de un panel administrador.







## Flujo de trabajo típico

**1. Registro e inicio de sesión**

* El usuario se registra mediante el formulario de RegisterForm y sus datos se guardan en la tabla usuario.
* Luego inicia sesión con LoginForm, lo que crea una sesión y lo redirige a su panel.

**2. Publicación de departamentos**

* Desde el panel de usuario, accede a la ruta /publish\_depto y completa el formulario PublishDeptoForm.
* El sistema valida los datos, guarda el departamento en la tabla departamento, las fotos en foto, y la ubicación en coordenadas.

**3. Visualización y búsqueda**

* Los departamentos publicados se muestran en la página principal y en /listings, con filtros de búsqueda.
* El usuario puede ver detalles en /viewProperty/<id>, donde se muestran fotos, ubicación y datos del propietario.

**4. Favoritos y reseñas**

* El usuario puede agregar departamentos a favoritos (/add\_favorite) y verlos en /favorites.
* Puede dejar una reseña en un departamento ajeno (/departamento/<id>/add\_resena), editarla o eliminarla.

**5. Notificaciones**

* El sistema envía notificaciones al propietario cuando recibe una reseña, visibles en /notifications.

**6. Edición y eliminación**

* El usuario puede modificar o eliminar sus departamentos desde el panel (/modify\_depto/<id> y /delete\_publication/<id>).
* Puede editar su perfil en /edit\_profile.

**7. Panel de administración**

* El administrador accede a /admin\_panel para gestionar usuarios y departamentos, editando o eliminando registros según sea necesario.

Este flujo asegura que los usuarios puedan publicar, buscar, interactuar y gestionar sus datos de forma segura y ordenada.

# Solución de Problemas

## Errores Comunes

1. No puedo iniciar sesión

* Causa: Usuario o contraseña incorrectos, o el usuario no está registrado.
* Solución: Verifica que el usuario y la contraseña sean correctos. Si olvidaste la contraseña, usa la opción "¿Olvidaste tu contraseña?", para recuperarla por email.

1. No se envía el correo de recuperación de contraseña

* Causa: El correo ingresado no está registrado o hay un problema con la configuración del correo en `.env`.
* Solución: Asegúrate de que el correo esté registrado y que las variables `EMAIL\_SENDER` y `EMAIL\_PASSWORD` estén correctamente configuradas en el archivo `.env`.

1. No puedo publicar un departamento

* Causa: No has iniciado sesión, faltan campos obligatorios, no seleccionaste ubicación en el mapa o no subiste imágenes.
* Solución: Inicia sesión, completa todos los campos requeridos, selecciona la ubicación y sube entre 1 y 5 imágenes.

1. Error al subir imágenes

* Causa: Formato de archivo no permitido o intentas subir más de 5 imágenes.
* Solución: Solo se permiten imágenes JPG, PNG, WEBP o JPEG. Sube entre 1 y 5 imágenes.

1. No puedo editar o eliminar mi publicación

* Causa: No eres el propietario de la publicación o no has iniciado sesión.
* Solución: Inicia sesión con el usuario correcto. Solo puedes modificar o eliminar tus propias publicaciones.

1. No aparecen localidades en el formulario

* Causa: Problema de conexión con la base de datos o la tabla `localidad` está vacía.
* Solución: Verifica la conexión a la base de datos y que existan localidades cargadas.

## Preguntas Frecuentes (FAQ)

* ¿Cómo me registro en la plataforma?

Ve a la opción "Registrarse" en el menú principal, completa el formulario y sigue las instrucciones.

* ¿Puedo publicar un departamento sin iniciar sesión?

No. Debes iniciar sesión para poder publicar, modificar o eliminar departamentos.

* ¿Cómo recupero mi contraseña?

Haz clic en "¿Olvidaste tu contraseña?" en la pantalla de inicio de sesión, ingresa tu correo y sigue el enlace que recibirás por email.

* ¿Qué tipos de archivos puedo subir como fotos?

Solo se permiten imágenes en formato JPG, PNG, WEBP o JPEG.

* ¿Puedo modificar mis datos personales?

Sí, desde el panel de usuario accede a "Editar perfil" para cambiar tu nombre, email, teléfono o contraseña.

* ¿Por qué no veo mis publicaciones en la página principal?

La página principal muestra solo los últimos departamentos publicados. Puedes ver todas tus publicaciones en tu panel de usuario.

* ¿Cómo elimino una publicación?

Desde tu panel de usuario, selecciona la publicación y haz clic en "Eliminar". Confirma la acción para completar el proceso.

# Pruebas y Validación

## Ejecución de Tests

No se implementaron pruebas automáticas como tests unitarios o de integración.  
En su lugar, se realizaron pruebas manuales, usando la aplicación como un usuario común, para verificar que todas las funcionalidades funcionen correctamente:

* Registro e inicio de sesión
* Publicación y edición de departamentos
* Búsqueda con filtros
* Agregar y quitar favoritos
* Recuperación de contraseña

## Reporte de Bugs

No se utilizó un sistema automático para reportar errores como GitHub Issues.  
Los errores encontrados durante el desarrollo se detectaron y corrigieron de forma manual al ir usando y probando la aplicación.

# Referencias

## Enlaces a documentación oficial de herramientas/librerías usadas

<https://www.youtube.com/@FaztCode>

<https://www.youtube.com/@roelcode>

<https://www.youtube.com/@prettyprinted>

<https://www.zonaprop.com.ar>

<https://docs.python.org/es/3.13/tutorial/index.html>

https://www.w3schools.com/sql/

# Apendice

## Glosario de Términos Técnicos y del proyecto

* Flask: Framework web ligero para Python, utilizado para crear la aplicación web.
* MySQL: Sistema de gestión de bases de datos relacional utilizado para almacenar usuarios, departamentos, fotos, etc.
* CSRF: Cross-Site Request Forgery, ataque que se previene usando tokens en formularios (implementado con Flask-WTF).
* bcrypt: Algoritmo de hash seguro para almacenar contraseñas.
* Token: Cadena segura utilizada para validar acciones sensibles, como la recuperación de contraseña.
* LoginManager: Componente de Flask-Login que gestiona la autenticación de usuarios.
* Session: Objeto que almacena información temporal del usuario autenticado.
* Form (Formulario): Estructura para recolectar datos del usuario, validada con Flask-WTF y WTForms.
* Flash: Sistema de mensajes temporales para mostrar notificaciones al usuario.
* API: Interfaz de programación de aplicaciones; en este proyecto se usa para endpoints como favoritos y localidades.
* Render Template: Función de Flask para mostrar páginas HTML usando datos dinámicos.
* .env: Archivo de variables de entorno para credenciales y configuraciones sensibles.

Licencia del Software

Este proyecto está licenciado bajo la **Licencia MIT**.Esto significa que cualquier persona puede usar, copiar, modificar y distribuir el software, incluso con fines comerciales, siempre que mantenga el aviso de derechos de autor y la licencia original.

Podés ver el contenido completo de la licencia en el archivo llamado LICENSE, incluido en la raíz del proyecto.

Contacto

## Correo Autores

Barrios Schallmoser, Facundo – facuschall875@gmail.com

Mora, Maia Lujan – moramaia18@gmail.com

## Enlace a Repositorio de GitHub

https://github.com/Toukhay/MyDeptos-Origins.git

# Conclusión final

El desarrollo de **MyDeptosFinal** permitió construir una plataforma web robusta, segura e intuitiva para la publicación, búsqueda y gestión de departamentos en venta o alquiler. A lo largo del proyecto se implementaron funcionalidades clave como:

* Autenticación de usuarios
* Publicación con imágenes y ubicación en mapa
* Lista de favoritos
* Edición de perfil
* Recuperación de contraseña
* Filtros avanzados de búsqueda

Todo fue desarrollado utilizando tecnologías actuales como **Flask, MySQL, SQLAlchemy y Jinja2**, junto con diversas extensiones que mejoran la seguridad y la usabilidad.

Este sistema fue pensado como una solución web **moderna y adaptable**, que facilita tanto a propietarios como a interesados el proceso de encontrar o publicar propiedades.

Además, el sistema está diseñado para ser **escalable**, con posibilidad de incorporar futuras mejoras como:

* Integración con pasarelas de pago
* Sistema de mensajería interna entre usuarios
* Panel de analíticas para propietarios

## Agradecimientos

Agradecemos a los compañeros y usuarios que aportaron ideas, ayudaron a detectar errores y propusieron mejoras durante el desarrollo.  
También reconocemos el valor de la documentación oficial de las librerías utilizadas, así como el gran aporte de la comunidad de Python y Flask.

Agradecemos especialmente al **I.P.E.T. 1308**, y a su equipo directivo y docente, por el acompañamiento, la guía y el apoyo brindado a lo largo de la cursada, que hicieron posible la realización de este proyecto.