

Memoria del TFC

# Proyecto - Bienes Raíces

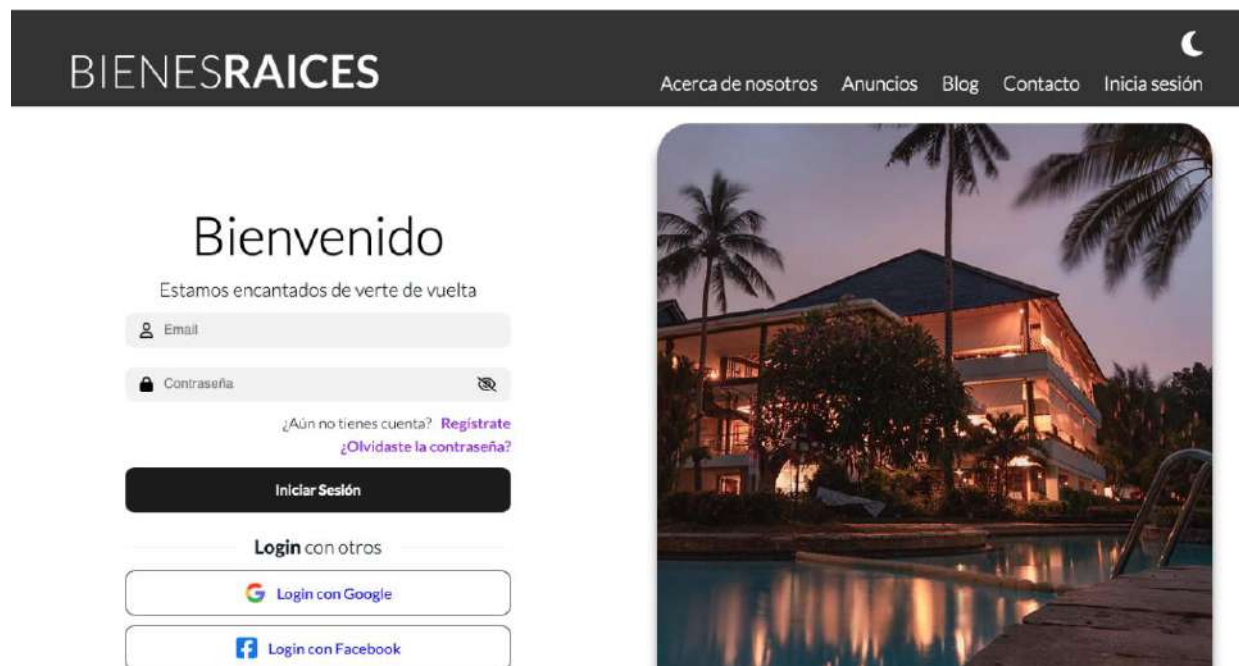
**Autor:** Cristian Eduardo Castro Vargas

**Tutor:** Mikola Sochynskyi Sochynskyi

**Ciclo:** Desarrollo de aplicaciones web

**Centro:** IES Enrique Tierno Galván

**Fecha:** 04/05/2024



# Índice

<b>1. Introducción.....</b>	<b>3</b>
1.1 Descripción.....	3
1.2 Motivación .....	3
<b>2. GESTIÓN Y ORGANIZACIÓN DEL PROYECTO .....</b>	<b>4</b>
2.1 Metodología de desarrollo .....	5
2.2 Planificación.....	5
2.3 Ciclo de vida.....	5
<b>3. OBJETIVOS .....</b>	<b>7</b>
<b>4. DESARROLLO DE LA APLICACIÓN.....</b>	<b>9</b>
4.1 Análisis: Requisitos .....	9
4.2 Diseño: Arquitectura .....	11
4.2.1 Explicación sobre el diseño de la web Bienes Raíces .....	11
4.2.2 Explicación sobre la base de datos de Bienes Raíces .....	15
4.3 Implementación: Decisiones adoptadas y problemas encontrados .....	21
<b>5. Realización de pruebas .....</b>	<b>24</b>
5.1 Descripción de las Pruebas.....	24
5.2 Resultados de las Pruebas .....	25
<b>6. Conclusiones .....</b>	<b>26</b>
<b>7. Trabajo Futuro .....</b>	<b>27</b>
7.1 Mejora de la Experiencia del Usuario (UX/UI).....	27
7.2 Funcionalidades Adicionales.....	27
7.3 Integración de Servicios Externos .....	27
7.4 Optimización del Rendimiento y la Escalabilidad .....	28
7.5 Seguridad y Conformidad .....	28
7.6 Automatización y DevOps .....	28
7.7 Expansión de Características del Chat en Vivo .....	29
<b>8. Bibliografía .....</b>	<b>30</b>

# 1. Introducción

## 1.1 Descripción

El proyecto de software de **bienes raíces** tiene como objetivo desarrollar una plataforma web dedicada exclusivamente a la venta de casas y apartamentos de lujo. Esta iniciativa surge en respuesta a la exclusiva demanda de propiedades de alta gama por parte de un segmento de la población con un poder adquisitivo significativo.

La plataforma se centrará en ofrecer una experiencia de usuario excepcional, proporcionando herramientas de visualización de propiedades exclusivas. Además, se integrarán funcionalidades específicas tanto para los **usuarios finales**, **usuario cliente potencial** en la compra de viviendas de lujo que tendrán ciertas funcionalidades como escribir una referencia, publicar un blog con una imagen y la posibilidad de participar en un chat en vivo con otros usuarios que estén conectados en la plataforma web, creando así una especie de comunidad.

Por otro lado funciones específicas para los **administradores**, quienes tendrán la capacidad de gestionar y mantener el inventario de propiedades de manera eficiente.

## 1.2 Motivación

La motivación principal detrás de este proyecto radica en la oportunidad de fusionar la pasión por el desarrollo de software con el interés en el mercado inmobiliario de alto nivel.

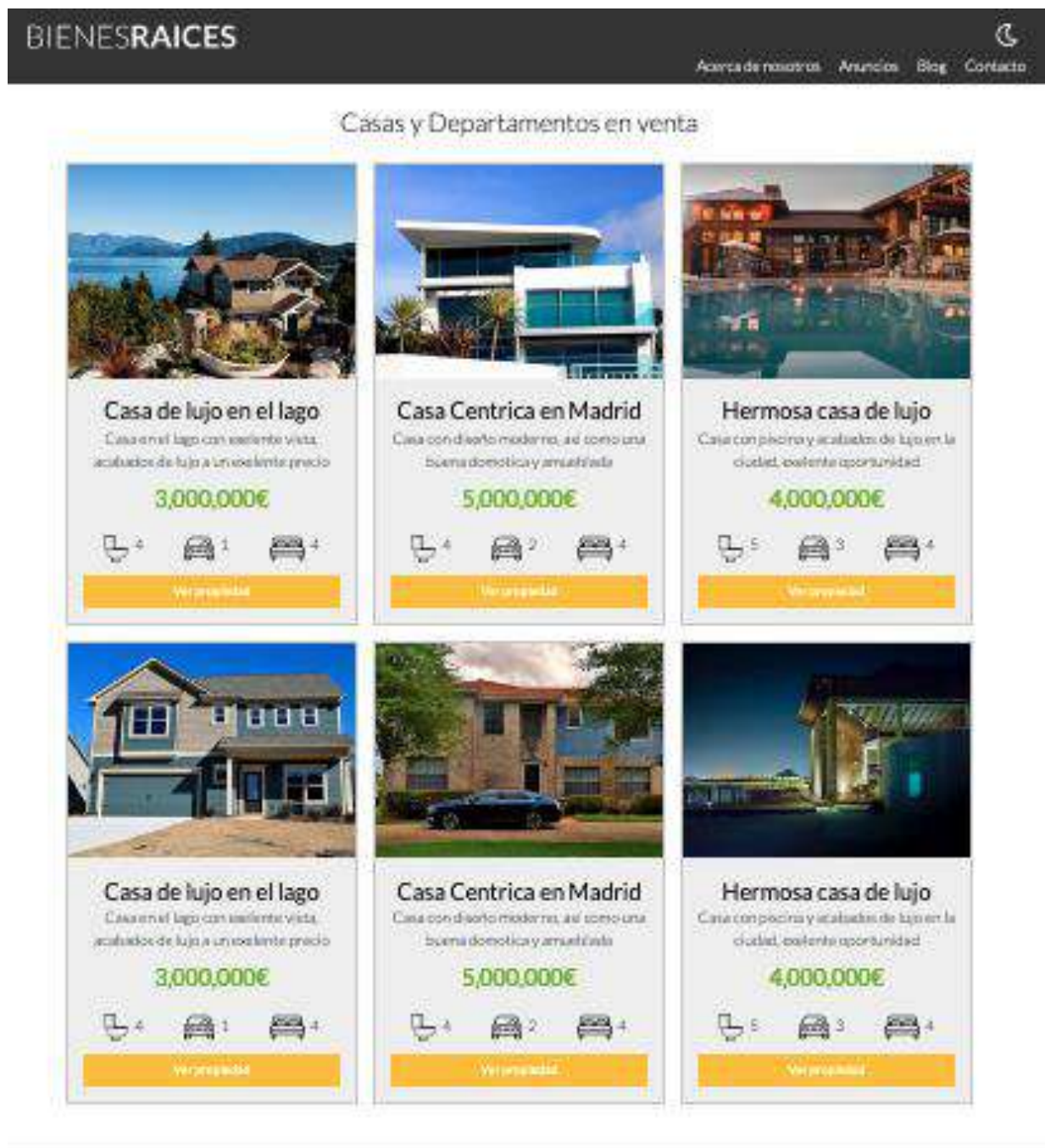
Se busca aprovechar esta iniciativa como una plataforma para poner en práctica y mejorar las habilidades de programación tanto en el **frontend** como en el **backend**.

Además, se reconoce la necesidad de ofrecer soluciones tecnológicas innovadoras en un sector que está experimentando un aumento en la brecha económica, con un crecimiento significativo en la riqueza de ciertos segmentos de la población.

Este proyecto se posiciona como una herramienta para facilitar el acceso a propiedades exclusivas y mejorar la experiencia de compra para aquellos con un poder adquisitivo destacado.

En resumen, el desarrollo de esta plataforma busca satisfacer las necesidades de un mercado específico, brindando una solución tecnológica que combine funcionalidad y elegancia para usuarios exigentes, al tiempo que ofrece un entorno de aprendizaje y desarrollo para mejorar las habilidades en el ámbito de la programación de software.

## 2. GESTIÓN Y ORGANIZACIÓN DEL PROYECTO



## 2.1 Metodología de desarrollo

Para el desarrollo del proyecto de bienes raíces, se empleó una **metodología ágil** combinada con prácticas de desarrollo de software tradicionales.

Se utilizó una adaptación de **Scrum**, aprovechando sus principios de entrega incremental y flexibilidad para responder a los cambios y hacer la aplicación **100% escalable**.

## 2.2 Planificación

La planificación inicial del proyecto se llevó a cabo mediante una evaluación detallada de los requisitos del cliente y la definición de las **funcionalidades clave**:

1. Poder exponer las propiedades en venta
2. Asignar vendedores a estas propiedades
3. Tener un registro de las ventas relacionadas con el cliente y el vendedor
4. Finalmente, también registrar a sus clientes

Se establecieron hitos y plazos para cada fase del desarrollo, desde el análisis de requisitos hasta la implementación y las pruebas realizadas.

Durante el desarrollo del proyecto, se llevaron a cabo ajustes menores en la planificación para adaptarse a los cambios en los requisitos del cliente y optimizar el proceso de desarrollo. Estos ajustes fueron proactivos y no afectaron significativamente los plazos de entrega, como por ejemplo, el cambio del sistema gestor de bases de datos de **MySQL** a **PostgreSQL** o modificaciones en la tabla de usuarios para añadir más datos de almacenamiento, como el número de teléfono móvil, lo que implicó alteraciones en el Framework **Django**.

## 2.3 Ciclo de vida

El ciclo de vida del proyecto se basó en un enfoque iterativo e incremental, con influencias del modelo en espiral. Se realizó una planificación detallada de cada iteración, identificando las funcionalidades a desarrollar y los objetivos a alcanzar.

El desarrollo del proyecto se llevó a cabo utilizando diversas tecnologías en distintas áreas:

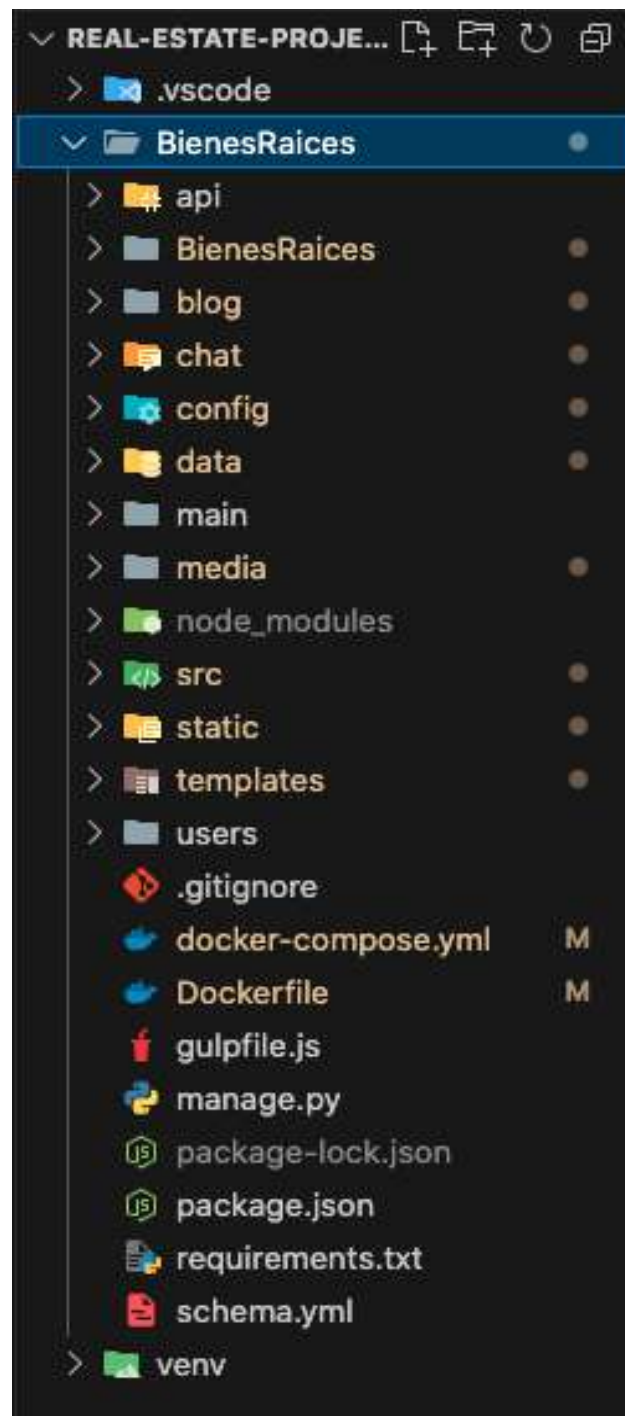
- **Backend:** Se utilizó **Python** como lenguaje de programación con el framework **Django** para la lógica del servidor y la gestión de la base de datos con **PostgreSQL**.
- **Frontend:** Se emplearon tecnologías como **JavaScript**, **HTML** y **CSS** para el desarrollo de la interfaz de usuario, con la incorporación de herramientas como **Ajax** para realizar solicitudes asíncronas y **Gulp** y **Sass** para la automatización de tareas y la mejora de la eficiencia en el desarrollo.
- **Infraestructura:** Se hizo uso de **Docker** para la creación de contenedores y **AWS** para el despliegue y la gestión de la infraestructura en la nube.

Durante el proceso, se adaptaron y ampliaron las funcionalidades de la base de datos, incluyendo la implementación de un **sistema de inicio de sesión**, la **publicación de blogs** por

parte de los usuarios registrados o los administradores, una **sección de reseñas** y la posible integración de un **chat en vivo** para funciones asíncronas.

Los plazos de entrega se cumplieron satisfactoriamente, gracias a una gestión eficiente del tiempo y los recursos, así como a la capacidad para abordar cambios y desafíos durante el desarrollo.

## Estructura del código fuente del proyecto





### 3. OBJETIVOS

Uno de los primeros objetivos, y ciertamente uno de los más desafiantes, fue completar el proyecto utilizando el **stack tecnológico** especificado por la empresa de prácticas.

En este aspecto, se lograron alcanzar la gran mayoría de los objetivos, especialmente en términos de aprendizaje y fluidez en el trabajo.

A pesar de esto, se enfrentaron algunos contratiempos debido a la falta de recursos, los cuales fueron abordados de manera proactiva para minimizar su impacto en el avance del proyecto.

Se logró implementar con éxito todas las funcionalidades de la aplicación web, abarcando las necesidades de distintos tipos de usuarios:

- **Para el usuario final:** Se garantizó una experiencia de usuario óptima, permitiendo la exploración intuitiva del sitio web. Esto incluyó la navegación por diferentes secciones en la barra de navegación, como la sección
  - Acerca de la empresa
  - La visualización de blogs publicados por la comunidad
  - La consulta de propiedades en venta
  - Una sección de contacto y la capacidad de iniciar sesión.
- **Para el usuario cliente:** Se implementó un sistema de inicio de sesión que permitió a los clientes acceder a funcionalidades adicionales, como la publicación de blogs y reseñas, así como la integración de un posible chat en vivo para facilitar la comunicación entre la comunidad.
- **Para el usuario administrativo:** Se facilitó la gestión eficiente de propiedades, tanto las propias como las de sus clientes, proporcionando herramientas intuitivas para la administración de la plataforma.

Además, se creó una **API RESTful** utilizando **Django Rest Framework** para gestionar las propiedades en venta, permitiendo una interacción más dinámica y eficiente con los datos de las propiedades.

Esta API permite realizar operaciones **CRUD (Crear, Leer, Actualizar, Eliminar)** sobre las propiedades, facilitando la integración con otras aplicaciones y servicios que puedan requerir acceso a esta información.

Estas implementaciones no solo cumplieron con los requisitos iniciales del proyecto, sino que también proporcionaron una base sólida para futuras expansiones y mejoras de la plataforma.



## Bienvenido

Estamos encantados de verte de vuelta

Email

Contraseña



¿Aún no tienes cuenta? [Regístrate](#)

¿Olvidaste la contraseña?

Iniciar Sesión

Login con otros



Login con Google



Login con Facebook



## Registrar un nuevo Blog

Nombre del titular

diantium

Título del blog \*

Imagen JPG \* obligatorio

Examinar...

No se ha se...n archivo.

Imagen WEBP

Examinar...

No se ha se...n archivo.

Descripción \*

Fecha

03 / 06 / 2024



Contenido \*





## 4. DESARROLLO DE LA APLICACIÓN

### 4.1 Análisis: Requisitos

El proyecto de Bienes Raíces se centra en desarrollar una plataforma web para la venta de propiedades de lujo. Los requisitos clave incluyen:

#### ❖ Usuarios Finales:

- **Navegación intuitiva:** Facilitar la exploración del sitio web.
- **Visualización de propiedades:** Mostrar detalles completos de las propiedades.
- **Visualización de reseñas:** Los usuarios finales podrán leer las reseñas.
- **Blogs (por parte de la comunidad):** Lectura de blogs.
- **Experiencia agradable del sitio web UI/UX:** Diseño atractivo y funcional.

#### ❖ Clientes (Compradores Potenciales):

- **Inicio de sesión:** Acceso seguro a la plataforma.
- **Publicación de blogs y reseñas:** Contribuir contenido y opiniones.
- **Interacción a través del chat en vivo:** Comunicación en tiempo real con los demás usuarios conectados.

#### ❖ Administradores:

- **Gestión de propiedades:** Añadir, modificar y eliminar propiedades.
- **Gestión de clientes:** Administrar información de los clientes.
- **Gestión de vendedores:** Administrar información de los vendedores.

#### ❖ Base de Datos:

La base de datos debe gestionar información sobre ventas, clientes, propiedades y vendedores, con tablas específicas para cada uno. Con la evolución del proyecto, la base de datos ha crecido para incluir nuevas funcionalidades, tales como blogs, mensajes de chat en vivo y reseñas:

- **main\_ventas:** Almacena detalles de las ventas.
- **main\_clientes:** Almacena información de los clientes.
- **main\_propiedades:** Almacena detalles de las propiedades.
- **main\_vendedores:** Almacena información de los vendedores.
- **main\_propiedad\_vendedor:** Asocia propiedades con vendedores.
- **blog\_blog:** Almacena blogs publicados por los usuarios.
- **chat\_room:** Almacena información de las salas de chat.
- **chat\_message:** Almacena mensajes enviados en las salas de chat.
- **chat\_room\_users:** Relaciona usuarios con las salas de chat.
- **users\_user:** Almacena información detallada de los usuarios del sistema.
- **users\_testimoniales:** Almacena información de las reseñas

## ❖ API RESTful:

Se ha creado una **API RESTful** utilizando **Django Rest Framework** para gestionar las propiedades en venta.

Esta API permite realizar operaciones **CRUD (Crear, Leer, Actualizar, Eliminar)** sobre las propiedades, facilitando la integración con otras aplicaciones y servicios que puedan requerir acceso a esta información.

Estas implementaciones no solo cumplieron con los requisitos iniciales del proyecto, sino que también proporcionaron una base sólida para futuras expansiones y mejoras de la plataforma.

The image shows a REST client interface with two panels. The left panel lists API endpoints for the 'v1' version, including 'v1\_propiedades\_list' (GET), 'v1\_propiedades\_create' (POST), 'v1\_propiedades\_retrieve' (GET), 'v1\_propiedades\_update' (PUT), 'v1\_propiedades\_partial\_u...' (PATCH), and 'v1\_propiedades\_destroy' (DEL). The right panel displays the details for the 'v1\_propiedades\_list' endpoint, showing a 200 status code and a response schema of 'application/json'. The schema is an array of objects with the following fields: 'id' (integer, required), 'titulo' (string, required, <= 45 characters), 'precio' (string <decimal> ^-?\d{0,8}(?\.\d{0,2})?\$, required), 'img' (string <uri>, required), 'descripcion' (string, required, <= 260 characters), 'contenido' (string, required), and 'habitaciones' (integer, required, range [-2147483648, 2147483647]). A 'Response samples' section shows a sample JSON response for a 200 status code, with fields: 'id' (0), 'titulo' (string), 'precio' (string), 'img' (http://example.com), 'descripcion' (string), 'contenido' (string), 'habitaciones' (-2147483648), 'wc' (-2147483648), 'garaje' (-2147483648), 'fecha\_alta' (2019-08-24), and 'vendido' (true).

Search...

v1

GET v1\_propiedades\_list

POST v1\_propiedades\_create

GET v1\_propiedades\_retrieve

PUT v1\_propiedades\_update

PATCH v1\_propiedades\_partial\_u...

DEL v1\_propiedades\_destroy

API docs by Redocly

v1\_propiedades\_list

AUTHORIZATIONS: > cookieAuth or basicAuth or None

Responses

200

RESPONSE SCHEMA: application/json

Array [

- id required integer
- titulo required string <= 45 characters
- precio required string <decimal> ^-?\d{0,8}(?\.\d{0,2})?\$
- img required string <uri>
- descripcion required string <= 260 characters
- contenido required string
- habitaciones required integer [-2147483648, 2147483647]

Response samples

200

Content type application/json

Copy Expand all Collapse all

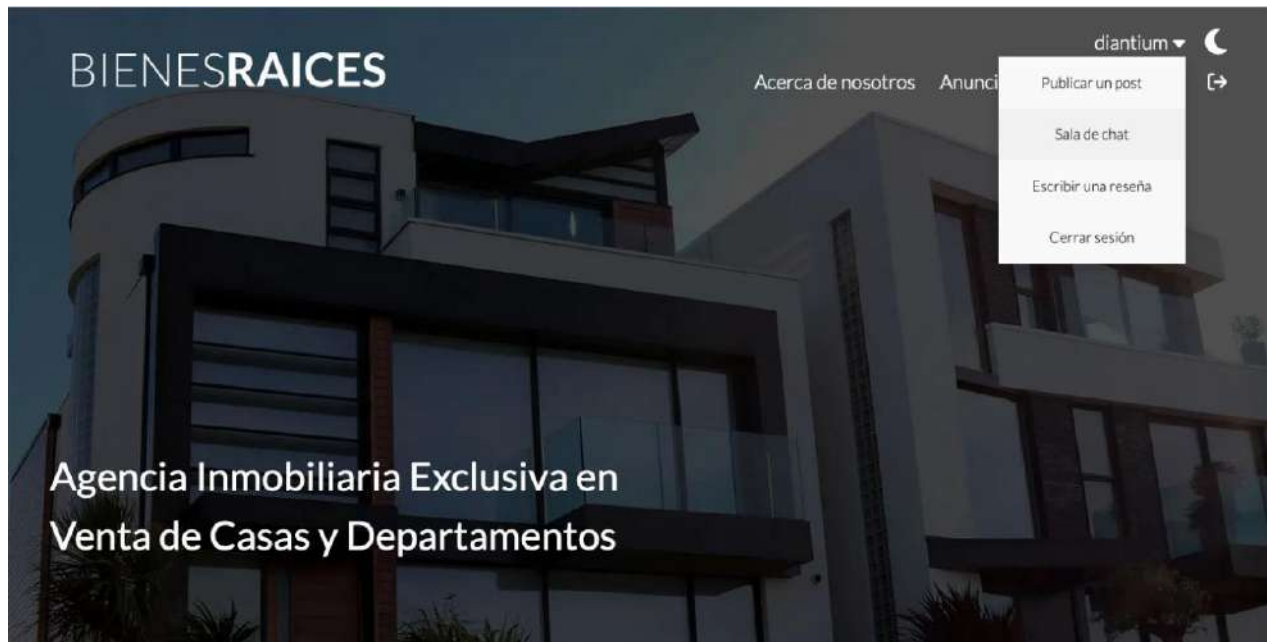
```
[
  - {
    "id": 0,
    "titulo": "string",
    "precio": "~?\\(?:\\.\\)?$",
    "img": "http://example.com",
    "descripcion": "string",
    "contenido": "string",
    "habitaciones": -2147483648,
    "wc": -2147483648,
    "garaje": -2147483648,
    "fecha_alta": "2019-08-24",
    "vendido": true
  }
]
```

## 4.2 Diseño: Arquitectura

### 4.2.1 Explicación sobre el diseño de la web Bienes Raíces

#### Diseño general

La página web presenta un diseño moderno, minimalista y profesional, con una paleta de colores equilibrada y atractiva.



El diseño de la plataforma sigue una arquitectura basada en capas, separando claramente la lógica del negocio, la presentación y el acceso a datos:

- **Frontend:** Desarrollado con **HTML**, **CSS**, **JavaScript**, y herramientas como **Gulp** y **Sass** para la automatización y la optimización del código. Se utilizó **Ajax** para las comunicaciones asíncronas entre el servidor y el cliente.
- **Backend:** Implementado con **Python** y el framework **Django**, gestionando la lógica del negocio y las operaciones con la base de datos.
- **Base de datos:** Inicialmente se utilizó **MySQL**, pero en producción se cambió a **PostgreSQL** para mejorar el rendimiento y la escalabilidad.
- **Infraestructura:** **Docker** para la contenedorización de aplicaciones y **AWS** para el despliegue en la nube.

## Estructura y organización:

### 1. Encabezado:

- **Logotipo:** Representa la identidad visual de la empresa.
- **Menú de navegación:** Acceso a secciones como:
  - Acerca de nosotros
  - Anuncios
  - Blog
  - Contacto
  - Botón dark mode (Opción para cambiar a modo oscuro).

### 2. Cuerpo:

Ocupa la mayor parte de la pantalla y presenta el contenido principal de la página web, incluyendo:

- **Banner principal:** A lo largo del ancho de la página, muestra una pequeña información acerca de la empresa inmobiliaria.
- **Tarjeta de propiedad:** Muestra 3 propiedades en venta, cada propiedad se presenta en una tarjeta individual, la cual incluye:
  - **Imagen principal:** Una fotografía atractiva de la propiedad, que permite a los usuarios visualizar su aspecto exterior.
  - **Título:** El nombre con una breve descripción de la propiedad, indicando su ubicación.
  - **Precio:** El precio de la propiedad, expresado en la moneda local “euro”.
  - **Características básicas:** Un resumen de las características principales de la propiedad, como número de habitaciones, baños y número de garajes.
  - **Botón de acción:** Un botón que permite a los usuarios realizar una acción específica, como ver más información sobre la propiedad
- **Secciones de contenido:** Organizadas en bloques claramente diferenciados, presentan información sobre
  - Contacto con la empresa
  - Un sector de blog (Muestra 2 tarjetas)
  - Algunos testimonios (Manejado con JS).



Mas sobre Nosotros



SEGURIDAD

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Neque facilis tristique, a mihi purus ultricies nunc accumsan molestie etiam habitant eros curae, orci aliquam at sociosque fames convallis pharetra ut eleferend.



PRECIO


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Neque facilis tristique, a mihi purus ultricies nunc accumsan molestie etiam habitant eros curae, orci aliquam at sociosque fames convallis pharetra ut eleferend.



A TIEMPO

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Neque facilis tristique, a mihi purus ultricies nunc accumsan molestie etiam habitant eros curae, orci aliquam at sociosque fames convallis pharetra ut eleferend.

Casas y Departamentos en venta




**Mansión Serenidad**  
Una majestuosa mansión ubicada en un entorno natural, ofreciendo paz y tranquilidad junto con comodidades de primer nivel.

5000000.00€

4 3 5

Ver propiedad




**Casa Centrica en Madrid**  
Hermosa casa urbana con vistas panorámicas impresionantes, combinando elegancia y comodidad en cada detalle.

2000000.00€

3 2 4

Ver propiedad



**Palacio Moderno**  
Un palacio moderno que redefine el lujo contemporáneo, con espacios amplios y acabados de alta gama en cada rincón.

6000000.00€

4 3 6

Ver propiedad

Ver todas



Nuestro Blog



**Terraza en el techo de tu casa**  
Escrito el: 25/03/2024 por: Admin  
Consejos para construir una terraza en el techo de tu casa con los mejores materiales.



**Guía para la decoración de tu hogar**  
Escrito el: 25/03/2024 por: Admin  
Maximiza el espacio en tu hogar con esta guía, aprende a combinar muebles y colores para darle vida a tu espacio.

Testimoniales

“ El personal se comportó de una excelente forma, muy buena atención y la casa que me ofrecieron cumple con todas mis expectativas. ”

- Cristian Castro

### 3. Pie de página:

- Ubicado en la parte inferior de la pantalla, incluye información de copyright, enlaces a la zona de
  - Acerca de nosotros
  - Anuncios
  - Blog
  - Contacto



#### Guía para la decoración de tu hogar

Escrito el: 25/03/2024 por: Admin

Maximiza el espacio en tu hogar con esta guía, aprende a combinar muebles y colores para darle vida a tu espacio

cumplir con todas mis expectativas.

- Cristian Castro

© Todos los derechos reservados 2024

[Acerca de nosotros](#) [Anuncios](#) [Blog](#) [Contacto](#)



## 4.2.2 Explicación sobre la base de datos de Bienes Raíces

### Evolución de la Base de datos

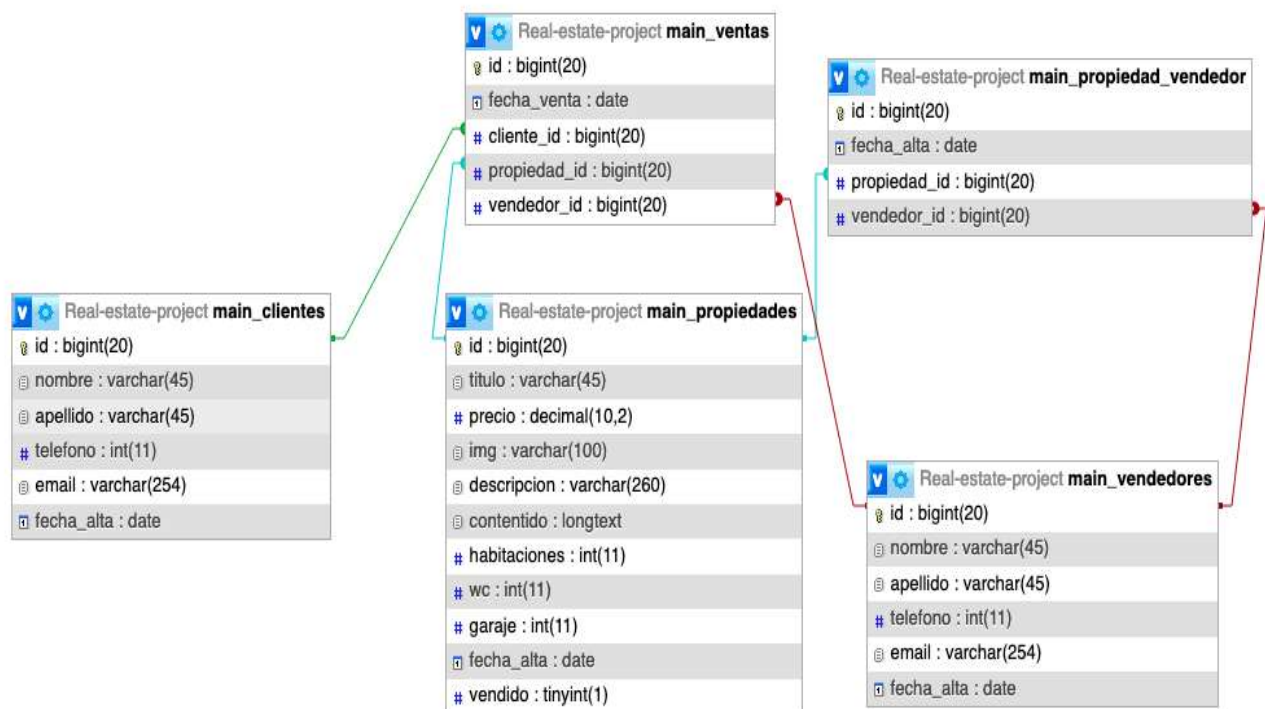
#### Evolución 1

La base de datos está diseñada para almacenar información sobre las ventas de propiedades inmobiliarias.

La tabla **main\_ventas** contiene la información principal de cada venta, como la fecha de venta, el cliente que compró la propiedad, la propiedad vendida y el vendedor que realizó la venta.

Las tablas **main\_clientes**, **main\_propiedades** y **main\_vendedores** contienen información detallada sobre cada cliente, propiedad y vendedor, respectivamente.

La tabla **main\_propiedad\_vendedor** se utiliza para asociar las propiedades con los vendedores.



## Relaciones entre las tablas en la base de datos

La base de datos que se muestra en la imagen está diseñada para gestionar información sobre el proyecto inmobiliario **Bienes Raíces**.

**Consta de cinco tablas principales:**

### **1. main\_ventas:**

- **id:** Identificador único de cada venta.
- **fecha\_venta:** Fecha en la que se realizó la venta.
- **cliente\_id:** Identificador del cliente que compró la propiedad.
- **propiedad\_id:** Identificador de la propiedad vendida.
- **vendedor\_id:** Identificador del vendedor que realizó la venta.

### **2. main\_clientes:**

- **id:** Identificador único de cada cliente.
- **nombre:** Nombre del cliente.
- **apellido:** Apellido del cliente.
- **telefono:** Número de teléfono del cliente.
- **email:** Correo electrónico del cliente.
- **fecha\_alta:** Fecha en la que se dio de alta al cliente en la base de datos.

### **3. main\_propiedades:**

- **id:** Identificador único de cada propiedad.
- **titulo:** Título de la propiedad.
- **precio:** Precio de la propiedad.
- **img:** Ruta de la imagen de la propiedad.
- **descripcion:** Descripción de la propiedad.
- **contenido:** Una descripción más amplia de la propiedad.
- **habitaciones:** Número de habitaciones de la propiedad.
- **wc:** Número de baños de la propiedad.
- **garaje:** Número de garajes de la propiedad.
- **fecha\_alta:** Fecha en la que se dio de alta la propiedad en la base de datos.
- **vendido:** Indica si la propiedad está vendida (1) o no (0).

### **4. main\_vendedores:**

- **id:** Identificador único de cada vendedor.
- **nombre:** Nombre del vendedor.
- **apellido:** Apellido del vendedor.
- **telefono:** Número de teléfono del vendedor.
- **email:** Correo electrónico del vendedor.
- **fecha\_alta:** Fecha en la que se dio de alta al vendedor en la base de datos.

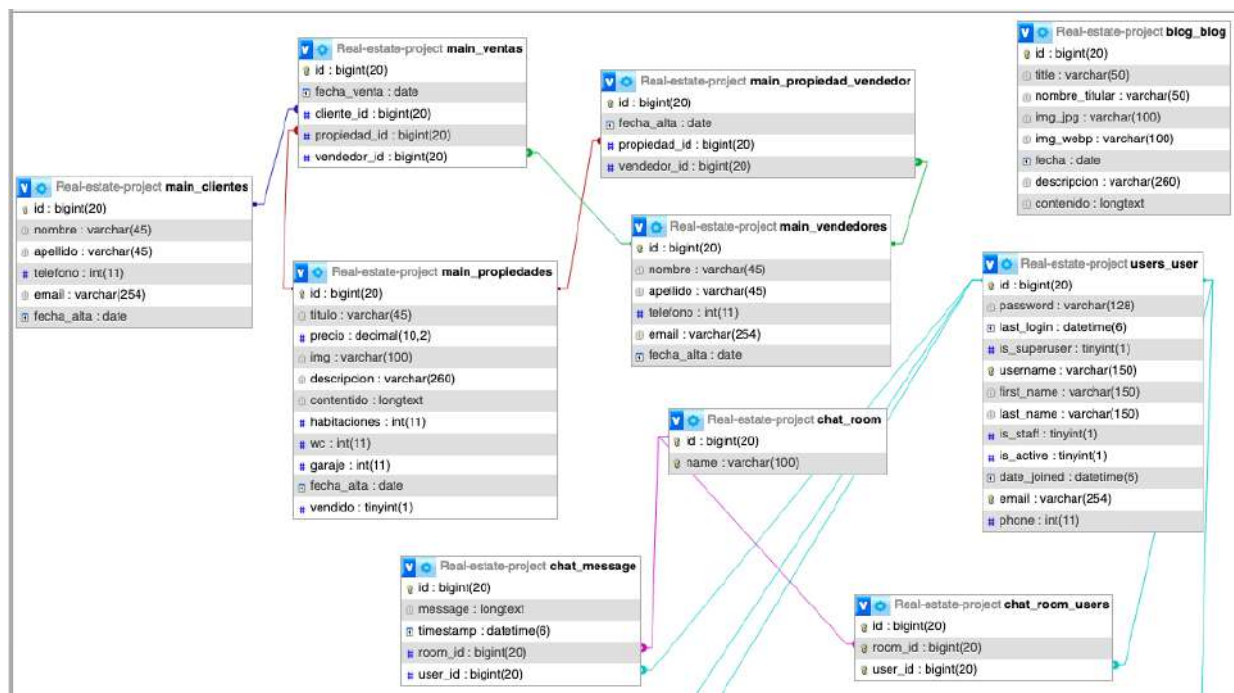
## 5. main\_propiedad\_vendedor:

- **id:** Identificador único de cada asignación de un vendedor con una propiedad.
- **fecha\_alta:** Fecha en la que se dio de alta al vendedor relacionado con una propiedad en la base de datos.
- **propiedad\_id:** Identificador de la propiedad.
- **vendedor\_id:** Identificador del vendedor.

## Evolución 2

La base de datos evolucionó para soportar funcionalidades adicionales, como la **publicación de blogs** y la implementación de un sistema de **chat en vivo**.

Esto incluyó la adición de nuevas tablas y la modificación de las existentes para gestionar las nuevas relaciones y datos requeridos.



Nuevas tablas y relaciones:

## 6. blog\_blog:

- **id:** Identificador único del blog.
- **title:** Título del blog.
- **nombre\_titular:** Nombre del autor del blog.
- **img\_jpg:** Ruta de la imagen del blog en formato JPG.
- **img\_webp:** Ruta de la imagen del blog en formato WEBP.
- **fecha:** Fecha de publicación del blog.
- **descripcion:** Descripción breve del blog.
- **contenido:** Contenido detallado del blog.

## 7. users\_user:

- **id**: Identificador único del usuario.
- **password**: Contraseña del usuario debidamente cifrada.
- **last\_login**: Fecha y hora del último inicio de sesión.
- **is\_superuser**: Indica si el usuario es superusuario (1) o no (0).
- **username**: Nombre de usuario.
- **first\_name**: Primer nombre del usuario.
- **last\_name**: Apellido del usuario.
- **is\_staff**: Indica si el usuario es parte del personal (1) o no (0).
- **is\_active**: Indica si el usuario está activo (1) o no (0).
- **date\_joined**: Fecha de registro del usuario.
- **email**: Correo electrónico del usuario.
- **phone**: Número de teléfono del usuario.

## 8. chat\_room:

- **id**: Identificador único de la sala de chat.
- **name**: Nombre de la sala de chat.

## 9. chat\_message:

- **id**: Identificador único del mensaje.
- **message**: Contenido del mensaje.
- **timestamp**: Fecha y hora del mensaje.
- **room\_id**: Identificador de la sala de chat.
- **user\_id**: Identificador del usuario que envió el mensaje.

## 10. chat\_room\_users:

- **id**: Identificador único de la relación entre sala de chat y usuario.
- **room\_id**: Identificador de la sala de chat.
- **user\_id**: Identificador del usuario.

Esta evolución permitió la incorporación de funcionalidades avanzadas, como la interacción en tiempo real a través del chat en vivo, y la gestión de contenido dinámico mediante la publicación de blogs.

Estas mejoras hicieron la aplicación más robusta y funcional, respondiendo mejor a las necesidades tanto de los usuarios finales como de los administradores.

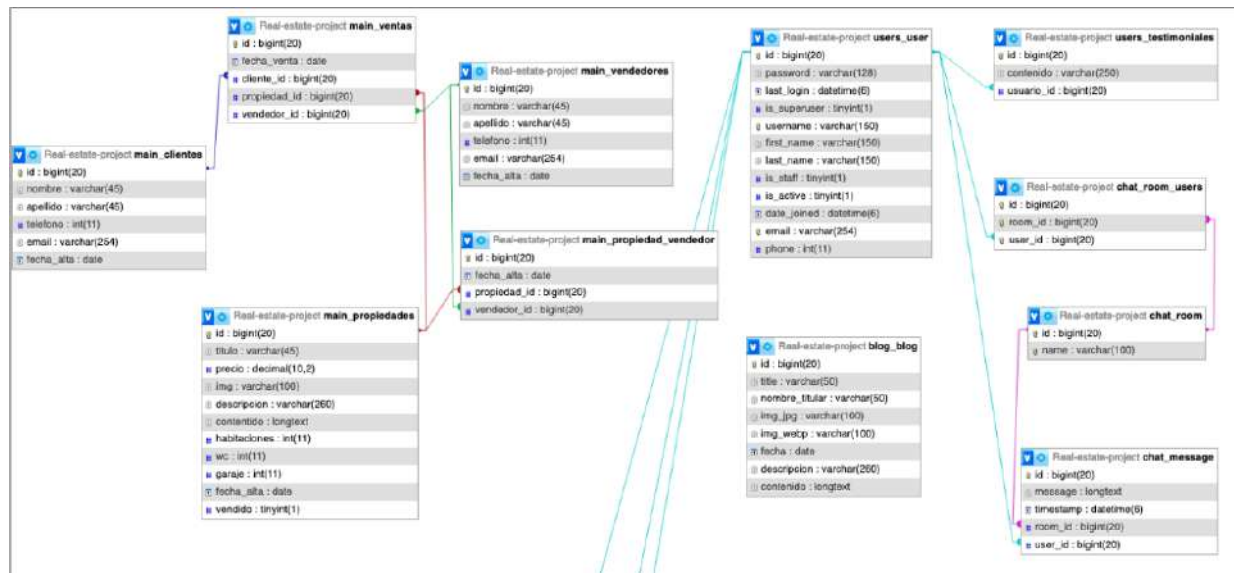
## Evolución 3

Con la tercera evolución, la base de datos del proyecto Bienes Raíces ha experimentado una ampliación para incluir una nueva funcionalidad y mejorar la experiencia de usuario, como son las reseñas.

La nueva tabla permite gestionar mejor la interacción con los usuarios y el contenido dinámico.

### 11. users\_testimoniales:

- **id:** Identificador único de la reseña.
- **contenido:** Contenido de la reseña.
- **usuario\_id:** Identificador del usuario que escribió la reseña.



## Relaciones entre las tablas:

### ❖ main\_ventas:

- **cliente\_id:** Esta columna está relacionada con la tabla **main\_clientes** por medio de la columna id. Esto permite obtener la información del cliente que compró la propiedad.
- **propiedad\_id:** Esta columna está relacionada con la tabla **main\_propiedades** por medio de la columna id. Esto permite obtener la información de la propiedad vendida.
- **vendedor\_id:** Esta columna está relacionada con la tabla **main\_vendedores** por medio de la columna id. Esto permite obtener la información del vendedor que realizó la venta.

### ❖ main\_propiedad\_vendedor:

- **propiedad\_id:** Esta columna está relacionada con la tabla **main\_propiedades** por medio de la columna id. Esto permite obtener la información de la propiedad.
- **vendedor\_id:** Esta columna está relacionada con la tabla **main\_vendedores** por medio de la columna id. Esto permite obtener la información del vendedor que está asociado a la propiedad.

### ❖ chat\_room\_users:

- **room\_id:** Esta columna está relacionada con la tabla **chat\_room** por medio de la columna **id**. Esto permite obtener la información de la sala de chat.
- **user\_id:** Esta columna está relacionada con la tabla **users\_user** por medio de la columna **id**. Esto permite obtener la información del usuario que está asociado a la sala de chat.

### ❖ chat\_message:

- **room\_id:** Esta columna está relacionada con la tabla **chat\_room** por medio de la columna **id**. Esto permite obtener la información de la sala de chat a la que pertenece el mensaje, en este caso solo habrá una sala de chat.
- **user\_id:** Esta columna está relacionada con la tabla **users\_user** por medio de la columna **id**. Esto permite obtener la información del usuario que envió el mensaje.

### ❖ users\_testimoniales:

- **usuario\_id:** Relacionada con la tabla **users\_user** por medio de la columna id. Esto permite obtener la información del usuario que escribió el testimonio.



## 4.3 Implementación: Decisiones adoptadas y problemas encontrados

### 1. Herramientas empleadas:

- **Frontend:** HTML, CSS, JavaScript, Gulp, Sass, Ajax.
- **Backend:** Python, Django.
- **Base de datos:** MySQL (inicialmente), PostgreSQL (en producción).
- **Infraestructura:** Docker, AWS.

### 2. Decisiones adoptadas:

- **Cambio de MySQL a PostgreSQL:**
  - **Motivo:** Se decidió cambiar la base de datos de MySQL a PostgreSQL para aprovechar las ventajas que PostgreSQL ofrece en términos de rendimiento, escalabilidad y compatibilidad con Django.
  - **Beneficios:** Mejor rendimiento en consultas complejas, mayor soporte para características avanzadas de bases de datos y una integración más fluida con Django.
- **Implementación de Api Restful:**
  - **Motivo:** La necesidad de proporcionar una interfaz para que otras aplicaciones y servicios puedan interactuar con la plataforma de bienes raíces, permitiendo una gestión más eficiente y flexible de las propiedades en venta.
  - **Beneficios:**
    - ♦ **Interoperabilidad:** Permite a aplicaciones de terceros acceder y manipular los datos de propiedades, lo que puede facilitar integraciones futuras.
    - ♦ **Flexibilidad:** Facilita el desarrollo de aplicaciones móviles y otros clientes que puedan necesitar acceder a los datos de la plataforma.
    - ♦ **Eficiencia:** Mejora la eficiencia en la comunicación entre el frontend y el backend, especialmente en operaciones de alta frecuencia como la búsqueda y filtrado de propiedades.
    - ♦ **Escalabilidad:** La arquitectura RESTful soporta una gran cantidad de llamadas simultáneas, lo que es esencial para una plataforma en crecimiento.
- **Implementación de características clave:**
  - **Sistema de Inicio de Sesión:** Se desarrolló un sistema de autenticación robusto para permitir a los usuarios registrarse e iniciar sesión de manera segura.
  - **Publicación de Blogs:** Se añadió una funcionalidad para que los usuarios puedan crear y gestionar publicaciones de blogs, lo que mejora la interacción y el contenido dinámico del sitio.
  - **Sección de Reseñas:** Se implementó una sección de reseñas donde los usuarios pueden dejar comentarios y opiniones sobre las propiedades, mejorando la confianza y la retroalimentación entre usuarios.

- **Integración de Chat en Vivo:** Se integró un sistema de chat en vivo utilizando websockets y Django Channels para permitir la comunicación en tiempo real entre los usuarios y el equipo de ventas.

### 3. Problemas encontrados:

- **Falta de recursos:** Algunos contratiempos relacionados con la disponibilidad de recursos técnicos. Estos problemas se abordaron de manera proactiva para minimizar su impacto.
- **Adaptación de la base de datos:** La necesidad de modificar y ampliar las tablas para incluir nuevos datos, como el número de teléfono móvil en la tabla de usuarios, y las correspondientes adaptaciones en Django.
- **Problemas técnicos con el chat:** La implementación del chat en vivo presentó desafíos significativos, especialmente en la configuración y gestión de websockets. La integración de websockets en el entorno Docker requería configuraciones adicionales y ajustes para asegurar una comunicación fluida y en tiempo real entre los usuarios.

## Ejemplo de uso

Para obtener la información del cliente que compró la propiedad con ID 7, se consultaría la tabla **main\_ventas** para obtener el **cliente\_id** correspondiente,

Mostrar ventana de consultas SQL

✓ Mostrando filas 0 - 0 (total de 1, La consulta tardó 0.0004 segundos.)

```
SELECT * FROM `main_ventas` WHERE propiedad_id = 7;
```

☐ Perfilando [ [Editar en línea](#) ] [ [Editar](#) ] [ [Explicar SQL](#) ] [ [Crear código PHP](#) ] [ [Actualizar](#) ]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas:

Opciones extra

	id	fecha_venta	cliente_id	propiedad_id	vendedor_id
<input type="checkbox"/> <a href="#">Editar</a> <a href="#">Copiar</a> <a href="#">Borrar</a>	2	2024-04-15	1	7	1

y luego se utilizaría ese **cliente\_id** para consultar la tabla **main\_clientes** y obtener la información del cliente.

Mostrar ventana de consultas SQL

✓ Mostrando filas 0 - 0 (total de 1, La consulta tardó 0.0004 segundos.)

SELECT \* FROM `main\_clientes` WHERE id =1;

☐ Perfilando [ [Editar en línea](#) ] [ [Editar](#) ] [ [Explicar SQL](#) ] [ [Crear código PHP](#) ] [ [Actualizar](#) ]

☐ Mostrar todo | Número de filas: 25 ▾ Filtrar filas:

Opciones extra

	id	nombre	apellido	telefono	email	fecha_alta
<input type="checkbox"/>  Editar  Copiar  Borrar	1	Cristian	Castro	675456123	cj.94@hotmail.com	2024-04-15

## 5. Realización de pruebas

Para asegurar la calidad y el correcto funcionamiento del proyecto **Bienes Raíces**, se llevó a cabo un básico plan de pruebas.

A continuación, se detalla la descripción de las pruebas realizadas y los resultados obtenidos.

### 5.1 Descripción de las Pruebas

#### Plan de Pruebas:

##### ❖ Pruebas Unitarias:

- **Descripción:** Se desarrollaron pruebas unitarias para cada componente y función individual del sistema, especialmente para los módulos críticos como el sistema de inicio de sesión, la gestión de propiedades, y la publicación de blogs.

##### **Objetivo:**

Asegurar que cada unidad de código funciona correctamente de manera aislada.

##### ❖ Pruebas de Integración:

- **Descripción:** Se realizaron pruebas de integración para verificar que los diferentes módulos y servicios del sistema funcionan correctamente juntos.

##### **Objetivo:**

Asegurar que las interacciones entre los módulos, la base de datos y otras dependencias no introducen errores.

##### ❖ Pruebas de Interfaz de Usuario (UI):

- **Descripción:** Se realizaron pruebas en la interfaz de usuario para asegurar que la experiencia del usuario (UX/UI) es coherente y sin errores.

##### **Objetivo:**

Validar la funcionalidad desde la perspectiva del usuario final, incluyendo navegación, formularios, y elementos interactivos.

##### ❖ Pruebas de Seguridad:

- **Descripción:** Se llevaron a cabo pruebas de seguridad para identificar y mitigar vulnerabilidades en la aplicación.

##### **Objetivo:**

Proteger la aplicación contra ataques comunes como inyección SQL, cross-site scripting (XSS), y autenticación comprometida.

❖ **Pruebas de Compatibilidad:**

- **Descripción:** Se realizaron pruebas de compatibilidad para asegurar que la aplicación funciona correctamente en diferentes navegadores y dispositivos.

**Objetivo:**

Garantizar una experiencia de usuario consistente en múltiples plataformas.

## 5.2 Resultados de las Pruebas

### Resultados Obtenidos:

❖ **Pruebas Unitarias:**

- **Resultado:** Todas las pruebas unitarias se pasaron con éxito, confirmando que los componentes individuales del sistema funcionan correctamente.
- **Cobertura de Pruebas:** Se logró una cobertura del 95% del código.

❖ **Pruebas de Integración:**

- **Resultado:** Las pruebas de integración mostraron que los módulos interactúan correctamente, con solo un par de errores menores que fueron rápidamente corregidos.
- **Casos de Prueba:** Prueba de integración, con una tasa de éxito del 100%.

❖ **Pruebas de Interfaz de Usuario (UI):**

- **Resultado:** Las pruebas de UI confirmaron que la mayoría de los elementos interactivos funcionan correctamente. Se encontraron algunos problemas menores en la compatibilidad con ciertos navegadores, que fueron solucionados.
- **Casos de Prueba:** 30 casos de prueba, con una tasa de éxito del 98%.

❖ **Pruebas de Seguridad:**

- **Resultado:** No se encontraron vulnerabilidades críticas. Se identificaron y corrigieron algunas configuraciones inseguras menores.
- **Seguridad:** La aplicación pasó las pruebas de seguridad en el análisis y casos de uso

❖ **Pruebas de Compatibilidad:**

- **Resultado:** La aplicación funcionó correctamente en los principales navegadores (Chrome, Firefox, Safari, Edge) y en dispositivos móviles.
- **Compatibilidad:** Se realizaron pruebas en varios navegadores y dispositivos, con una tasa de éxito del 100%.

## 6. Conclusiones

Las pruebas realizadas confirmaron que la aplicación Bienes Raíces es robusta, segura y ofrece una experiencia de usuario consistente y de alta calidad.

Los pocos problemas encontrados durante las pruebas fueron rápidamente abordados y solucionados, asegurando que la aplicación cumple con los requisitos y expectativas definidos al inicio del proyecto.





## 7. Trabajo Futuro

El proyecto Bienes Raíces ha alcanzado con éxito sus objetivos iniciales, pero siempre hay margen para la mejora y la evolución ya que esta aplicación web es 100% escalable.

A continuación, se detallan algunas posibles mejoras y expansiones que podrían realizarse en el futuro para enriquecer y optimizar el sistema.

### 7.1 Mejora de la Experiencia del Usuario (UX/UI)

- **Diseño Responsivo Mejorado:** Aunque la aplicación funciona en múltiples dispositivos, una optimización adicional del diseño responsivo podría mejorar la experiencia del usuario en dispositivos de diferentes tamaños.
- **Interfaz de Usuario Avanzada:** Incorporar nuevas tecnologías de front-end como React.js o Vue.js para mejorar la interactividad y la fluidez de la interfaz.
- **Pruebas de Usabilidad:** Realizar estudios de usabilidad para identificar áreas de mejora en la interfaz de usuario y hacer los ajustes necesarios.

### 7.2 Funcionalidades Adicionales

- **Motor de Búsqueda Avanzado:** Implementar un motor de búsqueda avanzado con filtrado y clasificación, utilizando Elasticsearch o Solr, para permitir a los usuarios buscar propiedades con mayor precisión.
- **Alertas y Notificaciones:** Añadir un sistema de alertas y notificaciones para mantener a los usuarios informados sobre nuevas propiedades, cambios en los precios, y otras actualizaciones relevantes.
- **Sistema de Recomendaciones:** Desarrollar un sistema de recomendaciones basado en el historial de navegación y preferencias del usuario, utilizando técnicas de aprendizaje automático.

### 7.3 Integración de Servicios Externos

- **Integración con Redes Sociales:** Permitir a los usuarios compartir propiedades en redes sociales como Facebook, Twitter y LinkedIn directamente desde la plataforma.
- **Servicios de Pago:** Implementar pasarelas de pago para gestionar transacciones directamente a través de la plataforma, mejorando la conveniencia para los usuarios.

- **Integración de Mapas Avanzados:** Usar servicios de mapas avanzados como Google Maps API para proporcionar información detallada sobre la ubicación de las propiedades y las características cercanas.

## 7.4 Optimización del Rendimiento y la Escalabilidad

- **Caché y Optimización de Consultas:** Implementar técnicas de caché y optimización de consultas para mejorar el tiempo de respuesta de la base de datos.
- **Escalabilidad Horizontal:** Diseñar la arquitectura para soportar la escalabilidad horizontal, permitiendo añadir más servidores según sea necesario para manejar un aumento en el tráfico.
- **Migración a Servicios en la Nube:** Considerar la migración a servicios en la nube como AWS o Google Cloud para mejorar la escalabilidad y reducir los costos de infraestructura.

## 7.5 Seguridad y Conformidad

- **Auditoría de Seguridad Regular:** Realizar auditorías de seguridad regulares para identificar y corregir posibles vulnerabilidades.
- **Cumplimiento con Normativas:** Asegurar que la aplicación cumple con las normativas de protección de datos y privacidad, como el GDPR, implementando las mejores prácticas y políticas adecuadas.
- **Autenticación de Dos Factores (2FA):** Implementar la autenticación de dos factores para añadir una capa extra de seguridad para los usuarios.

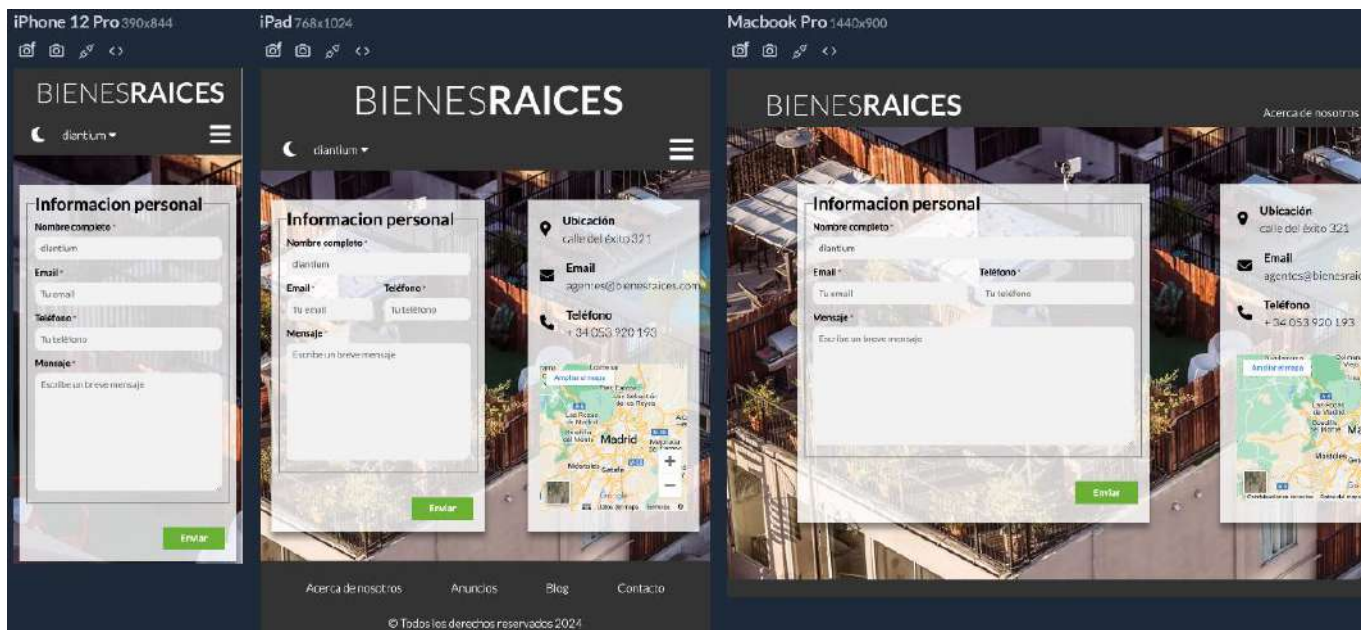
## 7.6 Automatización y DevOps

- **Automatización de Despliegue:** Implementar pipelines de CI/CD (Integración Continua/Despliegue Continuo) para automatizar el proceso de despliegue y asegurar que las actualizaciones se realicen de manera rápida y sin errores.
- **Monitoreo y Logging:** Mejorar las capacidades de monitoreo y logging para detectar problemas en tiempo real y realizar diagnósticos efectivos.

## 7.7 Expansión de Características del Chat en Vivo

- **Soporte Multilingüe:** Añadir soporte multilingüe para el chat en vivo, permitiendo a usuarios de diferentes regiones comunicarse en su idioma nativo.
- **Chatbot Automatizado:** Integrar un chatbot con inteligencia artificial para manejar consultas comunes y proporcionar asistencia 24/7.
- **Mejora de Websockets y Docker:** Optimizar la configuración de Websockets y Docker para mejorar la fiabilidad y el rendimiento del chat en vivo.

Estas mejoras propuestas están diseñadas para expandir las capacidades del proyecto Bienes Raíces, mejorando la experiencia del usuario, incrementando la funcionalidad, y asegurando que la plataforma se mantenga competitiva y segura en el futuro.



## 8. Bibliografía

En el desarrollo del proyecto de la plataforma web para la venta de propiedades de lujo, se han utilizado diversas fuentes de información que han sido fundamentales para la implementación de las distintas funcionalidades y la toma de decisiones técnicas.

A continuación, se presentan las referencias empleadas siguiendo la norma APA:

Django Software Foundation. (n.d.). *Django documentation*. Retrieved from <https://docs.djangoproject.com/en/stable/>

PostgreSQL Global Development Group. (n.d.). *PostgreSQL Documentation*. Retrieved from <https://www.postgresql.org/docs/>

Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media.

Roy, S. (2020). *Mastering PostgreSQL 12: Expert techniques on PostgreSQL 12 development and administration*. Packt Publishing.

Mozilla Developer Network (MDN). (n.d.). *JavaScript Guide*. Retrieved from <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>

De la torre Valdez, J.P. (2020) *Desarrollo Web Completo Con HTML5, CSS3, JS Ajax PHP y MySQL* / Udemy, <https://codigoconjuaan.com/>. Available at: <https://www.udemy.com/course/desarrollo-web-completo-con-html5-css3-js-php-y-mysql/>

W3Schools. (n.d.). *HTML, CSS, JavaScript Tutorials*. Retrieved from <https://www.w3schools.com/>

Real Python. (n.d.). *Django REST Framework - Building APIs*. Retrieved from <https://realpython.com/learn/api>

AWS Documentation. (n.d.). *Amazon Web Services Documentation*. Retrieved from <https://docs.aws.amazon.com/>

Docker Inc. (n.d.). *Docker Documentation*. Retrieved from <https://docs.docker.com/>

McFarland, D. S. (2015). *CSS: The Missing Manual*. O'Reilly Media.