EncuentraPiso

La forma más sencilla de gestionar tus ventas y alquileres de inmuebles.

Índice

EncuentraPiso

1. Justificación

Este proyecto desarrolla un portal inmobiliario orientado a la venta y alquiler de pisos. Hoy en día el tiempo es un bien escaso y la búsqueda de pisos puede resultar agotadora, el objetivo de este proyecto es facilitar la comunicación entre las empresas dedicadas a la venta y alquiler de inmuebles y los clientes y potenciales clientes.

El proyecto se plantea para que puedan participar cuatro tipos de roles diferenciados, el usuario anónimo, el cliente (o potencial cliente), el agente inmobiliario y el administrador del portal, para cada uno de estos roles hemos diseñado pantallas diferentes con diferentes funcionalidades.

Los roles para los que más funcionalidades se han desarrollado en la aplicación son el cliente y el agente inmobiliario, de forma que las acciones que realizan los clientes se verán reflejadas en las pantallas del agente inmobiliario y viceversa.

1.1 Estructura externa de la aplicación.

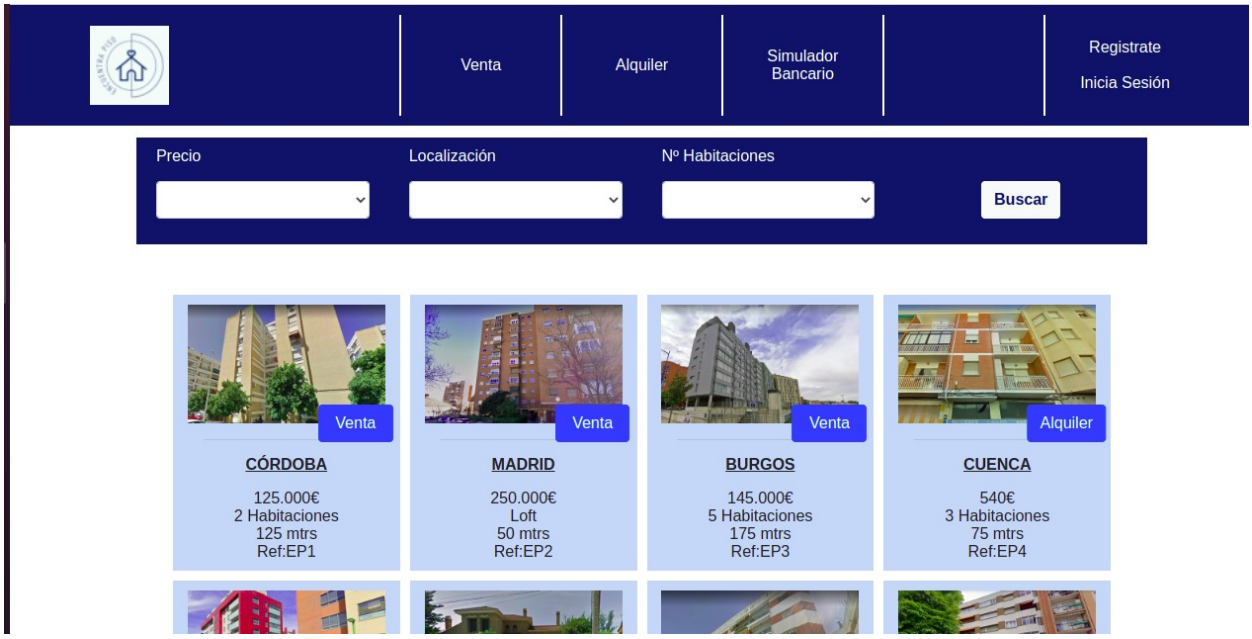
La aplicación da servicio a distintos perfiles de usuario:

- **Usuario Anónimo:** Este podrá acceder a la página principal donde se mostrarán todas las ofertas tanto de venta como de alquiler y se podrá ordenar por precio (ascendente o descendente) y filtrar por localidad y número de habitaciones.

  El usuario anónimo también podrá acceder a la pestaña de venta o de alquiler y podrá igualmente ordenar por precio y filtrar pero en este caso solo las ventas o los alquileres en función de la pestaña en la que se encuentre (venta o alquiler)

  Si la búsqueda no obtiene resultados se le facilitará al cliente un listado de las últimas ofertas registradas, si la búsqueda se ha realizado en la home las ofertas serán tanto de venta como alquiler y si se ha realizado la búsqueda en la pestaña de venta o alquiler se le presentarán las ultimas ofertas en función de la pestaña donde se encuentre.

  También podrá el usuario anónimo acceder al calculo de la cuota de una hipoteca facilitando el capital, el interés y el plazo se informa que todos los campos son obligatorios y si se intentan enviar los datos y no están completos, el servidor nos devuelve a la misma pantalla y nos informa del error. En esta pantalla presentamos también una lista de opciones, en este caso de venta, ordenadas por precio.



Img.1.1: Home. Página principal para todos los tipos de usuarios.

Cualquier usuario puede visitar la página como usuario anónimo y tendrá la posibilidad de registrarse, para ello en la cabecera aparece la opción de registrarse o iniciar sesión. En el registro se podrá elegir registrarse como cliente o como trabajador.

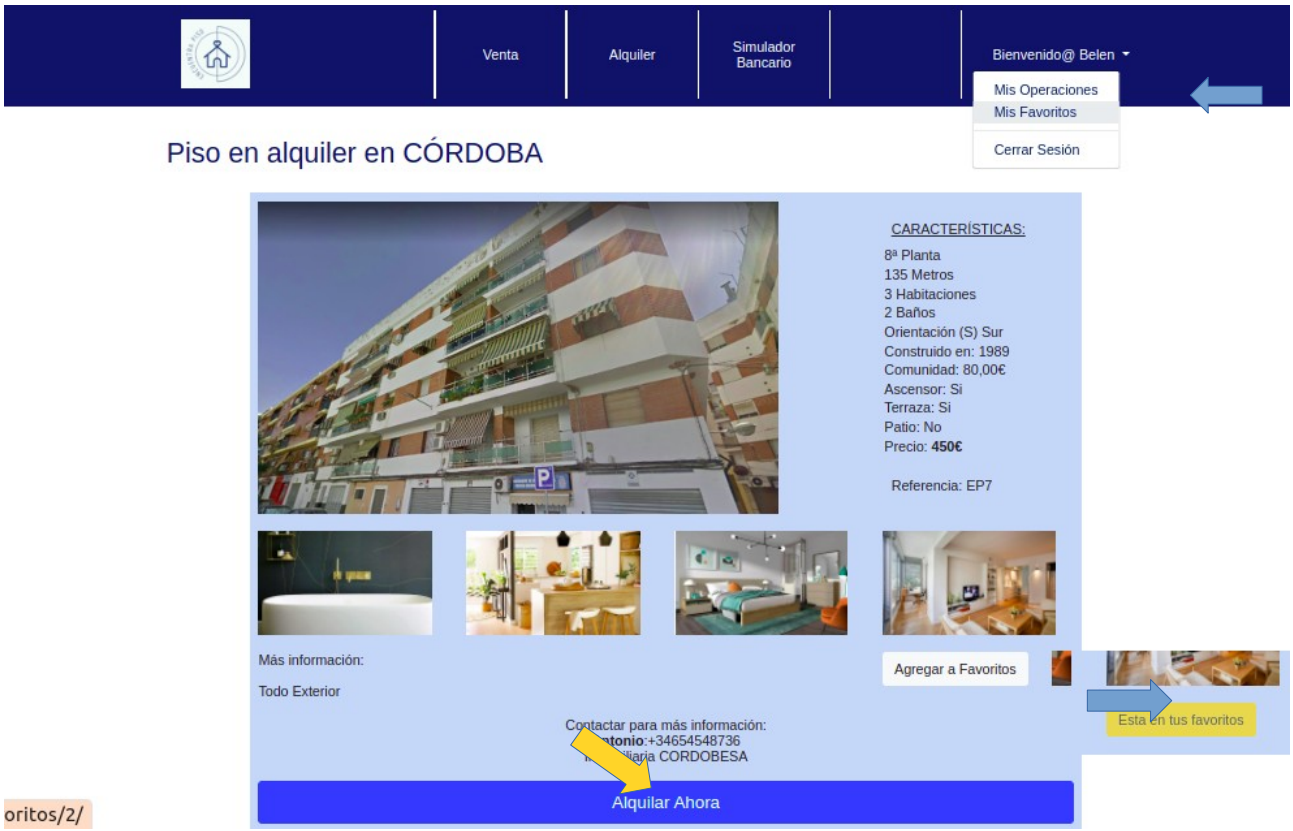
- **Cliente:** El cliente se registra facilitando sus datos personales, un nombre de usuario que debe ser único en toda la base de datos y una contraseña. Una vez que un usuario se registre como cliente tendrá acceso a todo lo que tiene acceso un usuario anónimo, y además podrá:

  - Añadir a favoritos: En la página de información de una oferta aparecerá un botón que le permitirá añadir esa oferta a favoritos, si ya lo tiene añadido el botón aparecerá en amarillo, deshabilitado e indicando que ya está agregado a favoritos. En todo momento el cliente tendrá acceso a las ofertas añadidas a favoritos desde un desplegable en la cabecera.

  - Solicitar una Compra: Si el cliente ve una oferta de compra y le interesa, puede solicitar una reserva de Compra sobre esa oferta.

  - Solicitar un Alquiler: Si el cliente ve una oferta de alquiler, y le interesa, podrá ver la fecha de salida del último inquilino, en el caso de que actualmente se encontrase alquilado, y solicitar una reserva de Alquiler.

  El cliente siempre podrá visitar sus operaciones desde el desplegable de la cabecera y podrá ver si sus solicitudes han sido aprobadas o siguen pendientes.



Img.1.2:Vista Información de una oferta para el Cliente.

Img.1.3:Pantalla Operaciones Cliente con la información de sus operaciones

- **<u>Trabajador:</u>** Un trabajador , Agente Inmobiliario, deberá pertenecer a una empresa, cada empresa registrada en el sistema tendrá un código de acceso. Para que un usuario se registre como Trabajador previamente deberá facilitar la razón social de la empresa a la que pertenece y el código de acceso, si ambos coinciden con lo registrado en la base de datos podrá darse de alta como trabajador.

Una vez registrado el trabajador podrá acceder a su panel donde verá las últimas ofertas de su empresa, las últimas ofertas subidas por el podrá crear nuevas ofertas o gestionarlas solicitudes de los clientes.

Img.1.4:Panel trabajador

Desde la pestaña de crear podrá dar de alta un inmueble y crear una oferta, y desde gestionar una oferta podrá ver si hay clientes interesados en sus ofertas y proceder a contactar con ellos por teléfono.

- **Administrador:** Por último tenemos al administrador del portal, que tendrá acceso al panel de control de toda la aplicación y podrá administrar tanto los clientes, trabajadores y ofertas. Solo el administrador del portal podrá registrar empresas en la base de datos y esto se diseña así porque se entiende que las empresas contratarán los servicios de este portal y para ello deberán contactar con el administrador del sitio para consensuar las condiciones de contratación.

1.2 Estructura interna de la aplicación.

La aplicación esta hecha en Django, y los proyectos realizados en este framework suelen dividir el código en aplicaciones que ayudan a organizarlo de forma que sea modulable y reutilizable para futuras aplicaciones.

Siguiendo estos principios, yo he creado este proyecto con tres aplicaciones diferentes:

- Administración: engloba los modelos de los distintos usuarios que se contemplan en este proyecto: Propietarios, Clientes, Trabajadores y Empresas.

- Inmuebles: En este caso incorporamos en esta aplicación el modelo de inmueble y el modelo de favoritos puesto que este no se verá afectado por las operaciones.

- Operaciones: Esta aplicación engloba todas las acciones que se van a desarrollar en el proyecto, las ofertas, las ventas y los alquileres.

El motivo por el cual he optado por esa división es porque en un futuro se pueden añadir nuevos modelos de inmuebles (chalet, locales, cocheras…..), nuevas operaciones, como por ejemplo permutas, o nuevos roles de usuarios y la aparición de estos elementos no debe de afectar al conjunto del proyecto sino a la aplicación propiamente dicha.

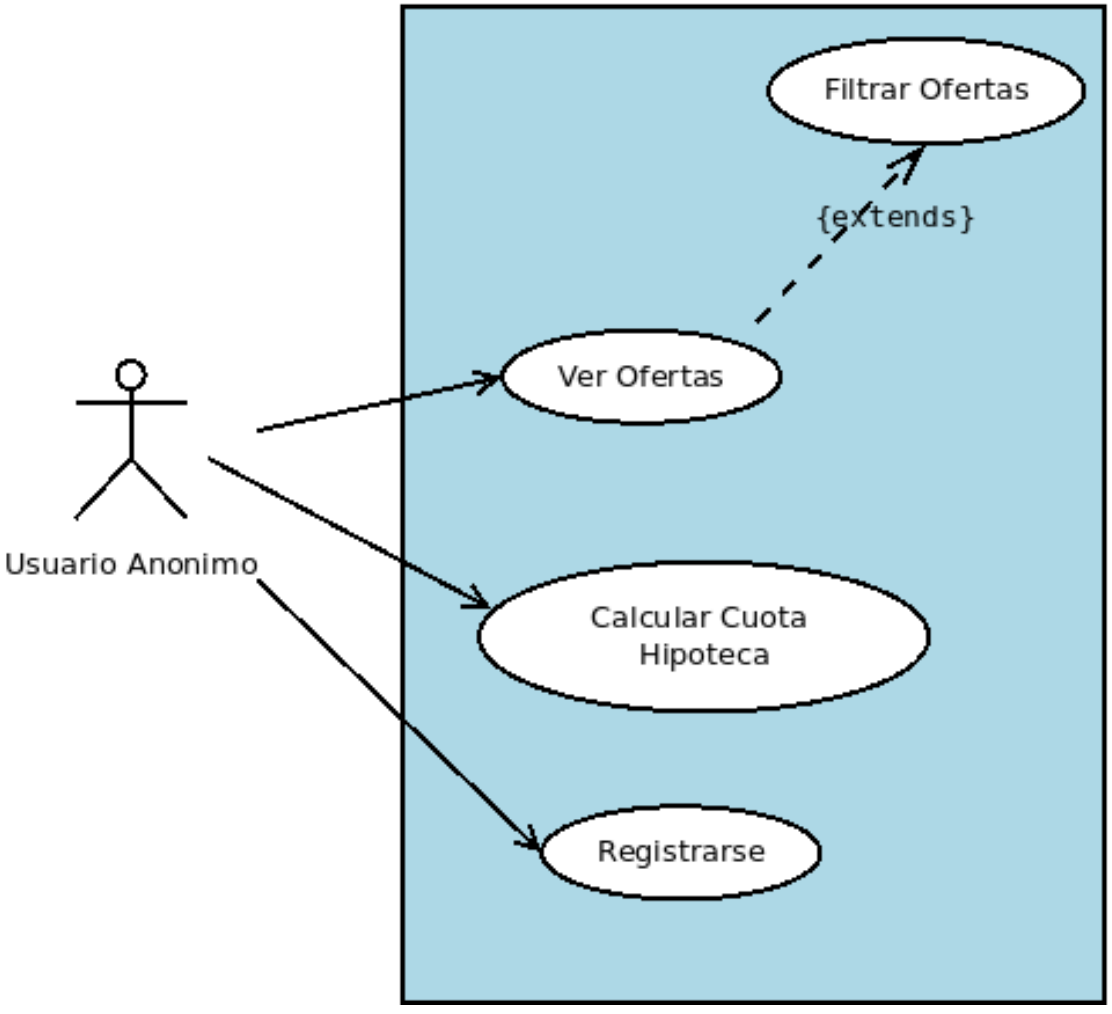
1.3 Posibles mejoras

Como posibles mejoras se podría añadir en un futuro una pasarela de pago para poder abonar una señal en el momento de solicitar la reserva. Para este propósito Django tiene varios paquetes que nos pueden ayudar a gestionarlo como son Django Paypal, Django Merchant o Django Payu

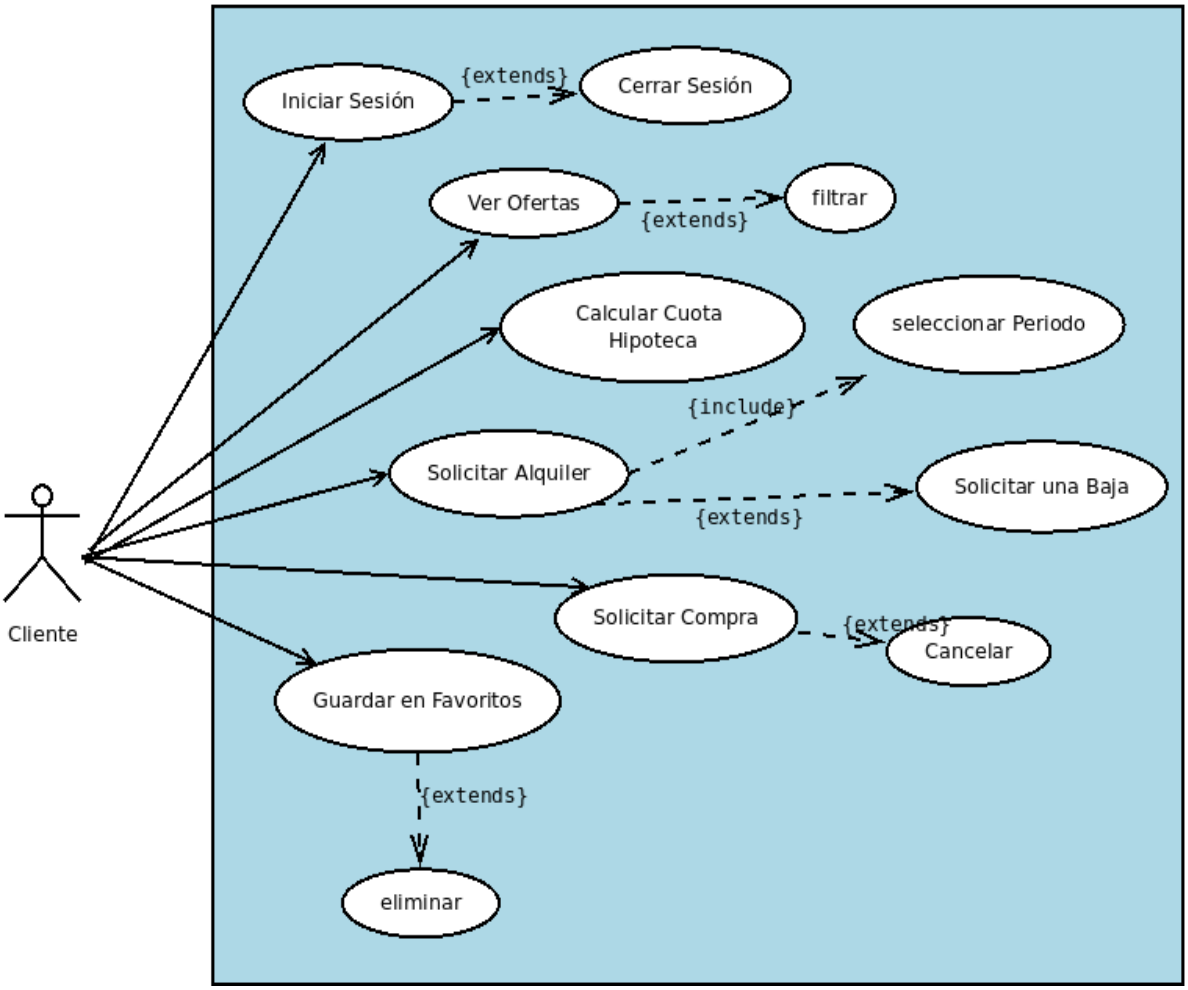
2. Desarrollo del Proyecto

2.1 Diagramas de Casos de Uso.

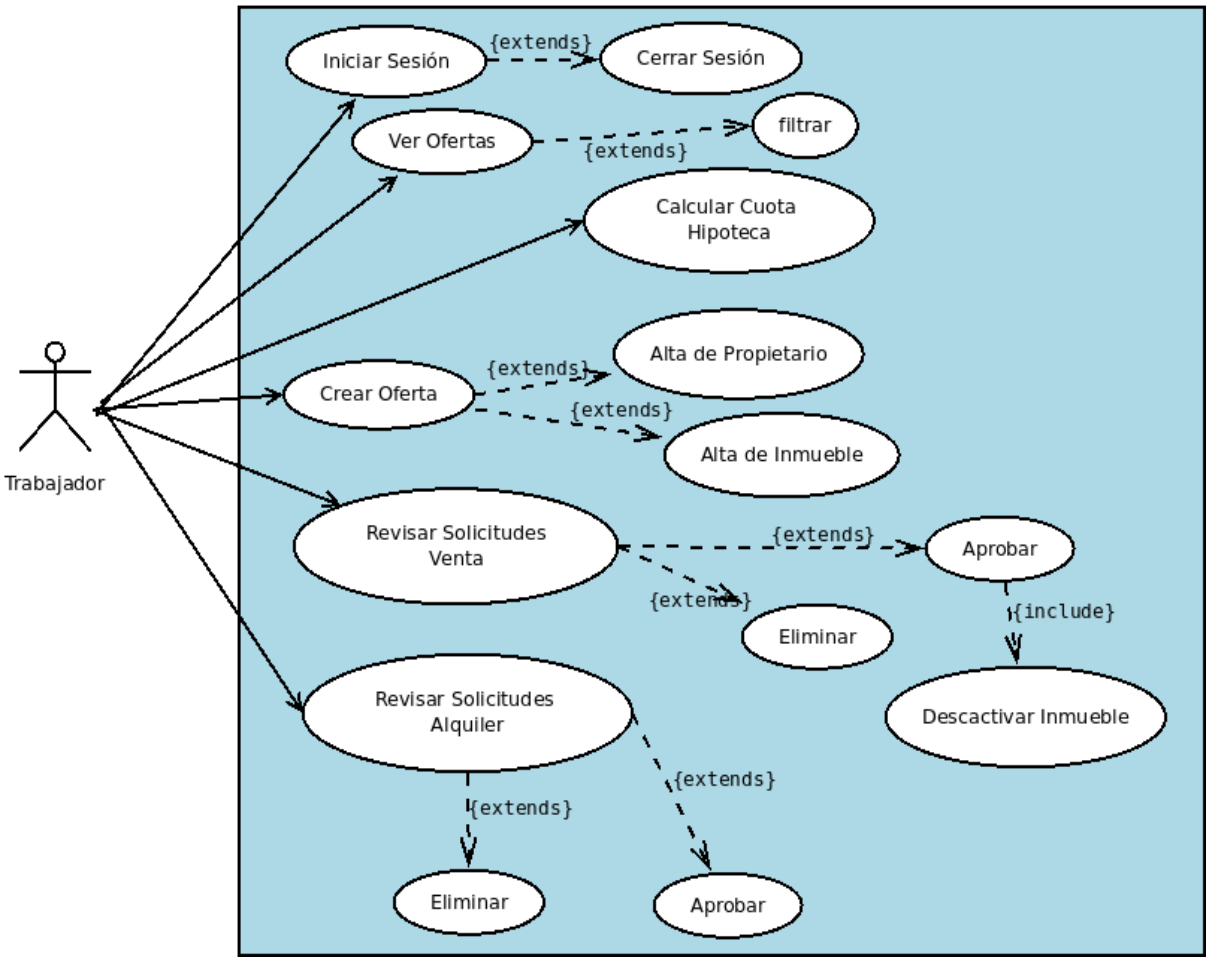
Casos de Uso para el Usuario Anónimo:



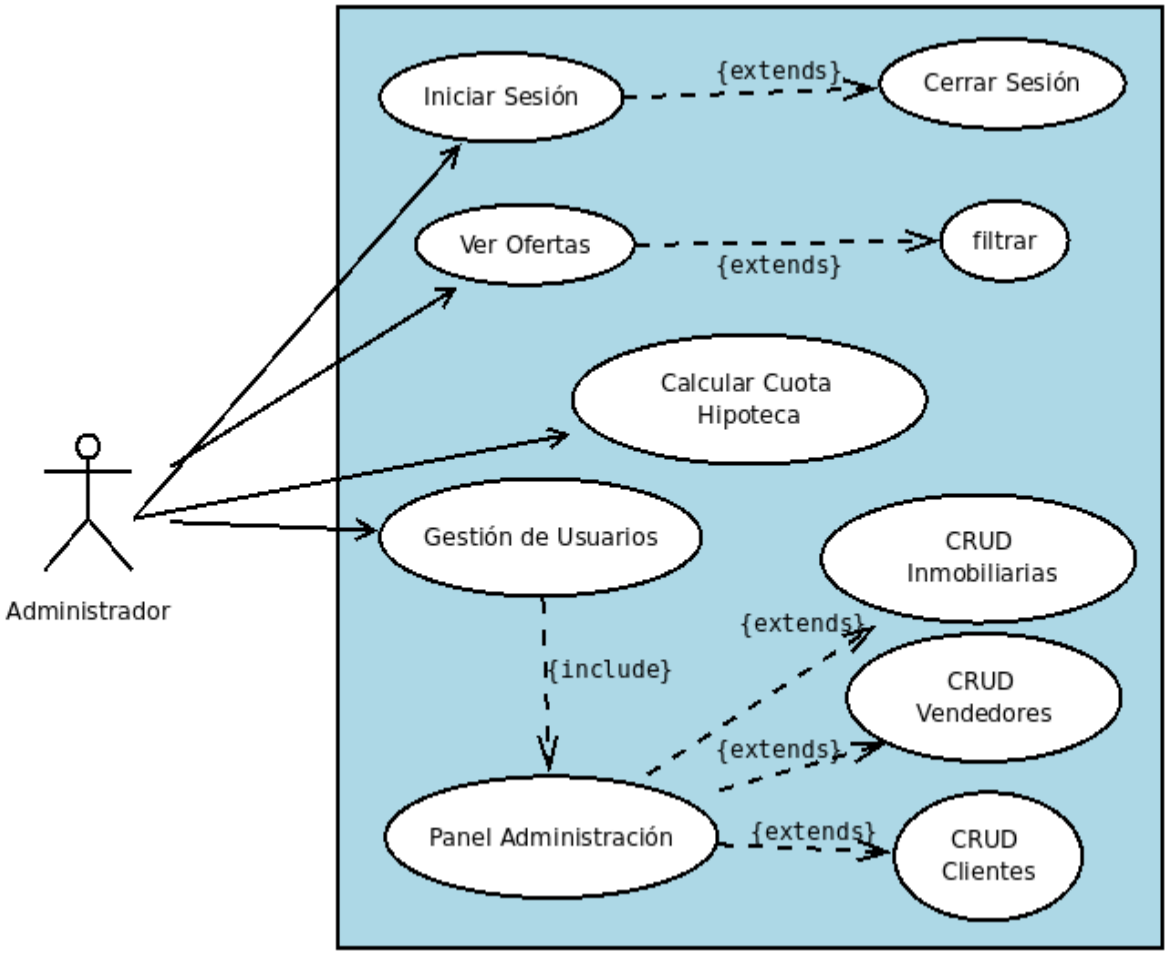
Casos de Uso para el Cliente:



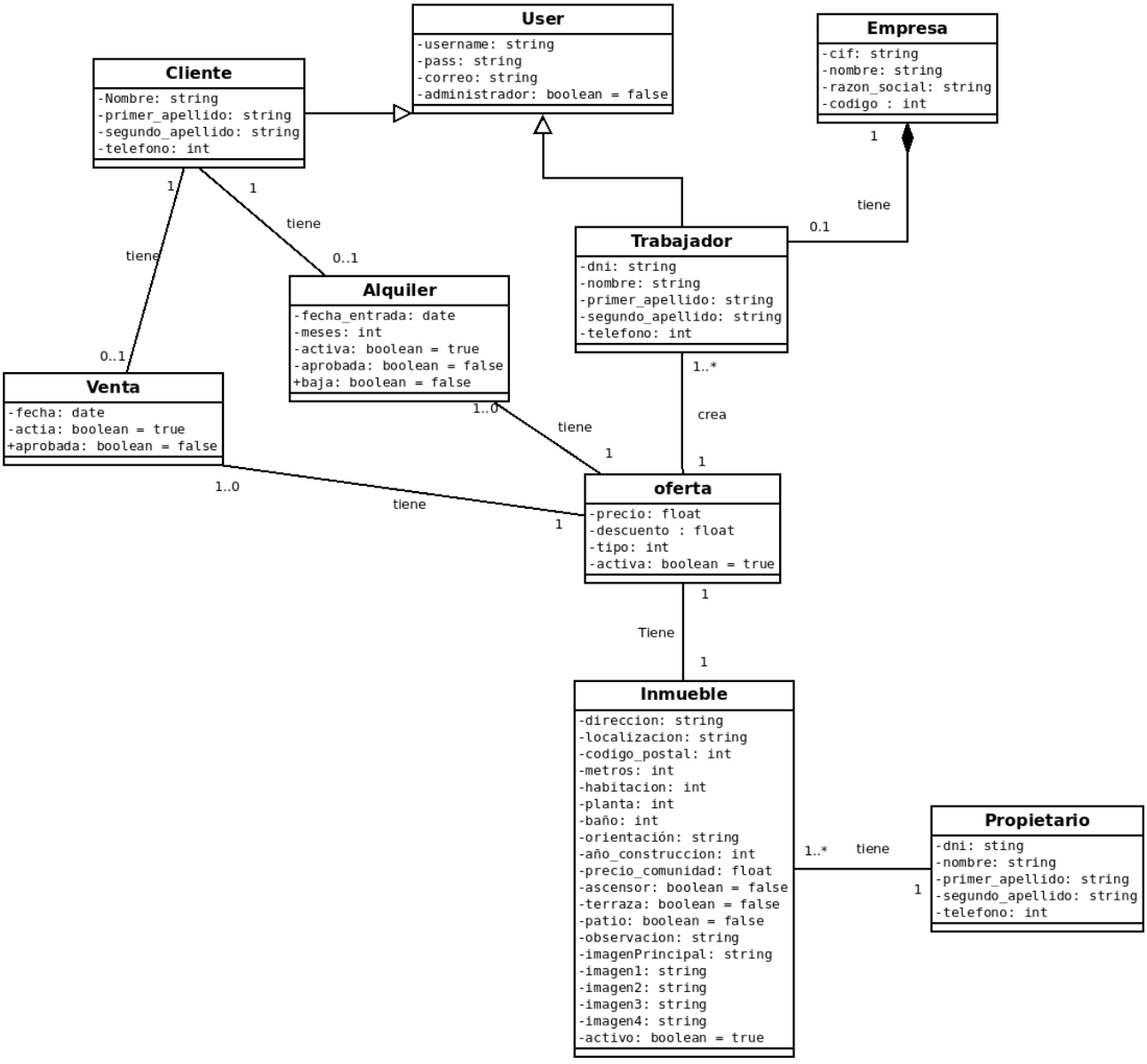
Casos de Uso para el Trabajador



Casos de Uso para el Administrador:

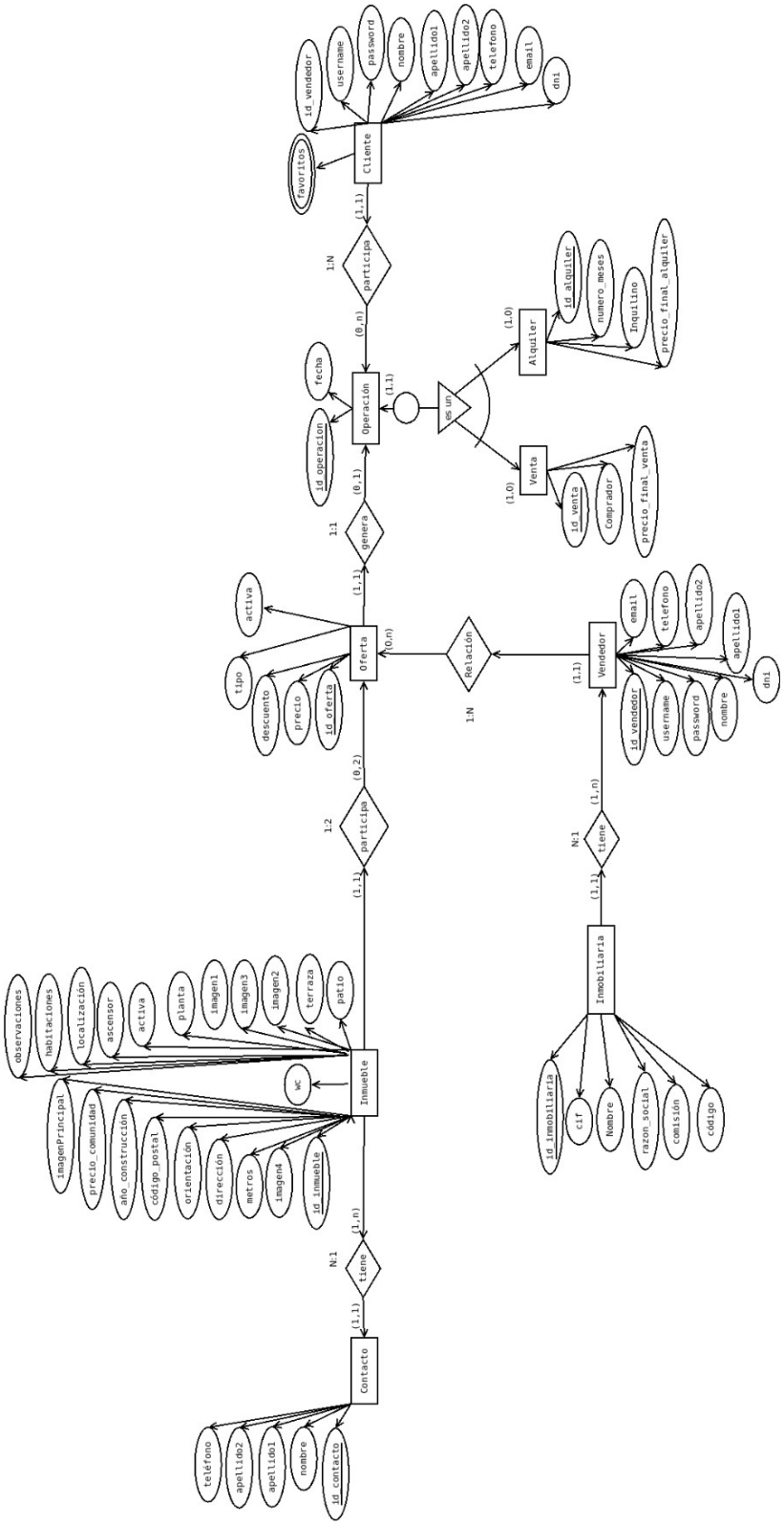


2.3 Diagramas de Clases



2.3 Diseño de la Base de Datos.

2.3.1 Modelo Conceptual



2.3.2 Modelo Lógico

USER: (id_user, username, pass, email)

CONTACTO: (id_contacto, dni, nombre, apellido1, apellido2, teléfono)

INMUEBLE: (id_inmueble, dirección, orientación, código_postal, año_construcción, precio_comunidad, habitaciones, localización, ascensor, planta, terraza, patio, wc, imagenPrincipal, imagen1, imagen2, imagen3, imagen4, activo, **id_contacto**)

INMOBILIARIA: (id_inmobiliaria, cif, nombre, razón_social, comisión, código)

TRABAJADOR: (id_trabajador , nombre, dni, apellido1, apellido2, teléfono, **id_inmobiliaria, id_user**)

OFERTA: (id_oferta, descuento, precio,tipo, activo, **id_trabajador,** i**d_inmueble**)

CLIENTE: (id_cliente, dni, nombre, apellido1, apellido2, teléfono,**id_user**)

VENTA: (id_venta, fecha, precio_final_venta, activa, aprobada, **id_cliente**, **id_oferta**)

ALQUILER: (id_alquiler, numero_meses, precio_fin_alquiler,activa, aprobada,baja **id_cliente, id_oferta**)

FAVORITO: (id_favoritos, **id_cliente**, i**d_inmueble**)

3. Implementación del Proyecto

3.1 Requisitos hardware y software

No será necesario ningún requisito hardware adicional solamente ordenador o dispositivo móvil. Para acceder al proyecto solo será necesario tener acceso a internet.

La realización del proyecto se ha ejecutado utilizando Ubuntu 20.04 y Python 3.8.10, se ha creado un entorno virtual donde se han instalado los siguientes paquetes:[1]

- Django==4.0.5
- django-bootstrap4==22.1 →(para incluir bootstrap en el proyecto)
- django-phonenumber-field==6.1.0 → (para trabajar con números de teléfono)
- django-phonenumbers==1.0.1 → (para trabajar con números de teléfono)
- phonenumbers==8.12.49 → (para trabajar con números de teléfono)
- Pillow==9.1.1 →(para trabajar con imágenes)
- python-dateutil==2.8.2 →(para trabajar con fechas)
- weasyprint==55.0 →(para trabajar con pdf)

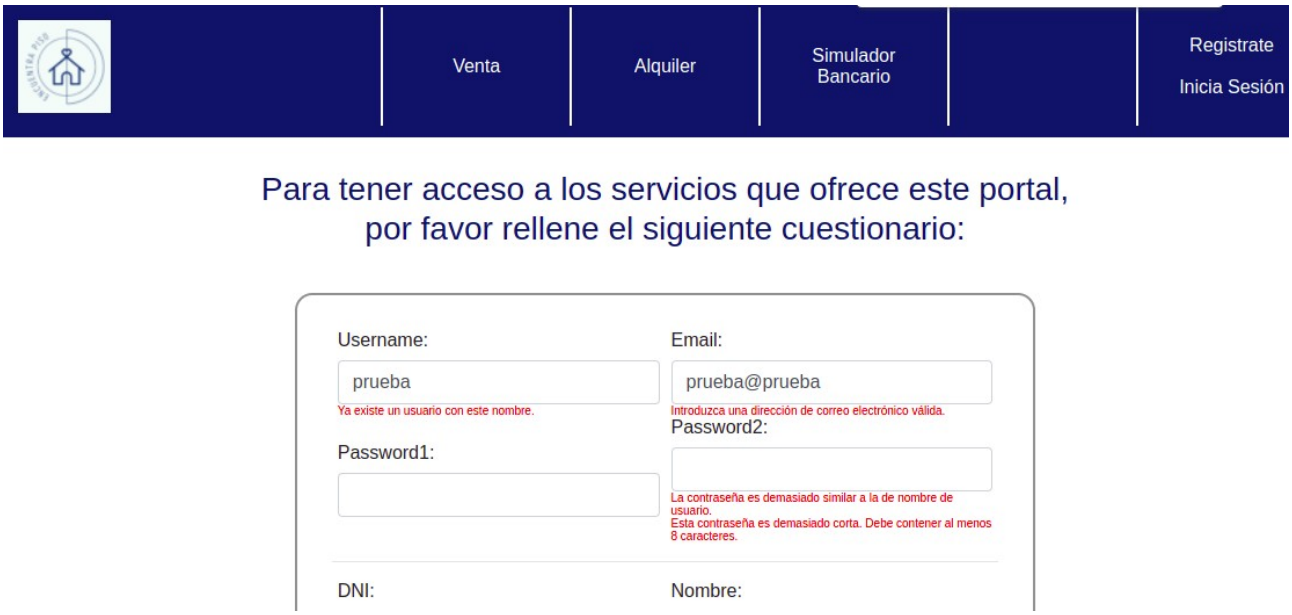
Para la base de datos se ha utilizado SQLite que es un sistema de gestión de bases de datos relacional y viene por defecto en Django.

3.2 Justificar la integridad y disponibilidad de los datos, especificando medidas de seguridad hacia los mismos.

- Registro de Usuarios:
  - Los campos pertenecientes a la clase User: Django dota a estos campos de su propia validación, y es por eso por lo que he decidido heredar las clases de usuario de esta. Por lo cual la contraseña se encripta automáticamente y el propio sistema comprueba que no haya otro usuario registrado con el mismo nombre. No obstante, con

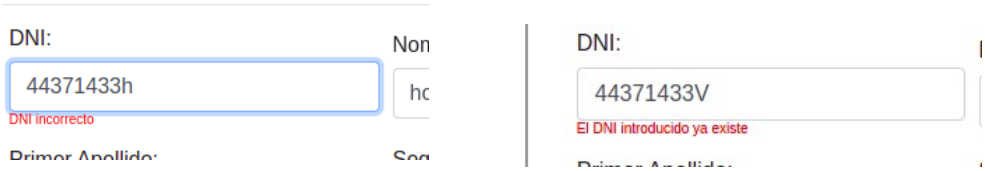
1 En el proyecto se adjuntará un fichero de requirements donde aparecerán todos los paquetes instalados en el desarrollo del proyecto.

JavaScript ,comprobamos que los dos campos de contraseña sean iguales antes de enviar al servidor.



Img.3.1: Comprobación de las alertas enviadas por el servidor en lo referente al registro del modelo User

- Otros campos del registro: el resto de campos del registro: dni, nombre, apellidos y teléfonos son validados con JavaScript antes de enviar al servidor, una vez en el servidor se volverá a validar si el dni es correcto y si no está duplicado antes de crear un nuevo usuario. Si hubiera errores tanto en la parte del cliente como del servidor se enviarán mensajes orientativos al usuario.



Img.3.2: A la izquierda vemos el mensaje enviado tras la comprobación por parte del cliente de que el formato del DNI no es correcto, a la derecha vemos como el servidor devuelve a la pantalla indicando que DNI introducido ya existe en el sistema

Para evitar la producción de errores el usuario solo introducirá datos a mano cuando es estrictamente necesario. Cada vez que se recogen datos manualmente se validan en el cliente y cuando se envían al servidor vuelven a comprobarse que los parámetros son correctos antes de crear nuevos registros en la base de datos.

3.3. Testeo

Para la realización de pruebas se han cargado en la base de datos los siguientes datos:

- 4 Inmobiliarias
- 6 Trabajadores
- 4 Propietarios
- 20 Inmuebles
- 20 Ofertas de Venta
- 20 Ofertas de Alquiler
- 2 Clientes.

Se han realizado pruebas de testeo con compañeros para probar las distintas funcionalidades y gracias a ello se han encontrado algunos errores que se han podido solucionar. Actualmente el código no presenta errores.

4. Bibliografía

- https://docs.python.org/es/3/
- https://docs.djangoproject.com/es/4.0/