

Índice de contenidos

Índice de contenidos	0
OBJETIVOS Y ALCANCE DEL PROYECTO	1
STACK TECNOLÓGICO Y ALTERNATIVAS EVALUADAS	2
Python:	2
Django:	2
HTML y CSS:	3
Bootstrap:	3
Java Script:	4
SQLite:	4
MODELO DE DATOS	5
Auth_user:	5
Usuarios:	5
Canchas:	6
Salones:	6
Alquileres:	6
Contactos:	6
Esquema:	7
REQUISITOS DE LA APLICACIÓN	8
Funciones principales:	10
Registro e inicio de sesión:	10
Alquilar:	12
Eliminar alquileres:	16
Usuario administrador:	17
Edición de usuario:	19
Eliminar usuario:	20
MANUAL DE INSTALACIÓN	22
CONCLUSIONES	24
EVOLUTIVOS DEL PROYECTO	25

DESCRIPCIÓN GENERAL DEL PROYECTO

En Uruguay, la cultura del fútbol está muy arraigada y es común entre grupos de amigos jóvenes (y no tanto), compañeros de trabajo, vecinos del barrio, compañeros de clase, etc., alquilar canchas en complejos deportivos para jugar como amigos, hacer torneos amateurs, etc. En este caso, contamos con una aplicación web que pertenece a un complejo deportivo (ficticio) de canchas de fútbol 5, 7 y 11. Este complejo contará con varias canchas de distintas capacidades y también con un salón de fiestas que se alquila para cumpleaños, eventos y demás.

OBJETIVOS Y ALCANCE DEL PROYECTO

La aplicación facilitará la agenda y el alquiler de las canchas de forma online a usuarios previamente registrados. Contará con dos tipos de usuarios, normales y administradores. Los usuarios normales, luego de registrarse, podrán generar reservas de canchas para días y turnos específicos. Los administradores podrán eliminar o editar estos alquileres, así como eliminar y editar usuarios desde un panel de administración dentro de la propia aplicación.

STACK TECNOLÓGICO Y ALTERNATIVAS EVALUADAS

Python:

Se utilizó Python para escribir los modelos de la base de datos, formularios, vistas y lógica de negocio. Django facilita la integración de Python en el desarrollo web mediante el uso de plantillas que emplean Jinja2, permitiendo integrar Python directamente en los archivos HTML cuando es necesario. En este caso, no se evaluaron alternativas, ya que Python era el lenguaje elegido sin dudas, aunque se utilizaron lenguajes complementarios detallados más adelante.

Django:

Como framework de desarrollo. Django cuenta con todo lo necesario (y más) para la creación rápida, limpia y eficiente de aplicaciones web usando Python. Ofrece una arquitectura MVC (Modelo Vista Controlador) con la que podemos diseñar nuestras aplicaciones basándonos en esos pilares principales. Además, proporciona una serie de herramientas integradas que brindan seguridad en el desarrollo y facilidad para implementarlas. Pudimos utilizar en este caso el sistema de inicio de sesión y registro que integra Django, junto con sus respectivos controles de seguridad y acceso, además del ORM que nos permite interactuar de manera sencilla con la base de datos. También utilizamos los modelos de formularios y Crispy Forms para utilizar formularios preestablecidos de Django. Como alternativa se evaluó el uso de Flask, que si bien es más minimalista y flexible, requiere más configuración y la integración de bibliotecas externas para

características avanzadas como la administración de bases de datos. Decidí usar Django debido a su estructura más completa, ofrece funcionalidades integradas y facilita el desarrollo rápido y seguro de aplicaciones web complejas, lo que puede ser más eficiente y menos propenso a cometer errores que construir todo desde cero en Flask.

HTML y CSS:

Se utilizaron para la creación y el diseño de las plantillas de la aplicación. CSS se empleó en conjunto con la siguiente herramienta para estilizar los elementos HTML y mejorar la apariencia de la aplicación.

Bootstrap:

Un framework front-end que se utilizó para construir y diseñar las vistas de usuario. Comencé el proyecto utilizando solo CSS personalizado, pero posteriormente se decidió integrar Bootstrap, ya que ofrece un conjunto de herramientas integradas y predefinidas que permiten crear diseños atractivos y, más importante aún, adaptables y responsivos, lo cual hoy en día es un estándar obligatorio dado que la mayoría de los consumidores utilizan dispositivos móviles. Se usaron varios elementos incluyendo botones, navbar, toggle buttons, contenedores, formularios y ventanas modales. La web es 90% responsiva (adaptable a pantallas móviles), excepto por las tablas del panel de administración que, si bien se adaptan a la pantalla, por la cantidad de datos no se ajustan al 100%.

Java Script:

Se utilizaron algunas líneas de JavaScript en el HTML para integrar funcionalidades que no eran alcanzables de manera óptima solo con HTML y CSS.

SQLite:

Se utilizó como motor de base de datos relacional para almacenar datos como la información de usuarios, alquileres y otros datos relevantes para la aplicación. Como alternativa, luego de estudiar diferentes tipos de BD, se consideró PostgreSQL. Para una app web pequeña o en desarrollo inicial, SQLite es preferible debido a que es más simple y requiere una configuración mínima. Es un motor de base de datos liviano y sin servidor, lo que facilita el desarrollo y la implementación rápida. A futuro sería posible migrar la BD en caso de que el proyecto escale al punto que ya no sea óptimo el uso de SQLite.

Este stack tecnológico permitió desarrollar una aplicación web robusta y escalable, ofreciendo una experiencia de usuario responsiva, fluida y atractiva.

MODELO DE DATOS

La base de datos utiliza 6 tablas de forma activa (tiene más que las crea Django de forma predeterminada, pero de estas solo usamos una que ya está incluida en las 6 mencionadas).

Auth_user:

Esta tabla es creada por Django de forma predeterminada para gestionar los usuarios. En ella se guardan los datos de registro del usuario, además de la fecha de registro y último inicio de sesión. Decidí usar esta tabla de forma predeterminada, ya que al ser nativa debería adaptarse con menos modificaciones para usar los datos y hacer sinergia con Django, lo que me ahorraría tiempo y evitaría posibles errores. Esta tabla también tiene una columna "is_staff" que distingue entre usuarios normales y administradores, y que se utilizará más adelante para restringir el acceso a ciertas páginas de la web y mostrar información solo a los administradores.

Usuarios:

Esta tabla se creó para que el usuario, al registrarse, ingrese su número de celular, dato que no está presente en la tabla auth_user. En lugar de modificar la tabla de Django, preferí crear esta y relacionarla con la tabla auth_user mediante una clave foránea (FK).

Canchas:

Aquí se almacenan las canchas del complejo con sus datos asociados como nombre, capacidad, precio por hora y tipo (fútbol 5, 7 u 11).

Salones:

Almacena los datos de un salón, similar a las canchas, incluyendo nombre, capacidad y precio por persona.

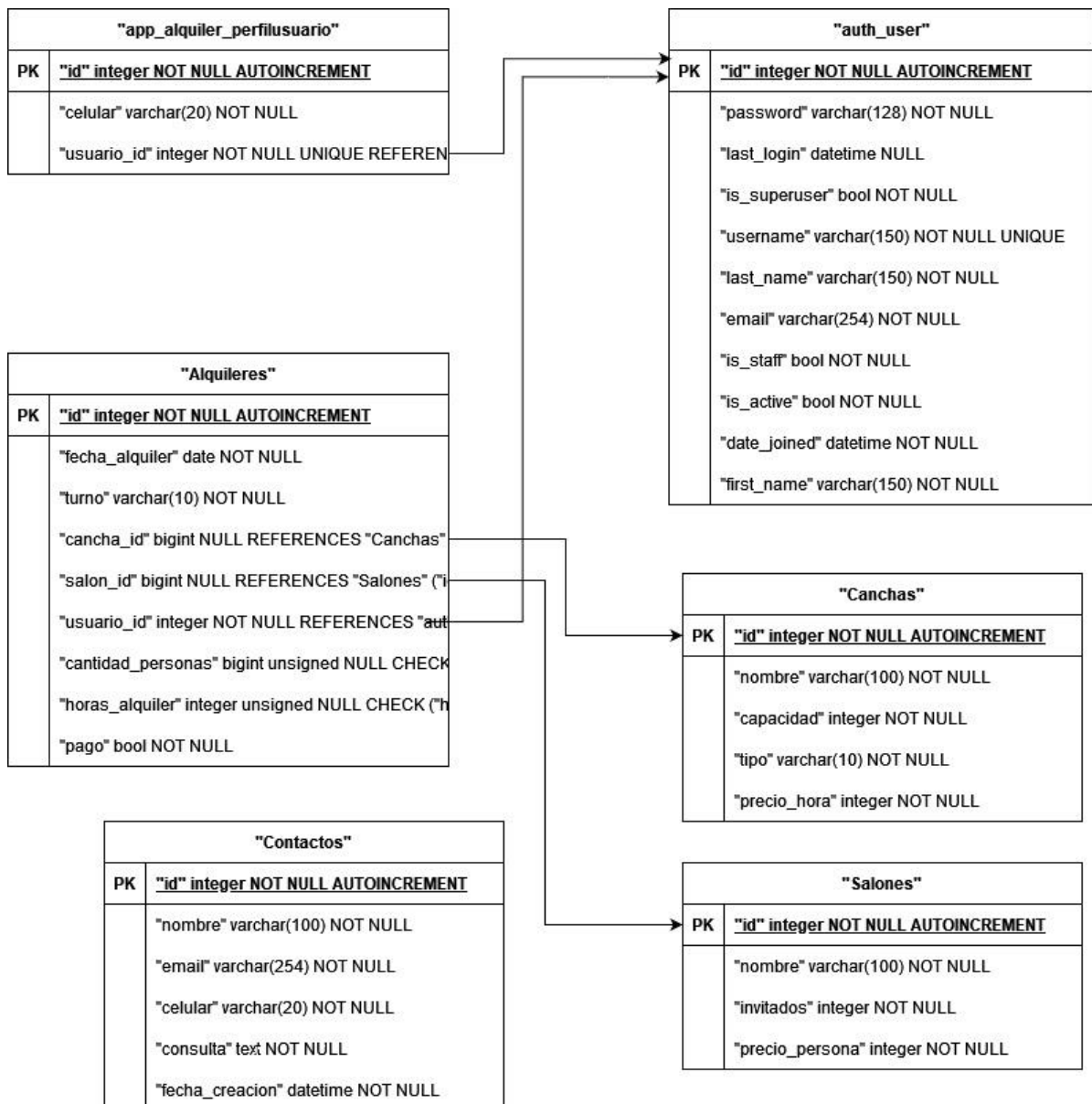
Alquileres:

Esta tabla guarda los datos de los alquileres generados. Cada alquiler se asocia a un usuario y una cancha vinculando los IDs correspondientes como FK. Guarda la fecha del alquiler, turno, ID de la cancha seleccionada o salón (con el otro campo nulo), ID del usuario que hizo la reserva, cantidad de personas para el salón u horas de alquiler para la cancha (uno de los dos campos es nulo) y un booleano para saber si el alquiler está pagado.

Contactos:

En esta tabla se almacenan las consultas realizadas mediante el formulario de contacto al final de la página principal. Guarda nombre, email, celular, consulta y fecha de creación. Este formulario es accesible para cualquier persona ya que no es necesario estar registrado para usarlo por ende no está relacionado con el resto de las tablas de la BD.

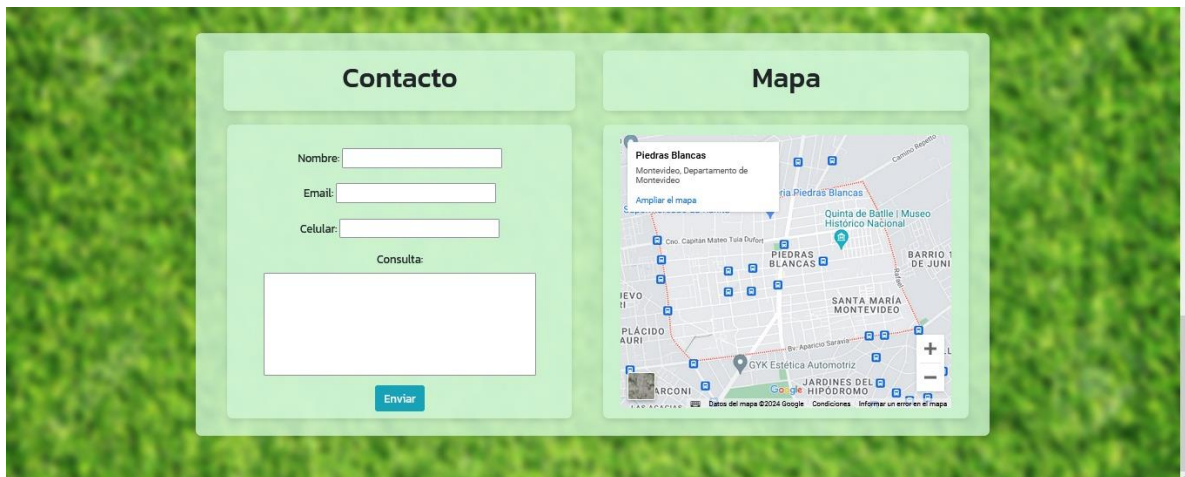
Esquema:



REQUISITOS DE LA APLICACIÓN

Página principal:





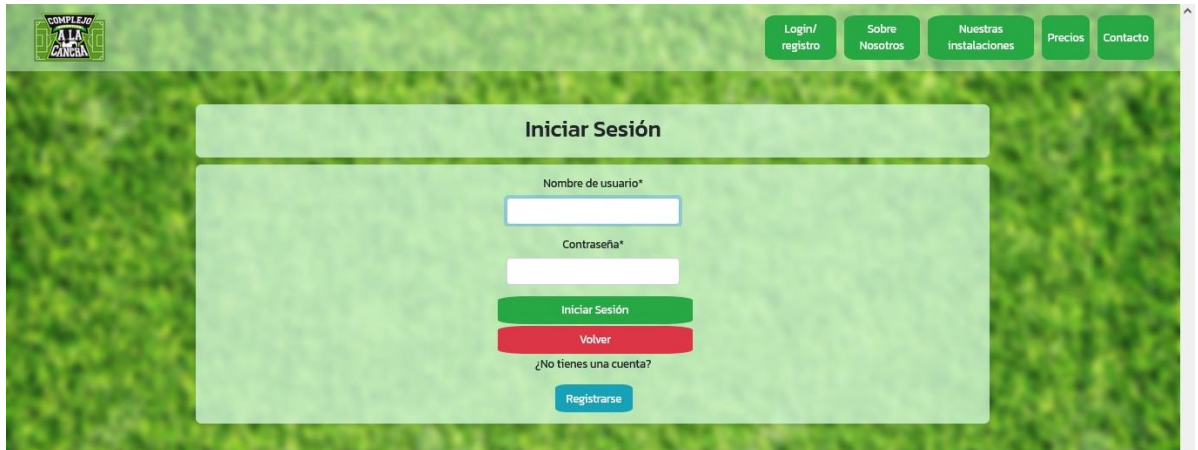
En la cabecera de la página principal se ve la barra de navegación que incluye:

- El logo a la izquierda, que al hacer clic desde cualquier vista de la web nos lleva a la página principal nuevamente.
- Botones "Sobre nosotros", "Nuestras instalaciones", "Precios" y "Contacto" que redirigen a diferentes secciones de la página principal automáticamente desde cualquier vista de la aplicación.
- Al final de la web tenemos el formulario de contacto, funcional al 100%, que almacena las consultas en la BD y el mapa de Google Maps, mostrando la ubicación del barrio donde se ubica el complejo.

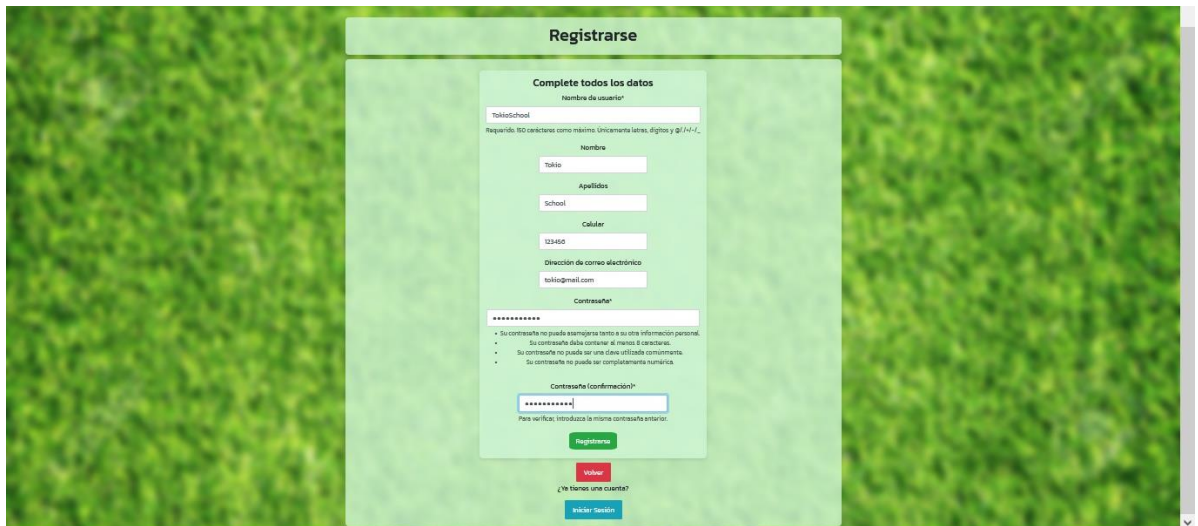
Funciones principales:

Registro e inicio de sesión:

Al acceder a "Login/registro" nos lleva a la página de inicio de sesión.



Si no disponemos de una cuenta, al hacer clic en el botón "registrarse" nos lleva a la página de registro.



En este caso crearemos un usuario, las credenciales de acceso serán:

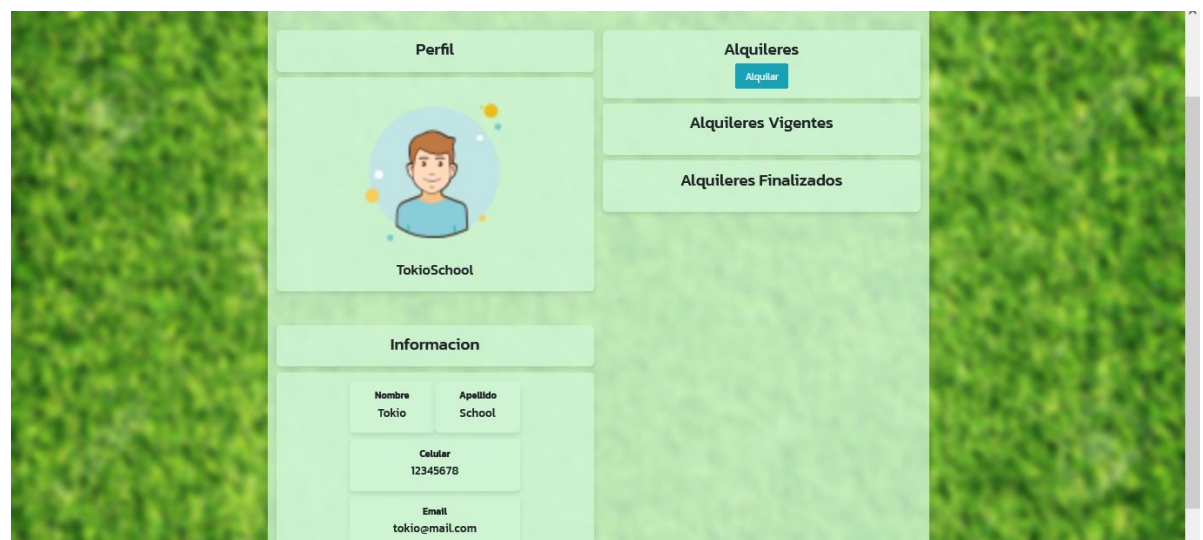
Usuario: TokioSchool

Contraseña: CursoPython

Al finalizar el registro nos devolverá a la página principal, pero con algunos cambios:



Como es de suponer "Cerrar sesión" desloguea al usuario y nos devuelve a la página principal inicial. Desde "Perfil" accedemos a la siguiente pantalla:



- En esta pantalla veremos una imagen de perfil con el nombre de usuario debajo, una sección de información donde muestra los datos del usuario logueado y una sección de alquileres donde se listarán los alquileres vigentes o aquellos cuya fecha aún no ha llegado, y otra sección donde se listarán los alquileres vencidos o cuya fecha ya haya pasado.

Alquilar:

Sobre esto, un botón de "Alquilar" que nos permitirá hacer lo propio en la siguiente pantalla:



Tendremos que clicar si queremos alquilar una cancha o un salón, en la misma página se desplegará el formulario correspondiente.

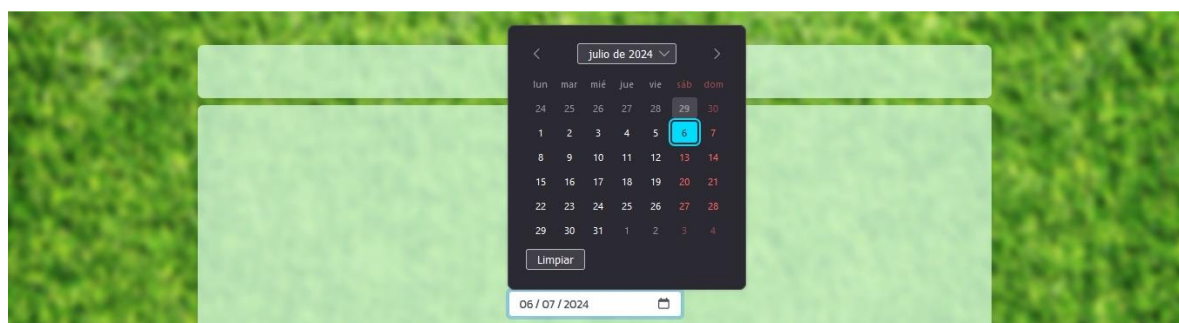
Vamos a alquilar una cancha:



Elegimos la cancha:



La fecha con el calendario:



El turno:

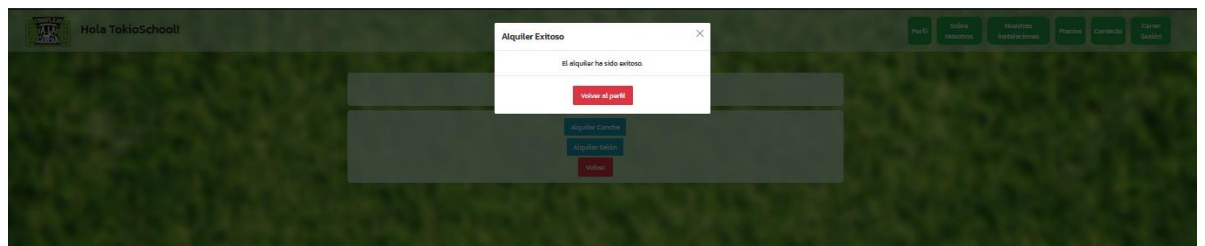


Por último, ingresamos las horas que vamos a utilizar la cancha y si la dejamos paga o pendiente:



The screenshot shows a web form titled "Alquiler" on a green grass background. The form has two tabs: "Alquiler Cancha" (active) and "Alquiler Salón". Below the tabs are buttons for "Volver" and "Cancha". A dropdown menu shows "Cancha F7 numero 1". The "Fecha alquiler*" field is set to "06 / 07 / 2024". The "Turno*" dropdown is set to "Tarde". The "Horas alquiler" field is set to "4". Under the "Pago" section, the "Paga" radio button is selected. At the bottom is a green "Alquiler Cancha" button.

Al clicar en "Alquilar cancha" nos mostrará una ventana modal de éxito y nos devuelve a la pantalla de alquiler.



Vamos a dejar hecho otro alquiler, pero esta vez de un salón y pago para ver las diferencias:



The screenshot shows the "Alquiler" form with the "Alquiler Salón" tab selected. The "Volver" button now says "Salon". The dropdown menu shows "Salon Principal". The "Fecha alquiler*" field is set to "05 / 07 / 2024". The "Turno*" dropdown is set to "Noche". The "Cantidad personas" field is set to "70". Under the "Pago" section, the "Pago" radio button is selected. At the bottom is a green "Alquiler Salón" button.

Ya en el perfil de usuario ahora nos aparecen listados los alquileres los resultados de estos listados son los 5 alquileres más recientes del usuario (esto lo limité desde el bucle for en el HTML):



Uno de los alquileres lo hicimos fuera de fecha para que quede en los finalizados.

Como podemos ver, los alquileres muestran el costo total del alquiler ya que, en base al precio/hora o precio/persona y las horas o cantidad de personas (respectivamente), calcula el monto a abonar. Esto es gracias a una función en el modelo alquiler que compara estos datos y hace los cálculos necesarios, al igual que otra función que distingue si el alquiler esta pasado de fecha o no.

Aquí vemos cómo están todos los alquileres en la BD incluyendo los últimos 3 que ingresamos al final de todos (68, 69, 70):

Table: Alquileres									
	id	fecha_alquiler	turno	cancha_id	salon_id	usuario_id	cantidad_personas	horas_alquiler	pago
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	5	2024-06-14	noche	2	NULL	8	NULL	2	0
2	6	2024-06-14	tarde	1	NULL	8	NULL	2	0
3	7	2024-06-02	tarde	NULL	1	8	6	NULL	0
4	18	2024-05-31	manana	1	NULL	5	NULL	1	0
5	55	2024-06-21	manana	NULL	1	5	60	NULL	0
6	56	2024-06-21	tarde	1	NULL	5	NULL	2	0
7	57	2024-06-15	manana	NULL	1	5	10	NULL	0
8	61	2024-06-29	tarde	NULL	1	5	76	NULL	0
9	62	2024-07-03	noche	2	NULL	5	NULL	3	0
10	63	2024-06-25	manana	4	NULL	5	NULL	1	1
11	68	2024-07-06	tarde	3	NULL	15	NULL	4	1
12	69	2024-07-05	noche	NULL	1	15	70	NULL	0
13	70	2024-06-14	noche	NULL	1	15	52	NULL	1

Eliminar alquileres:

Al darle click al icono de basura se elimina el alquiler en esa línea.
Antes de eliminarlo nos aparecerá una confirmación de si estamos seguros:



Al aceptar se elimina el alquiler y se actualiza la vista del perfil nuevamente, pero esta vez sin el alquiler que hemos eliminado.



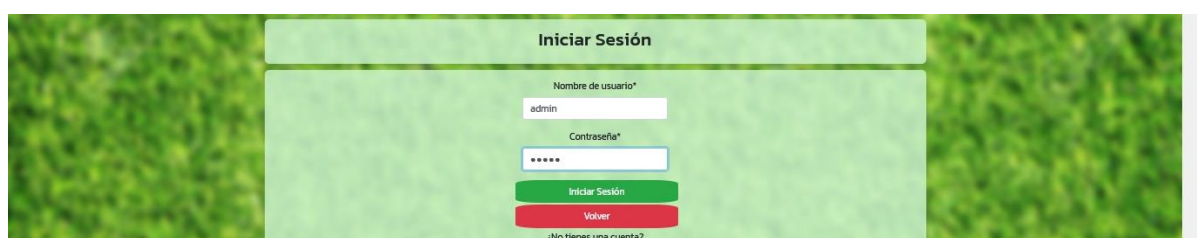
Por el momento, esto es lo que podemos hacer desde el perfil de usuario normal.

Usuario administrador:

Vamos a cerrar sesión e iniciar sesión como administrador, esto se determina porque en la BD El campo "is_staff" es True. Las credenciales de administrador son:

Usuario: admin

Contraseña: admin



Nos llevará a la página principal nuevamente, pero esta vez con un cambio más:



En esta pantalla veremos una tabla con TODOS los usuarios almacenados en la base de datos, TODOS los alquileres vigentes y TODOS los alquileres finalizados.

The screenshot shows the 'Administracion' (Administration) panel. It features a table titled 'Usuarios' (Users) with the following columns: Id, Usuario, Nombre completo, Celular, Email, Editar, and Eliminar. The table contains 12 rows of user data.

Id	Usuario	Nombre completo	Celular	Email	Editar	Eliminar
2	diegoprueba22	diegop claverie	095030309	diegoprueba@mail.com		
3	eldieego	diegop claverie	num095030309	diego@mail.com		
4	flavia	flavia hernandez	095725851	flavi@flavi.com		
5	ambarclave	ambar claverie	123456	ambar@mail.com		
6	dieguinho	diegop claverie	no	diegoprueba@mail.com		
8	martinR	Martin Roldan	095	mr@mail.com		
9	admin	Administrador administrador		admin@admin.com		
10	Ezio	Ezio Auditore	09503039	ezio@mail.com		
11	ramitagram	ramira patrona del mal	12345	ramira@mail.com		
15	TokioSchool	Tokio School	12345678	tokio@mail.com		

Alquileres								
Alquileres Vigentes								
Cancha/Salon	Fecha	Turno	Nombre	Horas	Invitados	Pago	Editar	Eliminar
Salón: Salon Principal	29 de junio de 2024	tarde	ambar	-	76	✗		
Cancha: Cancha FS numero 2	3 de julio de 2024	noche	ambar	3	-	✗		
Salón: Salon Principal	5 de julio de 2024	noche	Tokio	-	70	✅		
Alquileres Finalizados								
Cancha/Salon	Fecha	Turno	Nombre	Horas	Invitados	Pago	Editar	Eliminar
Cancha: Cancha FS numero 1	31 de mayo de 2024	manana	ambar	1	-	✗		
Salón: Salon Principal	2 de junio de 2024	tarde	Martin	-	6	✗		
Cancha: Cancha FS numero 2	14 de junio de 2024	noche	Martin	2	-	✗		
Cancha: Cancha FS numero 1	14 de junio de 2024	tarde	Martin	2	-	✗		
Salón: Salon Principal	14 de junio de 2024	noche	Tokio	-	52	✅		
Salón: Salon Principal	15 de junio de 2024	manana	ambar	-	10	✗		

Al igual que en el perfil de usuario, aquí podemos eliminar alquileres y usuarios, pero también editar los mismos.

Edición de usuario:

Vamos a editar un usuario, el de ID 6:

5	ambarclave	ambar claverie	123456	ambar@mail.com		
6	dieguinho	diegop claverie	no	diegoprueba@mail.com		
8	martinR	Martin Roldan	095	mr@mail.com		

Al darle click al botón editar nos lleva a esta página que nos muestra el mismo formulario de registro:

Editando informacion de dieguinho

Nombre de usuario*

Requerido. 150 caracteres como máximo. Únicamente letras, dígitos y @/./-/_/

Nombre

Apellidos

Celular

Dirección de correo electrónico

Contraseña*

- Su contraseña no puede asemejarse tanto a su otra información personal.
- Su contraseña debe contener al menos 8 caracteres.
- Su contraseña no puede ser una clave utilizada comúnmente.
- Su contraseña no puede ser completamente numérica.

Contraseña (confirmación)*

Para verificar, introduzca la misma contraseña anterior.

[Guardar Cambios](#)

Vamos a cambiar el nombre de usuario a "Dieguito" y las mayúsculas del nombre y apellido, también el celular, al guardar los cambios nos muestra una ventana de confirmación:



Y nos lleva de nuevo al perfil de administrador y ya se muestran los datos nuevos:

5	ambarclave	ambar claverie	123456	ambar@gmail.com		
6	dieguito	Diego Claverie	123456789	diegoprueba@gmail.com		
8	martinR	Martin Roldan	095	mr@gmail.com		

Eliminar usuario:

De la misma forma que eliminamos los alquileres podemos eliminar los usuarios. Solo hay que darle click al botón de eliminar, nos pedirá confirmar, eliminemos el usuario con ID 6:



En caso afirmativo se eliminará el usuario de la BD y al recargar la página este ya no aparecerá:

4	flavia	flavia hernandez	095726851	flavi@flavi.com		
5	ambarclave	ambar claverie	123456	ambar@gmail.com		
8	martinR	Martin Roldan	095	mr@gmail.com		

Como medida de seguridad, usamos la información de "is_staff" de la tabla auth_user para saber si el usuario que desea ingresar a "perfiladmin" es efectivamente un admin. En caso que no lo sea y por X razón alguien intenta acceder a esa URL de forma manual, aparecerá esta página en lugar de "perfiladmin":



Esta es toda la funcionalidad hasta ahora de la aplicación.

MANUAL DE INSTALACIÓN

Usuario normal:

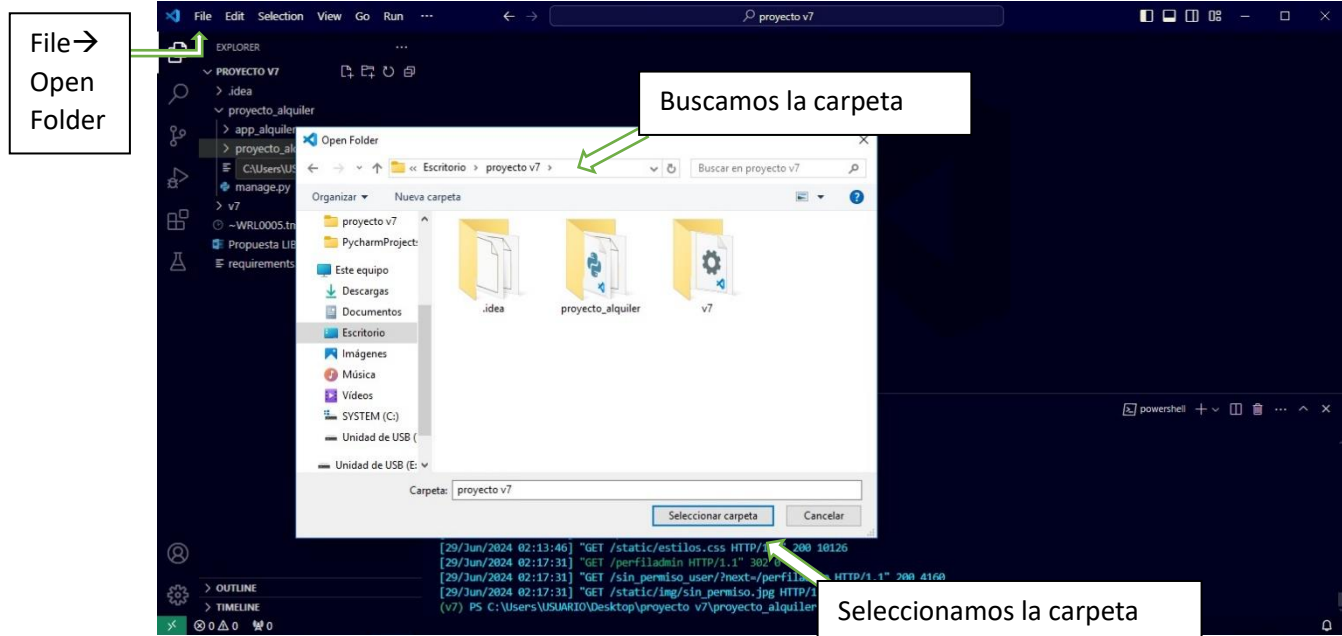
Usuario: TokioSchool Contraseña: CursoPython

Usuario administrador:

Usuario: admin Contraseña: admin

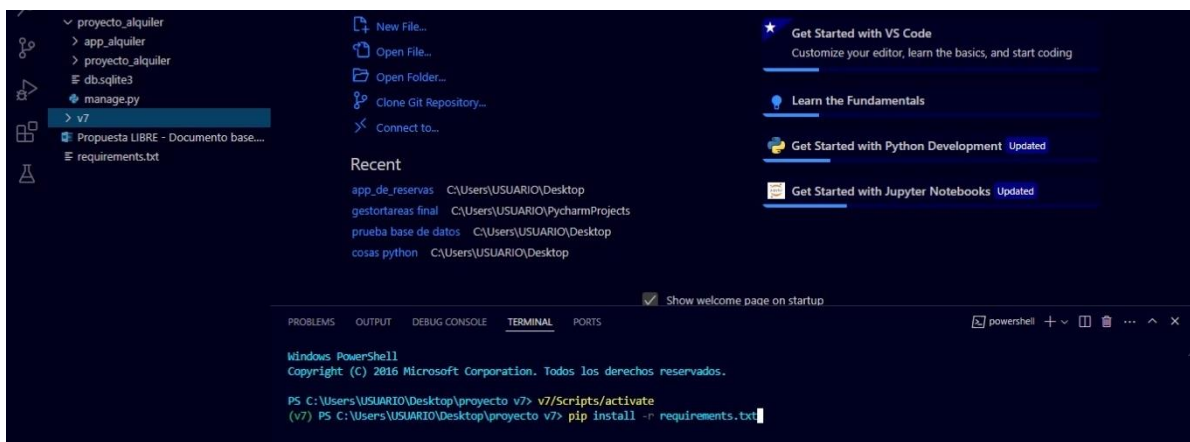
Hay más usuarios hechos en la BD para que se pueda probar la funcionalidad, la contraseña de todos ellos es "mayonesa".

La instalación de la app es bastante simple. En primer lugar, hay que abrir la carpeta del proyecto desde el IDE que vayamos a usar (yo usé VSC). Vamos a "File" -> "Open Folder..." -> buscamos la carpeta del proyecto (en mi caso es la carpeta "proyecto v7"):

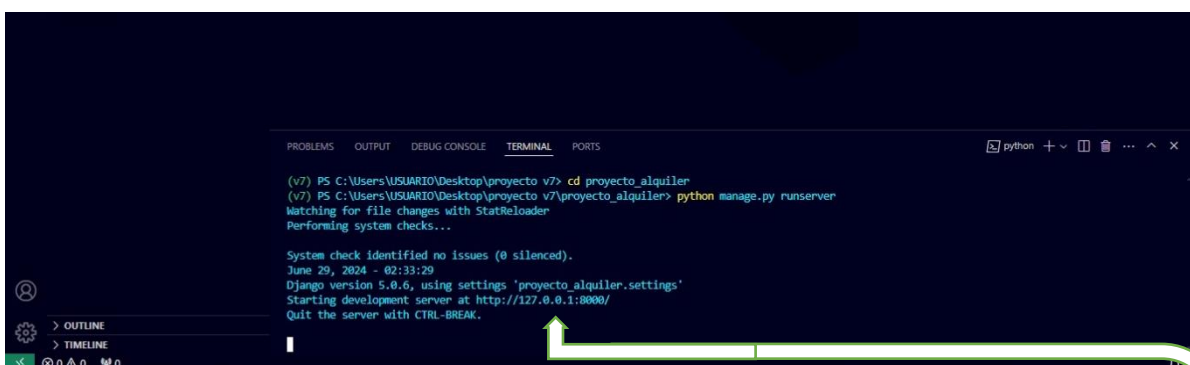


Ya con la carpeta abierta, hay que ejecutar el terminal y correr los siguientes comandos.

- Primero activamos el entorno virtual con "v7/Scripts/activate"
- Ya con el entorno activado, instalamos las dependencias necesarias con el requirements.txt:



- Con el entorno activado y todo instalado, procedemos a ir a la carpeta principal del proyecto con "cd proyecto_alquiler" y después ya podemos correr el server de Django con "python manage.py runserver":



Ya con esto, nos quedará el servidor corriendo y seremos capaces de abrir la página principal del proyecto con Ctrl y click sobre el servidor. ←

CONCLUSIONES

Como resultado tenemos una aplicación web funcional, con una base de datos que usa relaciones internas, en algunos casos con varias tablas a la vez, de fácil uso para el usuario y funcional para el administrador.

Disponemos de un buen nivel de seguridad con contraseñas cifradas gracias a Django.

El proyecto es escalable a futuro si la base de datos deja de ser funcional o se necesita mayor potencia, SQLite es fácilmente migrable a PostgreSQL lo que nos permitiría una mayor capacidad.

Es responsiva e intuitiva.

EVOLUTIVOS DEL PROYECTO

- Posibilidad de que el usuario pueda subir una foto de perfil personalizada.
- Agregar al panel del administrador una forma de acceder a las consultas echas desde el formulario, en este momento solo se podrían ver desde la BD o desde la pagina de administrador de Django (<http://127.0.0.1:8000/admin>) de la cual no quise disponer para el uso final, sino que quiero que se maneje todo desde la app.
- Agregar una forma de pago en línea para el usuario al marcar la opción Pago al momento de realizar el alquiler.
- Darle la posibilidad al usuario para poder modificar sus datos y sus alquileres desde su perfil. En este momento solo es posible modificar desde la vista del administrador.
- Se puede mejorar la interfaz gráfica, por ejemplo, las tablas de la vista administrador. A medida que se agrande la cantidad de datos en los registros van a quedar enormes en pantalla. Creo que sería posible agregarle páginas, limitar los resultados o filtrarlos sea por fecha, usuario, etc antes que se carguen. Quizás agregarle un buscador para ver cosas de un usuario especifico, fecha específica, cancha o salón.
- Agregar en el perfil de usuario una opción de "organizar torneo" o "liga" desde la cual el usuario mismo pueda organizar torneos con otros equipos de otros usuarios y pueda armar un "fixture" online de los partidos.

- Evitar que un mismo salón/cancha pueda ser reservado más de una vez el mismo día y turno lo que generaría errores de organización.
- Agregar una comprobación sobre la fecha para que no se puedan realizar alquileres con fechas en el pasado.

Estos últimos 2 puntos no los lleve a cabo por falta de tiempo.