

Manual del Backend

En este manual se explica el funcionamiento del backend de la aplicación.

Tecnologías

El backend está construido con:

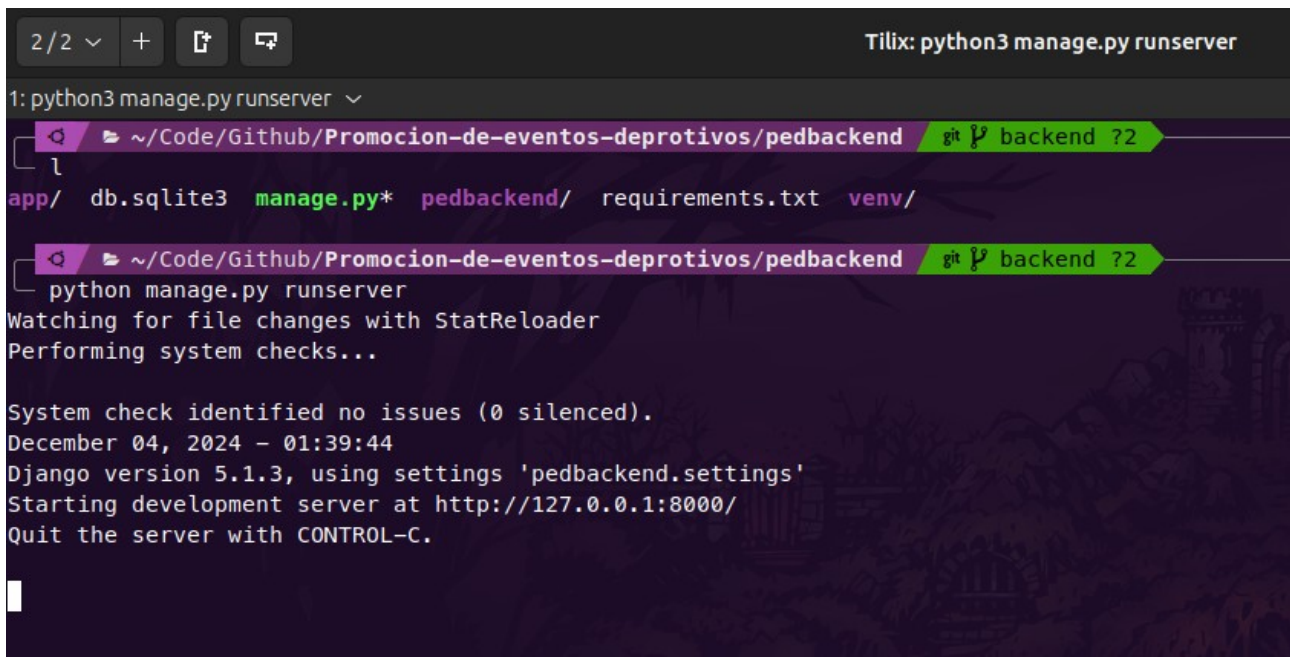
- Python 3.12.3
- Django 5.1.3
- SQLite 3

Ejecución

Para ejecutar el backend de la aplicación se debe tener en cuenta que se tienen las librerías de python requeridas. Las librerías necesarias se pueden instalar usando pip ejecutando el comando: *pip install -r requirements.txt*.

Una vez están instaladas las librerías necesarias, dentro del directorio llamado *pedbackend/* se encontrará un archivo llamado *manage.py*. Se debe situar allí la terminal y ejecutar el comando *python manage.py runserver*. Este comando ejecutará el servidor backend en el localhost (**127.0.0.1**) en el puerto **8000**, y quedará atento a solicitudes.

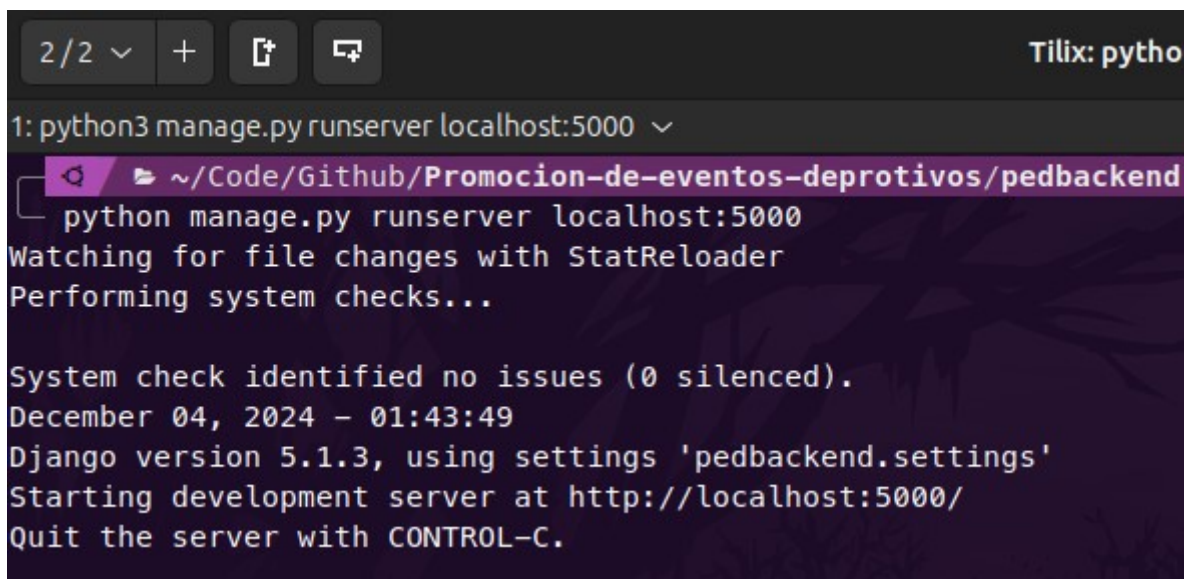
Se verá de esta forma:



```
Tilix: python3 manage.py runserver
1: python3 manage.py runserver
~/Code/Github/Promocion-de-eventos-deprotivos/pedbackend
app/ db.sqlite3 manage.py* pedbackend/ requirements.txt venv/
~/Code/Github/Promocion-de-eventos-deprotivos/pedbackend
python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
December 04, 2024 - 01:39:44
Django version 5.1.3, using settings 'pedbackend.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

El puerto puede ser cambiado a voluntad añadiendo como argumento al comando la IP y puerto deseado. Se vería de la siguiente forma:

A terminal window titled 'Tilix: python' showing the command 'python3 manage.py runserver localhost:5000' being executed. The terminal output includes the directory path '~ /Code/Github/Promocion-de-eventos-deprotivos/pedbackend', the command 'python manage.py runserver localhost:5000', and the Django development server startup sequence: 'Watching for file changes with StatReloader', 'Performing system checks...', 'System check identified no issues (0 silenced)', the date and time 'December 04, 2024 - 01:43:49', the Django version 'Django version 5.1.3, using settings \'pedbackend.settings\'', the server address 'Starting development server at http://localhost:5000/', and the instruction 'Quit the server with CONTROL-C.'

```
2/2  +  [ ]  [ ]  Tilix: python
1: python3 manage.py runserver localhost:5000  v
~/Code/Github/Promocion-de-eventos-deprotivos/pedbackend
python manage.py runserver localhost:5000
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
December 04, 2024 - 01:43:49
Django version 5.1.3, using settings 'pedbackend.settings'
Starting development server at http://localhost:5000/
Quit the server with CONTROL-C.
```

URLs

El backend responde ante 2 URLs, las cuales son:

- **/json_events/**: Solo solicitudes GET, no requiere argumentos y devuelve una respuesta en json con todos los datos registrados en la tabla *Events* de la base de datos. Cada elemento en la tabla será devuelto de la siguiente forma:

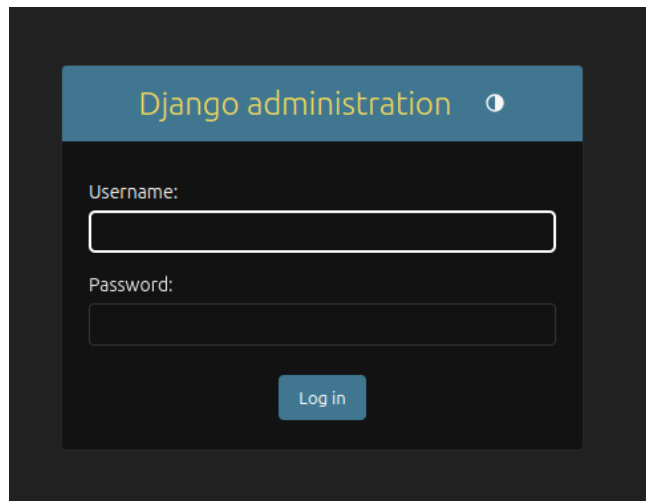
```
{
  "name": "Clásico Bogotano de Fútbol",
  "date": "2023-09-10",
  "location": "Parque Simón Bolívar",
  "requirements": "Inscripción previa",
  "sport": "Fútbol",
  "limit_date": "2025-01-01",
  "min_age": 18
},
```

- **/admin/**: Portal de administración de Django.

/admin/

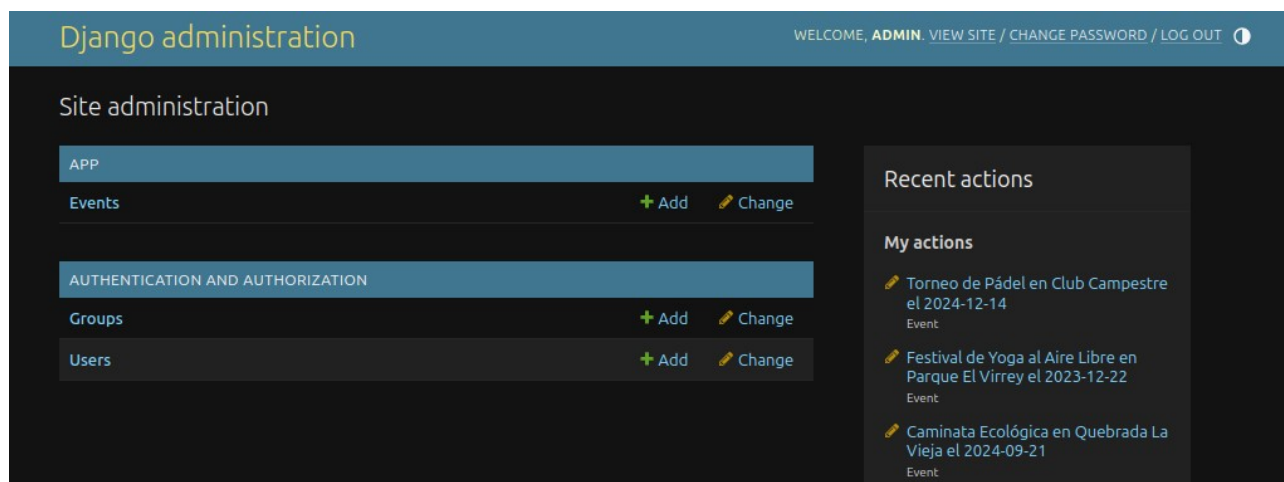
A través de la interfaz de administración de django se pueden manejar los registros de forma más segura e intuitiva, sin tener que modificar directamente la base de datos.

Al acceder a la ruta se mostrará la siguiente ventana de login:



Las credenciales para el inicio de sesión como administrador son **admin** como usuario y contraseña. Estas pueden ser cambiadas dentro de la interfaz de administración.

Una vez se ingresa a la interfaz, se verá de la siguiente forma:



A la izquierda se encuentra una opción llamada “Events” dentro de la aplicación “APP”. Al dar click izquierdo sobre “Events”, la interfaz mostrará todos los registros de la tabla “Events” de la base de datos. Se verá de la siguiente forma:

Django administration
WELCOME, ADMIN · [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home » App » Events

Select event to change
ADD EVENT +

Action: ----- 0 of 10 selected

<input type="checkbox"/>	NAME	DATE	1 ▲	LIMIT DATE	LOCATION	2 ▲	SPORT	3 ▲	MIN AGE
<input type="checkbox"/>	Clásico Bogotano de Fútbol 7	March 15, 2024		March 5, 2024	Canchas del Parque El Virrey		Fútbol		16
<input type="checkbox"/>	Torneo de Baloncesto 3x3	April 20, 2024		April 15, 2024	Canchas del Parque Nacional		Baloncesto		15
<input type="checkbox"/>	Torneo de Tenis Dobles	June 15, 2024		June 5, 2024	Club Campestre de Bogotá		Tenis		14
<input type="checkbox"/>	Acuación Parque El Virrey	July 20, 2024		July 10, 2024	Lago del Parque El Virrey		Natación		16
<input type="checkbox"/>	Torneo de Fútbol Playa	Sept. 14, 2024		Sept. 4, 2024	Playa Salitre		Fútbol		16
<input type="checkbox"/>	Copa Bogotá de Baloncesto	Oct. 19, 2024		Oct. 10, 2024	Coliseo El Campín		Baloncesto		18
<input type="checkbox"/>	Torneo de Tenis de Mesa	Nov. 16, 2024		Nov. 5, 2024	Coliseo cubierto de Teusaquillo		Tenis		12
<input type="checkbox"/>	Travesía a Nado	Feb. 15, 2025		Feb. 5, 2025	Lago de Guatavita		Natación		18
<input type="checkbox"/>	Copa Ciudad de Bogotá de Fútbol	April 12, 2025		April 2, 2025	Estadio Metropolitano		Fútbol		18
<input type="checkbox"/>	Torneo de Baloncesto Juvenil	May 17, 2025		May 7, 2025	Coliseo cubierto de Teusaquillo		Baloncesto		14

FILTER

Show counts

By date

Any date

Today

Past 7 days

This month

This year

By location

All

Canchas del Parque El Virrey

Canchas del Parque Nacional

Club Campestre de Bogotá

Coliseo El Campín

Coliseo cubierto de Teusaquillo

Estadio Metropolitano

Lago de Guatavita

Lago del Parque El Virrey

Playa Salitre

Crear un Nuevo Evento

Sí se desea ingresar un nuevo evento, se debe dar click sobre el botón “ADD EVENT +” que se encuentra en la parte superior derecha. Al hacerlo, se mostrará un formato que debe ser completado para crear un nuevo evento. Una vez se complete el formato, se debe dar click sobre el botón “SAVE”.

Django administration
WELCOME, ADMIN

Home » App » Events » Add event

Add event

Name:

Date: Today

Note: You are 5 hours behind server time.

Location:

Requirements:

Sport: ----- ▼

Limit date: Today

Note: You are 5 hours behind server time.

Min age:

Modificar y Borrar un Evento

En la página de eventos, los eventos aparecen con el nombre del evento en azul. Estos son enlaces clickeables que llevarán a la interfaz de modificación del evento seleccionado.

Se verá de la siguiente forma:

The screenshot shows the Django administration interface for a 'Change event' form. The form is titled 'Clásico Bogotano de Fútbol 7 en Canchas del Parque El Virrey el 2024-03-15'. It contains several fields: 'Name' (Clásico Bogotano de Fútbol 7), 'Date' (2024-03-15), 'Location' (Canchas del Parque El Virrey), 'Requirements' (Inscripción previa y equipo completo), 'Sport' (Fútbol), 'Limit date' (2024-03-05), and 'Min age' (16). At the bottom, there are buttons for 'SAVE', 'Save and add another', 'Save and continue editing', and 'Delete'.

Una vez aquí, los datos pueden ser reemplazados, sin embargo se debe tener en cuenta los tipos de datos que acepta la tabla en la base de datos.

Esta tabla fue creada usando el siguiente modelo, lo cual puede ayudar a aclarar que tipo de dato acepta cada campo:

```
File: app/models.py
1  from django.db import models
2
3  # Create your models here.
4
5  class Event(models.Model):
6
7      SPORT_CHOICES = [
8          ('Fútbol', 'Fútbol'),
9          ('Baloncesto', 'Baloncesto'),
10         ('Tenis', 'Tenis'),
11         ('Natación', 'Natación')
12     ]
13
14     id = models.AutoField(primary_key=True)
15     name = models.CharField(max_length=255) # Nombre del evento
16     date = models.DateField() # Fecha del evento
17     location = models.CharField(max_length=255) # Lugar del evento
18     requirements = models.CharField(max_length=255) # Requisitos del evento
19     sport = models.CharField(max_length=255, choices=SPORT_CHOICES, null=False) # Deporte del evento
20     limit_date = models.DateField() # Fecha fin de inscripcion
21     min_age = models.SmallIntegerField() # Edad mínima para participar
```

Es importante tener en cuenta que, debido a lo anterior, no se pueden guardar valores de tipo "texto" en los campos **date**, **limit_date** y **min_age**. Para **date** y **limit_date** solo aceptarán valores que concuerden con el formato de fechas establecido y **min_age** solo aceptará números enteros desde el -32768 hasta el 32767. Para el campo **sport** se han establecido 4 valores posibles, y se ha deshabilitado la posibilidad de aceptar un valor nulo.