

Sched It

Implementa un proyecto web Django que permita gestionar citas (no románticas – aunque también valdría –).

1. Puesta en marcha

Lleva a cabo los siguientes comandos para la puesta en marcha del proyecto:

```
just create-venv
source .venv/bin/activate
just setup
```

¿Qué ha ocurrido?

- Se ha creado un entorno virtual en la carpeta `.venv`
- Se han instalado las dependencias del proyecto.
- Se ha creado un proyecto Django en la carpeta `main`
- Se han aplicado las migraciones iniciales del proyecto.
- Se ha creado un superusuario con credenciales: `admin - admin`

2. Aplicaciones

Habrás que añadir las siguientes aplicaciones:

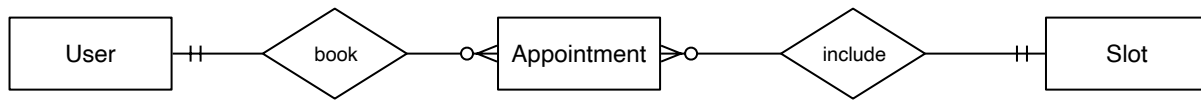
<code>accounts</code>	Gestión de autenticación.
<code>appointments</code>	Citas.
<code>slots</code>	Franjas horarias.

Se proporciona una *receta* `just` para añadir una aplicación:

```
just startapp <app>
```

Esta receta no sólo crea la carpeta de la aplicación sino que añade la línea correspondiente de configuración en la variable `INSTALLED_APPS` del fichero `settings.py`.

3. Modelos



Un usuario (`User`) reserva citas (`Appointment`) y una cita incluye franjas horarias (`Slot`).

3.1. `slots.Slot`

Campo	Tipo
<code>start_at</code>	<i>time</i>
<code>end_at</code>	<i>time</i>

Notas:

- La *hora de comienzo* y la *hora de finalización* **juntos** deben ser valores únicos.
- La ordenación por defecto debe ser por `start_at` y `end_at`, ambos de forma *ascendente*.
- Implementa la representación del objeto en formato: `hh:mm - hh:mm`

3.2. `appointments.Appointment`

Campo	Tipo
<code>date</code>	<i>datetime</i>

Notas:

- Debes tener en cuenta las claves ajenas necesarias (*si procede*). [No te olvides de `related_name`](#).
- La fecha y el “slot” **juntos** deben ser valores únicos.
- La ordenación por defecto debe ser por fecha y slot, ambos de forma *ascendente*.
- Implementa la representación del objeto en formato: `yyyy-mm-dd (hh:mm - hh:mm)`
- Implementa su URL canónica.

3.3. `User`

No hay que implementar este modelo. Se usará el modelo `User` que ofrece Django.

Carga de datos

Una vez que hayas creado los modelos y aplicado las migraciones, puedes cargar datos de prueba con la siguiente *receta* [just](#):

```
just load-data
```

4. URLs

/login/

- Inicio de sesión.
- Se debe solicitar nombre de usuario y contraseña.

/logout/

- Cierre de sesión.
- Mediante petición **GET**.

/signup/

- Registro de usuario.
- Campos a desplegar en el formulario:
 - Nombre de usuario.
 - Contraseña.
 - Nombre.
 - Apellidos.
 - Correo electrónico.

/

- Debe redirigir a `/appointments/`

/appointments/

- Listado con las citas del usuario logeado.
- El formato de cita debe ser: `yyyy-mm-dd (hh:mm - hh:mm)`
- Debe existir un enlace al detalle de cada cita.
- Debe existir un enlace para añadir una nueva cita.
- La ordenación de las citas vendrá *por defecto* a partir del modelo `Appointment`.
- Sólo debe ser accesible si el usuario está logeado.

/appointments/34/

- Detalle de la cita con `pk=34`
- Debe aparecer la fecha con formato: `Date: yyyy-mm-dd`
- Debe aparecer el “slot” con formato: `Slot: hh:mm - hh:mm`
- Debe aparecer un enlace para volver al listado de citas.
- Si un usuario trata de acceder a una cita que no le pertenece, hay que devolver un 403.
- Sólo debe ser accesible si el usuario está logeado.

</appointments/add/>

- Añadir una nueva cita.
- Desplegar un **formulario de modelo** con fecha y “slot”.
- Si la cita se añadió con éxito redireccionar al listado de citas.
- Sólo debe ser accesible si el usuario está logeado.

5. Administración

Los siguientes modelos deben estar accesibles desde la **interfaz administrativa** de Django:

- `appointments.Appointment`
- `slots.Slot`