

Yummi

Implementa un proyecto web Django para **recetas de cocina**.

Este software permitirá dar de alta recetas mediante la composición de ingredientes.

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:

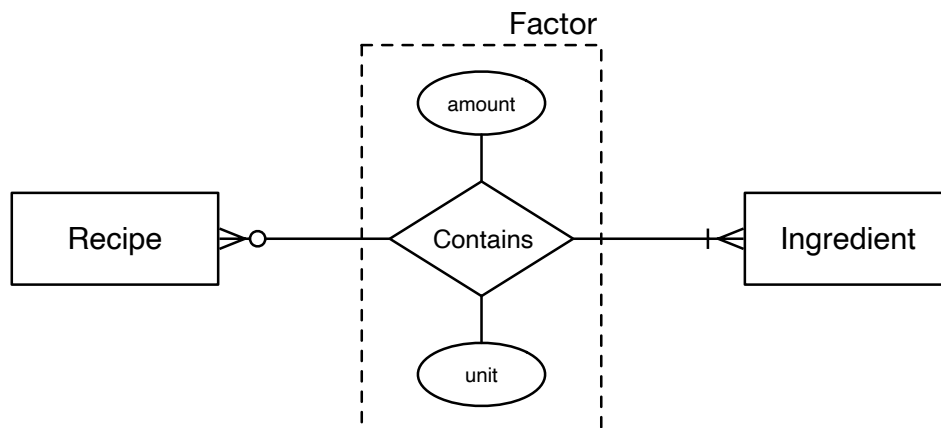
<code>shared</code>	Artefactos comunes.
<code>ingredients</code>	Gestión de ingredientes.
<code>recipes</code>	Gestión de recetas.

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



3.1. recipes.Recipe

Campo	Tipo
name ^(*)	str
script ^(*)	str
ingredients ^(∅)	m2m → Ingredient (Factor)

3.2. recipes.Factor

Campo	Tipo
recipe ^(*)	fk → Recipe
ingredient ^(*)	fk → Ingredient
amount ^(*)	float
unit ^(*)	enum

- La unidad (unit) será un [enumerado](#) con valores:
 - PIECE = 'P', 'Piece'
 - LITER = 'L', 'Liter'
 - GRAM = 'G', 'Gram'
- La receta y el ingrediente (juntos) deben ser únicos.

3.3. ingredients.Ingredient

Campo	Tipo
name ^(*)	str
description ^(∅)	str

3.4. 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

4.1. main.urls

/

- Contiene enlace a `/ingredients/`
- Contiene enlace a `/recipes/`

4.2. ingredients.urls

`/ingredients/`

- Listado de ingredientes.
- Muestra los nombres de todos los ingredientes.
- Cada ingrediente debe contener un enlace a su detalle.
- Muestra enlace para añadir un ingrediente.
- En el caso de que no haya ningún ingrediente en la base de datos hay que mostrar el mensaje: *No ingredients so far!*

`/ingredients/21/`

- Detalle del ingrediente con `pk=21`
- Muestra el nombre del ingrediente.
- Muestra la descripción del ingrediente.
- Si el ingrediente dado no existe, hay que [devolver un 404](#).
- En el caso de que el ingrediente no tenga descripción hay que mostrar el mensaje: *No description so far!*

`/ingredients/add/`

- Añadir un ingrediente.
- Se debe mostrar un formulario con nombre y descripción del ingrediente.
- Utiliza un [formulario de modelo](#).
- Tras procesar el formulario se debe redirigir al listado de ingredientes.

4.3. `recipes.urls`

`/recipes/`

- Listado de recetas.
- Muestra los nombres de todas las recetas.
- Cada receta debe contener un enlace a su detalle.
- Muestra enlace para añadir una receta.
- En el caso de que no haya ninguna receta en la base de datos hay que mostrar el mensaje:
No recipes so far!

`/recipes/add/`

- Añadir una receta.
- Se debe mostrar un formulario con nombre y gui3n de la receta.
- Utiliza un [formulario de modelo](#).
- Tras procesar el formulario se debe redirigir al detalle de la receta creada.

`/recipes/34/`

- Detalle de la receta con `pk=34`
- Muestra el nombre de la receta.
- Muestra el gui3n de la receta.
- Si la receta dada no existe, hay que [devolver un 404](#).
- Para cada **factor** de la receta, hay que mostrar:
 - Nombre del ingrediente.
 - Cantidad del ingrediente.
 - Unidad del ingrediente (*Piece, Liter, Gram*).
- En el caso de que la receta no tenga factores hay que mostrar el mensaje: *No factors so far!*
- Muestra un enlace para a3adir un factor a esta receta.

[/recipes/34/add-factor/](#)

- Añadir un factor a la receta con `pk=34`
- Muestra el nombre de la receta.
- Si la receta dada no existe, hay que [devolver un 404](#).
- Se debe mostrar un formulario con ingrediente, cantidad y unidad.
- Utiliza un [formulario de modelo \(escenario con claves ajenas\)](#).
- Tras procesar el formulario se debe redirigir a la receta con `pk=34`

5. Administración

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

- `ingredients.Ingredient`
- `recipes.Recipe`
- `recipes.Factor`