UNIVERSIDAD CATÓLICA ANDRÉS BELLO  
FACULTAD DE INGENIERÍA.  
ESCUELA DE INGENIERÍA INFORMÁTICA  
Electiva: Programación en Python

**Proyecto #2:**

**PizzeríaUCAB**

**Integrante:**

**Germán Li Chen**

**Febrero, 2021**

Tabla de Contenido

[**Funcionalidades del sistema** 3](#_Toc63283776)

[**Descripción funcional del sistema para la toma de pedidos** 4](#_Toc63283777)

[**Descripción funcional de los reportes del administrador** 10](#_Toc63283778)

[**Descripción modelo de datos** 13](#_Toc63283779)

[**Descripción funcional adicional del sistema para la toma de pedidos** 17](#_Toc63283780)

# **Funcionalidades del sistema**

La funcionalidad básica consiste en ordenar una pizza, escoger el tamaño, escoger los ingredientes y mostrar el monto de todo lo solicitado. El usuario tendrá la opción una vez solicitado una pizza volver a pedir otra hasta que decida que no.

Para ello en el enunciado se solicitad estos 3 casos particulares:

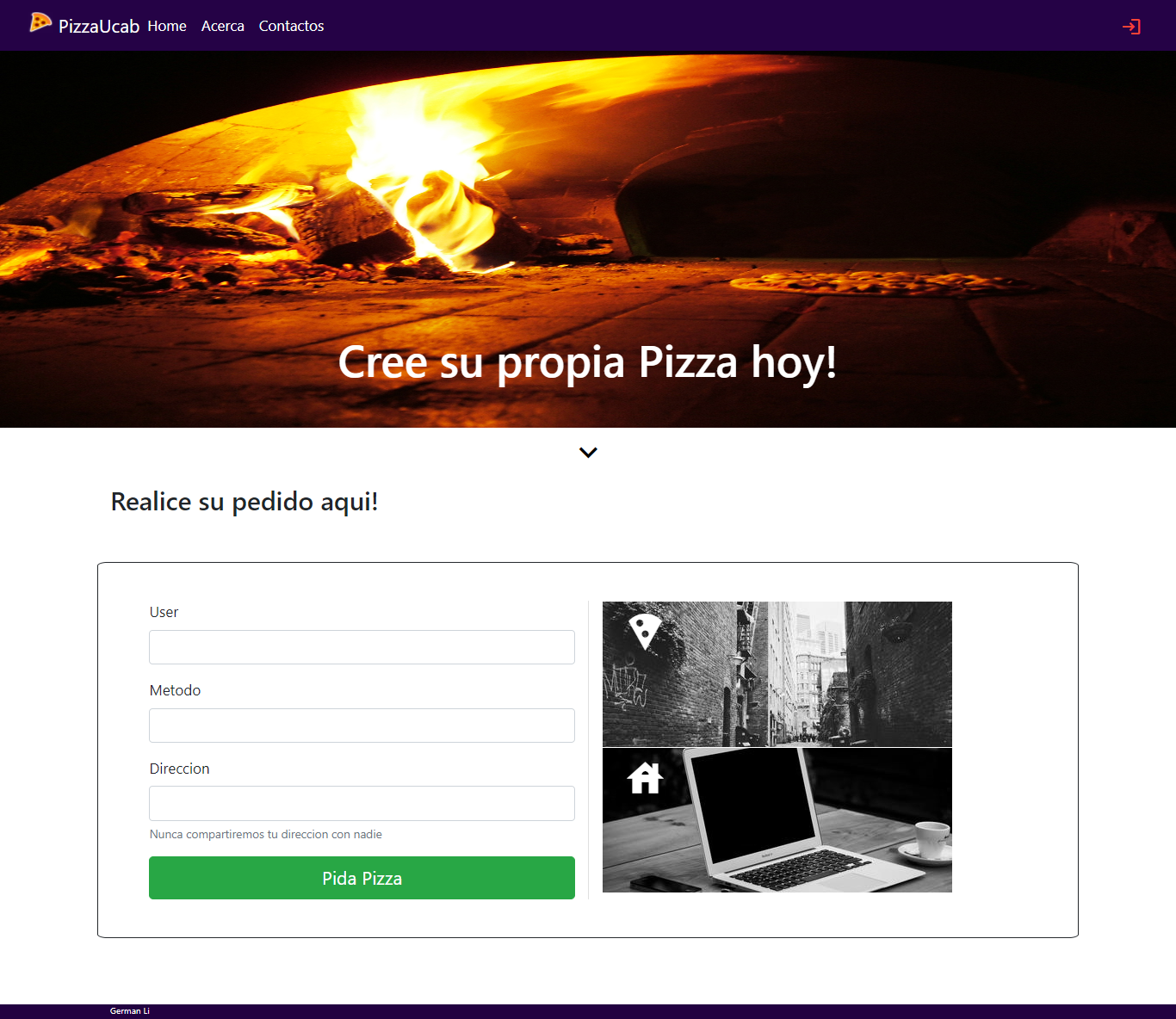
* **Caso 1:** El cliente sólo desea una pizza, de tamaño que desee, pero básica, es decir sin ingredientes adicionales
* **Caso 2:** El cliente sólo desea una pizza, de tamaño que desee, con uno o más ingredientes adicionales
* **Caso 3:** El usuario requiere de varias pizzas. En este caso al finalizar la primera orden, se le responde si al sistema y vuelve a empezar el ciclo

Se implemento Django Rest Framework para el trabajo con Angular.

Para empezar, se contarán con 2 sistemas funcionales, la parte del usuario y la parte administrativa.

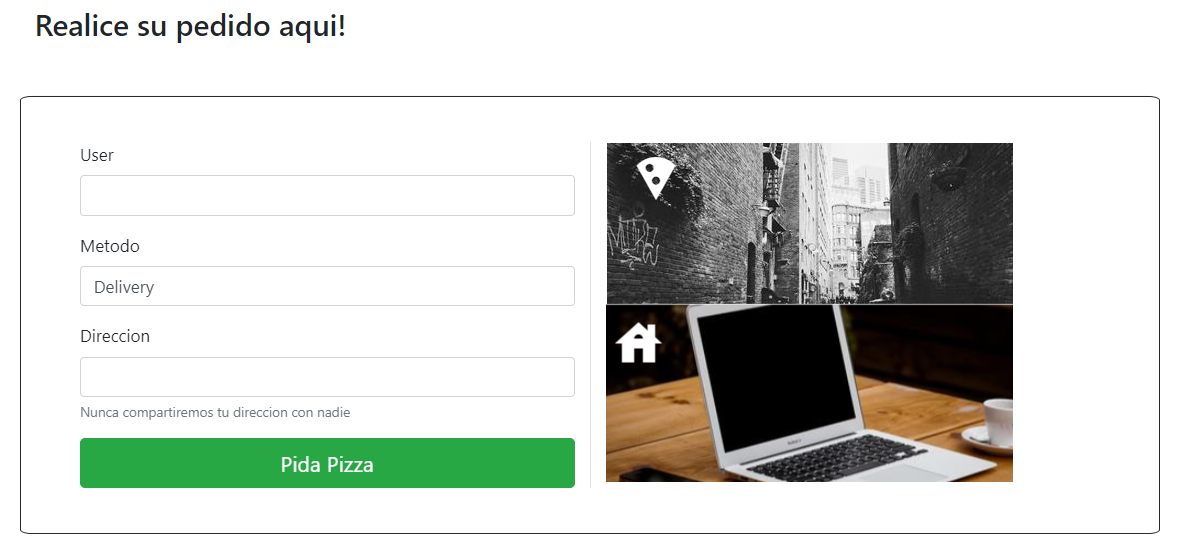
## **Descripción funcional del sistema para la toma de pedidos**

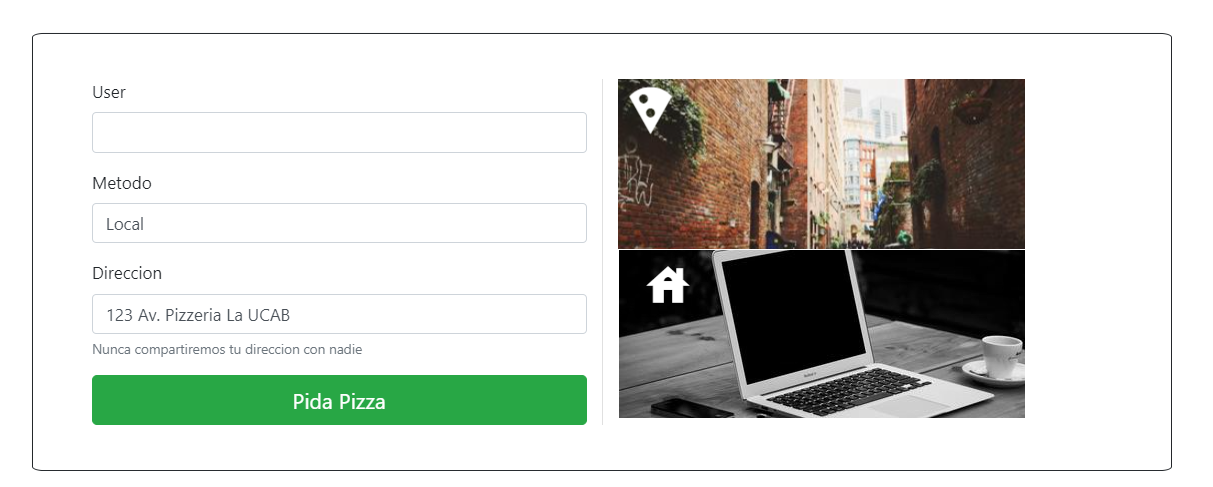
Se presenta de la siguiente manera, el Home, la página principal consta de la presentación y un formulario para la realización de pedidos:



En esta parte el usuario realizará el pedido, solo deberá escribir su nombre de usuario, y seleccionar algún método, de las que podemos encontrar entre

**Delivery,** en la que se le deberá indicar la dirección

**Local,** donde se le introducirá la dirección de la PIzzeriaUCAB

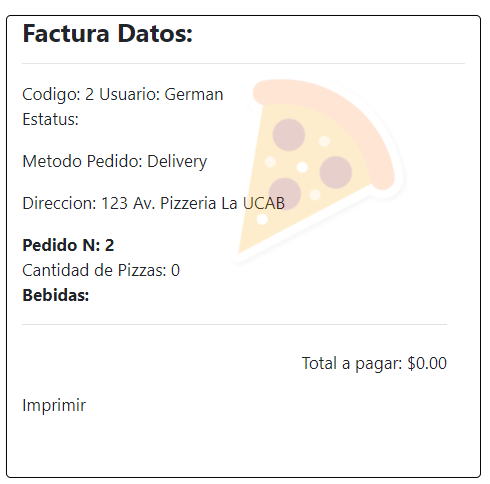


Una vez pedido la pizza, nos encontramos con la pagina de pedidos, en la cual consta de lo siguiente:

Graphical user interface, text, application, email

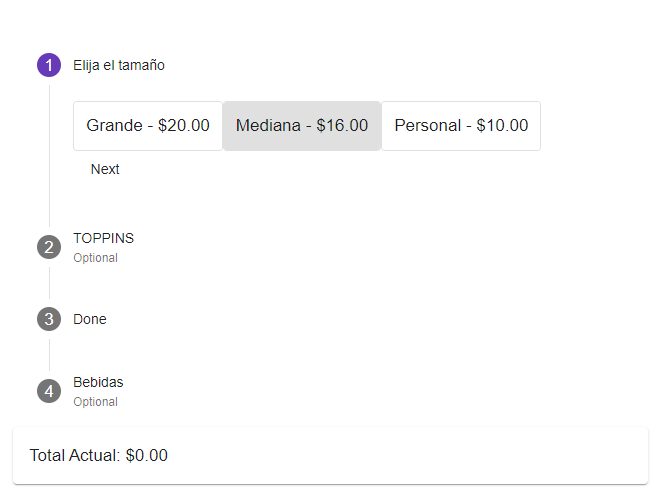
Description automatically generated

Un sistema por pasos, para seleccionar la pizza y los toppins, y la factura de datos, que ira recopilando la información del pedido, a medida que se vayan incluyendo más.

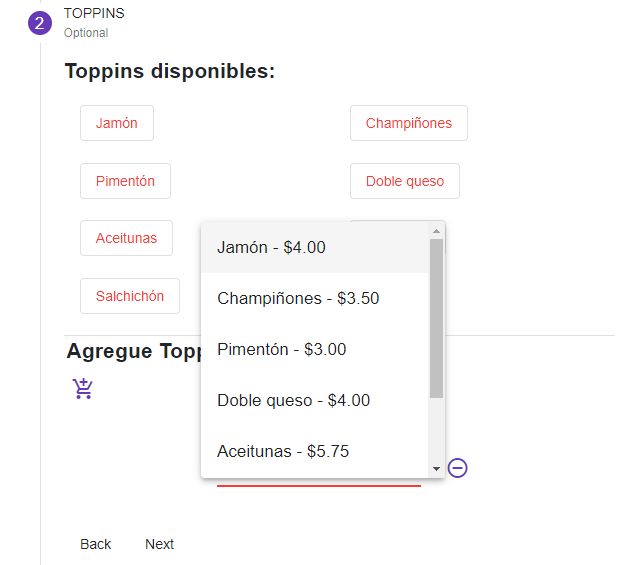


**Factura**

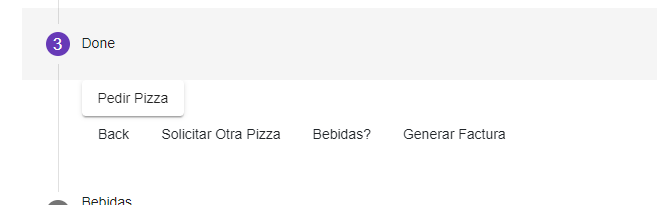
En los pedidos, se tiene que la primera opción se elige el tamaño de la pizza, con su respectivo monto.



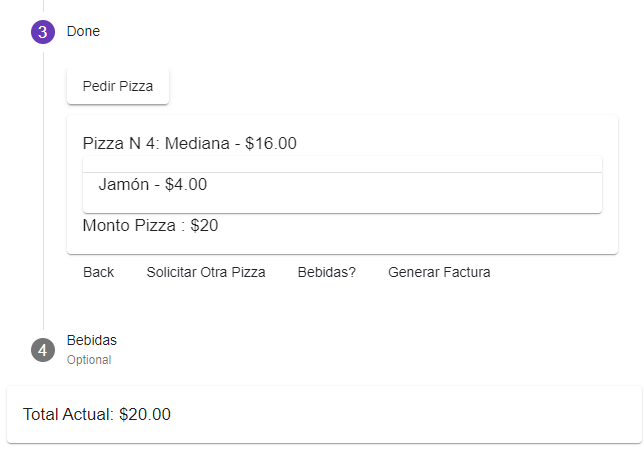
Seguido de eso, se pueden agregar los toppins en caso de que desee, siendo la casilla opcional, se va a poder pedir de 0 a varios toppins por pizza.



Finalizado todo, el usuario deberá seleccionar *Pedir Pizza* para solicitar la pizza con los toppins, en caso de haberlas agregado, y guardarlo en el pedido.



Esto nos genera una lista de la pizza y toppins elegidos, con su respectivo monto. Solo nos mostrara el monto de la pizza actual y sus datos. Por ejemplo, en la Pizza N`4 se tiene una Pizza Mediana con Jamón, con un monto de $20.



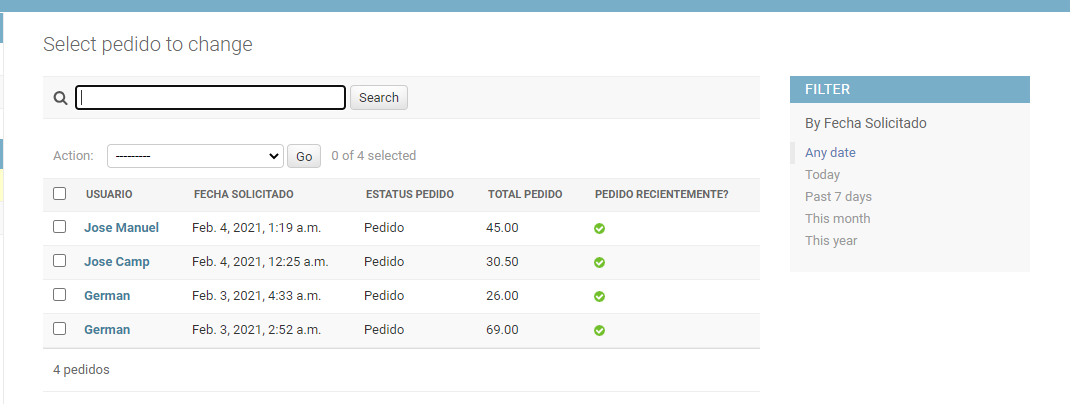
El ***Total Actual*** ira acumulando los montos de todo lo que se vaya a ir pidiendo.

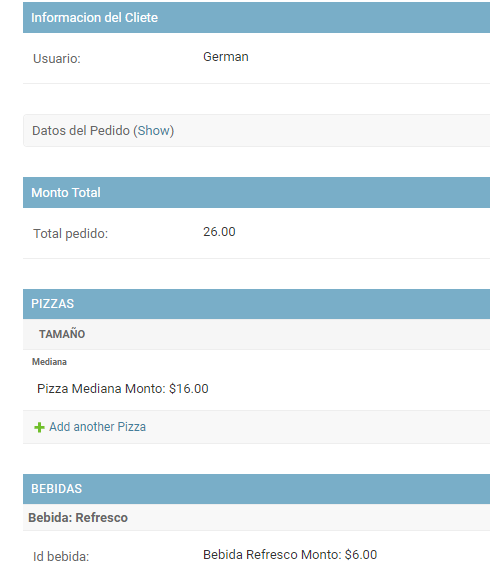
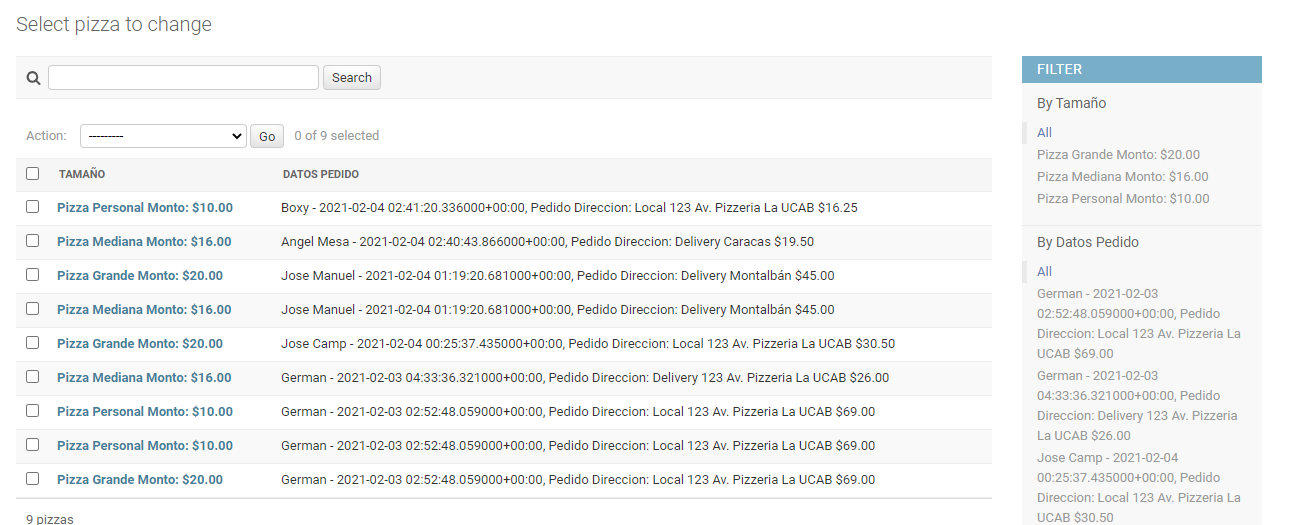
Para finalizar, seleccionamos *Generar Factura* lo que actualiza la pagina y nos genera la siguiente factura, indicando la cantidad de pizzas, los datos de las pizzas y toppins asociados a ese pedido, y en caso de pedir refrescos, sus datos. Y por último el monto total. (Si bien esto también se puede actualizar constantemente sin la necesidad de seleccionar *Generar Factura,* esto queda a criterio de la persona, colocar un actualizador al detectar cambios, o un intervalo para que vaya refrescando).



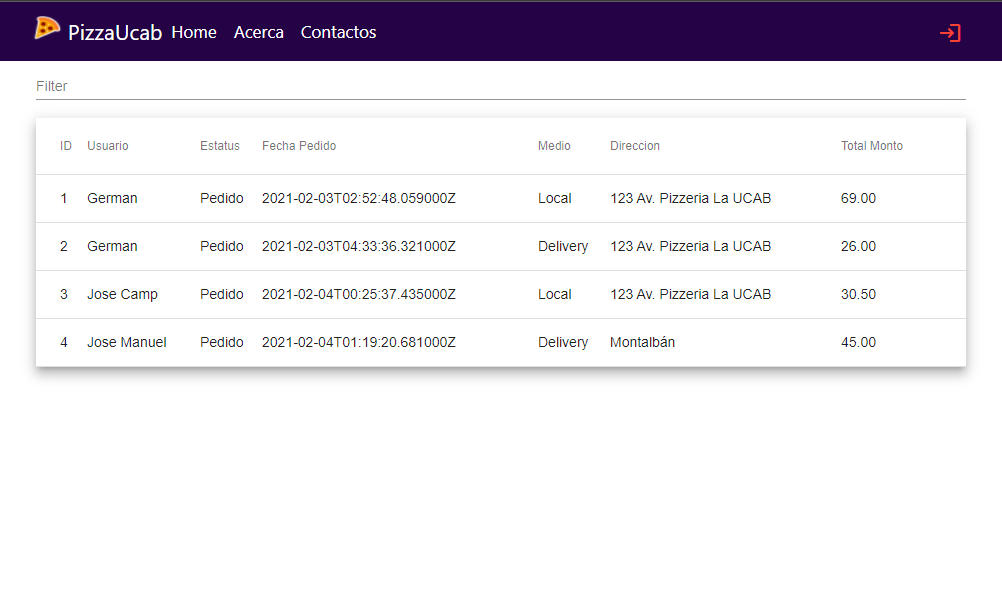
## **Descripción funcional de los reportes del administrador**

En los reportes consiste en el uso de la página por defecto /admin de Django para algunos reportes por día, o por usuario.

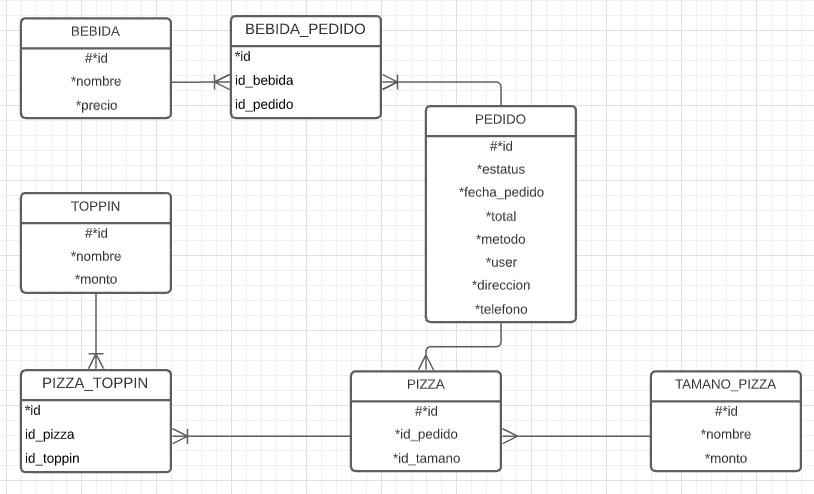




En la App se contará con una sección destinada al administrador de reportes.



## **Descripción modelo de datos**



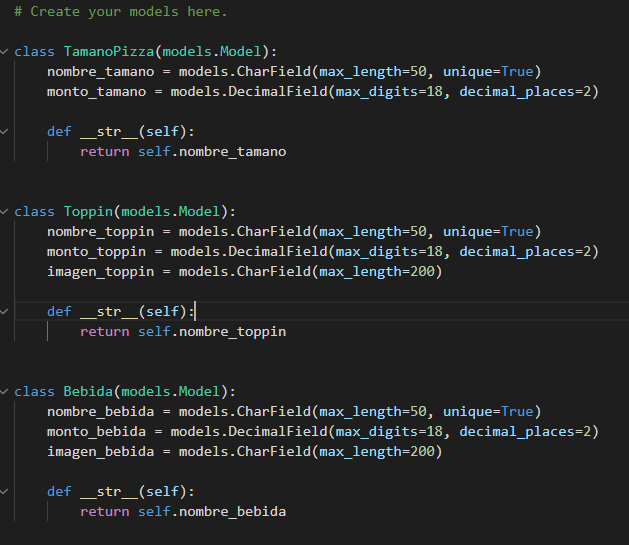
En este modelo este compuesto por las siguientes entidades. El pedido se obtienen los datos principales del usuario, y el ID del pedido. Un pedido posee una o varias pizzas, a la hora de crear el pedido, las pizzas que se creen se van a asociar al ID del pedido, para así asociar todas las pizzas y, bebidas también, de ese pedido.

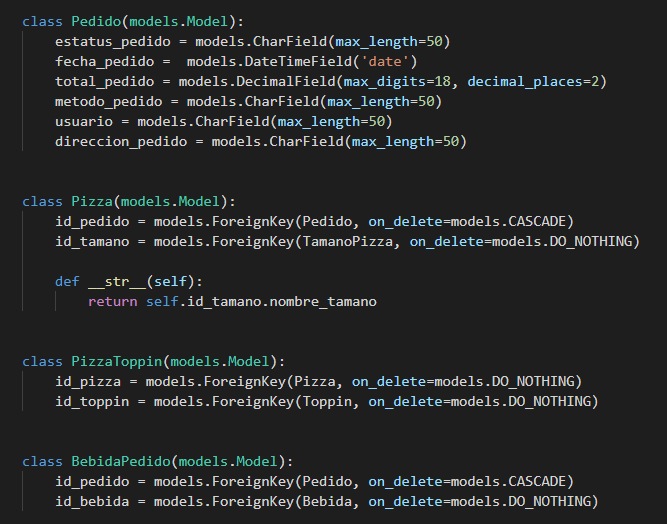
Lo demás, en la Pizza guardara los datos de la pizza, el tamaño, y los toppins, una pizza solo tiene un tamaño, y puede tener uno o más ingredientes/toppins, al igual que los toppins puede pertenecer a una o más pizzas, esta pasa igual con las bebidas.

Con este modelo de datos básico, se puede cumplir la lógica planteada para el proyecto.

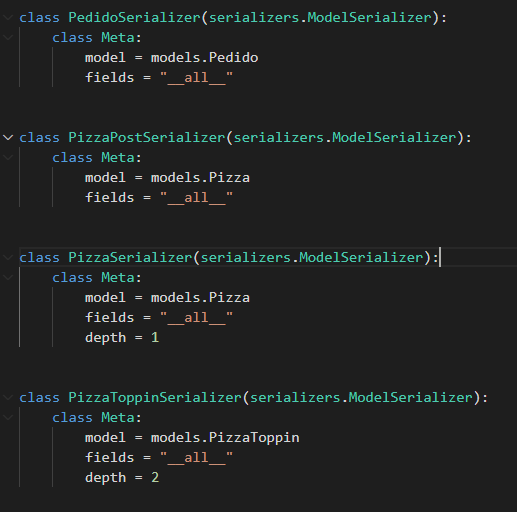
Models.py para una información extra:

Los modelos implementados:

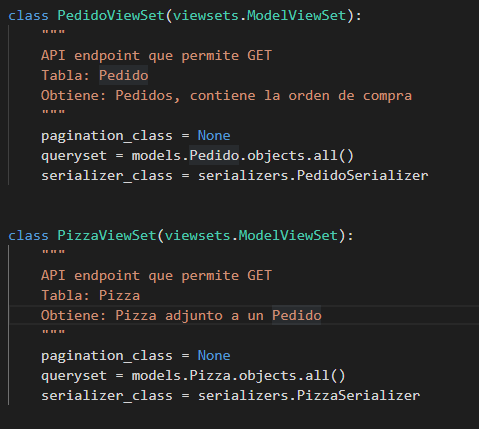




Con el uso de serializers de definen los campos del modelo a serializar



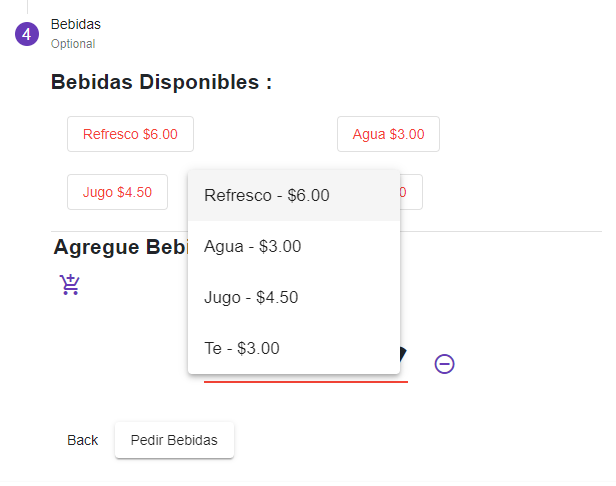
Para la vista recibe los datos del modelo y el serializer, para poder desplegar los endpoint que contiene la información de cada uno.



## **Descripción funcional adicional del sistema para la toma de pedidos**

En la parte de los *toppins*, al pasar el *mouse*(*hover*), se mostrará la imagen del producto, esto para dar una visual al usuario, al seleccionar. Esto fue implementado con CSS, e introduciendo en el *src* el URL de la imagen.





Esto también estará para la parte de bebidas. Lo segundo adicional, es que podrá pedir de 0 a varias bebidas, para acompañar en el pedido. Sigue un proceso parecido a los toppins, la cual se implementó un *form array.*

Para la 3ra funcionalidad, se aplicó *Angular* y *Bootstrap* para el *frontend* de las paginas diseñadas.

