

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
Descripción funcional del sistema para la toma de pedidos	4
Descripción funcional de los reportes del administrador.....	10
Descripción modelo de datos.....	13
Descripción funcional adicional del sistema para la toma de pedidos.....	17

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 solicitan 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 implementó 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:



Realice su pedido aqui!

User

Metodo

Direccion

Nunca compartiremos tu direccion con nadie.

Pida Pizza

Two stacked rectangular images. The top image is a black and white photograph of a narrow city street with buildings and a pizza icon overlaid in the top left corner. The bottom image is a black and white photograph of a laptop on a wooden table with a cup of coffee, with a home icon overlaid in the top left corner.

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

Realice su pedido aqui!


User

Metodo

Direccion

Nunca compartiremos tu direccion con nadie

Pida Pizza



Local, donde se le introducirá la dirección de la PizzeriaUCAB


User

Metodo

Direccion

Nunca compartiremos tu direccion con nadie

Pida Pizza



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

PizzaUcab

Home

Acerca

Contactos

1

Elija el tamaño

Grande - \$20.00

Mediana - \$16.00

Personal - \$10.00

Next

2

TOPPINS

Optional

3

Done

4

Bebidas

Optional

Total Actual: \$0.00

Factura Datos:

Codigo: 2 Usuario: German

Estatus:

Metodo Pedido: Delivery

Direccion: 123 Av. Pizzeria La UCAB

Pedido N: 2

Cantidad de Pizzas: 0

Bebidas:

Total a pagar: \$0.00

Imprimir



German Li

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

Factura Datos:

Codigo: 2 Usuario: German

Estatus:

Metodo Pedido: Delivery

Direccion: 123 Av. Pizzeria La UCAB

Pedido N: 2

Cantidad de Pizzas: 0

Bebidas:

Total a pagar: \$0.00

Imprimir



Factura

6

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

1 Elija el tamaño

Grande - \$20.00 Mediana - \$16.00 Personal - \$10.00

Next

2 TOPPINS
Optional

3 Done

4 Bebidas
Optional

Total Actual: \$0.00

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

2 TOPPINS
Optional

Toppins disponibles:

Jamón Champiñones

Pimentón Doble queso

Aceitunas

Salchichón

Agregue Toppings

🛒

Back Next

Jamón - \$4.00

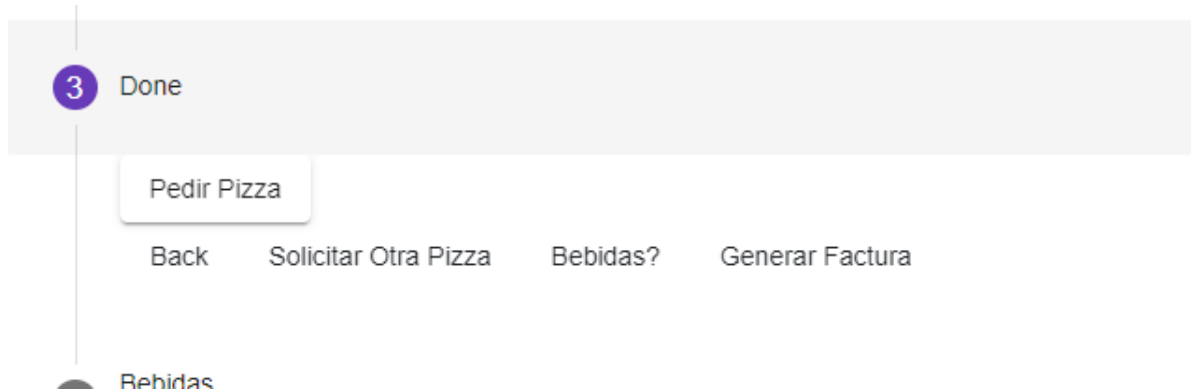
Champiñones - \$3.50

Pimentón - \$3.00

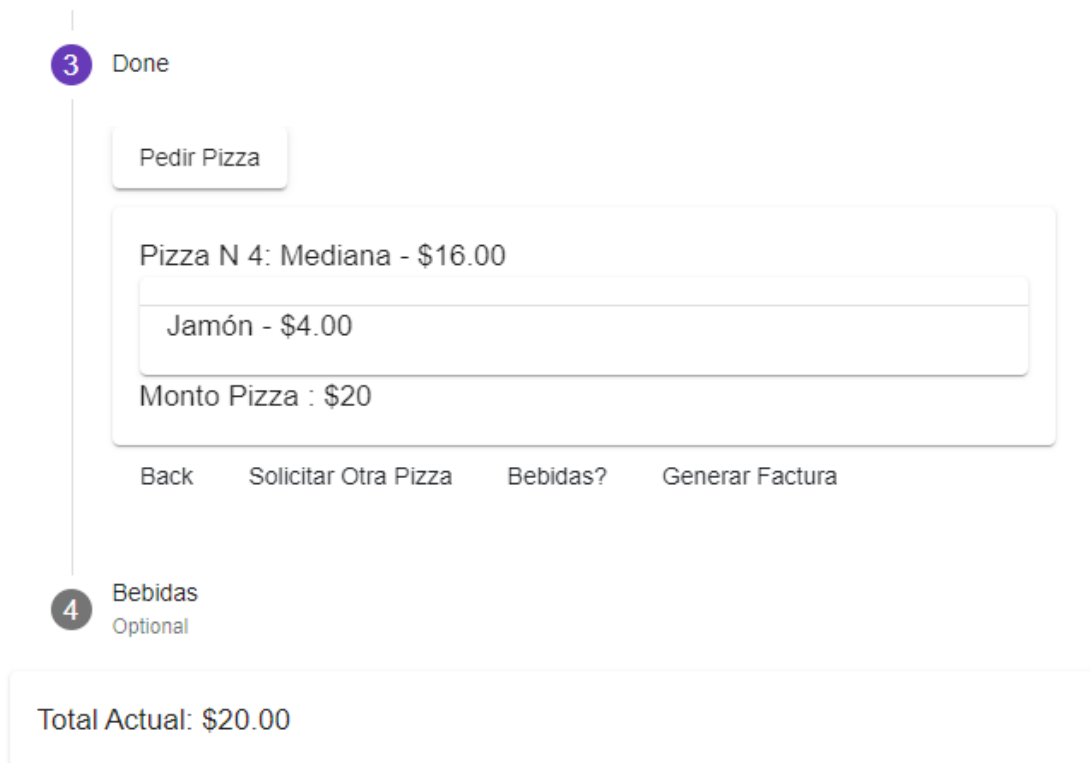
Doble queso - \$4.00

Aceitunas - \$5.75

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



Esto nos genera una lista de la pizza y toppings 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 toppings 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).

Factura Datos:

Codigo: 2 Usuario: German
Estatus:

Metodo Pedido: Delivery

Direccion: 123 Av. Pizzeria La UCAB



Pedido N: 2
Cantidad de Pizzas: 1

Pizza: Mediana	16.00
Toppins =	
Jamón	4.00
Bebidas:	
Refresco	6.00

Total a pagar: \$26.00

Imprimir

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.

Select pedido to change

Action: 0 of 4 selected

<input type="checkbox"/>	USUARIO	FECHA SOLICITADO	ESTATUS PEDIDO	TOTAL PEDIDO	PEDIDO RECIENTEMENTE?
<input type="checkbox"/>	Jose Manuel	Feb. 4, 2021, 1:19 a.m.	Pedido	45.00	✓
<input type="checkbox"/>	Jose Camp	Feb. 4, 2021, 12:25 a.m.	Pedido	30.50	✓
<input type="checkbox"/>	German	Feb. 3, 2021, 4:33 a.m.	Pedido	26.00	✓
<input type="checkbox"/>	German	Feb. 3, 2021, 2:52 a.m.	Pedido	69.00	✓

4 pedidos

FILTER

By Fecha Solicitado

Any date

Today

Past 7 days

This month

This year

Select pizza to change

Q

Search

Action: Go 0 of 9 selected

<input type="checkbox"/> TAMAÑO	DATOS PEDIDO
<input type="checkbox"/> Pizza Personal Monto: \$10.00	Boxy - 2021-02-04 02:41:20.336000+00:00, Pedido Direccion: Local 123 Av. Pizzeria La UCAB \$16.25
<input type="checkbox"/> Pizza Mediana Monto: \$16.00	Angel Mesa - 2021-02-04 02:40:43.866000+00:00, Pedido Direccion: Delivery Caracas \$19.50
<input type="checkbox"/> Pizza Grande Monto: \$20.00	Jose Manuel - 2021-02-04 01:19:20.681000+00:00, Pedido Direccion: Delivery Montalbán \$45.00
<input type="checkbox"/> Pizza Mediana Monto: \$16.00	Jose Manuel - 2021-02-04 01:19:20.681000+00:00, Pedido Direccion: Delivery Montalbán \$45.00
<input type="checkbox"/> Pizza Grande Monto: \$20.00	Jose Camp - 2021-02-04 00:25:37.435000+00:00, Pedido Direccion: Local 123 Av. Pizzeria La UCAB \$30.50
<input type="checkbox"/> Pizza Mediana Monto: \$16.00	German - 2021-02-03 04:33:36.321000+00:00, Pedido Direccion: Delivery 123 Av. Pizzeria La UCAB \$26.00
<input type="checkbox"/> Pizza Personal Monto: \$10.00	German - 2021-02-03 02:52:48.059000+00:00, Pedido Direccion: Local 123 Av. Pizzeria La UCAB \$69.00
<input type="checkbox"/> Pizza Personal Monto: \$10.00	German - 2021-02-03 02:52:48.059000+00:00, Pedido Direccion: Local 123 Av. Pizzeria La UCAB \$69.00
<input type="checkbox"/> Pizza Grande Monto: \$20.00	German - 2021-02-03 02:52:48.059000+00:00, Pedido Direccion: Local 123 Av. Pizzeria La UCAB \$69.00

9 pizzas

FILTER

By Tamaño

All

Pizza Grande Monto: \$20.00

Pizza Mediana Monto: \$16.00

Pizza Personal Monto: \$10.00

By Datos Pedido

All

German - 2021-02-03 02:52:48.059000+00:00, Pedido Direccion: Local 123 Av. Pizzeria La UCAB \$69.00

German - 2021-02-03 04:33:36.321000+00:00, Pedido Direccion: Delivery 123 Av. Pizzeria La UCAB \$26.00

Jose Camp - 2021-02-04 00:25:37.435000+00:00, Pedido Direccion: Local 123 Av. Pizzeria La UCAB \$30.50

Datos Pedido: Jose Manuel - 2021-02-04 01:19:20.681000+00:00, Pedido Direccion: Delivery Montalbán \$45.00

Tamaño Pizza

Tamaño: Pizza Grande Monto: \$20.00

TOPPINS

Toppin: Pimentón

Toppin: Bebida Pimentón Monto: \$3.00

+ Add another Toppin

Informacion del Ciente	
Usuario:	German

Datos del Pedido (Show)

Monto Total	
Total pedido:	26.00

PIZZAS	
TAMAÑO	
Mediana	
Pizza Mediana Monto: \$16.00	
+ Add another Pizza	

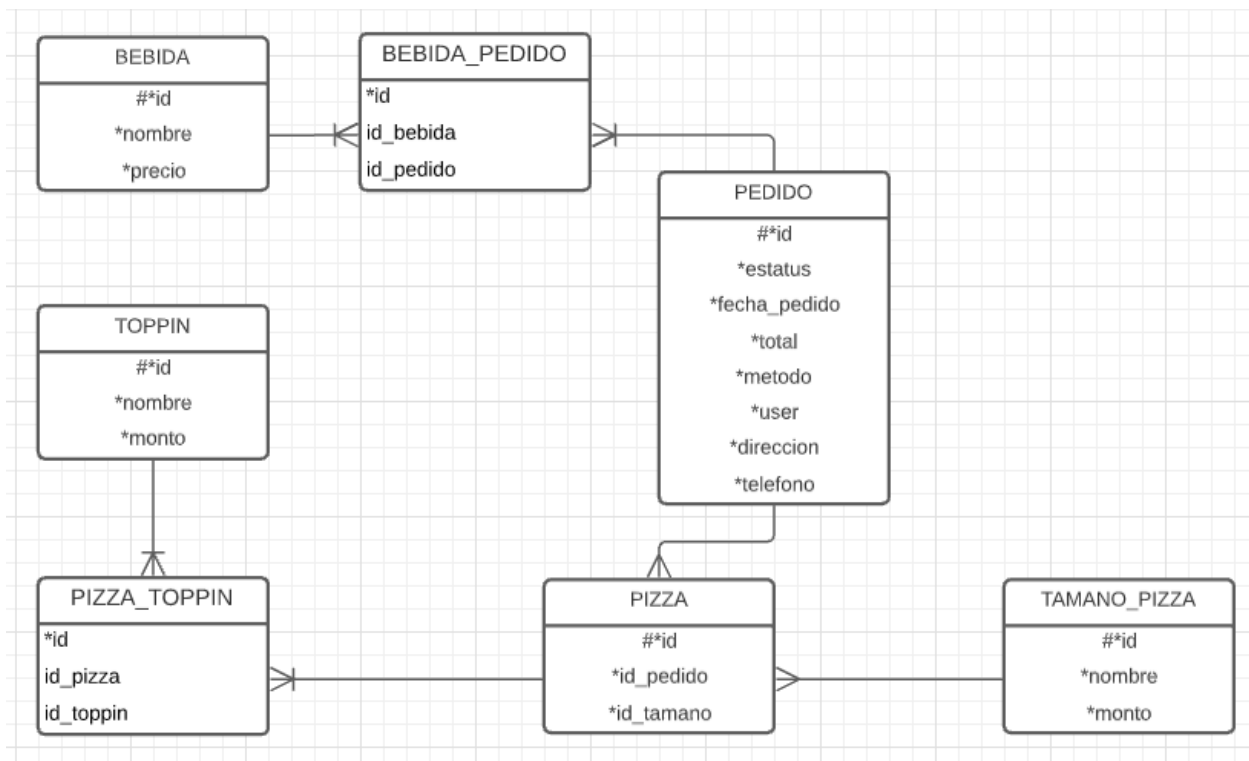
BEBIDAS	
Bebida: Refresco	
Id bebida:	Bebida Refresco Monto: \$6.00

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

Filter

ID	Usuario	Estatus	Fecha Pedido	Medio	Direccion	Total Monto
1	German	Pedido	2021-02-03T02:52:48.059000Z	Local	123 Av. Pizzeria La UCAB	69.00
2	German	Pedido	2021-02-03T04:33:36.321000Z	Delivery	123 Av. Pizzeria La UCAB	26.00
3	Jose Camp	Pedido	2021-02-04T00:25:37.435000Z	Local	123 Av. Pizzeria La UCAB	30.50
4	Jose Manuel	Pedido	2021-02-04T01:19:20.681000Z	Delivery	Montalbán	45.00

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 toppings, una pizza solo tiene un tamaño, y puede tener uno o más ingredientes/toppings, al igual que los toppings 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:

```
# Create your models here.

class TamanoPizza(models.Model):
    nombre_tamano = models.CharField(max_length=50, unique=True)
    monto_tamano = models.DecimalField(max_digits=18, decimal_places=2)

    def __str__(self):
        return self.nombre_tamano

class Toppin(models.Model):
    nombre_toppin = models.CharField(max_length=50, unique=True)
    monto_toppin = models.DecimalField(max_digits=18, decimal_places=2)
    imagen_toppin = models.CharField(max_length=200)

    def __str__(self):
        return self.nombre_toppin

class Bebida(models.Model):
    nombre_bebida = models.CharField(max_length=50, unique=True)
    monto_bebida = models.DecimalField(max_digits=18, decimal_places=2)
    imagen_bebida = models.CharField(max_length=200)

    def __str__(self):
        return self.nombre_bebida
```

```

class Pedido(models.Model):
    estatus_pedido = models.CharField(max_length=50)
    fecha_pedido = models.DateTimeField('date')
    total_pedido = models.DecimalField(max_digits=18, decimal_places=2)
    metodo_pedido = models.CharField(max_length=50)
    usuario = models.CharField(max_length=50)
    direccion_pedido = models.CharField(max_length=50)

class Pizza(models.Model):
    id_pedido = models.ForeignKey(Pedido, on_delete=models.CASCADE)
    id_tamano = models.ForeignKey(TamanoPizza, on_delete=models.DO_NOTHING)

    def __str__(self):
        return self.id_tamano.nombre_tamano

class PizzaToppin(models.Model):
    id_pizza = models.ForeignKey(Pizza, on_delete=models.DO_NOTHING)
    id_toppin = models.ForeignKey(Toppin, on_delete=models.DO_NOTHING)

class BebidaPedido(models.Model):
    id_pedido = models.ForeignKey(Pedido, on_delete=models.CASCADE)
    id_bebida = models.ForeignKey(Bebida, on_delete=models.DO_NOTHING)

```

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

```

class PedidoSerializer(serializers.ModelSerializer):
    class Meta:
        model = models.Pedido
        fields = "__all__"

class PizzaPostSerializer(serializers.ModelSerializer):
    class Meta:
        model = models.Pizza
        fields = "__all__"

class PizzaSerializer(serializers.ModelSerializer):
    class Meta:
        model = models.Pizza
        fields = "__all__"
        depth = 1

class PizzaToppinSerializer(serializers.ModelSerializer):
    class Meta:
        model = models.PizzaToppin
        fields = "__all__"
        depth = 2

```

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


```

class PedidoViewSet(viewsets.ModelViewSet):
    """
    API endpoint que permite GET
    Tabla: Pedido
    Obtiene: Pedidos, contiene la orden de compra
    """
    pagination_class = None
    queryset = models.Pedido.objects.all()
    serializer_class = serializers.PedidoSerializer

class PizzaViewSet(viewsets.ModelViewSet):
    """
    API endpoint que permite GET
    Tabla: Pizza
    Obtiene: Pizza adjunto a un Pedido
    """
    pagination_class = None
    queryset = models.Pizza.objects.all()
    serializer_class = serializers.PizzaSerializer

```

Descripción funcional adicional del sistema para la toma de pedidos

En la parte de los *toppings*, 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.

Toppins disponibles:

Jamón \$4.00

Champiñones \$3.50



Doble queso \$4.00

Pepperoni \$3.85

Salchichón \$6.25

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*.

4

Bebidas
Optional

Bebidas Disponibles :

Refresco \$6.00

Agua \$3.00

Jugo \$4.50

Agregue Bebidas



Refresco - \$6.00

Agua - \$3.00

Jugo - \$4.50

Te - \$3.00

Back

Pedir Bebidas

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

