

# Solución Ejercicio guiado

## "Pizza app"

1. Inicialmente, generaremos una carpeta que contenga nuestra app, esta carpeta se llamará `pizza_app`. Dado que esta app parece tener muchas tareas diversas aplicaremos un principio de modularización, para ello crearemos los siguientes módulos:

- `main.py`
- `add.py`
- `masa.py`
- `remove.py`
- `salsa.py`
- `show.py`
- `tiempo.py`

Cada uno de estos módulos albergarán las distintas funcionalidades.

2. Comenzaremos bosquejando lo que será nuestro esqueleto. Para ello en nuestro archivo `main.py` realizaremos lo siguiente:

Definimos nuestro formato para almacenar las pizzas. En este caso particular utilizaremos un diccionario con la siguiente estructura:

```
ingredientes_orden = {'masa': 'Masa Tradicional',  
                      'salsa': 'Salsa de Tomate',  
                      'ingredientes': ['Queso']  
                      }
```

Este diccionario es capaz de almacenar de manera diferenciada la Masa, la Salsa y una lista de los ingredientes. Estos serán nuestros ingredientes base.

Además definimos el Menú:

```
opcion = input('¿Qué desea realizar?')
    1. Cambiar tipo de Masa
    2. Cambiar tipo de Salsa
    3. Agregar Ingredientes
    4. Eliminar Ingredientes
    5. Ordenar con los Ingredientes Actuales

    0. Consultar ingredientes de la pizza

    Otro Número cancelará el pedido.
    > ''')
```

En este caso asociamos un número a cada capacidad que debe tener nuestra app. Además, generamos que en caso de utilizar otro número el pedido se cancelará.

3. Ahora comenzaremos a trabajar en cómo cambiar la masa en `masa.py`:

```
def tipo_masa(ingredientes, eleccion):
    if eleccion == 'T':
        ingredientes['masa'] = 'Masa Tradicional'
    elif eleccion == 'D':
        ingredientes['masa'] = 'Masa Delgada'
    elif eleccion == 'B':
        ingredientes['masa'] = 'Masa con Bordes de Queso'

    if eleccion in ['T','D','B']:
        print(f'Su masa se cambió a {ingredientes["masa"]}')
    else:
        print('No se ha cambiado su tipo de Masa')

    return ingredientes
```

4. Definimos la función tipo de Masa. Esta función permitirá escoger el tipo de masa modificando la clave masa del diccionario de ingredientes, para ello solicitará los ingredientes actuales y una elección.

Además informará que se cambió la masa. En caso de que se indique una opción inválida informará que no se cambió el tipo de masa.

Para probar el código se ejecutará la función utilizando un set de ingredientes de prueba para determinar si entrega el resultado esperado. El código completo luce así:

```
1 def tipo_masa(ingredientes, eleccion):
2     if eleccion == 'T':
3         ingredientes['masa'] = 'Masa Tradicional'
4     elif eleccion == 'D':
5         ingredientes['masa'] = 'Masa Delgada'
6     elif eleccion == 'B':
7         ingredientes['masa'] = 'Masa con Bordes de Queso'
8
9     if eleccion in ['T','D','B']:
10        print(f'Su masa se cambió a {ingredientes["masa"]}')
11    else:
12        print('No se ha cambiado su tipo de Masa')
13
14    return ingredientes
15
16
17 if __name__ == '__main__':
18     ingredientes_prueba = {'masa': 'Masa Tradicional',
19                             'salsa': 'Salsa de Tomate',
20                             'ingredientes': ['queso']}
21
22     eleccion = input("""Escoja el tipo de Masa:
23     T). Masa Tradicional
24     D). Masa Delgada
25     B). Masa con Bordes de Queso
26     > """).upper()
27     ingredientes = tipo_masa(ingredientes_prueba, eleccion)
28     print(ingredientes)
```

Imagen 1. Código completo módulo masa.py

Fuente: Desafío Latam

5. Ahora implementaremos la elección del tipo de salsa en el archivo `salsa.py`:

```
def tipo_salsa(ingredientes, eleccion):
    if eleccion == 'T':
        ingredientes['salsa'] = 'Salsa de Tomate'
    elif eleccion == 'A':
        ingredientes['salsa'] = 'Salsa Alfredo'
    elif eleccion == 'B':
        ingredientes['salsa'] = 'Salsa Barbecue'
    elif eleccion == 'P':
        ingredientes['salsa'] = 'Salsa Pesto'

    if eleccion in ['T','A','B','P']:
        print(f'Su salsa se cambió a {ingredientes["salsa"]}')
    else:
        print('No se ha cambiado su tipo de Salsa')

    return ingredientes
```

Definimos la función tipo de Salsa. Esta funcionalidad será muy parecida a la de elección de masa, solo que permitirá escoger el tipo de salsa modificando la clave salsa del diccionario de ingredientes. Para ello solicitará los ingredientes actuales y una elección.

Además informará que se cambió la salsa, en caso de que se indique una opción inválida informará que no se cambió el tipo de salsa.

Para probar el código se ejecutará la función utilizando un set de ingredientes de prueba para determinar si entrega el resultado esperado. El código completo luce así:

```
1
2 def tipo_salsa(ingredientes, eleccion):
3     if eleccion == 'T':
4         ingredientes['salsa'] = 'Salsa de Tomate'
5     elif eleccion == 'A':
6         ingredientes['salsa'] = 'Salsa Alfredo'
7     elif eleccion == 'B':
8         ingredientes['salsa'] = 'Salsa Barbecue'
9     elif eleccion == 'P':
10        ingredientes['salsa'] = 'Salsa Pesto'
11
12
13    if eleccion in ['T','A','B','P']:
14        print(f'Su salsa se cambió a {ingredientes["salsa"]}')
15    else:
16        print('No se ha cambiado su tipo de Salsa')
17
18    return ingredientes
19
20 if __name__ == '__main__':
21     ingredientes_prueba = {'masa': 'Masa Tradicional',
22                             'salsa': 'Salsa de Tomate',
23                             'ingredientes': ['Queso']}
24
25     eleccion = input('Seleccione su tipo de Salsa:
26     T). Salsa de Tomate
27     A). Salsa Alfredo
28     B). Salsa Barbecue
29     P). Salsa Pesto
30     > ')
31     ingredientes = tipo_salsa(ingredientes_prueba, eleccion)
32     print(ingredientes)
```

Imagen 2. Código completo módulo salsa.py

Fuente: Desafío Latam

6. Ahora se implementará la opción para agregar ingredientes en el archivo `add.py`:

```
def agregar_ingredientes(ingredientes, eleccion):  
  
    disponibles = ['Tomate', 'Champiñones', 'Aceituna', 'Cebolla', 'Pollo',  
                  'Jamón', 'Carne', 'Tocino', 'Queso']  
    nuevo_ingredientes = disponibles[eleccion-1]  
  
    if nuevo_ingredientes in ingredientes['ingredientes']:  
        print('El ingrediente ya existe')  
    else:  
        ingredientes['ingredientes'].append(nuevo_ingredientes)  
        print(f'Se ha agregado {nuevo_ingredientes}')  
  
    return ingredientes
```

En este caso, se define un diccionario de ingredientes disponibles. La función `agregar_ingredientes()` solicitará los ingredientes actuales y una elección numérica. Esta elección numérica se utilizará como índice para identificar qué ingrediente fue solicitado. Si el ingrediente ya ha sido agregado se devolverá un mensaje informando que el ingrediente ya existe. En caso contrario, el diccionario se agregará a la lista de ingredientes y se informará de esto al usuario. Finalmente la función devuelve los ingredientes actualizados.

Para probar el código se ejecutará la función utilizando un set de ingredientes de prueba para determinar si entrega el resultado esperado. El código completo luce así:

```
1
2 def agregar_ingrediente(ingredientes, eleccion):
3
4     disponibles = ['Tomate', 'Champiñones', 'Aceituna', 'Cebolla',
5                     'Pollo', 'Jamón', 'Carne', 'Tocino', 'Queso']
6     nuevo_ingrediente = disponibles[eleccion-1]
7
8     if nuevo_ingrediente in ingredientes['ingredientes']:
9         print('El ingrediente ya existe')
10    else:
11        ingredientes['ingredientes'].append(nuevo_ingrediente)
12        print(f'Se ha agregado {nuevo_ingrediente}')
13
14    return ingredientes
15
16 if __name__ == '__main__':
17     ingredientes_prueba = {'masa': 'Masa Tradicional',
18                             'salsa': 'Salsa de Tomate',
19                             'ingredientes': ['Queso']}
20
21     eleccion = int(input('Seleccione su Ingrediente:
22     1). Tomate
23     2). Champiñones
24     3). Aceituna
25     4). Cebolla
26     5). Pollo
27     6). Jamón
28     7). Carne
29     8). Tocino
30     9). Queso
31     > '''))
32
33     ingredientes = agregar_ingrediente(ingredientes_prueba, eleccion)
34     print(ingredientes)
```

Imagen 3. Código completo módulo `add.py`

Fuente: Desafío Latam

7. De manera análoga es posible generar la eliminación de un ingrediente en `remove.py`

```
def quitar_ingrediente(ingredientes, eleccion):
    disponibles = ['Tomate', 'Champiñones', 'Aceituna', 'Cebolla',
                    'Pollo', 'Jamón', 'Carne', 'Tocino', 'Queso']
    quitar = disponibles[eleccion-1]

    if quitar in ingredientes['ingredientes']:
        ingredientes['ingredientes'].remove(quitar)
        print(f'Se ha quitado {quitar}')
    elif len(ingredientes['ingredientes']) == 0:
        print('No hay ningún ingrediente que quitar')
    else:
        print('No se puede quitar ese ingrediente, porque no está incluido')

    return ingredientes
```

En este caso también se define un diccionario de ingredientes disponibles. La función `agregar_ingrediente()` solicitará los ingredientes actuales y una elección numérica. Esta elección numérica se utilizará de la misma manera que se utilizó al agregar un ingrediente.

Las opciones posibles acá son, si el ingrediente está presente se remueve y se informa. En el caso que no hayan ingredientes se avisará que no se puede quitar nada más porque no hay más ingredientes. Finalmente, en caso de que se quiera remover un ingrediente que no está también será informado, la función retorna los ingredientes actualizados.

Para probar el código se ejecutará la función utilizando un set de ingredientes de prueba para determinar si entrega el resultado esperado. El código completo luce así:

```
1
2 def quitar_ingrediente(ingredientes, eleccion):
3     disponibles = ['Tomate', 'Champiñones', 'Aceituna', 'Cebolla',
4                   'Pollo', 'Jamón', 'Carne', 'Tocino', 'Queso']
5     quitar = disponibles[eleccion-1]
6
7     if quitar in ingredientes['ingredientes']:
8         ingredientes['ingredientes'].remove(quitar)
9         print(f'Se ha quitado {quitar}')
10    elif len(ingredientes['ingredientes']) == 0:
11        print('No hay ningún ingrediente que quitar')
12    else:
13        print('No se puede quitar ese ingrediente, porque no está incluido')
14
15    return ingredientes
16
17 if __name__ == '__main__':
18     ingredientes_prueba = {'masa': 'Masa Tradicional',
19                             'salsa': 'Salsa de Tomate',
20                             'ingredientes': ['Queso']}
21
22
23     eleccion = int(input("""Seleccione qué ingrediente quitar:
24     1). Tomate
25     2). Champiñones
26     3). Aceituna
27     4). Cebolla
28     5). Pollo
29     6). Jamón
30     7). Carne
31     8). Tocino
32     9). Queso
33     > """))
34
35     ingredientes = quitar_ingrediente(ingredientes_prueba, eleccion)
36     print(ingredientes)
```

Imagen 4. Código completo módulo `remove.py`  
Fuente: Desafío Latam

8. Una vez que se ordena, se estimará el tiempo que toma. Esto irá en el archivo `tiempo.py`

```
def estimar_tiempo(ingredientes):  
    n_ingredientes = len(ingredientes['ingredientes'])  
    tiempo = 20 + n_ingredientes * 2  
    return tiempo
```

Esta función simplemente tomará los ingredientes actuales y los contará. A partir de eso reportará el tiempo según los requerimientos.

Para probar el código se ejecutará la función utilizando un set de ingredientes de prueba para determinar si entrega el resultado esperado. El código completo luce así:

```
1  
2 def estimar_tiempo(ingredientes):  
3     n_ingredientes = len(ingredientes['ingredientes'])  
4     tiempo = 20 + n_ingredientes * 2  
5     return tiempo  
6  
7 if __name__ == '__main__':  
8     ingredientes_prueba = {'masa': 'Masa Tradicional',  
9                             'salsa': 'Salsa de Tomate',  
10                            'ingredientes': ['Queso']  
11                             }  
12     print(estimar_tiempo(ingredientes_prueba))  
13
```

Imagen 5. Código completo módulo `tiempo.py`  
Fuente: Desafío Latam

9. La última funcionalidad a implementar contempla el mostrar los ingredientes, para ello el archivo `show.py`:

```
def mostrar_ingredientes(ingredientes):  
    print(f'Su masa es: {ingredientes["masa"]}')  
    print(f'Su salsa es: {ingredientes["salsa"]}')  
    print('Los ingredientes de su Pizza:')  
  
    for ing in ingredientes['ingredientes']:  
        print(f'- {ing}')
```

Esta función simplemente reportará la masa, la salsa y listará uno a uno los ingredientes línea por línea anteceditos de un guión (-).

Para probar el código se ejecutará la función utilizando un set de ingredientes de prueba para determinar si entrega el resultado esperado. El código completo luce así:



```
1
2 def mostrar_ingredientes(ingredientes):
3
4     print(f'Su masa es: {ingredientes["masa"]}')
5     print(f'Su salsa es: {ingredientes["salsa"]}')
6     print('Los ingredientes de su Pizza:')
7     for ing in ingredientes['ingredientes']:
8         print(f'- {ing}')
9
10
11 if __name__ == '__main__':
12     ingredientes_prueba = {'masa': 'Masa Tradicional',
13                             'salsa': 'Salsa de Tomate',
14                             'ingredientes': ['Queso']}
15
16     mostrar_ingredientes(ingredientes_prueba)
```

Imagen 6. Código completo módulo `show.py`

Fuente: Desafío Latam

Para terminar la app entonces ensamblamos las distintas partes dentro de `main.py`:

10. Para poder generar un Menú infinito que siempre muestre nuestro menú podemos hacer lo siguiente:

```
while True:
    opcion = opcion = input('¿Qué desea realizar?
    1. Cambiar tipo de Masa
    2. Cambiar tipo de Salsa
    3. Agregar Ingredientes
    4. Eliminar Ingredientes
    5. Ordenar con los Ingredientes Actuales

    0. Consultar ingredientes de la pizza

    Otro Número cancelará el pedido.
    > ''')
```

Esto permitirá que siempre que se ejecute alguna acción volvamos a este menú principal. Para determinar las acciones a ejecutar entonces se pueden utilizar condicionales que ejecutan cada una de nuestras características desarrolladas



**NOTA:** Usar `while True` debe ser utilizado con cautela ya que genera un Ciclo Infinito. En este programa los casos de salida vendrán dados por el uso de `exit()`.

Cambiar masa:

```
if opcion == '1':  
    eleccion = input("""Escoja el tipo de Masa:  
        T). Masa Tradicional  
        D). Masa Delgada  
        B). Masa con Bordes de Queso  
        > """).upper()  
    ingredientes_orden = tipo_masa(ingredientes_orden, eleccion)
```

Cambiar salsa:

```
elif opcion == '2':  
    eleccion = input('''Seleccione su tipo de Salsa:  
        T). Salsa de Tomate  
        A). Salsa Alfredo  
        B). Salsa Barbecue  
        P). Salsa Pesto  
        > ''').upper()  
    ingredientes_orden = tipo_salsa(ingredientes_orden, eleccion)
```

Agregar Ingrediente:

```
elif opcion == '3':  
    eleccion = int(input('''Seleccione su Ingrediente:  
        1). Tomate  
        2). Champiñones  
        3). Aceituna  
        4). Cebolla  
        5). Pollo  
        6). Jamón  
        7). Carne  
        8). Tocino  
        9). Queso  
        > '''))  
    ingredientes_orden =  
    agregar_ingrediente(ingredientes_orden,eleccion)
```

Eliminar Ingrediente:

```
elif opcion == '4':  
    eleccion = int(input('Selecione qué ingrediente quitar:  
        1). Tomate  
        2). Champiñones  
        3). Aceituna  
        4). Cebolla  
        5). Pollo  
        6). Jamón  
        7). Carne  
        8). Tocino  
        9). Queso  
        > '''))  
    ingredientes_orden = quitar_ingrediente(ingredientes_orden,  
    eleccion)
```

Ordenar:

```
elif opcion == '5':  
    tiempo = estimar_tiempo(ingredientes_orden)  
    print(f'Su Pizza estará lista en {tiempo} minutos')  
    ordenar = input('Desea ordenar ahora S/N: ').upper()  
  
    if ordenar == 'S':  
        print('Gracias por ordenar en Pizza JAT')  
        print('Disfrute su Pizza')  
        exit()
```

En este caso, `estimar_tiempo()` nos indica el tiempo que demorará en estar lista la Pizza. Luego de informar al usuario de esto, se confirma si se desea ordenar y en caso de aceptar se termina el programa.

Mostrar Ingredientes:

```
elif opcion == '0':  
    mostrar_ingredientes(ingredientes_orden)
```

Cancelación del Pedido:

```
else:
    print('Su pedido ha sido cancelado. Gracias por contactarse a
          Pizza JAT')
    exit()
```

Esto se implementa como una condición `else`, en caso de escoger opciones distintas a las dadas.

El código final se ve así:

```
1  from masa import tipo_masa
2  from salsa import tipo_salsa
3  from add import agregar_ingredient
4  from remove import quitar_ingredient
5  from show import mostrar_ingredientes
6  from tiempo import estimar_tiempo
7
8  ingredientes_orden = {'masa': 'Masa Tradicional',
9                        'salsa': 'Salsa de Tomate',
10                       'ingredientes': ['Queso']}
11
12
13 print('¡Gracias por ordenar con nosotros!')
14 while True:
15     opcion = input('¿Qué desea realizar?
16     1. Cambiar tipo de Masa
17     2. Cambiar tipo de Salsa
18     3. Agregar Ingredientes
19     4. Eliminar Ingredientes
20     5. Ordenar con los Ingredientes Actuales
21
22     0. Consultar ingredientes de la pizza
23
24     Otro Número cancelará el pedido.
25     > ')
26
27     if opcion == '1':
28         eleccion = input("""Escoja el tipo de Masa:
29         T). Masa Tradicional
30         D). Masa Delgada
31         B). Masa con Bordes de Queso
32         > """).upper()
33         ingredientes_orden = tipo_masa(ingredientes_orden, eleccion)
34
35     elif opcion == '2':
36         eleccion = input('Seleccione su tipo de Salsa:
37         T). Salsa de Tomate
38         A). Salsa Alfredo
39         B). Salsa Barbecue
40         P). Salsa Pesto
41         > ').upper()
42         ingredientes_orden = tipo_salsa(ingredientes_orden, eleccion)
```

Imagen 7. Código completo script `main.py` (Parte I)

Fuente: Desafío Latam

```
43
44     elif opcion == '3':
45         eleccion = int(input('Selecione su Ingrediente:
46         1). Tomate
47         2). Champiñones
48         3). Aceituna
49         4). Cebolla
50         5). Pollo
51         6). Jamón
52         7). Carne
53         8). Tocino
54         9). Queso
55         > '''))
56         ingredientes_orden = agregar_ingrediente(ingredientes_orden,eleccion)
57
58     elif opcion == '4':
59         eleccion = int(input('Selecione qué ingrediente quitar:
60         1). Tomate
61         2). Champiñones
62         3). Aceituna
63         4). Cebolla
64         5). Pollo
65         6). Jamón
66         7). Carne
67         8). Tocino
68         9). Queso
69         > '''))
70         ingredientes_orden = quitar_ingrediente(ingredientes_orden, eleccion)
71
72     elif opcion == '5':
73         tiempo = estimar_tiempo(ingredientes_orden)
74         print(f'Su Pizza estará lista en {tiempo} minutos')
75         ordenar = input('Desea ordenar ahora S/N: ').upper()
76
77         if ordenar == 'S':
78             print('Gracias por ordenar en Pizza JAT')
79             print('Disfrute su Pizza')
80             exit()
81
82
83     elif opcion == '0':
84         mostrar_ingredientes(ingredientes_orden)
85
86     else:
87         print('Su pedido ha sido cancelado. Gracias por contactarse a Pizza JAT')
88         exit()
```

Imagen 8. Código completo script `main.py` (Parte II)

Fuente: Desafío Latam

## 11. Vamos más allá (Refactorizando):

Si bien el programa comenzó con un buen diseño modularizado resulta ser un programa muy largo. Actualmente tiene 88 líneas. Cómo podríamos generar un `main.py` más conciso que tuviera lo justo y necesario para poder entender de buena manera el flujo.

Por ejemplo, es posible notar que en todas las opciones se despliega un menú que es, en general, bastante largo. Una cosa que podemos hacer es almacenar eso en variables que se definan en un script aparte. Lo mismo pasa con nuestra Pizza base, no es fundamental tenerla en el archivo principal.

Para ello crearemos un archivo llamado `datos.py`. Este contendrá variables que acortarán bastante el código.

```
ingredientes = {'masa': 'Masa Tradicional',
                'salsa': 'Salsa de Tomate',
                'ingredientes': ['Queso']}

T_SALSA = '''Seleccione su tipo de Salsa:
    T). Salsa de Tomate
    A). Salsa Alfredo
    B). Salsa Barbecue
    P). Salsa Pesto
> '''

T_MASA = '''Escoja el tipo de Masa:
    T). Masa Tradicional
    D). Masa Delgada
    B). Masa con Bordos de Queso
> '''

AG_INGREDIENTE = '''Seleccione su Ingrediente:
    1). Tomate
    2). Champiñones
    3). Aceituna
    4). Cebolla
    5). Pollo
    6). Jamón
    7). Carne
    8). Tocino
    9). Queso
> '''

QT_INGREDIENTE = '''Seleccione qué ingrediente quitar:
    1). Tomate
    2). Champiñones
    3). Aceituna
    4). Cebolla
    5). Pollo
    6). Jamón
    7). Carne
    8). Tocino
    9). Queso
> '''

menu = '''¿Qué desea realizar?
```

1. Cambiar tipo de Masa
2. Cambiar tipo de Salsa
3. Agregar Ingredientes
4. Eliminar Ingredientes
5. Ordenar con los Ingredientes Actuales

0. Consultar ingredientes de la pizza

Otro Número cancelará el pedido.

> '''

El texto asociado a cada menú será una variable, por lo tanto podríamos reescribir nuestro archivo `main.py` de la siguiente manera:

```
1  from masa import tipo_masa
2  from salsa import tipo_salsa
3  from add import agregar_ingrediente
4  from remove import quitar_ingrediente
5  from show import mostrar_ingredientes
6  from tiempo import estimar_tiempo
7  import datos as d
8  ingredientes_orden = d.ingredientes
9
10 print('¡Gracias por ordenar con nosotros!')
11 while True:
12     opcion = input(d.menu)
13
14     if opcion == '1':
15         eleccion = input(d.T_MASA).upper()
16         ingredientes_orden = tipo_masa(ingredientes_orden, eleccion)
17
18     elif opcion == '2':
19         eleccion = input(d.T_SALSA).upper()
20         ingredientes_orden = tipo_salsa(ingredientes_orden, eleccion)
21
22     elif opcion == '3':
23         eleccion = int(input(d.AG_INGREDIENTE))
24         ingredientes_orden = agregar_ingrediente(ingredientes_orden,eleccion)
25
26     elif opcion == '4':
27         eleccion = int(input(d.QT_INGREDIENTE))
28         ingredientes_orden = quitar_ingrediente(ingredientes_orden, eleccion)
29
30     elif opcion == '5':
31         tiempo = estimar_tiempo(ingredientes_orden)
32         print(f'Su Pizza estará lista en {tiempo} minutos')
33         ordenar = input('Desea ordenar ahora S/N: ').upper()
34
35         if ordenar == 'S':
36             print('Gracias por ordenar en Pizza JAT')
37             print('Disfrute su Pizza')
38             exit()
39
40
41     elif opcion == '0':
42         mostrar_ingredientes(ingredientes_orden)
43
44     else:
45         print('Su pedido ha sido cancelado. Gracias por contactarse a Pizza JAT')
46         exit()
```

Imagen 9. Código completo script `main.py` (Refactorizado)

Fuente: Desafío Latam

Se ha podido reducir el largo del código casi a la mitad, de 88 a 46 líneas. Además es posible entender de mejor manera el contenido de este.

Lo único que se agregó adicional fueron las líneas 7 y 8 que importan el script datos, y se redefine `ingredientes_orden` que ahora vendrá desde el módulo de datos.

## 12. Vamos aún más allá (Mejorando la experiencia de usuario):

Si comienza a realizar pruebas se puede ver que ciertos menú pasan demasiado rápido. Además, si se hacen muchos procesos en el código se notará que todo el historial de comandos quedará en la pantalla. Es por eso que acá se pueden utilizar pausas y limpiezas de pantalla que permitan un mejor flujo de la app.

Para esto se decide que:

- Cada menú que se despliegue se debe realizar en una ventana limpia.
- Entre cada elección se debe dejar una línea de separación para la reaparición del menú principal.
- Menú de muestra de Ingredientes debe permanecer por 5 segundos antes de que reaparezca el menú principal.
- Para los mensajes de finalización de programa por cancelación o por confirmación de orden se dejarán 3 segundos.

El programa final quedará así:

```
1  from masa import tipo_masa
2  from salsa import tipo_salsa
3  from add import agregar_ingredient
4  from remove import quitar_ingredient
5  from show import mostrar_ingredientes
6  from tiempo import estimar_tiempo
7  import datos as d
8  import time
9  import os
10 import sys
11 clear = 'cls' if sys.platform == 'win32' else 'clear'
12 ingredientes_orden = d.ingredientes
13
14 print('¡Gracias por ordenar con nosotros!')
15 while True:
16     os.system(clear)
17     opcion = input(d.menu)
18
19     if opcion == '1':
20         os.system(clear)
21         eleccion = input(d.T_MASA).upper()
22         ingredientes_orden = tipo_masa(ingredientes_orden, eleccion)
23         print('')
24     elif opcion == '2':
25         os.system(clear)
26         eleccion = input(d.T_SALSA).upper()
27         ingredientes_orden = tipo_salsa(ingredientes_orden, eleccion)
28         print('')
29     elif opcion == '3':
30         os.system(clear)
31         eleccion = int(input(d.AG_INGREDIENTE))
32         ingredientes_orden = agregar_ingredient(ingredientes_orden, eleccion)
33         print('')
```



Imagen 10. Código completo script `main.py` (Mejorando Experiencia Usuario Parte I)  
Fuente: Desafío Latam

```
35 elif opcion == '4':
36     os.system('clear')
37     eleccion = int(input(d.QT_INGREDIENTE))
38     ingredientes_orden = quitar_ingredientes(ingredientes_orden, eleccion)
39     print('')
40 elif opcion == '5':
41     os.system('clear')
42     tiempo = estimar_tiempo(ingredientes_orden)
43     print(f'Su Pizza estará lista en {tiempo} minutos')
44     ordenar = input('Desea ordenar ahora S/N: ').upper()
45     print('')
46     if ordenar == 'S':
47         print('Gracias por ordenar en Pizza JAT')
48         print('Disfrute su Pizza')
49         time.sleep(3)
50         exit()
51
52 elif opcion == '0':
53     os.system('clear')
54     mostrar_ingredientes(ingredientes_orden)
55     time.sleep(3)
56     print('')
57 else:
58     os.system('clear')
59     print('Su pedido ha sido cancelado. Gracias por contactarse a Pizza JAT')
60     time.sleep(3)
61     exit()
```

Imagen 11. Código completo script `main.py` (Mejorando Experiencia Usuario Parte II)  
Fuente: Desafío Latam

## Sugerencias de mejoras

Si bien lo que acabamos de hacer es un programa de bastante mayor complejidad es importante entender que aún hay bastantes cosas que se pueden hacer para mejorarlo:

Algunas ideas de qué implementar:

1. Validaciones, en el caso de que no se ingrese un valor permitido, mostrar nuevamente el menú indicando que se agregó un valor no permitido.
2. Opción para tener un stock con los ingredientes. En caso que el stock llegue a cero informar que el ingrediente no está disponible.
3. Opción para un ingrediente doble, como por ejemplo, doble queso.
4. Permitir la opción de Pizzas pre hechas, es decir que ya contengan una combinación de ingredientes.
5. Consultar por agregar o eliminar ingredientes continuamente (sin tener que volver al menú principal) hasta que el usuario indique que ya no desea hacer modificaciones.
6. Asignar un precio a cada ingrediente y calcular el valor final de la pizza, y mostrarlo al escoger la opción "ordenar".

Y por supuesto muchas más que pudieran aplicar para hacer la experiencia del usuario.