



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
ESCUELA DE INGENIERÍA  
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN  
IIC2233 — PROGRAMACIÓN AVANZADA 2021-2

# Actividad de Repaso

¡Bienvenid@ a la actividad de repaso de la semana! Esta actividad **no será evaluada**, ya que su objetivo es que sea realizada junto al profesor durante la clase.

Te recomendamos que solo **sigas la clase durante la realización del ejercicio**, y posteriormente a esta, practiques lo aprendido tratando de resolver la actividad por tu cuenta, utilizando este enunciado como guía. Como siempre, después de la clase se disponibilizará el video de esta para que puedas usarlo como apoyo en caso de que presentes dificultades con alguna parte del ejercicio.

## Introducción

Al no tener nada que programar de tu ramo favorito durante la semana de receso, decidiste dedicarte a otras aficiones para evitar el aburrimiento. En particular, te pusiste como meta el **mejorar tus habilidades de chef** y así poder cocinar las **mejores hamburguesas del mundo**. Sin embargo, al momento de preparar la cocina, te encontraste con un gran problema: ¡No había ningún ingrediente disponible en tu casa para cocinar!

Estabas a punto de rendirte con tu sueño cuando recordaste todo lo que habías visto en **Programación Avanzada** durante el semestre. No quiero desaprovechar lo aprendido, pensaste en una gran idea: crear un programa que, haciendo uso de **interfaces gráficas**, te permitiera practicar tus habilidades de cocina, y así apoyarte en tu camino a ser un **Master DCChef**.



Master DCChef:  
Burger Edition

Figura 1: Logo de Master DCChef

## Flujo del programa

El objetivo del programa es **simular una cocina de hamburguesas**, donde deberás **recibir pedidos, cocinar las hamburguesas y luego agregar los ingredientes necesarios** para cumplir con la orden del cliente.

Cuando el programa comienza, se abrirá automáticamente la **Ventana de pedidos**, donde podrás aceptar un pedido de hamburguesa, el cual consiste en una lista de ingredientes en un orden específico.

Una vez que aceptes dicho pedido, podrás avanzar a la **Ventana de Parrilla**, donde deberás cocinar las hamburguesas, procurando que estas alcancen su punto de cocción óptimo.

Luego, cuando hayas cocinado las hamburguesas necesarias, avanzarás a la **Ventana de Preparación**, en donde podrás armar tu hamburguesa según lo pedido en la orden y finalmente entregarla al cliente, quien te indicará si su orden fue acatada correctamente o no.

## Parte 1: Ventana de Pedidos

Para esta parte, primero deberás implementar la **Ventana de pedidos**, la cual corresponde a la ventana inicial que aparece en cuanto se ejecuta el programa.

La ventana de juego debe tener una sola funcionalidad: deberás poder presionar un botón para **recibir el pedido del cliente**. Un pedido se compone por un listado de ingredientes, en un orden específico, que deberá contener tu hamburguesa. Los pedidos siempre comenzarán y terminarán con una rebanada de pan, y tendrán como máximo dos hamburguesas dentro.

Una vez que recibas tu pedido y lo almacenes, automáticamente se cerrará esta ventana y podrás avanzar a la **Ventana de Parrilla** para comenzar a preparar tu hamburguesa.

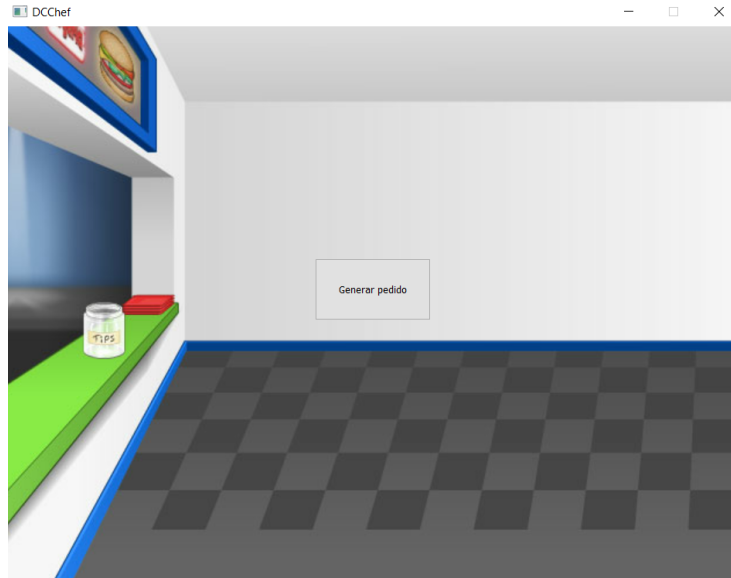


Figura 2: Ejemplo de Ventana de Pedido

En esta sección deberán implementar lo siguientes:

- Botón "Generar pedido": Deberás agregar un botón en el constructor de la clase `VentanaPedido` con el texto **"Generar pedido"**, ubicarlo en el centro de la ventana y conectarlo a al método `self.pedir_generar_pedido`.
- `self.pedir_generar_pedido`: También debes implementar este método, donde lo único que hay que hacer es emitir la señal `senal_generar_pedido`.
- `senal_generar_pedido`: Deberás conectar la señal `senal_generar_pedido` de `VentanaPedido` con la función `generar_pedido` de `DCChef` en el archivo `main.py`.

## Parte 2: Ventana de Parrilla

Esta ventana aparecerá una vez recibas y almacenes tu orden a cocinar. Es donde deberás **cocinar las hamburguesas**, teniendo ojo de que estas no se quemen ni les falte cocción. Para esto, tendrás disponible una parrilla con capacidad para **2 hamburguesas**.

Para colocar una hamburguesa en la parrilla, deberás poder hacer *click* sobre la **fuelle de hamburguesas** ubicada en el mesón. Esto provocará que automáticamente se ubique una hamburguesa sin cocinar en uno de los espacios disponibles de la parrilla.

Al estar en la parrilla, la hamburguesa comenzará a cocinarse, haciendo que cada fase de su cocción dure una cantidad de tiempo determinada por los valores indicados en un parámetro denominado `TIEMPOS_COCCION`. Cuando la hamburguesa se encuentre cocinada adecuadamente, podrás hacer *click* en ella para retirarla de la parrilla.

En caso de que no retires la hamburguesa a tiempo o la retires demasiado pronto, esta deberá poder ser desechada.

Una vez cocines las hamburguesas necesarias para la orden, podrás seleccionar el botón para avanzar a la **Ventana de Preparación**.

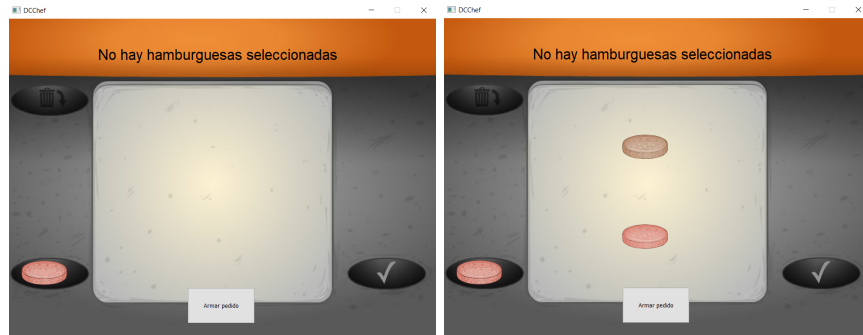


Figura 3: Ejemplos de la Ventana de Parrilla

En la clase `Hamburguesa` de `ventana_coccion.py` deberás implementar lo siguiente:

- `self.coccion`: Deberás completar este método como una *property* con *setter* y *getter*. El cual recibirá un valor correspondiente a un entero entre 1 y 5, el cual deberás guardar en un atributo privado y establecer como pixmap del objeto el *sprite* de la hamburguesa que corresponde al valor entregado. El *getter* solo retorna el valor guardado en el atributo oculto.
- `self.mousePressEvent`: Deberás completar este método que recibe un evento de click, si el atributo `self.en_estante` no está activo, entonces deberás emitir la señal `senal_seleccionar_hamburguesa` enviando el id. En otro caso, deberás ignorar el evento.

En la clase `VentanaCoccion` de `ventana_coccion.py` deberás implementar lo siguiente:

- `self.procesar_click`: Este método recibe como argumentos dos enteros `x`, `y` que representan coordenadas. En caso de que la *property* `hamburguesa_seleccionada` sea verdad, deberás verificar con el método `self.procesar_click_area` si es que `x`, `y` están dentro de `CLICK_AREA_BOTAR` o `CLICK_AREA_SACAR`, en caso de que se cumpla alguno deberás emitir la señal `senal_botar_hamburguesa` o `senal_sacar_hamburguesa` respectivamente, entregando el id de la hamburguesa seleccionada.
- `self.agregar_hamburguesa`: Recibe un id y un nivel de cocción, con los que deberás instanciar una clase `Hamburguesa`. Deberás verificar si la lista `self.hamburguesas` tiene espacio en 0 o 1, y guardar ahí la hamburguesa. Luego deberás mover la hamburguesa recién creada a las posiciones `POSICION_HAMBURGUESA_ARRIBA` o `POSICION_HAMBURGUESA_ABAJO` según corresponda su posición en la lista (0 o 1 respectivamente).
- `self.eliminar_hamburguesa`: Recibe un id, con el cual debes obtener la hamburguesa con el método `self.obtener_hamburguesa`. Debes ocultar la hamburguesa, llamar al método `self.vaciar_espacio_parrilla` y establecer la *property* `hamburguesa_seleccionada` como `None`.
- Señal `senal_botar_hamburguesa`: Debes conectar la señal con el método `botar_hamburguesa` de `DCChef` en `main.py`.
- Señal `senal_sacar_hamburguesa`: Debes conectar la señal con el método `terminar_cocer_hamburguesa` de `DCChef` en `main.py`.
- Señal `senal_agregar_hamburguesa`: Debes conectar la señal con el método `agregar_hamburguesa_parrilla` de `DCChef` en `main.py`.

En la clase `Hamburguesa` de `backend.py` deberás implementar lo siguiente:

- `self.avanzar_coccion`: Recibe un entero `tiempo`, el cual deberá esperar antes de continuar. Luego en caso de que el atributo `self.usuario_saco` sea falso, deberás aumentar en 1 la cocción de la hamburguesa y emitir la señal `senal_coccion` con el id y la cocción actual.
- `self.run`: Deberás recorrer los tiempos en `TIEMPOS_COCCION` y para cada uno llamar al método `self.avanzar_coccion` entregando el tiempo correspondiente.
- Señal `senal_coccion_hamburguesa`: Debes conectar la señal con el método `cocer_hamburguesa` de `VentanaCoccion` en `main.py`.

En la clase `DCChef` de `backend.py` deberás implementar lo siguiente:

- `self.agregar_hamburguesa_parrilla`: Deberás revisar que haya menos de dos hamburguesas en la lista `self.hamburguesas_parrilla` antes de continuar. Luego, si se puede, debes instanciar una `Hamburguesa`, agregarla a la lista mencionada, emitir la señal `senal_hamburguesa_agregada` con su id y nivel de cocción y por ultimo iniciar el `thread` de la hamburguesa.

### Parte 3: Ventana de Preparación

Esta es la última etapa de creación de tu hamburguesa. Aquí deberás crear una ventana que contenga la **información del pedido a crear**, todos los **ingredientes** disponibles para generar una hamburguesa.

La ventana debe permitir seleccionar los ingredientes que lleva el pedido, en el orden específico pedido, para armar la hamburguesa. Para esto, se deberá poder hacer *click* sobre la imagen de cada ingrediente necesario.

Una vez que coloques la rebanada superior de pan en tu hamburguesa, el programa debe detectar que tu hamburguesa ya fue finalizada. A continuación, deberás verificar si la hamburguesa creada coincide con el pedido del cliente, y notificar mediante la interfaz si tu preparación fue acertada o no.

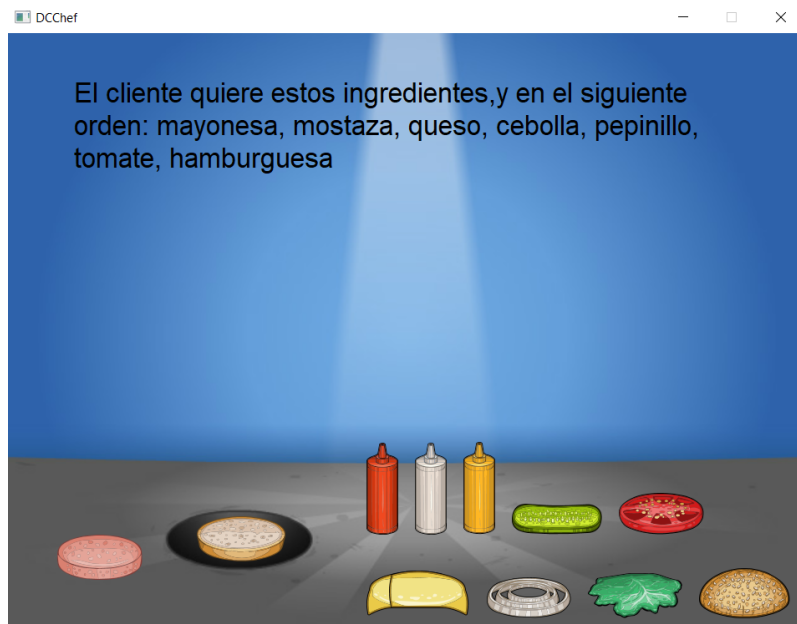


Figura 4: Ejemplo de Ventana de Preparación

En la clase `Ingrediente` de `ventana_armado.py` deberás implementar lo siguiente:

- `self.mousePressEvent`: Deberás comprobar si el Ingrediente es `clickable`, en cuyo caso deberás emitir la señal `senal_ingrediente_agregado` con el tipo e id del ingrediente. En otro caso, deberás ignorar el evento.

En la clase `VentanaArmado` de `ventana_armado.py` deberás implementar lo siguiente:

- `self.poner_ingredientes`: Recibirás una lista de tuplas con el id y nivel de cocción de las hamburguesas preparadas. Con lo cual, a partir de la posición `POSICION_HAMBURGUESAS` para cada hamburguesa en la lista deberás instanciar un `Ingrediente` del tipo `"hamburguesa_{nivel_coccion}"`, colocarlo en las coordenadas correspondientes y guardarlo en el diccionario `self.hamburguesas` con su id como llave, además debes disminuir en `TAMANO_INGREDIENTES[1]` el eje vertical entre cada hamburguesa.

Luego, a partir desde la posición `POSICION_INGREDIENTES_ARRIBA` para cada tipo de ingrediente en `SPRITES_INGREDIENTES`, debes instanciar un `Ingrediente`, luego verificar si es que la coordenada x actual más el ancho del ingredientes superan el ancho de la ventana, entonces cambiar las coordenadas a `POSICION_INGREDIENTES_ABAJO`, para luego mover el ingrediente a esa posición y aumentar en `ingrediente.width() + TAMANO_INGREDIENTES[0]` la coordenada x entre cada ingrediente.

Tanto hamburguesas como ingredientes deberás mostrarse con `show` para aparecer en pantalla.

En la clase `DCChef` de `backend.py` deberás implementar lo siguiente:

- `self.empezar_armado`: Deberás preparar un diccionario con la información requerida para comenzar a armar las hamburguesas. Debes generar un texto que indique el orden en que se quieren los ingredientes dados por `self.ingredientes_pedido` y una lista con tuplas que contengan el id y nivel de cocción de cada hamburguesa en `self.hamburguesas_cocidas`. Con esos dos datos, deberás generar un diccionario con el formato `{ "hamburguesas": lista, "pedido": texto }`, el cual debes enviar por la señal `senal_comenzo_armado`.
- Señal `senal_comenzo_armado`: Debes conectar la señal con el método `comenzo_armado` de `VentanaArmado` en `main.py`.

## Parte 4: Disfruta tu Hamburguesa

Si ya has completado las tres ventanas del ejercicio, ¡Has terminado! Ahora puedes disfrutar de tus nuevas habilidades de *Master DCChef* y mostrarle al mundo tus mejores preparaciones.