



**Certified Tech  
Developer**  
The Ultimate Degree

## Actividad de repaso

### PARTE TEÓRICA

Esta actividad cuenta con 20 preguntas teóricas, la idea es girar la ruleta y responder las preguntas sobre los diferentes temas que se revisaron durante el cursado.

Url: <https://wordwall.net/es/resource/31606616/repasemos>

### PARTE PRÁCTICA

El objetivo de la práctica es trabajar con la aplicación web de [Comida YA!](#)

Se tiene un software denominado **Comida Ya!** El cual nos permite realizar la compra y venta de productos. El mismo cuenta con las siguientes funcionalidades: un registro (tanto para un usuario administrador y un usuario cliente), un Login, un ABM de productos y pedidos (Alta-Baja-Modificación) y Listar productos y pedidos.

## REQUERIMIENTOS

### ADMINISTRADOR

- El sistema debe permitir registrar un tipo de usuario administrador. Los datos necesarios son: nombre, apellido, email y contraseña. Se debe validar que todos los campos estén completos y tengan el formato correcto.
- El sistema debe permitir loguear a los usuarios. En caso de que el email o la contraseña sean incorrectos, se debe mostrar un mensaje de error.
- El usuario administrador podrá crear, modificar y eliminar productos. Los campos obligatorios de los mismos son: nombre, url de la imagen y precio. En caso de que algún campo esté incompleto, indicarlo con un mensaje de error.
- El usuario administrador podrá cambiar el estado del pedido. Si el pedido llega al estado Confirmado, este no se podrá eliminar.
- La aplicación debe ser responsive, es decir debe adaptarse a las diferentes resoluciones del navegador hasta llegar a la versión móvil.

### CLIENTE

- El sistema debe permitir registrar un tipo de usuario cliente. Los datos necesarios son: nombre, apellido, email y contraseña. Se debe validar que todos los campos estén completos y tengan el formato correcto.
- El sistema debe permitir loguear a los usuarios. En caso de que el email o la contraseña sean incorrectos, se debe mostrar un mensaje de error.
- El usuario cliente podrá ver todos los productos disponibles y agregarlos al carrito de compras. Al hacer clic en un producto se debe mostrar un mensaje indicando que el producto fue añadido al carrito.
- El usuario cliente podrá ver su carrito de compras, elegir su forma de pago y confirmar la compra. Si el carrito no posee productos el botón “confirmar compra” debe aparecer deshabilitado.
- La aplicación debe ser responsive, es decir debe adaptarse a las diferentes resoluciones del navegador hasta llegar a la versión móvil.

## Ejercicios

1) Redactar un caso de prueba aplicando partición de equivalencia y otro caso de prueba aplicando la técnica de tabla de decisión.

### Partición de Equivalencia

- **Rango válido:** \$0 a \$1000 (inclusive).
- **Rango inválido:** precios menores a \$0 (negativos) y superiores a \$1000.

Se verificará que los precios de los productos sean mayores a \$0 y con tope en \$1000 finales (inclusive). Fuera de este rango, esos precios serán rechazados.

### Tablas de decisión

Contemplando que el precio del producto seleccionado es superior a \$0, entonces se podrá agregar al carrito y verificar en esta última opción. Posteriormente, se podrá también confirmar la compra ingresando al historial de órdenes.

2) Identificar tres (3) defectos del sistema en cualquiera de los menús disponibles. (No se requiere escribirlos en formato de template)

- ☒ El sistema admite precios negativos.
- ☒ El sistema permite agregar productos que no sean comida (por ejemplo: tornillos).
- ☒ Existen productos que no tienen imagen.

3) Escribir un (1) caso de prueba positivo y un (1) caso de prueba negativo. No se requiere escribirlos en formato de template.

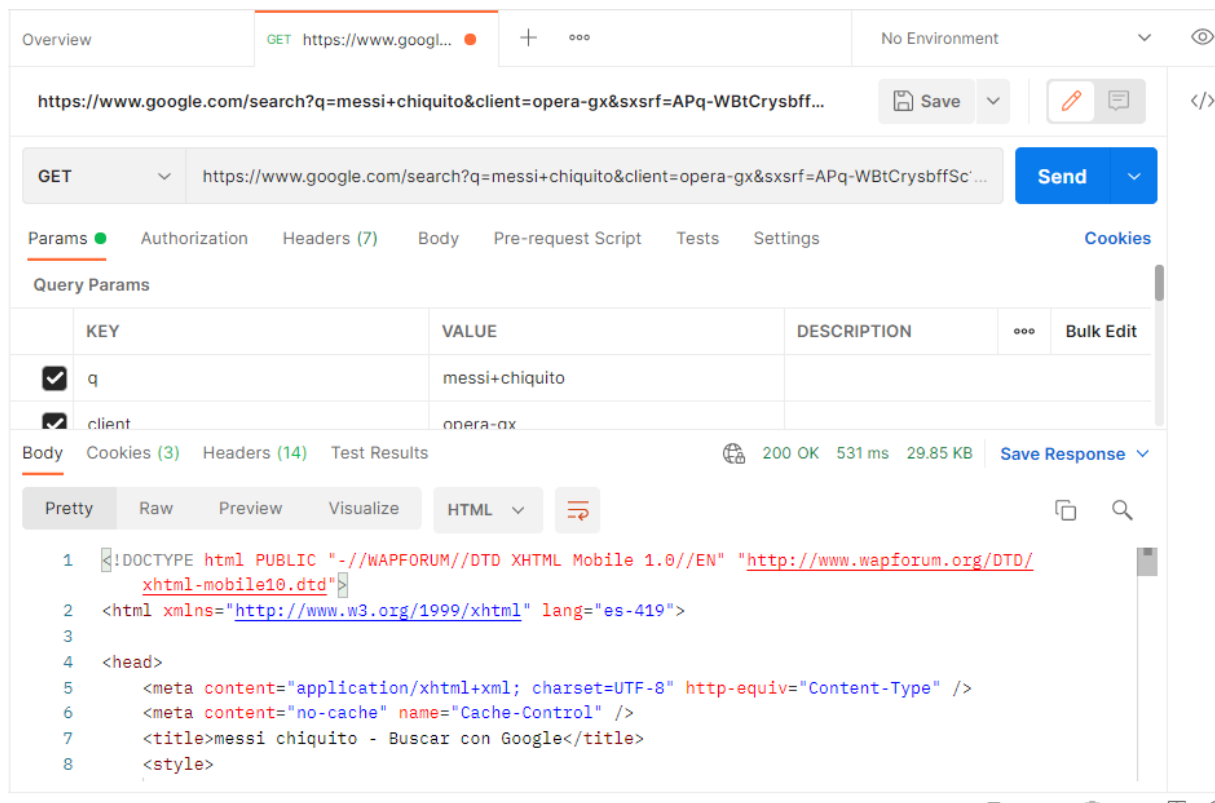
- ✓ **CP (+):** realizar una confirmación de compra con el carrito con al menos un producto.
- ✓ **CP (-):** forzar a realizar una compra con el carrito vacío (sin productos).

4) Redactar brevemente una (1) prueba funcional y una (1) prueba no funcional. No se requiere escribirlos en formato de template.

**Funcional:** al agregar un producto al carrito, se espera que pueda visualizarse dentro de la sección “Carrito”.

**No Funcional:** que el sistema sea responsive, pudiendo adaptarse a las diferentes resoluciones, ya sea escritorio como mobile.

5) Identificar en la aplicación de [Comida YA!](#) una petición GET.



The screenshot shows the Chrome DevTools network panel. A GET request to `https://www.google.com/search?q=messi+chiquito&client=opera-gx&sxsrf=APq-WBtCrysbf...` is selected. The 'Query Params' tab is active, displaying a table of query parameters:

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	q	messi+chiquito			
<input checked="" type="checkbox"/>	client	opera-gx			

The 'Body' tab is also active, showing the HTML response. The first few lines of the HTML are:

```

1 <!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.0//EN" "http://www.wapforum.org/DTD/xhtml-mobile10.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml" lang="es-419">
3
4 <head>
5   <meta content="application/xhtml+xml; charset=UTF-8" http-equiv="Content-Type" />
6   <meta content="no-cache" name="Cache-Control" />
7   <title>messi chiquito - Buscar con Google</title>
8   <style>

```