

Easy-Order: Aplicación web para realizar pedidos en restaurantes

Abdullah Bashir Yasmin

Escola Politècnica Superior d'Enginyeria de Vilanova i la Geltrú

1. RESUMEN

Actualmente las colas de espera en los restaurantes de comida rápida son largas. A pesar de tener máquinas para poder hacer el pedido, la espera puede variar y ser un problema.

Una posible solución habría sido tener más máquinas, pero también requiere más espacio.

La otra solución es disponer de alguna aplicación que agilice este proceso. Easy-order es una idea que tiene como objetivo proporcionar una forma cómoda y rápida para que un consumidor pueda hacer el pedido en el restaurante mediante una aplicación Web. La finalidad es que estando en un restaurante se haga uso de una aplicación Web, que ésta misma nos permitiría escanear un código QR (éste siendo el identificador del restaurante) y acceder al menú del restaurante para poder hacer el pedido.

2. MOTIVACIÓN PERSONAL

La idea de este proyecto nace de una experiencia personal. Como asistente asiduo a un restaurante bastante concurrido de comida rápida, me di cuenta de que la mayoría de las veces tengo que soportar una larga espera para hacer el pedido en las máquinas TPV. Una vez hecho el pedido, el TPV imprime el "ticket" en el que se indica el número de recogida o seguimiento del pedido. A continuación, tengo que permanecer un buen rato observando un monitor, situado en el propio restaurante, donde aparecerá tarde o temprano la indicación de que el pedido está listo, y entonces podré pasar a recogerlo. Con tanta multitud de gente, se hace difícil e incómodo tener que esperar mirando el monitor a cada rato para ver cómo va mi pedido.

De aquí es donde me surge la idea de tener una aplicación que pudiera hacer que el proceso de realizar un pedido fuera mucho más simple y fácil. Y a la misma vez estar notificado lo que sucede con el pedido.

3. ESTRUCTURA DE LAS APLICACIONES

En el lado del consumidor/cliente es una aplicación Web ya que así no "molestamos" al usuario para que descargue una aplicación móvil. Sino que desde el mismo sitio Web le damos la opción de escanear el código QR. Una vez escaneado el QR, nos dirigiría al menú del restaurante donde podríamos seleccionar los platos que deseamos consumir (figura 1),

próximamente se procederá a hacer el pago. A continuación, la cocina procesa el pedido y una vez preparado el pedido, se le notificará (mediante websockets) al cliente en el mismo navegador conforme el pedido está listo para recoger. La idea es que en cada paso en el que se encuentre el pedido se le notifique al cliente (figura 2 y figura 3).



Figura 1 – Carta para un restaurante

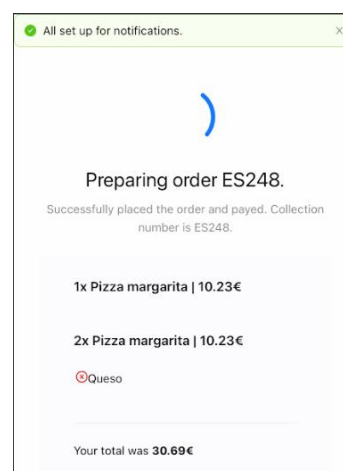


Figura 2 – Pedido realizado

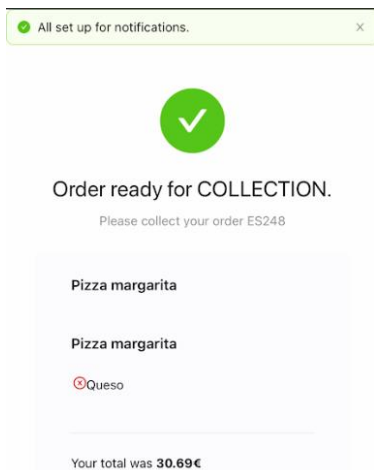


Figura 3 – Pedido listo para recoger

En cuanto a la administración tendríamos varias vistas:

- Ver estadísticas sobre el rendimiento del restaurante
- Ver, añadir, editar o eliminar platos (figura 4)
- Ver, añadir, editar o eliminar categorías
- Ver y gestionar pedidos
- Ver el código QR del restaurante

Nombre	Descripción	Precio	Ingredientes	Etiquetas	Foto	Activo
Big Mac	Ever wondered what's on a Big Mac®? The McDonald's Big Mac® is a 100% beef burger with a taste like no other. The mouthwatering perfection starts with	10.5€	[Burger, Bun]	[]		Si
Cheese burger	Enjoy that delicious and cheesy McDonald's Cheeseburger! Our easy classic cheeseburger starts with a 100% beef burger patty.	3.75€	[Burger, Bun]	[]		Si
Coca-Cola	Enjoy a cold, refreshing Coca-Cola® soda from McDonald's that complements all your menu favorites.	5.0€	[Burger, Bun]	[]		Si
Dr Pepper	Satisfy your taste buds with the unique taste of Dr Pepper® at McDonald's. Wondering what flavor is Dr Pepper®? This classic and refreshing fountain d	4.0€	[Burger, Bun]	[]		Si
Gloves	Gloves	5.5€	[wool, yarn]	[]		Si

Figura 4 – Vista principal de platos

4. TECNOLOGÍAS

Se ha utilizado Django, una librería de Python, para la aplicación de administración del restaurante. Django permite crear páginas web estáticas de forma rápida y sencilla, incluye validación de datos y mecanismos de autenticación y permisos. Es seguro y popular para desarrollo web [1]. También se ha utilizado Django Rest Framework para la parte de la API.

Para el desarrollo front-end se ha utilizado ReactJS, la librería más popular [2] que facilita el renderizado de páginas y el manejo de datos sin recargar la página. Se ha utilizado AntDesign para el diseño de la interfaz de usuario.

Se ha optado por una aplicación web en lugar de una App móvil para simplificar las cosas para el usuario y evitar descargas y actualizaciones. Se ha utilizado PostgreSQL como base de datos relacional debido a la naturaleza de la información del proyecto.

Tanto el panel como la API están montados en contenedores Docker [3], lo que hace que las aplicaciones sean eficientes, portables y fáciles de manejar.

5. BASE DE DATOS

La base de datos escogida fue PostgreSQL. Ya que permite datos como JSON o UUID que para este proyecto han sido útiles para guardar ingredientes o etiquetas de platos.

Tanto la aplicación para el consumidor, en panel de administración como la API interactúan con ésta única base de datos.

Se optó por el uso de UUID para prevenir borrados masivos de información en caso de que ocurra una filtración en la base de datos. Es importante tener en cuenta que uno de los ataques más simples consiste en recorrer los registros de la tabla, incrementar gradualmente el ID y eliminar todos los registros en el proceso.

6. ALOJAMIENTO, DOMINIO Y SERVIDOR

El servidor usado para alojar ha sido el que pone en venta DigitalOcean [5]. El host tiene las siguientes especificaciones:

- 2 GB de memoria RAM
- 50 GB de disco SSD
- CPU intel regular
- 1 vCPU

No es una máquina potente, pero basta para aguantar las conexiones y peticiones.

En cuanto a la promoción de la aplicación, la idea inicial sería encontrar un distribuidor de los restaurantes. Ya sea un distribuidor de comida o de publicidad. Se hablaría con éste distribuidor para promocionar la aplicación de Easy-order. Los primeros días serían de prueba para ver el rendimiento y los posibles fallos en un escenario más real. También se implementaría el pago real y todo estaría más controlado para poder encontrar fallos.

La compra del dominio ha sido de la web GoDaddy [6].

Próximamente se ha configurado el servidor Proxy. Para ello he usado la tecnología HaProxy [7].

Se ha seguido una guía [8] para configurar una mínima configuración en el proyecto. Después de la instalación, se ha configurado el protocolo HTTPS para garantizar la seguridad de la aplicación y permitir el uso de la cámara en la aplicación del cliente. Esto se ha logrado generando certificados utilizando la herramienta LetsEncrypt [9] y asociándolos para configurar el HTTPS.

Se realizó una prueba de carga utilizando la herramienta Locust [10] para obtener métricas del sistema y evaluar el rendimiento de la aplicación. Se utilizó Grafana [11] para visualizar estas métricas. Locust permitió simular múltiples usuarios realizando diferentes solicitudes al servidor simultáneamente, lo que ayudó a evaluar tanto el rendimiento del servidor como el de la aplicación Django.

Para promocionar la aplicación, la idea inicial es encontrar un distribuidor para promocionar la aplicación EasyOrder en restaurantes. Se realizarían pruebas iniciales para evaluar el rendimiento y posibles fallos en un entorno real.

Además, se implementaría el pago real y se mejorarían los controles para identificar cualquier problema.

7. MEJORAS

Como todo, esta aplicación aun requiere de bastantes mejoras tanto en el lado de administración como en la aplicación. Algunas de estas mejoras son:

- En lugar de guardar el identificador del pedido en el "local storage", se sugiere establecer una sesión segura utilizando HTTPS para mayor seguridad.
- Implementar notificaciones en tiempo real mediante *Push Notifications* para que los usuarios puedan recibir actualizaciones sin tener que permanecer en la aplicación web.
- Permitir a los usuarios realizar más de un pedido, lo cual requeriría ajustar el código actual.
- Separar la lógica de cambio de estado del pedido del panel de administración y asignar esa responsabilidad a una "tercera persona".
- Evitar pedidos realizados desde ubicaciones diferentes al restaurante almacenando y verificando la ubicación exacta.
- Gestionar el pago real a través de la aplicación utilizando herramientas como Stripe, que facilitan los pagos en línea.

8. APLICACIONES SIMILARES

Actualmente existen aplicaciones similares a Easy-order. Es la aplicación Grab [4], pero es más orientada a servicio de restaurantes para aeropuertos. Pero aun así carece de algunas características comparado con Easy-order.

9. AGRADECIMIENTOS

Quisiera agradecer al profeso Guillem Godoy por su ayuda y constante seguimiento del proyecto. También me ayudó bastante a la hora de tomar las decisiones; por ejemplo, hicimos que cuando llegaran los nuevos pedidos por websocket, el pedido actual con el que el administrador interactúa no se pierda de vista. Se marca de color azul. Tal y como se puede ver en la **figura 5**.

Hora del pedido	Platos	Código entrega	Estado	Importe
2023-05-23T09:46:32.417Z	Big Mac Cheese burger Burger McDouble Burger	ES99	ORDER 1	21.25€

Figura 5 – La fila del pedido con el que interactúa se marca de color azul para no perder lo de vista.

10. CONCLUSIONES

En este trabajo hemos resuelto lo propuesto inicialmente. Solamente carece de la parte de dejar reseñas. Permitted al administrador del restaurante administrar los platos, las categorías y manejar los pedidos. Y permitimos a los clientes de restaurantes acceder a ellos, ver platos por categorías, añadir de la cesta, eliminar de la cesta, excluir ingredientes y hacer el pedido.

También hemos conseguido que el usuario sepa en tiempo real

que está pasando con el pedido, si está preparado, o todavía en proceso.

En las pruebas podemos observar que se hace un uso alto de recursos para 30 usuarios. Pero esto también se debe a que la máquina no solamente atiende peticiones de Django y de usuarios de la aplicación, sino también otros procesos como la base de datos y el Redis.

En conclusión, el producto propuesto para el proyecto cumple con los objetivos iniciales, aunque falta implementar que los usuarios de la aplicación puedan dejar reseña al restaurante, pero unas mejoras más harían que este proyecto sea más funcional y adaptado a las necesidades reales de los restaurantes.

11. BIBLIOGRAFIA

[1] Librería Django para Python
<https://www.djangoproject.com/>

[2] Frameworks más populares para desarrollo frontend
<https://www.sam-solutions.com/blog/best-frontend-framework/>
<https://www.linkedin.com/pulse/top-popular-frontend-frameworks-2023-zuraiz-khan/>

[3] Página oficial de Docker
<https://www.docker.com/>

[4] Aplicación Grab
<https://getgrab.com/>

[5] Compra de servidores cloud
<https://cloud.digitalocean.com/>

[6] Página de GoDaddy para la compra de dominios
<https://www.godaddy.com/>

[7] Haproxy
<https://www.haproxy.org/>

[8] Guía para configurar haProxy
<https://help.clouding.io/hc/es/articles/360010289000-Balancear-servicio-web-con-HAProxy-con-certificado-SSL-en-Ubuntu-18-04>

[9] Página oficial de la herramienta LetsEncrypt para generar certificados

<https://letsencrypt.org/es/>

[10] Herramienta para realizar pruebas de carga y usuario

<https://locust.io/>

[11] Herramienta de monitorización

<https://grafana.com/>