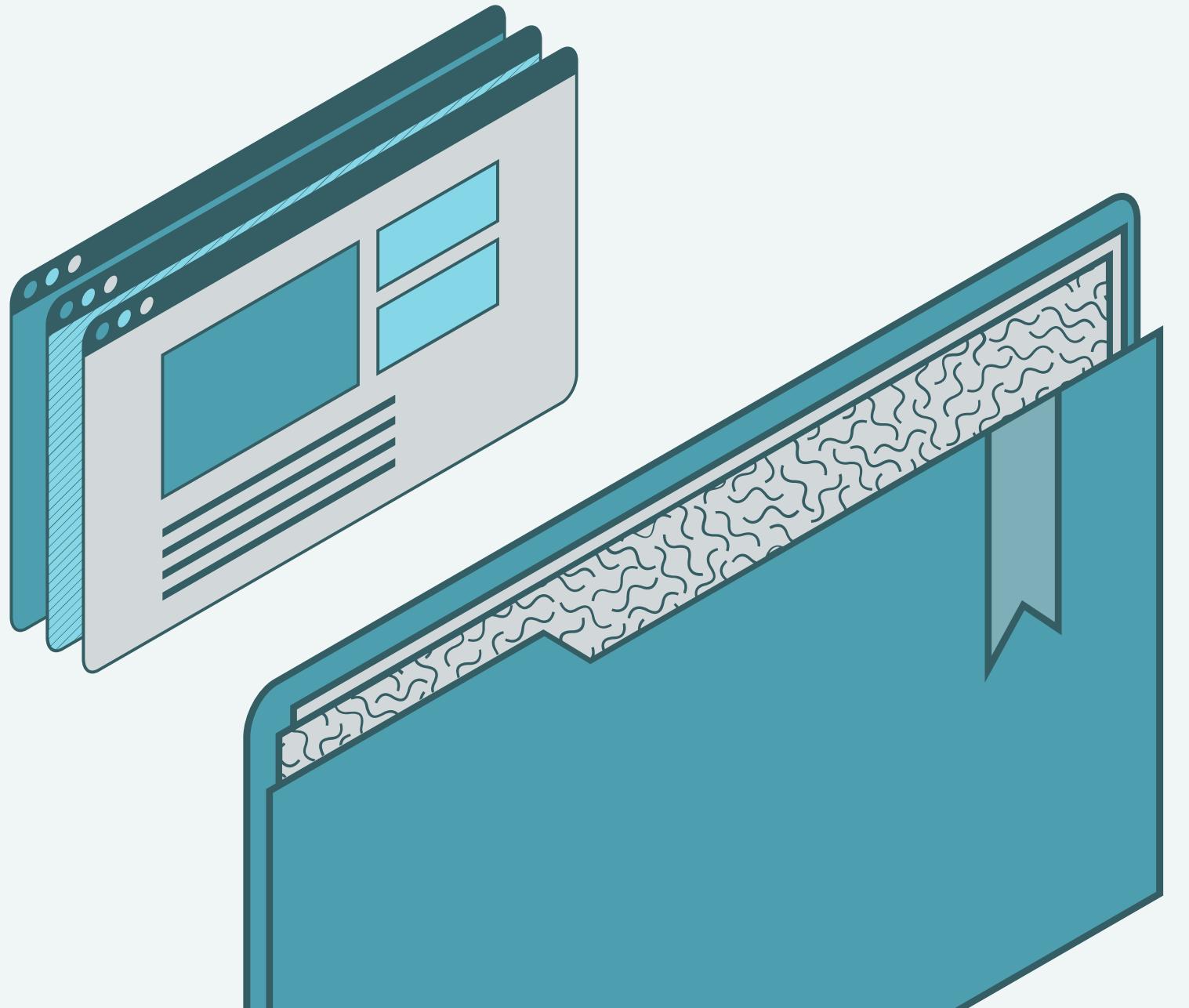


## Presentación de Daniel Peñaranda

# OrderNow

# íNDIC



---

**01. Introducción**

---

**02. Objetivos**

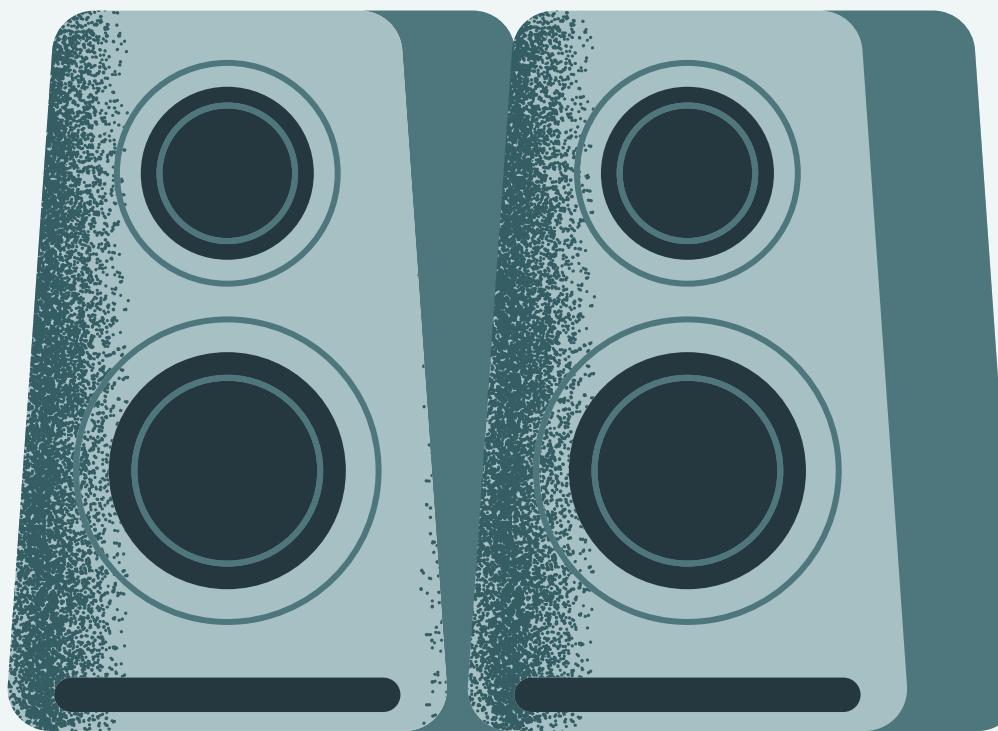
---

**03. Flujo de trabajo**

---

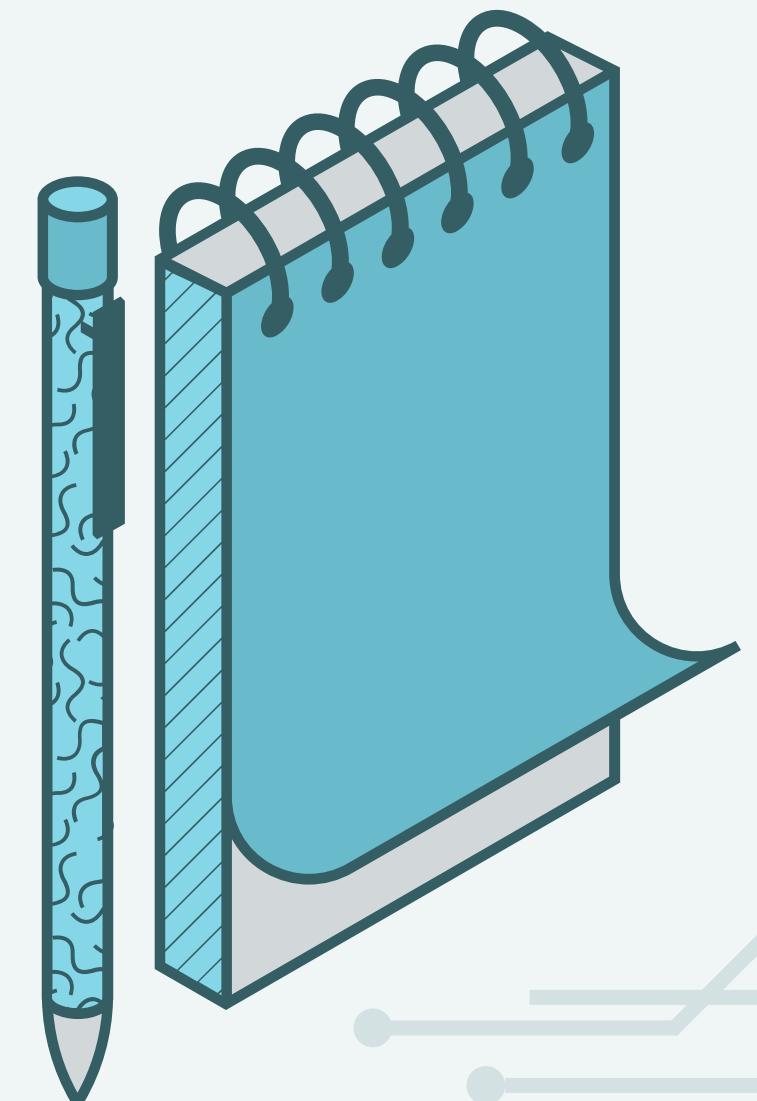
**04. Lecciones aprendidas**

---



# INTRODUCCIÓN

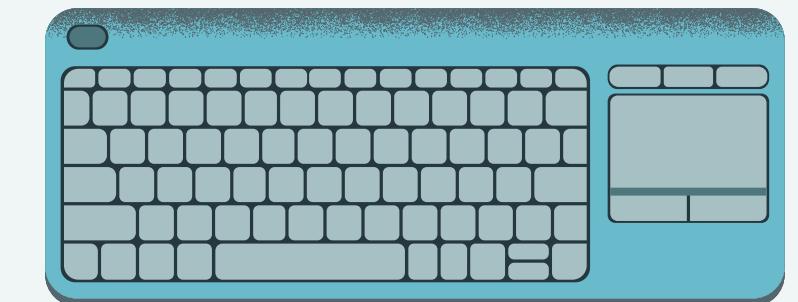
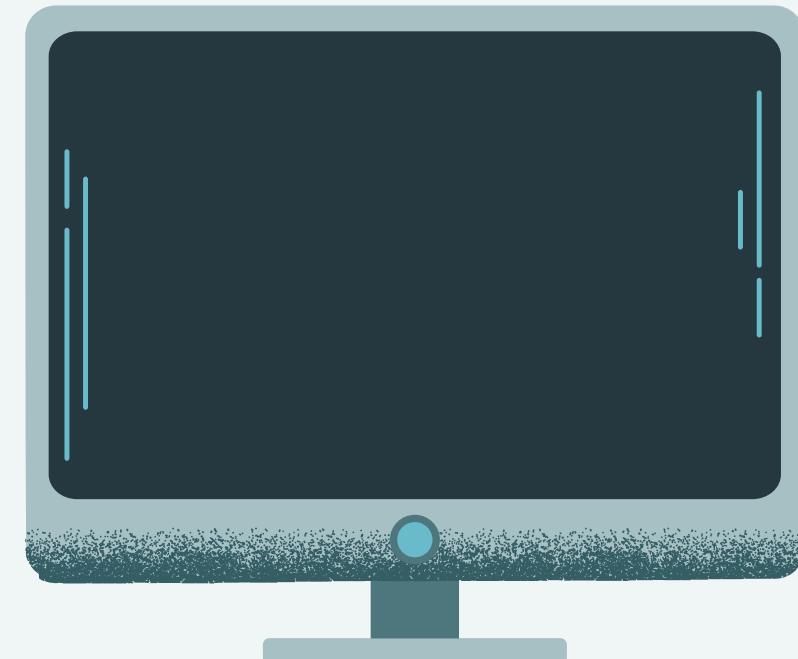
Este proyecto consiste en una página de pedidos de comida a restaurantes, consta de dos partes: El lado del cliente, donde un consumidor puede ver los restaurantes y productos disponibles y hacer pedidos cómodamente.



# OBJETIVOS

## OBJETIVO PRINCIPAL

- Establecer un nuevo servicio de pedidos de comida en la Universidad mediante dos aplicaciones en conjunto tanto para restaurantes como clientes universitarios con el fin de facilitar la reserva de almuerzos y reducir las filas de espera.



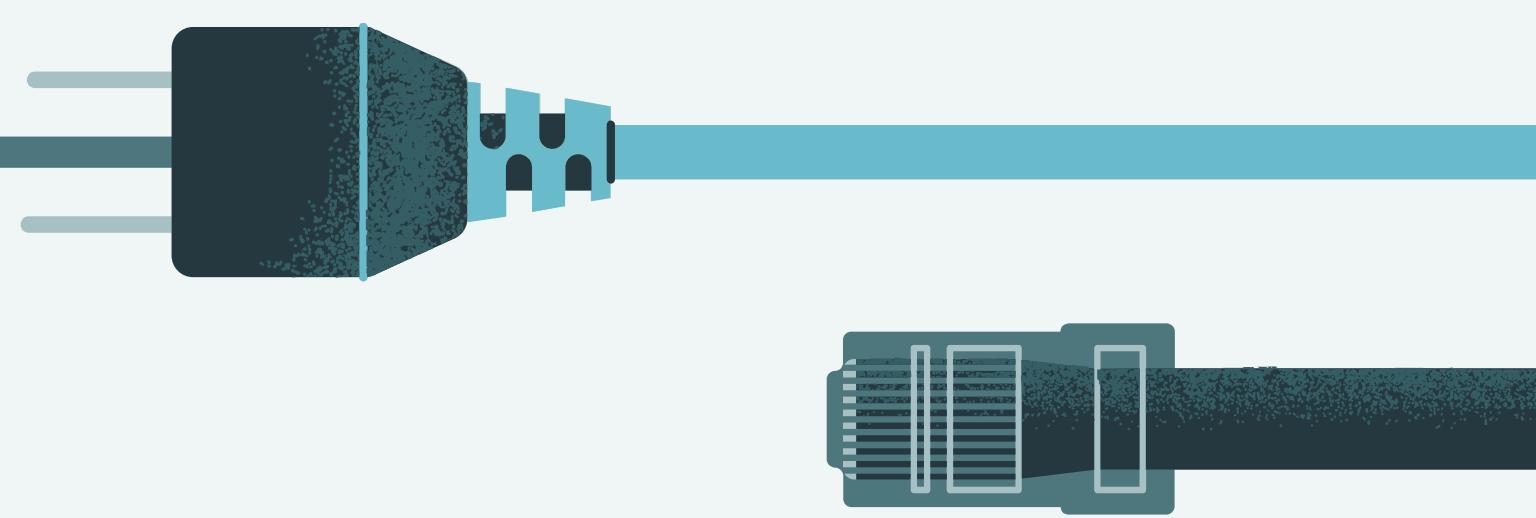
## OBJETIVOS SECUNDARIOS

- Analizar el funcionamiento de los servicios de la competencia
- Registrar las preferencias de los clientes y dueños de restaurantes objetivo
- Determinar las falencias de los restaurantes objetivo

# FLUJO DE TRABAJO

- Mi trabajo inició investigando acerca de los storybooks, elemento que ayudaría al equipo a comprobar la funcionalidad de los componentes en el frontend, por falta de tiempo, la implementación completa de storybooks fue descartada.
- Posteriormente me dediqué a trabajar en el carrito: Obtención de productos del almacenamiento del navegador, gestión de cantidad de productos en el carrito y envío del carrito a la página de Checkout.





# FLUJO DE TRABAJO

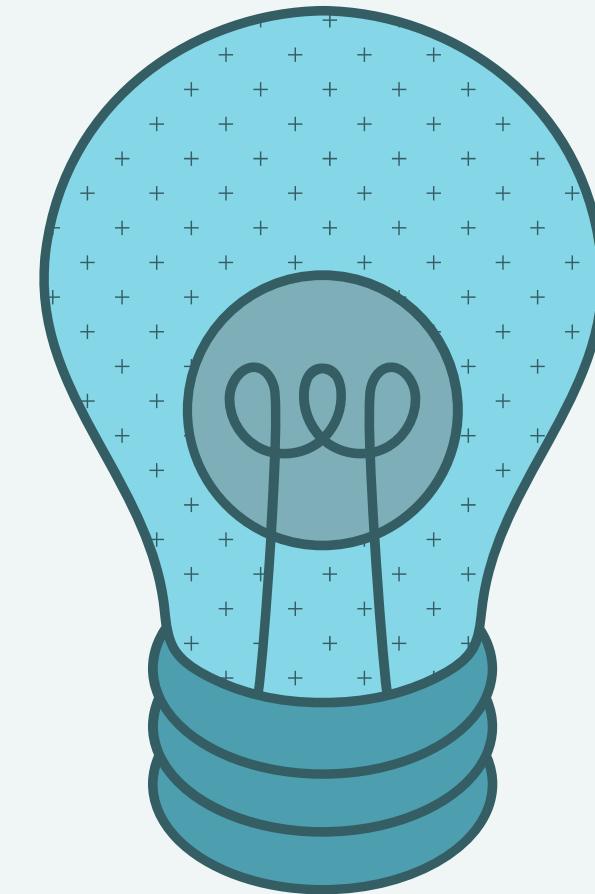
- Cada vez más cerca del final del ciclo de desarrollo, me fue asignada la tarea de obligar al usuario a iniciar sesión antes de poder avanzar con el Checkout (y envío de pedido al restaurante), además de ser encargado de QA del equipo, lo que consistía en validar que todos los PR cumplen sus criterios de aceptación y no tengan errores.
- Como último trabajo asignado, hice visible el ícono del carrito para poder acceder a él siempre que hayan productos en este.

## COMUNICACIÓN

Para evitar malentendidos, sobreposiciones con las tareas e incluso para prevenir dependencias

## ORGANIZACIÓN

Igual que la comunicación, organizar bien las tareas previene retrasos futuros con estas



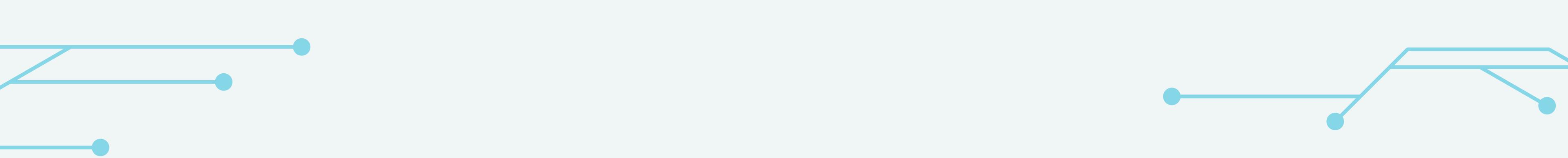
## INDEPENDENCIA

Mock-ups y saltos de pasos para poder probar flujos o componentes por cuenta propia y posteriormente unir los componentes sin haber perdido tiempo.

## DISTINGUIR HU Y ÉPICA

Atomizar HUs para mejorar la distribución de tareas y así avanzar más rápido.

# LECCIONES APRENDIDAS



[www.unsitiogenial.es](http://www.unsitiogenial.es)

# MUCHAS GRACIAS

