Foto del alumno

(foto actual)

MÓDULO PROYECTO

Ciclo Superior Desarrollo de Aplicaciones Web

Departamento: Informática y Comunicaciones

IES “María Moliner”

Curso: 2023/2024

Grupo: S2I

Proyecto: Comida En Marcha

**Fernando Juan Estrada Gallardo**

Email: fern.j.e.g@gmail.com

Tutor individual: María José González García

Tutor colectivo: Enrique Carballo Albarrán

Fecha de presentación: (fecha de presentación)

Contenido

[Descripción general del proyecto 3](#_Toc164545452)

[Objetivos 3](#_Toc164545453)

[Cuestiones metodológicas 3](#_Toc164545454)

[Entorno de trabajo 3](#_Toc164545455)

[Descripción general del producto 4](#_Toc164545456)

[Planificación y presupuesto 4](#_Toc164545457)

[Planificación 4](#_Toc164545458)

[Metodología 4](#_Toc164545459)

[Bocetos 4](#_Toc164545460)

[Presupuesto 5](#_Toc164545461)

[Documentación técnica 6](#_Toc164545462)

[Entidades 6](#_Toc164545463)

[Rutas API 6](#_Toc164545464)

[Hosting 7](#_Toc164545465)

[Servidor 9](#_Toc164545466)

[Manuales de usuario 10](#_Toc164545467)

[Instalación 10](#_Toc164545468)

[Uso 10](#_Toc164545469)

[Conclusiones y ampliaciones 11](#_Toc164545470)

[Conclusiones 11](#_Toc164545471)

[Ampliaciones 11](#_Toc164545472)

[Bibliografía 12](#_Toc164545473)

[Glosario 13](#_Toc164545474)

# Descripción general del proyecto

## Objetivos

Se pretende la realización de una aplicación web que sirva para varios establecimientos y consiga reemplazar al empleado tomando la nota por una aplicación.

Será una aplicación que funcionará por suscripción, aunque esta no estará integrada para la fase de este proyecto si no que se hará de forma manual entre las partes.

Los pedidos se relacionan con las mesas, y al finalizar la comida el resultado total del ticket estará disponible para el cobro.

Los cocineros verán en todo momento los platos que se han pedido y podrán ocultar platos en caso de que se hayan agotado.

El esquema es el siguiente:

Cliente 🡪 pide comida con pantalla 🡪 llega al cocinero en tiempo real 🡪 se hace la comida y se entrega 🡪 se puede pedir más comida o finalizar la comida 🡪 se finaliza la comida 🡪 se cobra en una caja a la salida o donde el establecimiento quiera.

Se intentará hacer lo más escalable y modular la aplicación por si en un futuro algún cliente quisiera añadirle funcionalidades o crear un software personalizado para este.

Se reservará el dominio principal para si sobrase tiempo hacer una página estática explicando el servicio, esto será solo si sobrase tiempo y funcionase todo correctamente antes de la entrega.

## Cuestiones metodológicas

## Entorno de trabajo

La herramienta de control de versiones usadas es GitHub. El enlace al repositorio es el siguiente: <https://github.com/FerZeg/TFG>

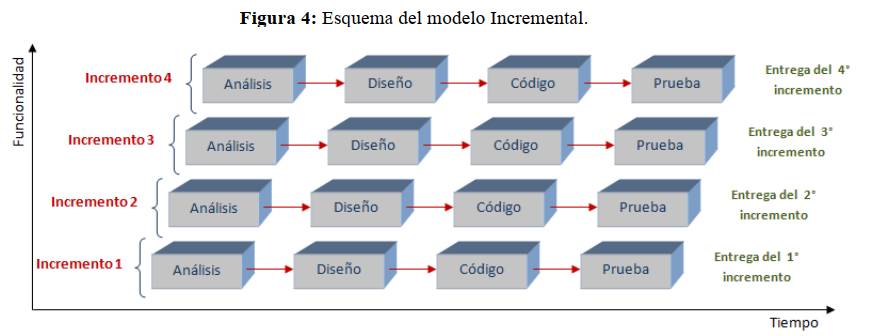
# Descripción general del producto

# Planificación y presupuesto

## Planificación

### Metodología

La metodología usada es incremental iterativa.



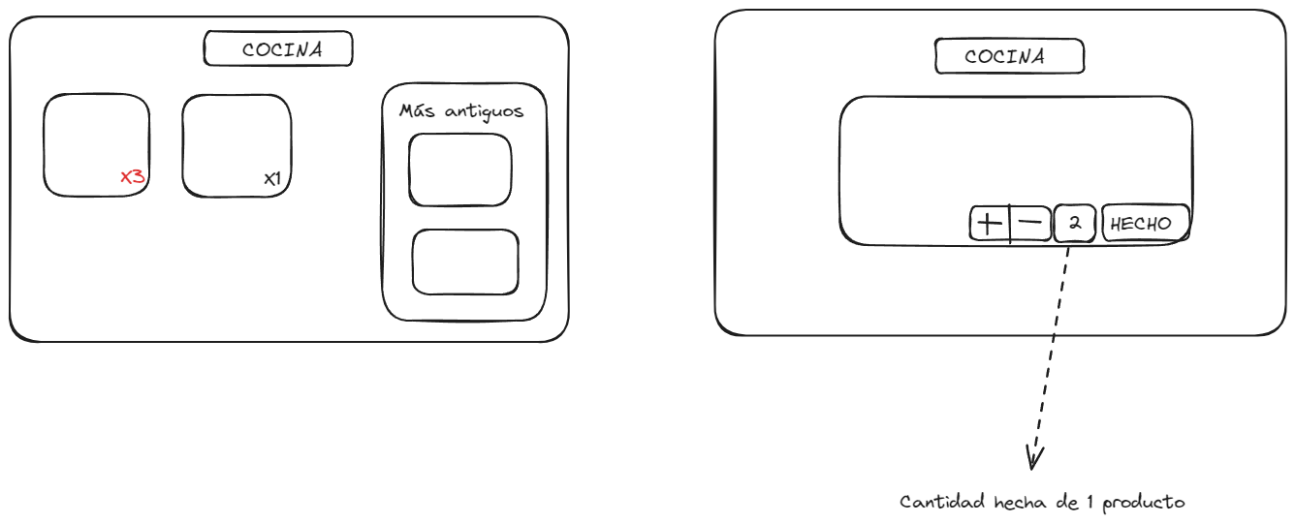
### Bocetos

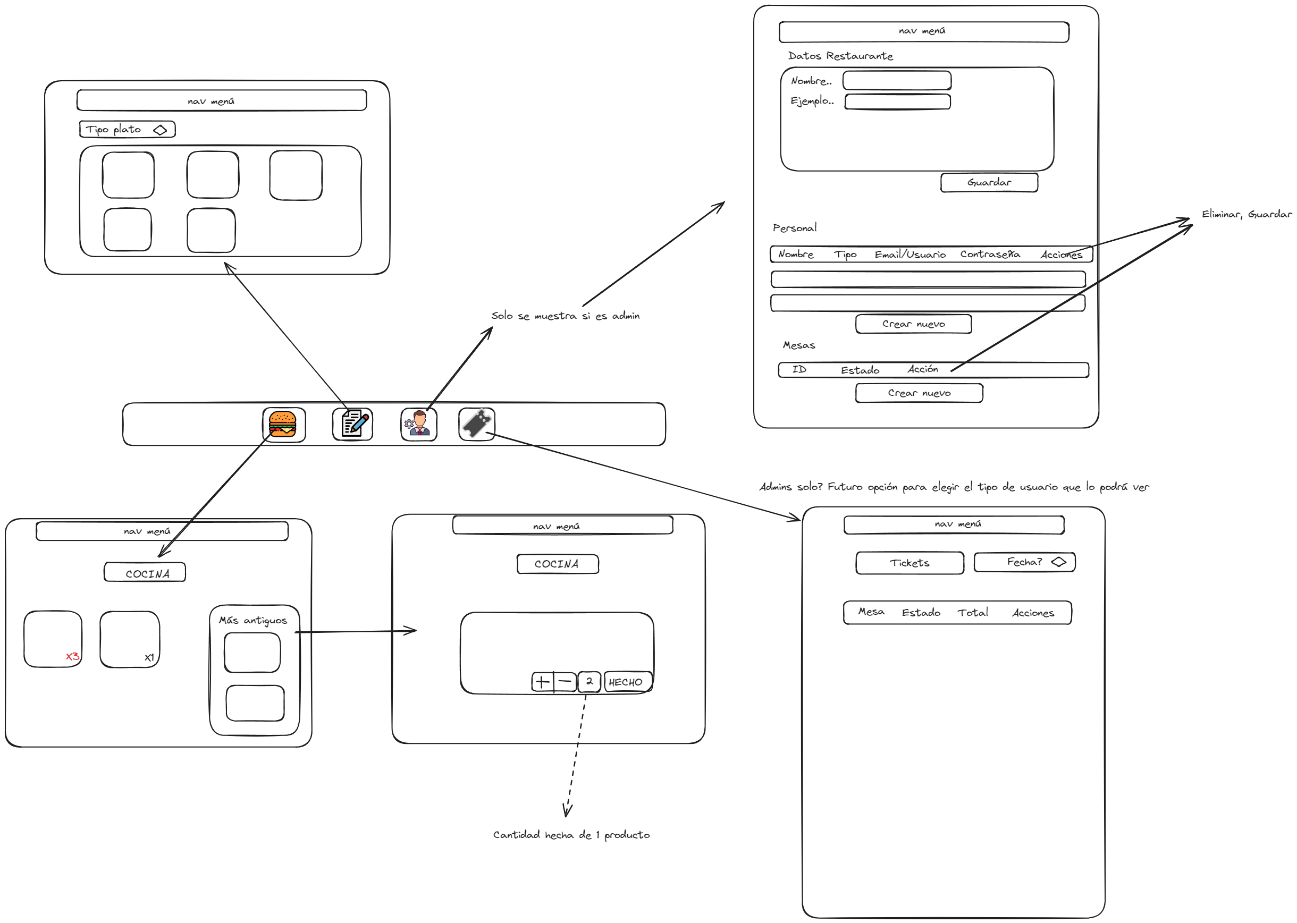
1.

Forma, Rectángulo

Descripción generada automáticamente

2.





## Presupuesto

# Documentación técnica

## Entidades

He definido las diferentes entidades existentes de la aplicación mediante un archivo de typescript para visualizar los “tipos” claramente. La base de datos en las que trabajaran estas entidades será MongoDB; una base de datos no relacional muy flexible.

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente

## Rutas API

**Restaurantes**

**POST /restaurantes**

**GET /restaurantes/:id**

**PUT /restaurantes/:id**

**DELETE /restaurantes/:id**

**Mesas**

**GET /mesas**

**POST /mesas**

**GET /mesas/:id**

**PUT /mesas/:id**

**DELETE /mesas/:id**

**Tickets**

**GET /mesas/:mesa\_id/tickets**

**POST /mesas/:mesa\_id/tickets**

**GET /mesas/:mesa\_id/tickets/:id**

**DELETE /mesas/:mesa\_id/tickets/:id**

**Pedidos**

**GET /tickets/:ticket\_id/pedidos**

**POST /tickets/:ticket\_id/pedidos**

**GET /tickets/:ticket\_id/pedidos/:id**

**PUT /tickets/:ticket\_id/pedidos/:id**

**DELETE /tickets/:ticket\_id/pedidos/:id**

**Platos**

**GET /platos**

**POST /platos**

**GET /platos/:id**

**PUT /platos/:id**

**DELETE /platos/:id**

**Cocineros**

**GET /cocineros**

**POST /cocineros**

**GET /cocineros/:id**

**PUT /cocineros/:id**

**DELETE /cocineros/:id**

**WebSocket para Cocineros**

**WS /cocineros/:id/socket**

## Hosting

Hay 3 hostings utilizados, diferenciando la parte del front(interfaces de usuario), la del servidor y la de la base de datos.

* Para las diferentes interfaces se usa Cloudflare.
* Para el servidor se usará Northflank.
* Para la base de datos se usará Mongo Atlas.

Ambos son de uso gratuito, siendo Cloudflare un *CDN (Content Delivery Network)* bastante popular y ampliamente usado por muchas aplicaciones. Al ser el frontend una compilación en archivos estáticos nos podemos aprovechar de su red mundial para servir los archivos con muy baja latencia en cualquier parte del mundo.

Northflank es una plataforma Cloud relativamente nueva, que trae muchas funcionalidades para facilitar el despliegue de la aplicación. Con un dockerfile sencillo defino el entorno de ejecución de mi servidor.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Captura de pantalla de un celular

Descripción generada automáticamente

Entre otras funcionalidades este servicio ofrece:

* Panel de control donde ver los logs de la aplicación
* Diferentes tipos de métricas
* Una consola de la instancia que se está ejecutando
* Sistema para exponer puertos diferentes según las necesidades
* Herramienta para configurar dominios customizados, certificados de SSL automáticos al conectar el domino.
* Ejecutar código cada cierto tiempo (Cron Jobs) o manualmente.
* Bases de datos.
* Usuarios y equipos.
* Notificaciones.

Todo esto con una capa gratuita bastante generosa para probar proyectos sin un tráfico excesivo.

Captura de pantalla de computadora

Descripción generada automáticamente

Ambas plataformas ofrecen un servicio de *CD (Continuos Deployment),* lo que facilita muchísimo el despliegue de la aplicación al ser automatizado al hacer un push a la rama Main del repositorio.

## Servidor

Librerías usadas principales del proyecto:

* Express: Framework de desarrollo para facilitar la elaboración del servidor, consta de un sistema avanzado de routing y librerías para parsear las cookies u otros.
* Mongoose: Interfaz para la base de datos mongo, permite crear Schemas (Estructuras de objetos) y muchas funciones añadidas.
* Bcrypt para encriptar las contraseñas.

# Manuales de usuario

## Instalación

## Uso

# Conclusiones y ampliaciones

## Conclusiones

## Ampliaciones

# Bibliografía

Modelar los datos con MongoDB:

* <https://www.youtube.com/watch?v=YsaOcUDUJKY>
* <https://www.youtube.com/watch?v=Hidk36H6hBY>

Información sobre metodología incremental:

* <https://mwebs.com.uy/blog/qu%C3%A9-es-el-desarrollo-incremental/23>

# Glosario

* **CDN (Content Delivery Network):** Red de servidores interconectados que contienen copias locales de contenidos para distribuirlos rápidamente desde el nodo más cercano al usuario.
* **CD (Continuos Deployment):** Es una estrategia por la cual se publican automáticamente los cambios hechos a producción (producto final).