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](#_Toc161905763)

[Objetivos 3](#_Toc161905764)

[Cuestiones metodológicas 3](#_Toc161905765)

[Entorno de trabajo 3](#_Toc161905766)

[Descripción general del producto 4](#_Toc161905767)

[Planificación y presupuesto 4](#_Toc161905768)

[Planificación 4](#_Toc161905769)

[Metodología 4](#_Toc161905770)

[Bocetos 4](#_Toc161905771)

[Presupuesto 5](#_Toc161905772)

[Documentación técnica 6](#_Toc161905773)

[Entidades 6](#_Toc161905774)

[Rutas API 6](#_Toc161905775)

[Hosting 7](#_Toc161905776)

[Servidor 7](#_Toc161905777)

[Manuales de usuario 9](#_Toc161905778)

[Instalación 9](#_Toc161905779)

[Uso 9](#_Toc161905780)

[Conclusiones y ampliaciones 10](#_Toc161905781)

[Conclusiones 10](#_Toc161905782)

[Ampliaciones 10](#_Toc161905783)

[Bibliografía 11](#_Toc161905784)

[Glosario 12](#_Toc161905785)

# 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 ágil Kanban, es decir, divido las tareas en columnas; por hacer, en proceso y hechas. Hay un flujo constante de tareas pendiente y se van haciendo una a una.

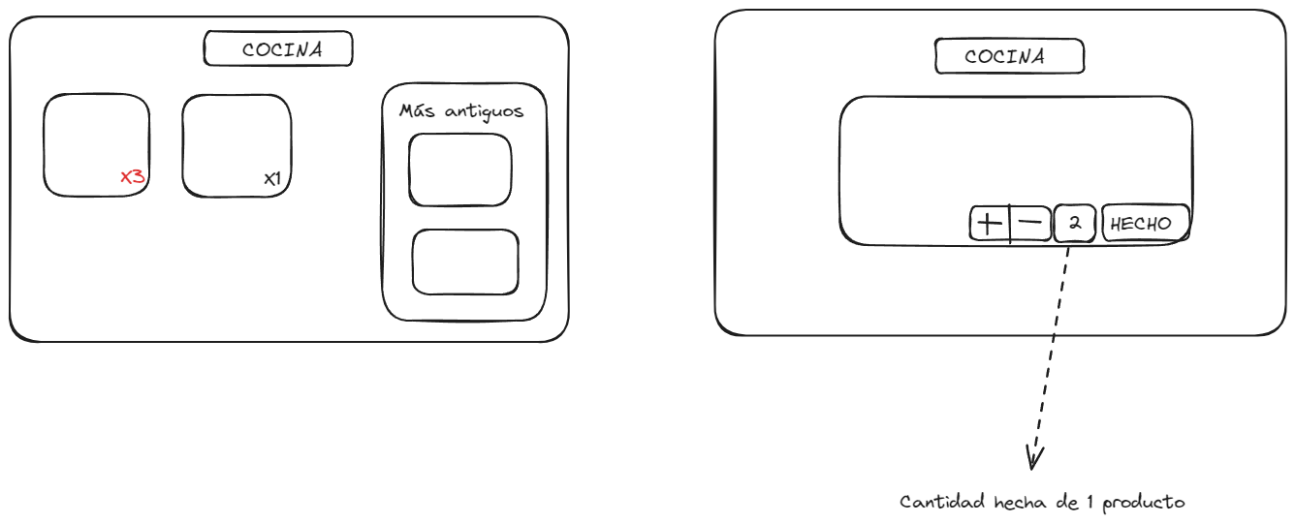
### Bocetos

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 2 hostings utilizados, diferenciando la parte del front(interfaces de usuario) y la del servidor.

* Para las diferentes interfaces se usa Vercel
* Para el servidor se usará Fly.io o AWS

## 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>

# Glosario