**Centro educativo**

|  |  |  |  |
| --- | --- | --- | --- |
| **Código** | **Centro** | **Concello** | **Ano académico** |
| 15005397 | I.E.S. Fernando Wirtz Suárez | A Coruña | 2022-2023 |

**Ciclo formativo**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Código da familia profesional** | **Familia profesional** | **Código do ciclo formativo** | **Ciclo formativo** | **Grao** | **Réxime** |
| FP16 | Informática e comunicacións | CSIFC01 | Desenvolvemento de Aplicacións Web | Superior | Adultos |

**Módulo profesional e unidades formativas de menor duración (\*)**

|  |  |
| --- | --- |
| **Código**  **MP/UF** | **Nome** |
| MP0374 | Proxecto de Desenvolvemento de Aplicacións Web  Equivalencia en créditos ECTS: 5. |

**Profesorado responsable**

|  |  |
| --- | --- |
| **Tutor** | Rafael Ferreiro López |

**Alumno**

|  |  |
| --- | --- |
| **Alumno** | Iván García García |

**Datos do Proxecto**

|  |  |
| --- | --- |
| **Título** | *GGBowling* |

**CONTROL DE VERSIÓNS:** [**https://github.com/ivangarciagarcia/GGBowling**](https://github.com/ivangarciagarcia/GGBowling)

|  |  |  |
| --- | --- | --- |
| **Versión** | **Data** | **Observacións** |
| **0.0.1 Snapshot** | **01/02/23** | **Inicio del proyecto** |
| **0.0.2** | **01/03/23** | **Investigacion completa, toma de requisitos tomada y estructura del proyecto diseñada** |
| **0.1** | **22/03/23** | **Base de datos completa, proyecto base de backend listo y base de interfaz grafica completada** |
| **1.0.0** | **31/03/23** | **Primera entrega de seguimiento del proyecto** |
| **1.0.1** | **12/04/23** | **Correcciones del back-end, correcciones de la interfaz grafica y desarrollo de la logica de front.** |
| **1.1** | **01/05/23** | **Segunda entrega de seguimiento del proyecto** |
| **1.1.1** | **03/05/23** | **Integracion de front-end y back-end, pruebas de integracion y corecciones en la integracion** |
| **1.1.2** | **20/05/23** | **Implementacion de restriciones en el front, implementacion de excepciones en el back y correccion de errores y pruebas adicionales** |
| **1.1.3** | **29/05/23** | **Implementacion de SpringSecurity, features menores, correcciones finales y documentacion.** |
| **1.2** | **30/05/23** | **Entrega final proyecto** |

**Índice:**

[1. Objetivo 3](#__RefHeading___Toc992_3466438741)

[2. Descripción 3](#__RefHeading___Toc994_3466438741)

[3. Alcance 3](#__RefHeading___Toc996_3466438741)

[4. Planificación 3](#__RefHeading___Toc998_3466438741)

[5. Medios a utilizar 4](#__RefHeading___Toc1000_3466438741)

[6. Presupuesto 4](#__RefHeading___Toc1002_3466438741)

[7. Título 4](#__RefHeading___Toc1004_3466438741)

[8. Trabajo en grupo 4](#__RefHeading___Toc1006_3466438741)

[9. Ejecución 4](#__RefHeading___Toc1008_3466438741)

# Objetivo

El objetivo principal del proyecto es tener una web que gestione las reservas (tanto de pistas para jugar o de mesas para ir al restaurante) de los usuarios que quieran ir al local.

Ademas, otros objetivos generales del sitio web son:  
  
- Aumentar el conocimiento de la marca. Con esto me refiero a que el sitio web de la bolera es una parte clave en la estrategia de marketing, debería ayudar a aumentar el conocimiento de la bolera y atraer nuevos clientes.

- Proporcionar servicio al cliente. Este objetivo se centra en proporcionar información al cliente sobre la bolera y el restaurante como el horario, las normas, información de contacto, dirección, etc

También hay algunos objetivos mas específicos como:

- El sitio web sea compatible con dispositivos móviles.

- Navegación sencilla e intuitiva del mismo.

- Mantener el sitio web actualizado.

# Descripción

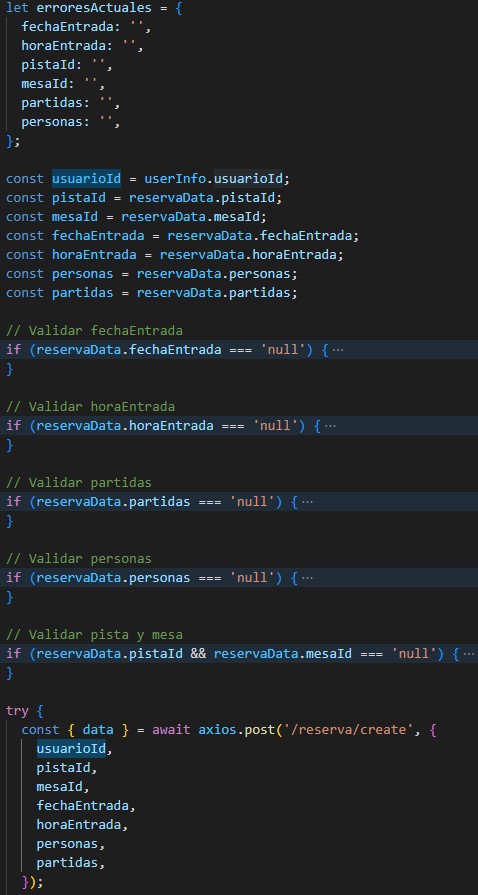
El proyecto de GGBowling está hecho principalmente para poder gestionar las reservas de una Bolera-Restaurante de una manera automática. Aunque también tiene otros objetivos que se describen en el anterior punto.

El proyecto está pensado para que una parte pueda funcionar sin la otra (La bolera sin el restaurante y viceversa) lo que ofrece mucha adaptabilidad para distintos tipos de negocios.

La creación de este proyecto nace por la necesidad de pequeñas empresas en tener una página web la cual pueda captar nuevos clientes y gestionar reservas mediante un sistema de usuarios. Como he dicho anteriormente, la pagina funciona sin una de las 2 partes lo que permite adaptarse tanto a un restaurante como a una bolera o a las 2 combinadas, lo que da muchas mas posibilidades en el mercado.

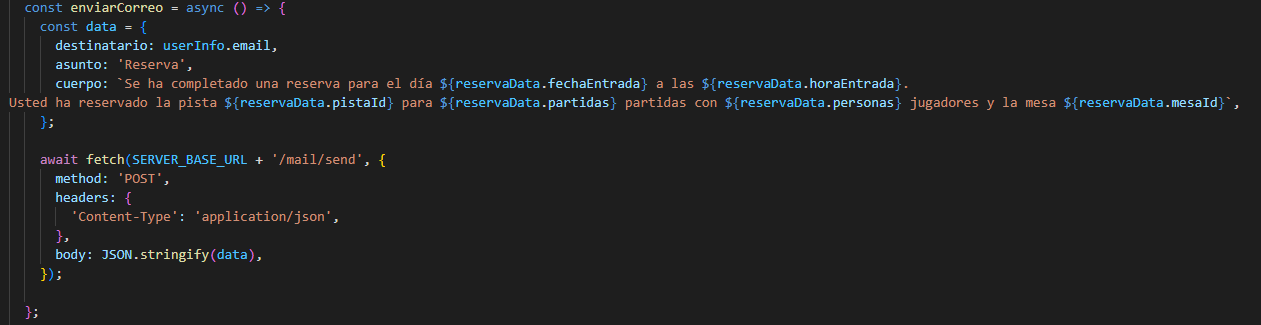
La parte mas grande del proyecto es la lógica de reservas y todo lo que eso conlleva así que en eso es en lo que mas me voy a centrar.

Uno de los problemas con los que me encontré fue que la reserva se hiciera desde el usuario que está logueado sin la necesidad de introducir el id del usuario manualmente. Para solucionar este problema hice un estado global de redux el cual lo reutilizo para otras partes del proyecto. Ese estado guarda toda la información del usuario que se loguea. Al realizar la reserva, el id se pasa desde ese estado a la url de la petición directamente



Otro de los problemas fué como mostrar esa reserva cuando el usuario la realiza. Mi primera idea fue mostrarla en el perfil pero al final me decanté por mandar un email con los datos de la reserva para asi mostrarla en recepción para así demostrar que tienes reservado.

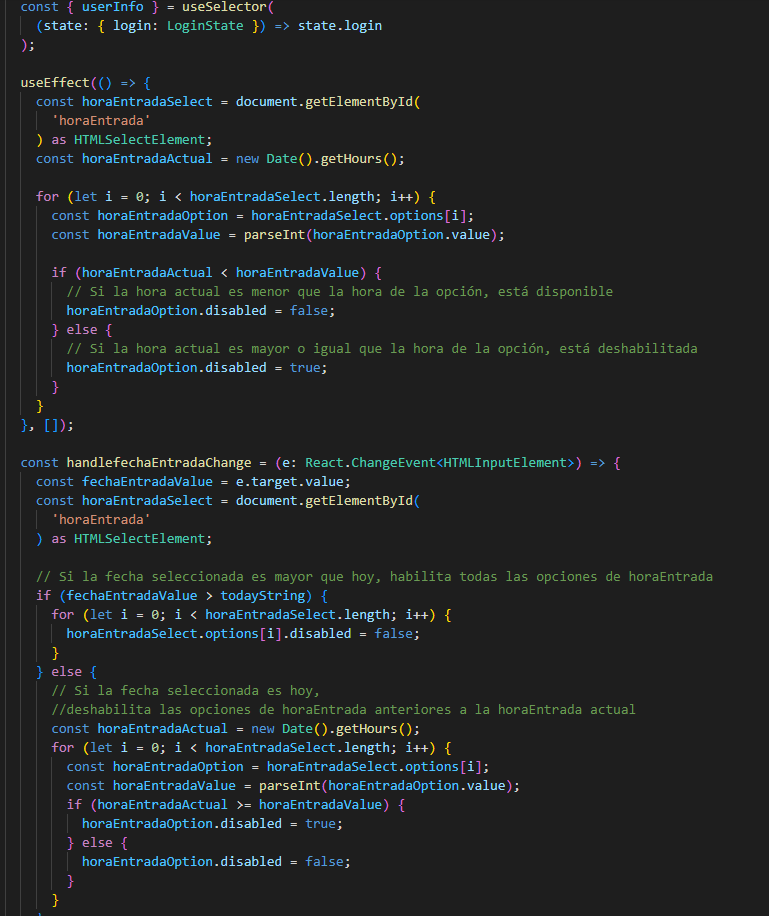
Lógica en el front:



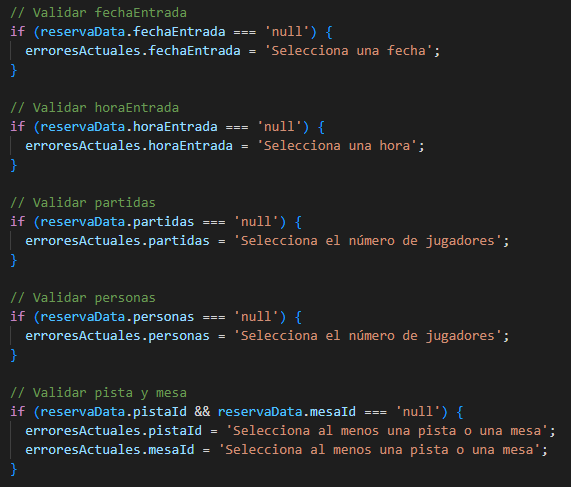


Lógica en el back:

De esta forma los problemas de la reserva están solucionados pero me surgieron otros como todas las restricciones que debería de hacer para que la reserva se efectúe correctamente y que no choque con otra reserva ya registrada.



Aquí reviso los posibles nulos



Ahora si que si está la reserva hecha.

La segunda parte con mas complicaciones fué el perfil pero al implementar el estado global que comente anteriormente se me simplificó bastante. El único problema mayor fue el hecho de editar algunos campos del usuario porque tenia que ponerlos en el mismo formato que los anteriores y por ejemplo para la encriptación de la contraseña me costo un poco.

Y el ultimo error que me llevó varios quebraderos de cabeza fue la política de CORS. Al principio no me dio muchos problemas pero al implementar Spring Security tuve que dedicarle bastante tiempo a esa tarea que en un principio no tenia prevista.

# Alcance

El alcance del proyecto es poder gestionar una bolera-restaurante mediante las reservas de los usuarios y la capacidad de poder atraer nuevos clientes. Debe de tener:

- Sistema de registro e identificación de usuarios

- Apartado de perfil de usuario

- Modificación de usuarios

- Creación de reservas mediante interfaz grafica

- Navegación amigable e intuitiva

- Paginas de información (Como ofertas, precios, horarios, etc)

- Adaptabilidad para móviles

Funcionalidades que quedaron fuera del proyecto por tiempo:

- Vista de administrador

- Pasarela de pago

- Edición y eliminación de reserva desde la vista de usuario

- Página de feedback

Las funcionalidades no implementadas no afectan al correcto funcionamiento de la página, hay apartados que si están operativos solo que no tienen una interfaz gráfica.

# Planificación

El proyecto lo inicia el 01/02/23 y se entregará el 30/05/23. Para ello planifiqué 6 fases del proyecto que se dividen en una fase inicial y 5 sprints.

- Fase inicial: Donde se investigará sobre que tecnologías y herramientas utilizar y como utilizarlas, se hará la toma de requisitos y se diseñara la base sobre la que construir el proyecto. Desde 01/02/23 hasta 1/03/23

- Sprint 1: En el cual me dediqué a desarrollar la base sobre la que construir el proyecto tanto de back como de la base de datos. También diseñé la interfaz grafica de algunas paginas.

-Sprint 2: Muchas correcciones tanto de backend como de la interfaz y además empecé con la lógica del frontend.

-Sprint 3: Lo dedique exclusivamente a la integración del front y el back. Tanto las pruebas de integración como la integración misma como las correcciones.

-Sprint 4: Comencé a implementar las restricciones en el front y las excepciones en el back. Además hice algunas pruebas adicionales para testearlas y correcciones

-Sprint 5: Implemente Spring Security, hice features menores para la comodidad del usuario, terminé las ultimas correcciones del proyecto y acabé con la documentación del mismo.

# Medios a utilizar

Los medios utilizados para este proyecto fueron:

**RECURSOS MATERIALES:**

**Hardware:**

**Asus TUF Gaming F15 FX506HCB-HN200 Intel Core i5-11400H/16GB/512GB SSD/RTX 3050/15.6"**

Procesador Intel Core i5-11400H (6C/HexaCore 4.5GHz, 12MB)

Memoria RAM 16GB SO-DIMM DDR4 3200MHz

Disco duro 512GB SSD M.2 NVMe PCIe

Display 15.6" Full HD 1920 x 1080 pixeles, 144Hz, IPS

**Software:**

Diseño: Figma

Base de datos: MySQL, Dbeaver

Backend: SpringBoot (Java), IntelliJ

Frontend: React (TypeScript, JavaScript, HTML, SASS), Visual Studio Code

# Presupuesto

El calculo del costo de la aplicación lo hice en base a lo que gana un programador full stack promedio en España. La cifra es de 18€ la hora.  
  
He trabajado al rededor de 300 horas en el proyecto. Si multiplicamos los 2 valores, los costes fijos por el desarrollo de la pagina da un total de 5400€.  
  
Si el cliente quiere un extra por un año de mantenimiento los costes serian:

Hosting + Dominio + Seguridad: 49,99€/año (el primer año, luego sube a 174,49€)

Mantenimiento de la web + Servicio al cliente personalizado: 100€/año

# Título

GGBowling

# Trabajo en grupo

Realizado de forma individual por Iván García García

# Ejecución

Ejecución en directo el jueves 1. Video de demostración en la carpeta plantillas