



I.E.S. LAGUNA DE JOATZEL

TRABAJO FIN DE CICLO

Euskal Etxea

CICLO FORMATIVO DE GRADO SUPERIOR

DESARROLLO DE APLICACIONES WEB

Nombre del Alumno/a:

Pablo Bermejo Bados
Daniel Galeano Freijo

Tutor:

Alain Fernández Fernández

Abstract y palabras clave

La web planteada en este proyecto contiene una aplicación diseñada para que los clientes del restaurante/txoko “Euskal Etxea” puedan realizar sus reservas online y para que los administradores puedan agilizar el proceso de gestión total de usuarios y reservas. La aplicación está desarrollada completamente en PHP y desplegada en un servidor local que hemos conectado al dominio tfgpolydaniel.site.

The website proposed in this project contains an application designed for "Euskal Etxea" restaurant/txoko customers being able to make their online reservations as for administrators to streamline the overall user and reservation management process. The application is fully developed in PHP and deployed on a local server that we have connected to the domain tfgpolydaniel.site.

Palabras clave: CRUD, PHP, servidor web, gestor BBDD, mySQL, github, vsCode, Trello, hosting web, DNS.

Keywords: CRUD, PHP, web server, DB manager, mySQL, github, vsCode, Trello, hosting web, DNS.

ÍNDICE

SECCIÓN I. INTRODUCCIÓN	4
1. Presentación	4
2. Contexto	4
3. Objetivos	4
 SECCIÓN II. DESARROLLO	5
4. Temporalización y ciclo de vida	5
5. Trabajo de Análisis	5
5.1. Descripción general	6
5.2. Casos de uso	6
6. Trabajo de Diseño	7
6.1. Modelo de datos	7
6.2. Diseño de módulos/ficheros de código...	8
6.3. Diagrama de navegación	9
7. Trabajo de Implementación	10
8. Trabajo de Pruebas	10
9. Trabajo de Implantación	11-12

SECCIÓN III. PRODUCTO	13
10. Descripción externa	13-16
11. Infraestructura	16
12. Mantenimiento	16
<hr/>	
SECCIÓN IV. CONCLUSIÓN	17
13. Grado de consecución de objetivos	17
14. Posible evolución futura del proyecto	17
15. Lecciones aprendidas	17-18
16. Bibliografía y webgrafía	18

SECCIÓN I. INTRODUCCIÓN

1. Presentación

Euskal Etxea es una aplicación web que permite a sus usuarios acceder y gestionar los diferentes servicios que se ofrecen a la hora de concertar una reserva en un negocio dedicado a la hostelería. Así mismo, también permite a los propios empleados gestionar las reservas y perfiles de los clientes, ramificándose en dos secciones de la aplicación diferenciadas, dependiendo del rol del usuario que desee acceder.

2. Contexto

La idea de Euskal Etxea surge de un proyecto relacionado a un negocio real homónimo, que quería implementar una gestión de reservas dentro de su página web ya existente, tras una serie de problemas de comunicación decidimos desvincularnos del proyecto y seguir trabajando por nuestra cuenta, creando algo sumamente mejor de lo que nos habían propuesto, pero conservando gran parte de los requisitos originales.

3. Objetivos

- Creación de un repositorio de GitHub para trabajar de forma paralela.
- Crear y conectar la página web a una base de datos MySQL.
- Tener la capacidad de registrar usuarios, con los que iniciar sesión en la aplicación.
- Implementar cookies y sesiones para los usuarios.
- Formar dos tipos de usuarios separados en roles de usuario y administrador.
- Dentro del usuario crear un CRUD que gestione las reservas relacionadas a ese usuario propio.
- Dentro del admin poder gestionar a través de CRUDs, tanto las reservas de todos los usuarios, como los perfiles de este rol.
- Implementar el proyecto en un servidor web con un dominio propio, desplegado desde un ordenador local.

SECCIÓN II. DESARROLLO

4. Temporalización y ciclo de vida

Semanas	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Lluvia de ideas	✓	✓																		
Análisis de requisitos			✓	✓																
Diagrama de clases					✓	✓														
Modelado de BBDD y aplicación							✓	✓	✓	✓										
Despliegue del servidor										✓	✓									
Desarrollo de la aplicación												✓	✓	✓	✓	✓	✓	✓	✓	✓
Pruebas																	✓	✓	✓	
Documentación																	✓	✓		

5. Trabajo de Análisis

Todo buen desarrollo debe comenzar con un buen análisis de los requisitos necesarios para el adecuado funcionamiento de la aplicación y de los pasos a seguir a lo largo de este, eligiendo las tecnologías más eficientes y creando de antemano un mapa conceptual de lo que sería la futura aplicación. Por eso nosotros decidimos utilizar una metodología en cascada, de tal forma que el inicio de la siguiente etapa debe esperar a la finalización de la anterior. Comenzamos con el modelado de la base de datos, adaptándonos a lo que ya existía y exigía el cliente. Una vez terminado, creamos la aplicación en su totalidad, no hacía falta que estuviera vistoso y bonito, pero si había que darle forma, sentando las bases de lo que sería. A continuación, decidimos realizar la conexión entre la base de datos y los comienzos de la aplicación, tras eso nos dispondríamos a codificar todos y cada uno los elementos de la aplicación que le otorgarían la funcionalidad y por último solo quedaría el tramo final pero no menos importante, la parte estética con la elección de colores, elementos gráficos y distribución estos a lo largo y ancho de toda la aplicación, al terminar con el modelado realizaremos las pruebas para comprobar todas las posibles inconvenientes que pueden llevar al crasheo de la aplicación, forzando los límites de esta y comprobando que todos los manejos de errores funcionan correctamente.

5.1. Descripción general

El mayor inconveniente que puede encontrar un cliente habitual de la hostelería, es llegar a su local favorito y encontrarlo totalmente abarrotado, teniendo que sufrir grandes tiempos de espera para poder probar su comida favorita. Para solucionar este problema existen las reservas, pero gracias a las tecnologías actuales estas reservas se pueden realizar con tan solo un click, esa es la finalidad de nuestra aplicación, que el restaurante tenga controlados a sus clientes, pudiendo inclusive saber cuales son clientes habituales, generando un plan de negocio y que estos clientes a su vez puedan efectuar reservas sin la necesidad de realizar ningún esfuerzo, desde cualquier lugar y en cualquier momento, sin que sea necesario ajustarse a los horarios del restaurante.

5.2. Casos de uso

La aplicación está dividida en dos secciones claramente diferenciadas, que no interactúan entre ellas. Dentro de la zona de usuarios encontramos los siguientes casos de uso.

- Registro: para acceder al resto de la aplicación es necesario que tu usuario haya sido registrado con anterioridad.
- Autentificación: una vez el usuario está creado, tiene que iniciar sesión para continuar.
- Creación de reserva: el usuario podrá en cualquier momento crear una nueva reserva, siempre y cuando no solape alguna reserva ya creada y se mantenga en los horarios predeterminados.
- Gestión de reservas: el usuario podrá ver sus reservas pendientes y podrá tanto editarlas como cancelarla si le place.
- Perfil: El usuario podrá ver su perfil y si cuenta con su contraseña podrá editar ciertos valores de este.

Por otro lado los usuarios de rol administrador serán redirigidos automáticamente a su zona de la aplicación, contando con los siguientes casos de uso:

- Gestión de usuarios: creación de usuarios nuevos siempre y cuando sean del rol user, además de la edición y eliminación de estos mismos.
- Gestión de reservas: un usuario del perfil administrador podrá gestionar las reservas de cualquiera de los usuarios que hayan utilizado la aplicación.
- Crear reservas: se realizará la creación de una nueva reserva de un usuario ya existente.

6. Trabajo de Diseño

6.1. Modelo de datos

Modelo entidad-relación de las tablas existentes:

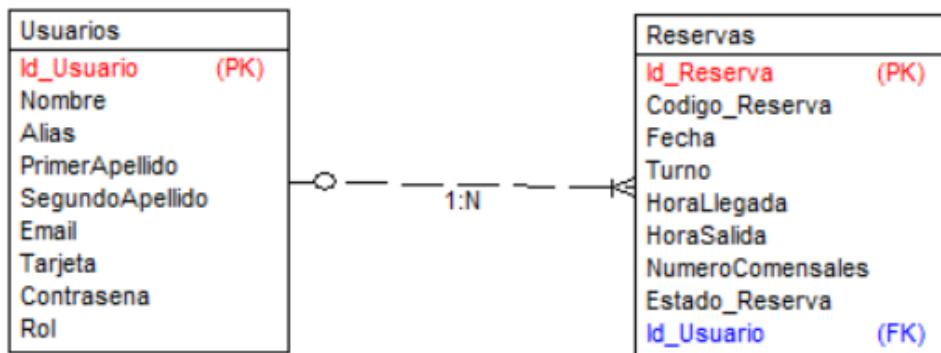


Tabla de Usuarios:

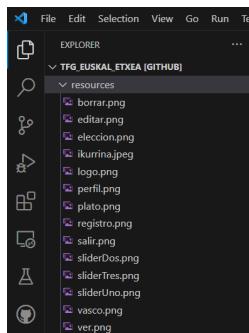
#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
□ 1	Id_Usuario 🔑	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar ▾ Más
□ 2	Nombre	varchar(20)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar ▾ Más
□ 3	Alias	varchar(15)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar ▾ Más
□ 4	PrimerApellido	varchar(20)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar ▾ Más
□ 5	SegundoApellido	varchar(20)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar ▾ Más
□ 6	Email	varchar(50)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar ▾ Más
□ 7	Tarjeta	varchar(16)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar ▾ Más
□ 8	Contrasena	varchar(30)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar ▾ Más
□ 9	Rol	varchar(5)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar ▾ Más

Tabla de Reservas:

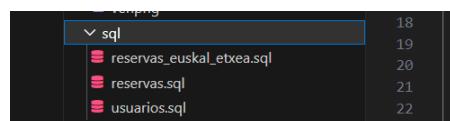
#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
□ 1	Id_Reserva 🔑	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar ▾ Más
□ 2	Id_Usuario 🔗	int(11)			No	Ninguna			Cambiar Eliminar ▾ Más
□ 3	CódigoReserva	int(11)			No	Ninguna			Cambiar Eliminar ▾ Más
□ 4	Fecha	date			No	Ninguna			Cambiar Eliminar ▾ Más
□ 5	Turno	varchar(6)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar ▾ Más
□ 6	HoraEntrada	time			No	Ninguna			Cambiar Eliminar ▾ Más
□ 7	HoraSalida	time			No	Ninguna			Cambiar Eliminar ▾ Más
□ 8	NumeroComensales	int(11)			No	Ninguna			Cambiar Eliminar ▾ Más
□ 9	EstadoReserva	tinyint(1)			No	Ninguna	Provisional o Definitivo		Cambiar Eliminar ▾ Más

6.2. Diseño de módulos/ficheros de código...

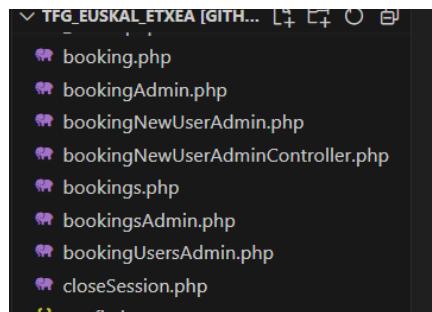
La estructura de los ficheros se divide en los recursos, son las imágenes que hemos utilizado a lo largo del proyecto:



Los ficheros de sql que contienen todo lo referente a la base de datos:



Los phps que se dividen en vistas y controladores siendo estos últimos los llamados como tal:



Además contamos con un script para alguna función de javascript, el json que sirve para la conexión a la base de datos y el fichero estilos que almacena todo el css del código:

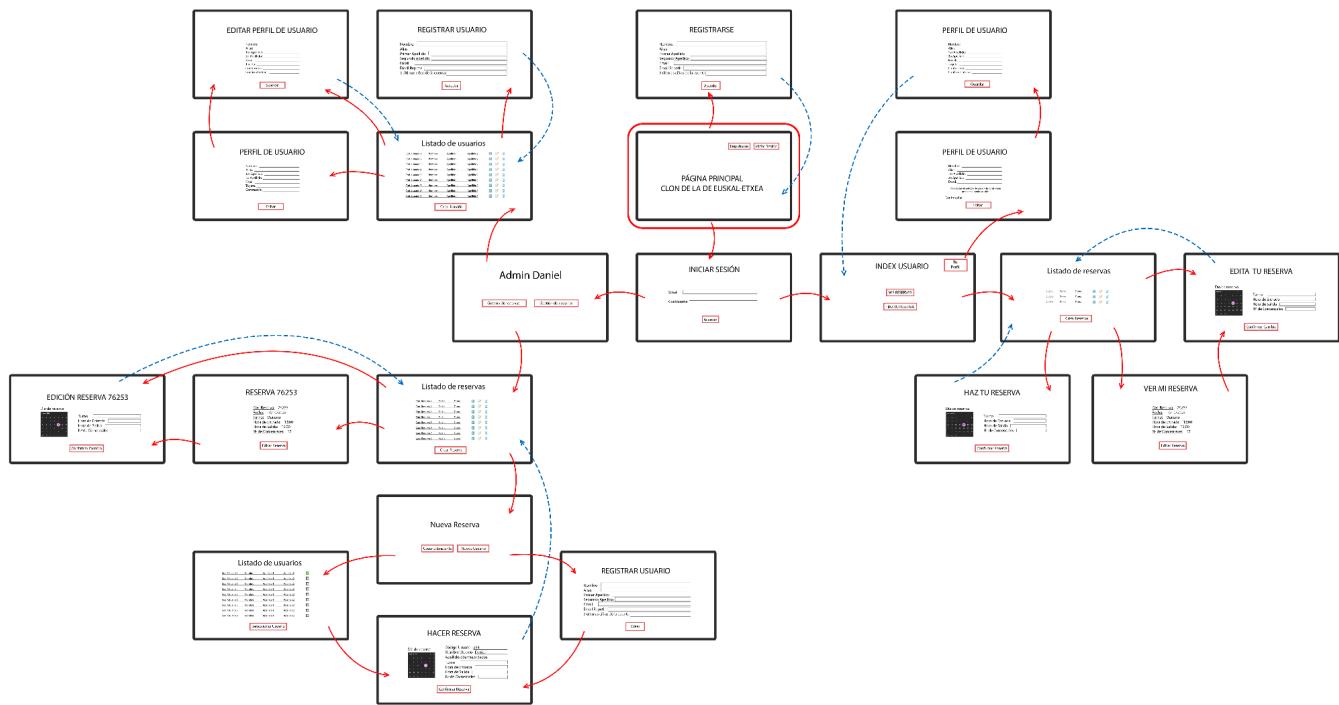
```
closeSession.php
config.json
createBooking.php
```

```
autoServer.com
estilos.css
home.php
```

```
script.js
sessionCheck
```

6.3. Diagrama de navegación

ESQUEMA WEB GESTIÓN DE RESERVAS EUSKAL ETXEA



La navegación por la página comienza en el fichero “home.php” donde el usuario podrá iniciar sesión o registrarse. A partir de ahí el movimiento entre las páginas viene definido por el rol del usuario que haya iniciado sesión.

Si el usuario tiene el rol “user” podrá navegar por aquellas páginas pensadas para el usuario habitual de la web, el cliente. Si por el contrario el usuario tiene el rol “admin” podrá acceder a la gestión de usuarios y reservas total.

7. Trabajo de Implementación

En el programa tenemos principalmente los siguientes tipos de ficheros de código:

- Ficheros de control que contienen los códigos PHP que formarán la parte backend de la aplicación. Estos ficheros controlarán entre otras cosas la gestión de cookies, conexión a BBDD, la obtención e inserción de datos en esta.
- Ficheros de vistas que contienen la parte principalmente HTML que mostrará los elementos visuales e interactivos de la aplicación web.
- Fichero JSON de configuración que contiene los datos de conexión a la BBDD.
- Fichero CSS que contienen el código que mejorará la forma y el color de la aplicación.
- Ficheros SQL en los que tendremos las estructuras SQL que construyen la BBDD y los elementos que rellenan cada una de las tablas de esta. Estos ficheros pueden extraerse de la carpeta del proyecto para que sean ejecutados en el gestor BBDD que sea necesario.

8. Trabajo de Pruebas

En primer lugar comprobamos que a la hora de registrar usuarios no ocurriera ninguna incongruencia, por lo que al final acabamos restringiendo que no pudiera haber usuarios con emails repetidos. Una vez solucionado el problema de los usuarios nos dimos cuenta que al iniciar sesión las cookies que se creaban podían generar errores, por lo que decidimos que al cerrar la sesión todas las cookies fueran borradas automáticamente. Además comprobamos que había un error de seguridad ya que alguien logueado como “user” podía cambiar todos sus datos por lo que implementamos que se necesitase introducir una contraseña para poder editar los datos. Una vez comprobado que toda la zona de gestión de usuarios funcionaba bien tocaba editar las reservas, en las cuales hicimos que las horas de reservas estuvieran acotadas en rango dependiendo del turno que se eligiese, así como no poder reservar en una fecha ya agenciada. Las pruebas nos sirvieron para arreglar todos los fallos e incongruencias que pudimos observar a lo largo de nuestro programa.

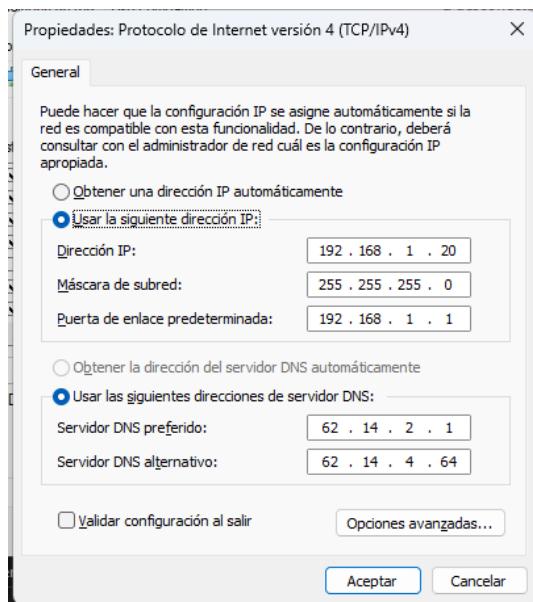
9. Trabajo de Implementación

La implantación de nuestra aplicación se ha organizado de la siguiente manera:

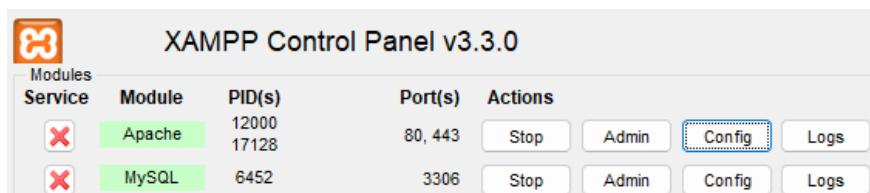
- Se crea un servidor web http en el administrador del router de la compañía de red. Para hacer esto vamos a la configuración de NAT/PAT/CGNAT y personalizamos una regla en la que definimos lo siguiente:

Personalizar reglas						
estado	aplicación / servicio	puerto interno	puerto externo	protocolo	IPv4 del dispositivo	
	FTP Server	21	21	TCP		<button>añadir</button>
	Web Server (HTTP)	80	80	TCP	192.168.1.20	<button>delete</button>

- A continuación configuraremos en el adaptador de red la configuración IP que hemos definido en nuestro servidor.



- El siguiente paso fue instalar XAMPP donde desplegamos usando Apache el servidor de aplicación y el servidor de base de datos usando mySQL.



Meteremos todos los archivos que conforman nuestra web en la carpeta htdocs dexampp.

- Una vez finalizado este proceso podremos proceder con el siguiente paso de conectar nuestra web a un dominio. Para hacer esto el primer paso realizado fue pagar el registro de dominio que posteriormente utilizaría nuestro sitio. En este caso utilizamos el gestor de dominios “Hostinger” donde obtuvimos la siguiente licencia de dominio.



- Finalmente se configuran los registros DNS apuntando a nuestra dirección ip del servidor (en este caso 87.222.165.111) (conectamos la ip 87.222.165.111 con el nombre “tfgpolydaniel.site” y “www.tfgpolydaniel.site”)

Administrador de registros DNS

Estos registros definen cómo se comporta tu dominio. Los usos comunes incluyen apuntar tu dominio a servidores web o configurar la entrega de correo electrónico para tu dominio.

Tipo*	Nombre*	Apunta a*	TTL*																				
A	@	Apunta a *	14400	Agregar registro																			
<table border="1"> <thead> <tr> <th>Tipo</th> <th>Nombre</th> <th>Prioridad</th> <th>Contenido</th> <th>TTL</th> <th></th> </tr> </thead> <tbody> <tr> <td>A</td> <td>www</td> <td>0</td> <td>87.222.165.111</td> <td>14400</td> <td>Borrar Editar</td> </tr> <tr> <td>A</td> <td>@</td> <td>0</td> <td>87.222.165.111</td> <td>14400</td> <td>Borrar Editar</td> </tr> </tbody> </table>						Tipo	Nombre	Prioridad	Contenido	TTL		A	www	0	87.222.165.111	14400	Borrar Editar	A	@	0	87.222.165.111	14400	Borrar Editar
Tipo	Nombre	Prioridad	Contenido	TTL																			
A	www	0	87.222.165.111	14400	Borrar Editar																		
A	@	0	87.222.165.111	14400	Borrar Editar																		

SECCIÓN III. PRODUCTO

10. Descripción externa

Página inicial, junto al registro y al iniciar sesión:



The image contains two side-by-side screenshots of the registration and login forms. The left screenshot shows the 'Registrar' (Register) form, which includes fields for Nombre, Alias, Primer Apellido, Segundo Apellido, Email, Teléfono, and Contraseña, followed by a 'Registrar' button. The right screenshot shows the 'Iniciar sesión' (Log in) form, which includes fields for Email and Contraseña, followed by an 'Entrar' (Enter) button.

Index del usuario, el perfil y la opción de editar perfil.



The screenshot shows the 'Perfil de Usuario' (User Profile) page. It displays basic user information: Nombre (Name) Pablo, Alias (Poli), Primer Apellido (First Last Name) Bermejo, Segundo Apellido (Second Last Name) Gómez, Email (Email) pablobermejogomez@gmail.com, and Contraseña (Password) (redacted). At the bottom is an 'Editar' (Edit) button.

The screenshot shows the 'Perfil de Usuario' (User Profile) page in edit mode. It lists all profile fields with their current values: Nombre (Name) Pablo, Alias (Poli), Primer Apellido (First Last Name) Bermejo, Segundo Apellido (Second Last Name) Gómez, Email (Email) pablobermejogomez@gmail.com, Teléfono (Phone) 1111111111, Contraseña (Password) (redacted), and Confermar Contraseña (Confirm Password) (redacted). At the bottom is a 'Guardar' (Save) button.

Lista de reservas del usuario y CRUD.

The image contains four screenshots of the EUSHAL ETHER application:

- Screenshot 1:** Shows the "Lista de Reservas" (List of Reservations) page. It features a table with columns: Código, Fecha, Turno, Ver, Editar, and Eliminar. Two rows are listed: one for Código 73330 on 2023-08-16 at Turno Matana, and another for Código 47427 on 2023-08-22 at Turno Matana.
- Screenshot 2:** Shows the "Ver Mi Reserva" (View My Reservation) page. It displays details for Reservation ID 73330, including Date (2023-08-16), Start Time (15:31:00), End Time (15:28:00), and Number of Participants (6). A "Editar" button is present.
- Screenshot 3:** Shows the "Haz Tu Reserva" (Make Your Reservation) page. It includes fields for Date, Turno, Start Time, End Time, and Number of Participants, along with a "Confirmar Reserva" (Confirm Reservation) button.
- Screenshot 4:** Shows the "Edita Tu Reserva" (Edit Your Reservation) page. It displays reservation details with input fields for Date, Turno, Start Time, End Time, and Number of Participants, and a "Confirmar Cambios" (Confirm Changes) button.

Index del admin

The image shows the Admin index page of the EUSHAL ETHER application. It features a large red and green Basque flag icon. Below the flag are two navigation buttons: "Gestion de Reservas" and "Gestion de Usuarios".

Gestión de usuarios y CRUD de usuarios desde el admin

The image shows the "Lista de Usuarios" (List of Users) page from the Admin section. It features a table with columns: Nombre, Alias, Apellidos, Rol, Ver, Editar, and Eliminar. The table lists various users with their details and management icons.

Nombre	Alias	Apellidos	Rol	Ver	Editar	Eliminar
Felipe		Rey Stoilo	user			
few	few	few wew	user			
fewf	few	oewfew	user			
Gabriel	gab22	Soto Garcia	user			
ger	grg	eror poger	user			
Javier	javi	Silva Peralta	user			
Jedris	jebi	Sánchez Sánchez	user			
Jorge	jig12	Silva Peralta	admin			
Jorge	George	Ruter Russell	user			
JoséJuan	squad	dawd dawd	user			
Laura	laural99	González Rojas	user			
Lorena	lone63	Gómez López	user			
Luis	lulito	Soto Navarro	user			

Gestión de reservas, con el editar y el ver.

Creación de una reserva desde el rol admin



11. Infraestructura

En cuanto al lenguaje de programación utilizado decidimos el uso de php puro, al comienzo nuestra intención era utilizar el framework laravel, pero debido a las evasivas del cliente decidimos comenzar a utilizar php ya que era lo mas fácil de modificar a la hora de recibir una respuesta, al momento de desvincularnos de la página oficial, el proyecto estaba en un estado tan avanzado que la mejor opción era terminarlo con la utilización de este lenguaje. En cuanto a la base de datos que hemos decidido utilizar es MySQL ya que nuestro conocimientos previos en este campo son bastante amplios por lo que se nos iba a hacer más sencillo utilizar esta BBDD. A la hora de comunicarnos e ir avanzando en el proyecto en paralelo sin muchas complicaciones utilizamos GitHub, una herramienta que cuanto mas conoces sobre ella más útil se vuelve. Para desplegar en local nuestra web utilizamos el xampp ya que sino tendríamos que haber utilizado un interprete de php, ya que el entorno que utilizamos es el que nos resultaba mas cómodo y con mas herramientas Visual Studio Code. Por último utilizamos un servidor nuestro que desplegamos a través de un ordenador.

12. Mantenimiento

Para el mantenimiento de la aplicación deben aplicarse parches y actualizaciones de seguridad para proteger la aplicación de posibles vulnerabilidades, realizar copias de seguridad periódicas de la base de datos y los archivos de la aplicación para evitar pérdidas de datos en caso de producirse algún error. Tratar de optimizar el código con mejoras constantes intentando encontrar potenciales errores. Hacer que los usuarios prueben la aplicación constantemente e implementar un apartado de contacto, para que los clientes puedan dar su opinión sobre posibles cambios que se puedan añadir a la aplicación. Mantener la documentación actualizada también es fundamental para facilitar futuros mantenimientos y para que otros desarrolladores puedan entender y trabajar en la aplicación si es necesario.

SECCIÓN IV.CONCLUSIÓN

13. Grado de consecución de objetivos

A día de hoy habiendo finalizado el proyecto podemos decir satisfactoriamente, que hemos logrado crear algo similar a lo que habíamos propuesto inicialmente, las dificultades encontradas en el proceso se han podido paliar de la manera más efectiva, aunque ha habido diferentes situaciones que nos han obligado a readaptar ciertas interpretaciones que teníamos de la aplicación. En cuanto a todo el contenido multimedia utilizado en la memoria, podemos decir orgullosos que es propio, gracias a nuestros conocimientos en Photoshop, no hemos tenido que recurrir a imágenes proporcionadas por internet. Obviamente somos conscientes que con más tiempo y mejor gestión de este, podríamos haber logrado no tan solo cumplir nuestras expectativas sino además haberlas superado con creces.

14. Posible evolución futura del proyecto

De cara al futuro, podrían realizarse una serie de mejoras al proyecto que lo hicieran crecer exponencialmente, como añadir la posibilidad de ver los distintos menús que ofrece el restaurante, incluso podría implementarse un sistema de descuentos para usuarios recurrentes, que ayudaría a la expansión del restaurante y mejoraría el feedback de los propios usuarios, pero más allá de mejoras técnicas, la evolución del proyecto natural, sería finalmente la implementación de este a la página oficial de Euskal Etxea, la elección del php puro como lenguaje de programación fue precisamente para evitar problemas de compatibilidad a la hora de implementarlo en la web real, ya que en caso de que ellos utilizaran algún tipo de framework podría llegar a complicarse su implantación. Por otro lado, para la mejora de la seguridad de la web, antes de implantarlo se debería implementar un control que realice validaciones de los datos obtenidos para poder evitar ataques externos, como por ejemplo las inyecciones SQL.

15. Lecciones aprendidas

Una vez acabado el proyecto y echando la vista atrás, no solo hemos sido capaces de mejorar nuestra velocidad a la hora de crear código nuevo, sino que además gracias a tener que gestionar un proyecto tan grande, nuestra capacidad de análisis se ha fortalecido significativamente. Aprendimos la importancia de una buena planificación, antes de comenzar el proyecto, dedicamos tiempo a definir los requisitos e identificar los objetivos dándonos cuenta de que una planificación adecuada nos permite anticipar problemas y tomar las mejores decisiones posibles. Al enfrentarnos a un proyecto tan grande, nos dimos cuenta de que dividirlo en

módulos más pequeños nos facilitaba el desarrollo y mantenimiento del código. Gracias al uso del GitHub pudimos trabajar en paralelo y minimizar los conflictos entre los miembros del equipo, a pesar de que al momento de unir las partes siempre había que sacrificar código. Además, tratamos de reutilizar el código existente, lo que nos permitió acelerar el desarrollo y mantener una base de código más limpia y mantenible a largo plazo.

16. Bibliografía y webgrafía

- Como crear un servidor web en tu pc local
 - ➡ Crear un servidor web en tu pc y hacerlo publico - 2020 !!
- Asignación de ip al dominio a tu web
 - ➡ Apuntar tu dominio a una dirección IP
- Creación de datos de ejemplo para las tablas

<https://chat.openai.com/>