

MEMORIA DEL PROYECTO CURSO 2018-2019 CFGS

TÍTULO DEL PROYECTO: RESERVED

NOMBRE DEL ALUMNO: DAVID NIN

HOJA DE CALIFICACIONES

(a rellenar por el Tribunal)

PROYECTO FINAL	PESO	CALIFICACIÓN
IDEA INNOVACIÓN	10%	
PRODUCTO	25%	
EXPOSICIÓN ORAL + DEMO	25%	
DOCUMENTO ESCRITO	40%	
OBJETIVOS	10%	
TECNOLOGÍAS USADAS	10%	
DISEÑO	30%	
DESARROLLO	10%	
TESTING	10%	
SEGUIMIENTO Y CONTROL	5%	
COSTES Y PLANIFICACIÓN	5%	
CONCLUSIONES, MEJORAS Y LÍNEAS FUTURAS	10%	
ANEXOS	5%	
EXTRAS	5%	
Q PROYECTO FINAL= <ul style="list-style-type: none"> • 10% Idea innovación • 25% Producto • 25% Exposición oral + demo • 40% Documento escrito 		
Nota final de la UF será calculada teniendo en cuenta los items de seguimiento, según la siguiente expresión: $Q_{UF1} = 0.1 \cdot Q_{FOLLOW_UPS} + 0.3 \cdot Q_{ITEMS_SEGUIMIENTO} + 0.6 \cdot Q_{PROYECTO_FINAL}$		

OBSERVACIONES (a rellenar por el tribunal)

Contenido

HOJA DE CALIFICACIONES.....	2
1 Introducción	7
2 Objetivos.....	8
2.1 Requisitos	8
2.1.1 Requisitos funcionales.....	8
2.1.2 Requisitos no funcionales.....	9
2.2 Valoración de requisitos.....	9
3 Tecnologías usadas.....	10
3.1 Lenguajes de Programacion	10
3.2 Herramientas, entornos de desarrollo y mas.....	11
3.2.1 Herramientas.....	11
3.2.2 Entornos de Desarrollo.....	11
3.2.3 Controlador de Versiones.....	12
4 Diseño.....	12
4.1 Diseño técnico	12
4.1.1 Esquema Código Fuente.....	12
4.1.2 Modelo de Datos	13
4.1.3 Diagrama de Clases.....	14
4.1.4 Casos de Uso.....	15
4.2 Diseño GUI.....	17
4.2.1 Arquitectura de la información	17
4.2.2 Diseño del layout.....	18
4.2.3 Diseño Visual	19
5 Desarrollo	21
5.1 Lenguajes de programación	21
5.1.1 Angular.....	21
5.1.2 Node	21
5.1.3 MongoDB.....	21
5.2 Herramientas, IDE, Controlador de Versiones	22
5.3 Aspectos Importantes de la Web	22

5.3.1	Registro / Inicio Sesión	22
5.3.2	Comentar	22
5.4	Implementaciones Externas para la Web.....	23
5.4.1	TripAdvisor.....	23
5.4.2	Google Maps.....	23
6	Testing	24
6.1	Registro.....	24
6.1.1	Introduciendo información en todos los campos.....	24
6.1.2	Introduciendo todos los campos excepto uno	24
6.1.3	Introduciendo todos los campos (con una cuenta de email ya registrada en nuestra base de datos) 24	
6.2	Inicio sesión	25
6.2.1	Introduciendo información en todos los campos.....	25
6.2.2	Introduciendo todos los campos excepto uno	25
6.2.3	Sin introducir ningún campo	25
6.3	Comentarios	26
6.3.1	Introduciendo información en todos los campos.....	26
6.3.2	Introduciendo todos los campos excepto uno	26
6.3.3	Sin introducir ningún campo	26
6.3.4	Numero introducido más grande de 10	26
6.4	Reservas.....	27
6.4.1	Introducir todos los campos correctos.....	27
6.4.2	Introducir solo la fecha (a la hora de buscar mesas).....	27
6.4.3	Introducir solo el turno (a la hora de buscar mesas).....	27
6.4.4	Click en mesa ya reservada.....	28
6.4.5	Introducir un número excesivo de personas	28
6.4.6	Introducir un número menos de personas	28
7	Seguimiento y control en la ejecución del proyecto	29
7.1	Apartado Reservas.....	29
7.2	Apartado Identificación	30
8	Costes y planificación	31
9	Conclusiones	32
10	Mejoras y líneas futuras	33
10.1	Mejoras.....	33

10.2	Línea futura.....	33
11	ANEXOS.....	34
11.1	Product Backlog.....	34
11.2	Herramientas.....	35
11.3	Entorno de Desarrollo	35
11.4	Controlador de Versiones.....	36
11.5	Cuerpo de la web.....	37
11.6	Registro / Inicio Sesión	38
11.7	Comentar.....	39
11.8	TripAdvisor.....	39
11.9	Error Reservas.....	40
11.10	Error Identificación	40
11.11	RoadMap	41

1 Introducción

Cuando empezó la asignatura de proyecto, yo me encontraba estudiando en la Salle y aparte trabajando en un restaurante/bar llamado Chumi Churri, restaurante de comida rápida y mexicana.

Todo esto lo cuento como contexto, ya que cuando empezó la época de pensar que íbamos a desarrollar para el proyecto final no tenía una idea fija, entonces me puse a pensar, sobre todo el sector de la hostelería y todo el respaldo informático que tiene este sector (ya sea por internet, web de opiniones o reservas, programas para gestionar comandas...) y pensé en la idea de ofrecer la posibilidad de poder reservar una mesa en un sitio concreto, por ejemplo, al lado del lavabo, en un día y un turno determinado.

Mi motivación a la hora de desarrollar este proyecto es poder ayudar o enfocar de una nueva manera de reservar mesa en restaurantes. Con mi idea creativa e innovadora, el cliente podrá ver desde su dispositivo (siempre que este disponga de conexión a internet) el mapa del restaurante, así como, la disponibilidad de dicho restaurante en la fecha y turno buscado.

Al ser algo innovador, ya que este método de reserva no lo he visto en ninguna página de reserva de restaurante/bar (de la competencia cercana al restaurante al cual he creado este nuevo servicio), creo que es una nueva manera de captar nuevos clientes de cara al restaurante y aparte para los clientes que son habituales, dadles una herramienta de reservas más fácil y útil que la que tiene por el momento este restaurante

2 Objetivos

2.1 Requisitos

2.1.1 Requisitos funcionales

El requisito principal de este proyecto es el hecho de que el usuario (una vez este identificado en la página web) tenga la posibilidad de poder reservar una mesa a partir de un gráfico representado por imágenes de mesas ya preparadas para X personas (hay imágenes de una mesa para dos, para cuatro...). Aparte de poder hacer una reserva, el usuario tendrá la capacidad también de borrar dicha reserva o consultar sus propias reservas.

Para llevar a cabo la acción que he explicado anteriormente, el usuario tendrá que cumplir un requisito mínimo el cual deberá estar registrado en la base de datos con una cuenta. Este sería otro requisito de la web que no es el principal pero que, sin este objetivo, no se puede llevar a cabo el requisito principal, por tanto, lo llamaría el requisito fundamental de esta web. Para que el usuario lleve a cabo esta acción, se ha creado una base de datos específica para esta web, así como una página de registro o en el caso de que el usuario ya tenga un correo electrónico y una contraseña asociada a este usuario pueda iniciar sesión.

Otro requisito funcional que hemos incorporado es el hecho de que el usuario tenga la capacidad de poder dejar un comentario en nuestra propia página web. Este comentario solo se puede crear si el usuario que accede a nuestra página web está identificado, sino es así deberá identificarse primero (como en el apartado anterior he explicado) y después ya podrá dejar su comentario.

2.1.2 Requisitos no funcionales

En los requisitos no funcionales de cara al usuario nos podemos encontrar la conexión entre los datos de mesas reservadas en un día específico (de la base de datos creada específicamente para esta web) y del día en el que el usuario a buscado una fecha para reservar una mesa. Este requisito no funcional sirve sobre todo para guiar al usuario de cara a reservar mesa (una vez se le muestre el grafico de imágenes como he comentado antes) para que el usuario tenga la idea de cuantas mesas quedan libres para esa fecha y para ese turno o al contrario, si tendrá que buscar otra fecha porque en la fecha que ha seleccionado el usuario ya está todo el restaurante lleno. Para que el usuario sepa si está o no reservada una mesa, la imagen se pondrá de color rojo cuando ya no esté disponible y de color blanco cuando si lo esté.

Otro requisito no funcional de la web creada para el Chumi Churri es la comprobación del inicio de sesión, es decir, si el usuario ha introducido un email y contraseña validas, dar-le permisos para entrar a la web con su cuenta y que el usuario ya tenga la opción de reservar o dejar un comentario. Así mismo si la información introducida por el usuario es errónea, mostrarle un error conforme la información que se requisa para iniciar sesión es incorrecta y que introduzca otra vez sus datos.

2.2 Valoración de requisitos

Objetivamente creo que los requisitos mínimos para este proyecto los he cumplido y con creces (no hay ningún error en los requisitos anteriormente nombrados) sin esconder que también ha habido requisitos funcionales que no he podido llegar a implementar por una organización de tiempo errónea de cara a realizar el proyecto o por falta de conocimientos.

Aquí os dejo una referencia hacia el anexo donde os dejo una imagen del Product Backlog con las versiones del primer y segundo Ítem así mismo una pequeña referencia de aquello que se ha

llevado a cabo y aquello que al final no se ha podido llevar a cabo.

ANEXO: 11.1 Product Backlog.

3 Tecnologías usadas

3.1 Lenguajes de Programacion

El proyecto de fin de curso, en la escuela de la Salle se lleva a cabo durante todo el curso, por lo tanto, cuando se acercaba el momento de decidir con que tecnología íbamos a hacer el proyecto, era en noviembre y aun estábamos aprendiendo a programar, eso quiere decir que no tenía una referencia muy clara sobre que lenguaje dominaba en el mercado global y me resultaría más fácil aplicarlo a mi proyecto.

Empecé a investigar sobre que lenguajes de programación se podría ajustar más a mis necesidades a la hora de hacer esta web. Descubrí bastantes lenguajes de programación muy interesantes y que prometían mucho potencial, como ahora React(de cara a Front) y Python/SQL (de cara a Back y Base de Datos), pero hubo una combinación de lenguajes que me llamo más la atención que las demás y esta fue Angular (por parte en el Front) y Node/MongoDB (por parte en el Back y en Base de Datos). Empecé a buscar información sobre dichos lenguajes de programación (ya era por la misma documentación de los creadores del lenguaje o por web, foros etc..) hasta que un día no sabía por dónde buscar más información para ponerme ya a implementar dichos lenguajes en mi proyecto, hasta que pregunté a Uri sobre alguna página donde poder aprender dichos lenguajes con facilidad.

Entonces me enseñó la página web de cursos online llamada Udemy, allí fue donde aprendí a manejar dichos lenguajes y poder implementarlo hacia mi proyecto.

Escogí Angular como lenguaje de programación dominante en mi Front-End por el simple hecho de que (una vez aprendido, que me costó bastante pillarle el truco) es tan comodo el hecho de cambiar de pestaña en tu web y que esta no se recargue entera, que solo se recargue la parte que quieres que me parecía una maravilla, pensé en que angular en mi página web podría cuadrar, podría funcionar y esa fue la razón principal por la que implemente Angular en mi proyecto.

El curso que me compre, especialmente lo hice porque la fuerza dominante del Front-End era angular y me enseñaba a implementarlo como tal, pero este curso fue una sorpresa, porque por parte del Back-end enseñaban Node y MongoDB, dos lenguajes de programación que no tenía en mente integrar en mi proyecto que, a raíz de seguir el curso, me pareció tan fácil de implementar y de relacionar con Angular que no dude ni un momento en implementarlo en mi proyecto.

3.2 Herramientas, entornos de desarrollo y mas

3.2.1 Herramientas

Para el diseño de las mesas he utilizado Adobe Illustrator por tal de crear unas mesas específicas y no coger imágenes de internet. Aparte de este programa de Adobe, utilice el Adobe Color Picker para crear una paleta de colores que luego me serviría para dar diseño a la web.

Aparte de utilizar programas de Adobe, utilice Trello como guía a la hora de seguir sprints

ANEXO: 11.2 Herramientas

3.2.2 Entornos de Desarrollo

Como bien he dicho antes, cuando empeze a programar código de cara al proyecto, aun estábamos un poco verdes en el mundo de la programación, por lo que empeze a programar con el IDE llamado Atom, un IDE sencillo y cómodo de utilizar. Hasta que llegó el momento de hacer prácticas en una empresa, gracias a ello descubrí un nuevo IDE para mi llamado Visual Studio que era tan cómodo, fácil de programar y tan fácil de implementar ayudas (ya sea auto rellenado de palabras, paletas de color para saber que en que función estas...)

Aparte de este IDE tan conocido, utilice Postman para hacer consultas a la base de datos desde el Back-End que había creado. Programa que descubrí gracias al curso de Udemy y que me resulto muy fácil y cómodo de utilizar.

ANEXO: 11.3 Entornos de Desarrollo

3.2.3 Controlador de Versiones

El programa que utilice como controlador de versiones es Git. Utilice Git en local ya que es el controlador de versiones estándar en la mayoría de proyectos que cualquier programador crea. Conocía su existencia gracias a las clases de Santi, así que su implementación en el proyecto fue fácil. Utilice GitHub para subirlo a un repositorio remoto.

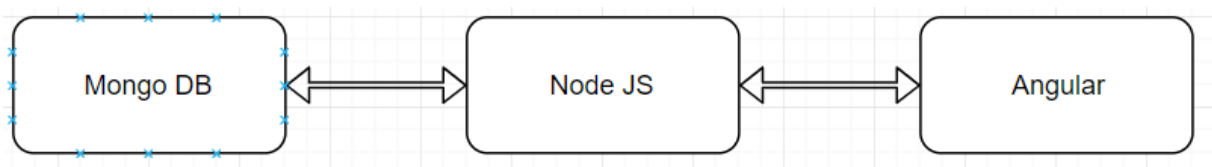
ANEXO: 11.4 Controlador de Versiones

4 Diseño

4.1 Diseño técnico

4.1.1 Esquema Código Fuente

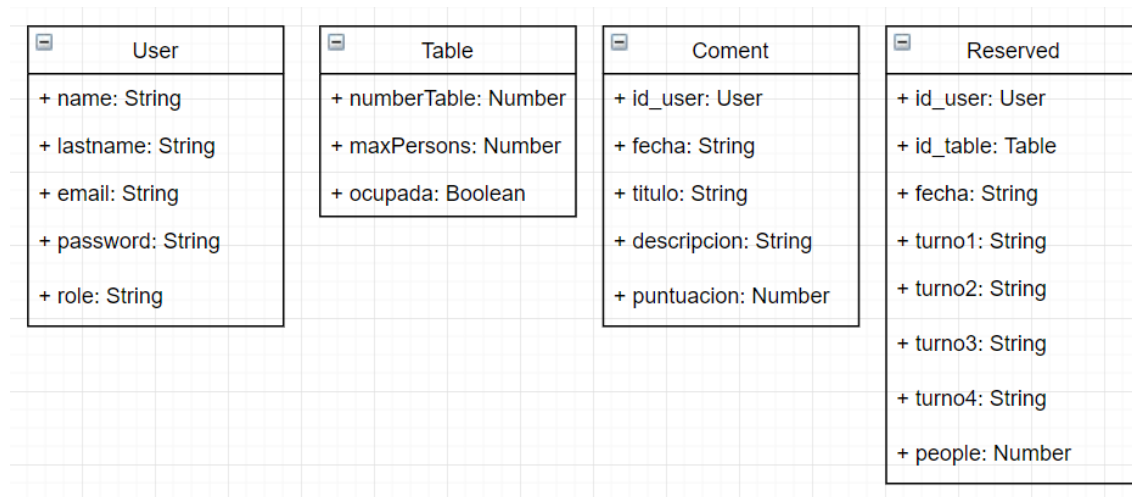
En este apartado os explicare la comunicación de la web que he creado. Esta comunicación es bastante simple y se representa de la siguiente manera:



La comentare un poco por encima aunque creo que una imagen vale mas que mil palabra y mas en esta foto, la conexion es muy simple, angular recibe la informacion del usuario, la cual se la pasa a Node y Node hace peticiones a la base de datos Mongo, Node recibe una respuesta que se la envia a Angular para que se lo muestre al usuario.

4.1.2 Modelo de Datos

Aquí hablaremos sobre el modelo de datos y que patrón he seguido a la hora de crear usuarios, mesas, comentarios y reservas.



Como podéis observar en la siguiente captura (la pongo aquí ya que vamos a hablar de ella en profundidad, sino la pondría en el anexo) tenemos cuatro modelos de datos. Hablaremos uno por uno empezando por el de usuario, el cual tiene la información que se conoce como información necesaria para poder registrarte en cualquier base de datos y el administrador de esta, poder saber un mínimo de cada usuario.

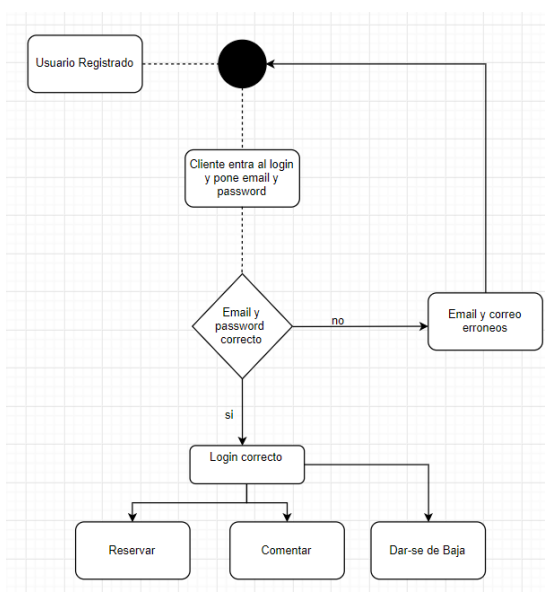
Para el modelo de mesa, he creado tres columnas de datos, numero de mesa, para que el administrador del sistema cuando mire las reservas sepa que mesa tiene que reservar, el número máximo de personas por mesa, para que los usuarios cuando tengan que reservar mesa no se pasen de personas asignadas a una mesa en concreto y la última columna de información y la que me llevo más tiempo para pensar en ponerla o no es la de ocupada, esta columna solo sirve para saber si la mesa está ocupada o no en los resultados de la búsqueda del usuario, con unos métodos que cree de código, modifica dicha variable en función si está reservada o no.

Para el modelo de comentario, solo la columna de dato a resaltar es la que coje el id del usuario para tener una referencia de quien ha hecho el comentario aparte de toda la información que creo que es necesaria sobre un comentario, como es la fecha en la cual se ha comentado, una puntuación...

Para el ultimo método, pero no menos importante como ahora es el de reserva recoge la id del usuario y de la mesa seleccionada por el usuario además del turno de la búsqueda del usuario (dicho turno se pone como valor positivo, es decir, en true, mientras que los demás obtendrán un valor negativo, es decir, en false) ya que por defecto todos son valores negativos, con un valor en false. La última columna de dato que es personas, guarda la cantidad de personas que van a ocupar la mesa reservada, la cual, compara con la columna anterior de tabla (maxPersons) y si el número de personas que van a ocupar la mesa es más grande que el número máximo de personas, no se podrá llevar a cabo la reserva

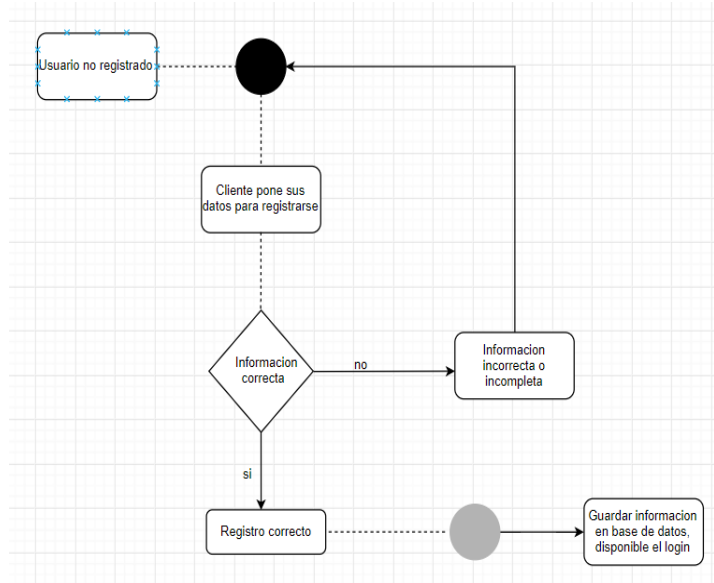
4.1.3 Diagrama de Clases

Este es el diagrama de clases sobre un usuario que está registrado, pero no ha iniciado sesión en nuestra página web.



En el cual, si un usuario que está registrado, pero no ha iniciado sesión, tendrá que iniciar con su email y su password, si es erróneo, volverá otra vez a la página del login, si los datos son correctos, se abrirá la sesión del usuario y podrá hacer las funciones principales de un usuario en nuestra web.

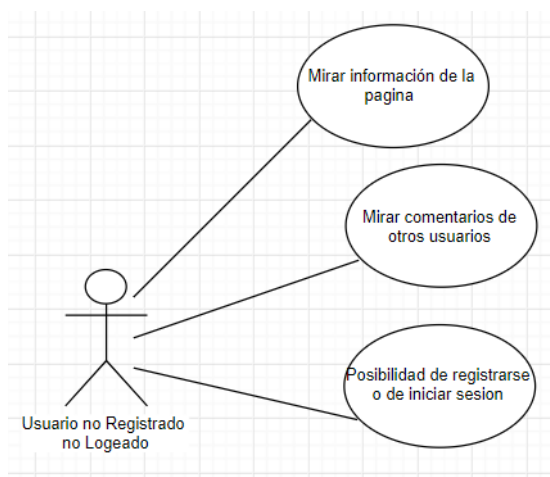
Este sería el diagrama de clase de un usuario el cual no ha hecho ningún registro, no tienen ninguna información personal en nuestra base de datos, por lo tanto, quiere ingresar sus datos, si hay algún dato erróneo o vacío, vuelve a la página de registro, sino, podrá proseguir al login.



4.1.4 Casos de Uso

Para los casos de uso he implementado tres tipos de usuario: El usuario no registrado o no logeado, el usuario identificado o el usuario administrador.

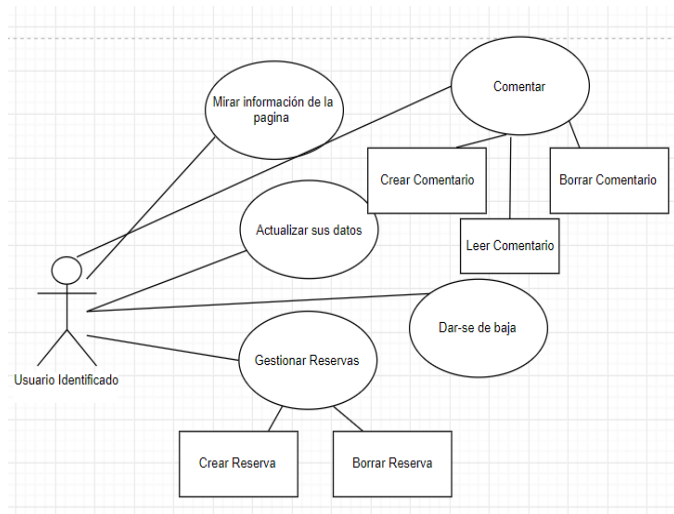
El usuario no registrado o no logeado:



Este tipo de usuario solo podra hacer tres tareas en nuestra página web:

- Mirar información sobre la página web (toda la página web excepto la pestaña de reservas)
- Mirar comentarios de otros clientes en nuestra página web
- Poder hacer login o password

El usuario identificado:



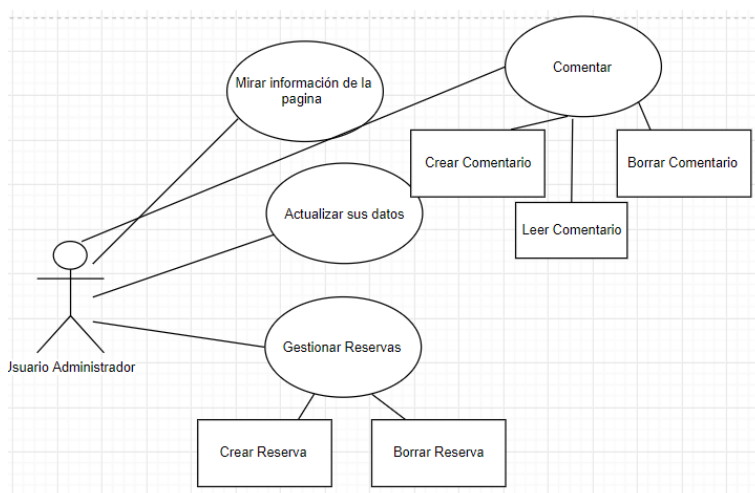
Este tipo de usuario podrá hacer todas las tareas de web:

- Mirar información sobre la página web
- Actualizar sus datos: El usuario logeado podrá actualizar su email así como su nombre o su contraseña.
- Gestionar Reservas: El usuario logeado podrá hacer una reserva o borrar sus reservas.

-Comentar: El usuario logeado podrá crear un comentario, así como borrar sus comentarios y leer todos los demás comentarios

-Dar-se de Baja: El usuario tendrá a su disposición una funcionalidad para dar-se de baja de nuestra base de datos así como de nuestra web

El usuario administrador:



Este tipo de usuario podrá hacer todas las tareas de web:

- Mirar información sobre la página web
- Actualizar sus datos: El administrador podrá actualizar su email, así como su nombre o su contraseña.
- Gestionar Reservas: El

administrado podrá hacer una reserva o borrar todas las reservas.

-Comentar: El administrador podrá crear un comentario, así como borrar todos los comentarios y leer todos los demás

4.2 Diseño GUI

4.2.1 Arquitectura de la información

La página web está distribuida en cinco secciones que son:

- **El restaurante:** Es la página donde además de explicar un poco de historia sobre el restaurante, encontraremos la función de poder compartir una opinión de un usuario identificado, así como todos los demás comentarios de usuarios. Además de los comentarios, también encontraremos una pequeña pestaña sobre los comentarios de chumi churri en TripAdvisor
- **La carta:** Página donde el usuario encontrara la oferta gastronómica del restaurante
- **Reservas:** Página principal y por la cual he desarrollado todo este proyecto, es el santo grial de esta web. Lo primero que observará el usuario que entre a este apartado será una serie de pasos que tendrá que leer saber cómo llevar a cabo una reserva. Tendrá que meter una fecha, seguidamente de elegir turno. Una vez le de al botón de buscar con toda la información introducida, le aparecerá el mapa gráfico del restaurante donde en las mesas que no tengan el fondo rojo se le permitirá al usuario reservar mesa siempre y cuando la mesa escogida y las personas que ocupen esta mesa no sean mayor al número máximo de sillas de la imagen de la mesa.
- **Contacto:** Página típica de cualquier restaurante donde el usuario podrá ver la localización del restaurante así como el horario, correo de crítica o sus redes sociales
- **Login:** Página para que los usuarios registrados puedan identificarse y los nuevos usuarios puedan registrarse

4.2.2 Diseño del layout

La web que he diseñado se basa en dos partes:

La Barra de Navegación: Consiste en una barra de navegación estática la cual no variara (excepto el botón de identificarte y el botón de logOut, ya que estos botones variaran según el usuario que haya entrado en la página web este identificado o no). La barra de navegación está compuesta por una imagen del logo del restaurante (cuando haces click en ella te redirecciona a la sección principal), 4 secciones (que las he nombrado y definido antes) y el botón que varía según el estado del usuario.



Esta es la barra de navegación cuando entras por primera vez a la web del restaurante. Como se puede apreciar, el botón de identificarse esta colocado a la derecha mientras que todas las opciones de navegación de la web están colocadas a la izquierda.

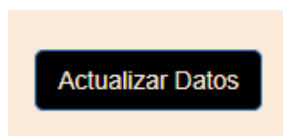


Esta es la barra de navegación cuando el usuario está identificado en la web. Como se puede apreciar, el botón de identificar-se ha sido substituido por un botón de cerrar sesión además de un botón a su derecha que forma parte de la barra de navegación como es el botón del perfil. Este botón aparecerá a la izquierda del botón de dar-se de baja y saldrá el nombre del usuario identificado.

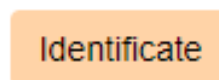
El Cuerpo: Es la parte de la web que depende de la sección elegida en la barra de navegación cambia de contenido. Gracias al lenguaje de programación Angular la página web no se recarga cada vez que cambias de sección, hecho que hace que la web sea más rápida a la hora de la navegación y hace que el usuario no pierda mucho tiempo entre que se carga una ventana y pierda la paciencia y se termine yendo de la web.

ANEXO: 1.15 Cuerpo de la Web

A la hora de hacer el diseño de los botones y de los desplegables de la carta, consulte varias páginas web sobre hostelería, y me fije que lo que predominaba en este sector son botones con texto muy breves y sencillos, así como botones no cuadrados, sino con un cierto nivel de borde redondo. Los botones son de interacción con la página web son de color negro con la letra de color del fondo del recuadro de interacción, salvo el botón de identificarte que es del mismo color que el color de fondo de pantalla con las letras en color negro. Por el contrario el botón de cerrar sesión o dar-se de baja es el típico botón con las letras de color negro y un fondo de color rojo.



Botón de interacción normal



Botón de login

Por parte del diseño del contenido de interacción, este está con un fondo de color más resaltado que el fondo de color de la web (detalle que hace que resalte con la página y cree más impresión a un simple vistazo rápido).

 A login form titled 'Iniciar Sesión' on a light orange background. It contains two input fields: 'email' with a person icon and 'password' with a lock icon. Below the fields is a dark orange 'Entrar' button. At the bottom, there is a link 'No tienes una cuenta? Regístrate Aquí' where 'Regístrate Aquí' is a dark orange button.

Efecto que hace que la web sea más atractiva y más útil de cara a usuarios con alguna discapacidad

4.2.3 Diseño Visual

Como he comentado en el punto anterior, el hecho de que aquello que sea interactivable con el usuario sea de un color un poco más vivo, un poco más fuerte que el color del fondo y de un pequeño contraste hace que la web sea más dinámica y no tan monocromática.

La paleta de colores que utilizado para hacer este efecto son tres colores, de menos a más intensos, todos siguiendo una línea de color que es un color mostaza clarito. El código de color que utilizado en el fondo es: #F9F3ED, el color para hacer contraste entre el fondo y la parte interactuable es: #FBEADA mientras que el color con más contraste y que menos he utilizado es: #FFD3A7.

rgb(249, 243, 237) # F9F3ED

rgb(251, 234, 218) # FBEADA

rgb(255, 211, 167) # FFD3A7

De cara a la tipografía utilizada, he cogido tres fuentes diferentes para cada sección de la web. Para los títulos de las secciones he utilizado una tipografía llamada “Open Sans, sans-serif” con efecto letra negrita. Para partes de descripción, he puesto la misma letra que para el título, pero con efecto “light” y cursiva. Y para el resto de página web he utilizado la misma letra del título y descripción, pero sin ninguna modificación.

Nuestra Carta

Tipo de letra de titulo

A continuación, podrán averiguar la nueva experiencia gastronómica que ofrece nuestro icónico restaurante Chumy Churry. Podrá degustar desde la comida más típica mediterránea como las delicias mexicanas siempre elaboradas y preparadas por nuestros expertos cocineros tanto en gastronomía mediterránea como en gastronomía mexicana.

Tipo de letra de descripción.

Enrollados de tortita

Butifarra con Queso y Roquefort

Franfurt con Bacon y Queso

Vegetal (Pimiento Rojo, Berenjena, Cebolla, Esparrago y Queso)

Tipo de letra del resto de la web

5 Desarrollo

5.1 Lenguajes de programación

5.1.1 Angular

Tuve que instalar Angular-CLI para agilizar la creación, generación, ejecución, testing, deploy de la aplicación web. A la hora de instalar Angular me dio bastantes problemas ya que empecé con la versión 2.0 y tuve que migrar en medio del proyecto a la última versión de Angular ya que la 2.0 está muy anticuada.

5.1.2 Node

Gracias a la instalación de Angular, Node ya está configurado e instalado para poder trabajar con este lenguaje de programación y Angular. En si, no me dio ningún problema a la hora de instalar Node. A parte de Node instale Nodemon por tal de que el servicio de back-end si este es modificado, se actualice solo.

5.1.3 MongoDB

Con la instalación de MongoDB solo me tuve que descargar un programa el cual fue durante todo el proceso de instalación darle al botón de siguiente y ya se configuraba solo. No me dio ningún problema a la hora de instalar-lo. Instale también la extensión de Mongoose por tal de definir objetos con un esquema fuertemente tipado y por su gran cantidad de funcionalidades para crear y trabajar con esquemas.

5.2 Herramientas, IDE, Controlador de Versiones

A la hora de instalar cualquier herramienta, cualquier IDE o Git no ha habido ningún problema que retrasara la planificación de las tareas de la web.

5.3 Aspectos Importantes de la Web

5.3.1 Registro / Inicio Sesión

Es uno de los requisitos fundamentales para la web, se trata del clásico registro de datos del cliente en la base de datos del usuario y de la introducción de email y password para iniciar sesión.

Lo más complicado, a nivel personal, ha sido el hecho de coger la password que introduce el usuario a la hora de registrarse, cifrar dicha password en la API y cuando el usuario inicie sesión, descifrarla y compararla.

Otra de las cosas que me han costado más, cuando el usuario se identifica correctamente, devolver un token para cada vez que haga una interacción en la web (ya sea reservar o comentar) obtener este token y una vez obtenido, sacar el id del usuario.

ANEXO: 11:16 Registro / Inicio Sesión

5.3.2 Comentar

Es una función extra, que bien es una función importante, en el primer Product Backlog no estaba planteada que fuera una página de comentarios internos, sino que fuera a través de terceros (como explicare en el siguiente punto).

Lo más complicado de esta funcionalidad fue el hecho de darle una visualización correcta, ya que tenía mucha información de los comentarios, pero no sabía cómo gestionar dicha información. Las funciones para crear esta posibilidad dentro de la página web son sencillas, pero igualmente os dejare un par de captura en el anexo.

ANEXO: 11:17 Comentar

5.4 Implementaciones Externas para la Web

5.4.1 TripAdvisor

En la web, hay una pequeña mención a la página líder en comentarios sobre bares, restaurantes, hoteles... vamos lo que viene siendo el rey del sector de la hostelería. Mi idea era de alguna manera enlazar los comentarios de mi página web con los comentarios de una aplicación tercera como es TripAdvisor.

En este caso, implemente un pequeño recuadro donde se muestra un pequeño resumen de comentarios de gente, una puntuación global sobre los servicios del restaurante y un pequeño enlace para salir de la página web del restaurante y entrar en la del TripAdvisor.

La manera de implementar esta pequeña API fue relativamente fácil ya que solo me tuve que registrar como el administrador web de este restaurante y la propia web de TripAdvisor me ofrecía diversos widgets de diversos tamaños, solo tuve que poner un iframe y un script para que funcionara.

ANEXO: 11.18 TripAdvisor

5.4.2 Google Maps

API muy fácil de implementar, ya que solo me tuve que ir a la página de google maps, introducir aquello que quiere saber, compartir como si fuera un iframe y listo.

Tamaño personalizabe.

6 Testing

Los testeos de comentarios y reservas dando por entendido que el user está identificado

6.1 Registro

6.1.1 Introduciendo información en todos los campos

Objetivo:

Conseguir que el usuario al introducir toda la informacion correctamente, ya tenga sus datos en nuestra base de datos y pueda realizar el login.

Resultado: ERROR / **CONSEGUIDO**

6.1.2 Introduciendo todos los campos excepto uno

Objetivo:

Conseguir que el usuario al introducir tres de los cuatro campos que se requieren para un buen registro observe que se ha dejado un campo por rellenar con un mensaje de error

Resultado: ERROR / **CONSEGUIDO**

6.1.3 Introduciendo todos los campos (con una cuenta de email ya registrada en nuestra base de datos)

Objetivo:

Conseguir que el usuario al introducir todos los campos que se requieren para un buen registro observe que se ha ocurrido un error y que esa cuenta de correo electrónico ya está registrada en la base de datos

Resultado: ERROR / **CONSEGUIDO**

6.2 Inicio sesión

6.2.1 Introduciendo información en todos los campos

Objetivo:

Conseguir que el usuario al introducir toda la información correctamente pueda iniciar sesión sin ningún problema y tenga accesibilidad a las funciones de la web

Resultado: ERROR / **CONSEGUIDO**

6.2.2 Introduciendo todos los campos excepto uno

Objetivo:

Conseguir que el usuario al introducir uno de los dos campos que se requieren para un buen inicio de sesión observe que se ha dejado un campo por rellenar con un mensaje de error

Resultado: ERROR / **CONSEGUIDO**

6.2.3 Sin introducir ningún campo

Objetivo:

Conseguir que el usuario al no introducir ninguna información en los campos que se requieren para un buen inicio de sesión observe que se ha dejado todos los campos por rellenar con un mensaje de error

Resultado: ERROR / **CONSEGUIDO**

6.3 Comentarios

6.3.1 Introduciendo información en todos los campos

Objetivo:

Conseguir que el usuario al introducir toda la información correctamente pueda dejar su comentario sin ningún problema.

Resultado: ERROR / **CONSEGUIDO**

6.3.2 Introduciendo todos los campos excepto uno

Objetivo:

Conseguir que el usuario al introducir dos de los tres campos que se requieren para un buen funcionamiento de la generación de comentarios, observe que se ha dejado un campo por rellenar con un mensaje de error

Resultado: ERROR / **CONSEGUIDO**

6.3.3 Sin introducir ningún campo

Objetivo:

Conseguir que el usuario al no introducir ninguna información en los campos que se requieren para un buen funcionamiento de la generación de comentarios observe que se ha dejado todos los campos por rellenar con un mensaje de error

Resultado: ERROR / **CONSEGUIDO**

6.3.4 Numero introducido más grande de 10

Objetivo:

Conseguir que el usuario al introducir un numero más grande de los permitidos, observe que se ha dejado todos los campos por rellenar con un mensaje de error

Resultado: ERROR / **CONSEGUIDO**

6.4 Reservas

6.4.1 Introducir todos los campos correctos

Objetivo:

Conseguir que el usuario al introducir todos los campos requeridos para hacer una reserva, observe que se ha generado una reserva correctamente y que se de cuenta con un mensaje afirmando dicha acción.

Resultado: ERROR / **CONSEGUIDO**

6.4.2 Introducir solo la fecha (a la hora de buscar mesas)

Objetivo:

Conseguir que el usuario al solo introducir la fecha, observe que se ha dejado por escoger el turno el cual quiere hacer la reserva y que se dé cuenta con un mensaje afirmando dicho error.

Resultado: ERROR / **CONSEGUIDO**

6.4.3 Introducir solo el turno (a la hora de buscar mesas)

Objetivo:

Conseguir que el usuario al solo introducir el turno pueda proseguir con la búsqueda de mesa para la fecha actual

Resultado: ERROR / **CONSEGUIDO**

6.4.4 Click en mesa ya reservada

Objetivo:

Conseguir que una vez la búsqueda de mesas haya sido correcta, a la hora de clickar en una mesa que ya está reservada no consiga proseguir con la reserva de dicha mesa

Resultado: ERROR / **CONSEGUIDO**

6.4.5 Introducir un número excesivo de personas

Objetivo:

Conseguir que una el usuario haya escogido una mesa, las personas que van a rellenar esa mesa no sea más grande que el número máximo de personas para dicha mesa, por lo tanto, cuando esto ocurra se dará cuenta por un mensaje de error en el que le ponga que el número introducido sobrepasa al número máximo de personas para la mesa

Resultado: ERROR / **CONSEGUIDO**

6.4.6 Introducir un número menos de personas

Objetivo:

Conseguir que una el usuario haya escogido una mesa, las personas que van a rellenar esa mesa no sea más pequeño que el número máximo de personas para dicha mesa, por lo tanto, cuando esto ocurra se dará cuenta por un mensaje de error en el que le ponga que el número introducido es inferior al número máximo de personas para la mesa

Resultado: ERROR / **CONSEGUIDO**

7 Seguimiento y control en la ejecución del proyecto

En este apartado me centrare en explicar los errores más graves que he tenido, ya que estos errores me han hecho retrasarme mucho más de lo previsto. Cuando en el primer Item de este proyecto nos pedisteis que generáramos una especie de estima de tardanza del proyecto, lo vi tan fácil que tire a lo bajo y dije que el proyecto estaría terminado en marzo, cosa que no fue así y gracias a estos errores se ha terminado en mayo.

7.1 Apartado Reservas

El primer error que tengo el gusto de explicaros es el peor que he tenido en el proyecto y tiene relación con el apartado de reservas. Se trata del hecho de hacer que cuando el usuario filtre por una fecha y un turno, estos dados cotejarlos con la base de datos y retornar las reservas de ese día y ese turno, luego, cotejarlas con el hecho de si estaban reservadas o no con el array de mesas que están todas las mesas.

Lo tuve a la hora de que el desplegable del datepicker no apareciera cada vez que se hacía click en la sección de reservas y el usuario estuviera identificado. Por lo tanto, investigando descubrí que el código que ejecutaba el datapicker se tenía que colocar en el ngOninit para cada vez que se entrara en la sección se cargara.

ANEXO: 11.9 Error Reservas

Otro error que tuve bastante grave fue el hecho de que el programa petaba a la hora de obtener el turno escogido. Antiguamente me salían como diez líneas de código erróneas y actualmente me salen dos, lo malo de estas dos líneas que no he conseguido arreglarlas es que al compilar el proyecto peta, pero aun asi funcionan, para conseguir que compile, tendremos que comentar un par de líneas.

ANEXO: 11.9 Error Reservas

7.2 Apartado Identificación

El error que tuve de vulnerabilidad y no resolví hasta hace poco (y suerte que me di cuenta ya que creo que era motivo para suspender este proyecto) y era el hecho de que un usuario podía registrarse dos veces con el mismo email. Esto lo solucione con una función muy sencilla en la parte del front que podéis ver en el siguiente “link” al anexo

ANEXO: 11.10 Error Identificación

Al arreglar el error anterior, me sucedió otro error que fue una catástrofe ya que, a la hora de enviar la información, no mostraba ningún error la consola y me ponía que no podía acceder al método de registro de la API, gracias al repaso del código línea por línea, conseguí averiguar el error que residía en un mal formato de código y una {cerrada mal.

Cada vez que realizaba un paso bastante importante dentro del proyecto (ya sea crear una nueva sección, arreglar un problema bastante gordo de estilos, crear una nueva funcionalidad etc etc...) veía prioritario el hecho de hacer un guardado rápido gracias a la herramienta de Git para ahorrar futuros infartos por si perdía todo el progreso del proyecto.

Por tal de tener el proyecto guardado en otro dispositivo que no sea mi ordenador de trabajo (en caso de pérdida o rotura del mismo) decidí colgar el desarrollo del proyecto en GitHub. En esta plataforma iba colgando siempre que veía que avanzaba bastante en el proyecto y tenía que subirlo por precaución.

URL de GitHub: <https://github.com/davidnin/proyectoFinCurso/tree/master>

8 Costes y planificación

El roadMap inicial, en el cual no ha cambiado gran cosa respecto al de ahora (por eso no pongo el nuevo y me dedico a explicar en qué ha cambiado el de ahora) el cual podréis observar en el anexo. Como he comentado en el punto anterior, gracias a todos los errores que me surgieron, tarde dos meses más de lo esperado.

ANEXO: 11.11 RoadMap

Para dar un costo aproximadamente del proyecto en global, supondremos que mi sueldo a la hora (limpio) es de 9€ y las horas que he dedicado al proyecto aproximadamente son de 200h (contando también la documentación). Por tanto, si queréis un número aproximado de lo que costaría realizar este proyecto, el número sería de unos **1.800€** solo la mano de obra de un programador junior, como lo soy yo. Ha esto hay que añadir-le que es una nueva idea revolucionara en el sector de la hostelería cosa que crecería la cifra final a unos **4.000€**. Todo esto contando que todo lo que se ha utilizado en la web ha sido free-software.

9 Conclusiones

Llegados a este punto toca hacer una pequeña reflexión sobre este proyecto. En este proyecto he aprendido a implementar funciones que tenía en mente en mi proyecto, y estoy muy orgulloso de mi mismo tanto por llevar a cabo esa acción como por el hecho de sacar adelante este proyecto con apenas el fin de semana para desarrollar-lo.

También he aprendido a auto gestionarme el poco tiempo que tenía para llevar a cabo todo esto igual que también he aprendido a buscarme la vida y aprender por mi banda (a la hora de desarrollar Angular en un nivel más profesional, por ejemplo) y estos dos roles son muy bueno para mi futuro dentro de esta industria.

Por último, mencionar que de lo que me siento más orgulloso es de poder enseñar al mundo la web que he creado, sé que no se puede comparar con páginas web de grandes dimensiones, pero personalmente, es mi pequeña obra de arte, la cual puedo enseñarse-la a cualquiera y sé que le gustara.

Hasta aquí mis conclusiones de todo el proyecto, quiero despedir esta pequeña reflexión con agradecimientos a todos los compañeros que me han ayudado en los lenguajes con los que he ido desarrollando y a los profesores, que siempre que teníamos una pequeña duda, nos la resolvían tan pronto como les era posible.

Espero que les guste mi web, ¡un saludo!

10 Mejoras y líneas futuras

En este último apartado comentare aquellas mejoras que le daría a la web ya creada y que podría implementar en un futuro.

10.1 Mejoras

La primera mejora que le haría al código de la web sería intentar optimizar algunas funciones que creo que con el nivel que tengo actual de Angular podría resolver (ya que, dichas funciones están creadas desde hace meses y no tenía el nivel de Angular que dispongo actualmente)

Una de las mejoras que le pondría a la web sería poner una especie de alerta a la hora de que el usuario, una vez en la sección de reservas, si la posibilidad no elegir una fecha a la hora de buscar mesas disponibles, como automáticamente se busca por el día actual, enseñar un mensaje explicando ese descuido.

Una vez el usuario ha encontrado una mesa disponible, cuando haces click en la mesa para seleccionar-la se abre el último de los pasos a seguir para crear una reserva, pero si el usuario es despistado, quizás no se da cuenta de dicho paso, entonces la idea sería forzar la página web a colocar-se a la altura de este último paso.

10.2 Línea futura

La función a la cual le veo aun que le queda mucho para crecer es la opción del mapa grafico del restaurante. Para el tiempo y el conocimiento que tengo estoy contento de como ha quedado, pero sé que si tuviera un conocimiento mucho más amplio de CSS podría hacer que una imagen fuese todo el mapo grafico de la web. Debería de investigar para llevar a cabo esta idea, pero creo que, con esta idea aplicada a la web actual, ganaría muchísimo más.

11 ANEXOS

11.1 Product Backlog

Versión Item 1

Version Item 2 / Final

Funcionalidad	Descripción
Sign up	Poder registrar-te utilizando nombre, DNI, email i password.
Log in/ Log out	Poder iniciar sesión con el email i password utilizado para el registro.
Connectar	Poder conectar la base de datos con los datos puestos por el cliente.
Contacto	Poder saber con un vistazo rápido donde está el restaurante, como llegar, horario, etc..
Funcionalidad	Poder hacer que la web sea lo más funcional posible y lo más orientativa posible
Responsive	Poder poner la web que sea para todos los dispositivos
Posibilidad de Eliminar Reserva	Poder dar la opción al usuario registrado de poder eliminar una reserva que previamente habrá realizado de forma sencilla.
Reservar	Poder reservar mediante un gráfico del restaurante.
Diseño de pagina	Poder un diseño de página atractivo a la vista
Opiniones	Poder poner opiniones de otros clientes .
Posibilidad de contestar valoraciones	EL hecho de que el administrador pueda contestar valoraciones/opiniones de los clientes para tener una relación más directa.
Formulario	Poder poner un formulario para consultas donde el cliente deje un comentario y el administrado lo vea y le pueda contestar.
Gestionar promociones	El administrador podrá variar las promociones de su sitio web dependiendo de su criterio
Logear con Google	Poder iniciar sesión con la cuenta de Google.
Conectar web	Poder conectar esta web con diferentes redes sociales.

Funcionalidad	Descripción
* Registrarse / Darse de Baja	El usuario podrá registrarse para poder acceder a las diferentes funciones de la página web. Con poner su email, su password (que tendrá que ser su id de google, ya que habrá la opción de vincularla con la de google) y su DNI. El usuario podrá darse de baja de mane
* Log in/ Log out	El usuario podrá iniciar sesión al entrar a la web , y cuando el usuario quiera desconectar-se , podrá hacerlo de manera fácil y sencilla.
* Enlazar informacio del user con mi bbdd	La información puesta por el usuario se lanzara a la base de datos en la cual se almacenara para que cada vez que quiera iniciar sesión lo haga sin problemas
* Crear reservar	El usuario que este reservado podrá reservar mesa gráficamente. Es decir, en la web tendrá el diagrama del restaurante para poder elegir una mesa específica. El usuario podrá elegir a través de un desplegable de calendario el día y la hora a través también de otro desplegable a la hora que elija mesa, así como las personas de que van a ser en esa mesa.
* Gestion de reserva: cliente	El cliente una vez tenga creada su reserva, la podrá modificar de la siguiente manera: podrá cambiar la fecha de reserva, así como la hora de reserva. Además podrá borrar una reserva.
* Gestion de reserva: Administrador	El administrador podrá ver a través de la página web todas las reservas de todos los clientes para poder gestionar y reservar esas mesas en el restaurante.
* Responsive	Poder poner la web que sea para todos los dispositivos
* Consultas / Dudas / Valoraciones	El usuario podrá a través de un formulario hacer una consulta, una duda o una valoración la cual se mandara a un email que solo podrá ver el administrador del sitio web.
* Opiniones	Los usuarios podran dejar sus opiniones con un texto y una puntuacion en un apartado de una la página web la cual podrán ver todas las personas que visiten la página del restaurante. El usuario podrá crear y borrar sus opiniones
* Diseño de pagina	Poder un diseño de página atractivo a la vista, que sea dinámica y atractiva a primera vista para el usuario
X Cambiar contraseña Usuario / Olvide mi contraseña	El usuario podrá cambiar su password de manera sencilla poniendo su password antigua por su password nueva. Además, tendrá la opción de que el usuario entre en la opción de olvide su contraseña para que en un pequeño formulario ponga su email y que le envíen un correo con su nueva contraseña temporal
X Gestionar promociones	El administrador podrá variar las promociones de su sitio web dependiendo de su criterio
X Logear con Google	Poder iniciar sesión con la cuenta de Google.
* Conectar web	Poder conectar esta web con diferentes redes sociales.

Marcado con una * los requisitos que si tengo realizados y marcado con una x los requisitos que no tengo realizados en el segundo product backlog ya que es el definitivo.

11.2 Herramientas

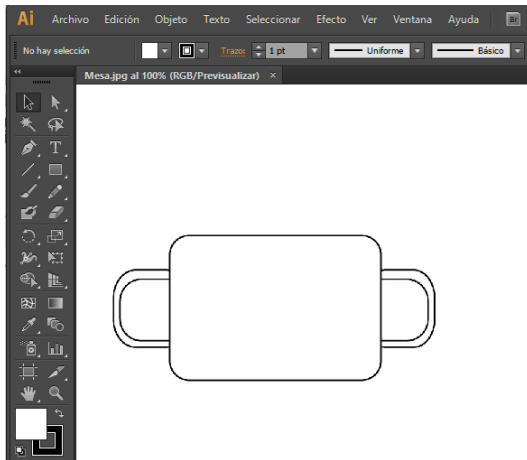
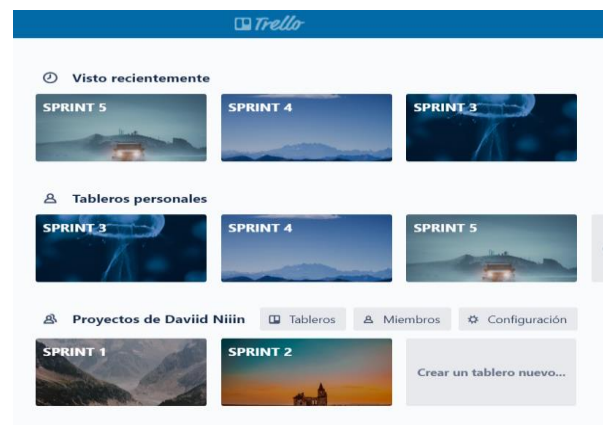


Imagen representativa de Illustrator, tuve un pequeño percance ya que cree las mesas en dos sesiones, la primera sesión fue de prueba así que cree dos mesas como la de la foto, y cuando todo funcionaba y fui a crear las demás, Illustrator había expirado y tenía que comprar versión.

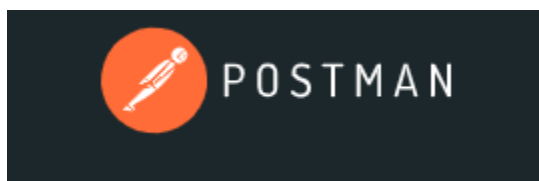
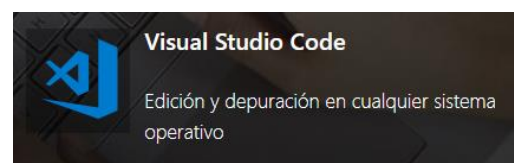
Fui listo y me descargué una pirata para poder seguir trabajando.

Imagen actual del Trello, en donde podéis ver que tengo 5 sprints. Tengo que añadir que la metodología tradicional de Trello no la seguí a raja tabla e hice un Trello personalizado a mi gusto



11.3 Entorno de Desarrollo

Este fue mi IDE durante la mayoría de tiempo de creación de proyecto.



Y este fue mi “Amigo” para poder comunicarme con la base de datos y saber que lo que estaba haciendo estaba bien, ¡así que gracias Postman!!

11.4 Controlador de Versiones

davidnin / proyectoFinCurso
Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

No description, website, or topics provided.

Manage topics

28 commits 2 branches 0 releases 1 contributor

⚠ We found a potential security vulnerability in one of your dependencies.

Only the owner of this repository can see this message.

[Manage your notification settings](#) or [learn more about vulnerability alerts](#).

See security alert

Branch: master New pull request
Create new file Upload files Find File Clone or download

This branch is 21 commits ahead of develop.

Pull request Compare

davidnin All works correctly, i have to do de desing of footer and home page ...
Latest commit 1df4362 6 days ago

API Commit secure 15 days ago
cliente All works correctly, i have to do de desing of footer and home page ... 6 days ago
.gitignore Initial commit with API of back almost finished 5 months ago

Help people interested in this repository understand your project by adding a README.

Add a README

Este sería mi GitHub tal y como os había explicado antes, y aquí abajo tendréis capturas de algunos de los commits que he ido haciendo a lo largo del proyecto.

```

commit 1df4362e873f91ea7a9764f1571d80c7895a3b2d (HEAD -> master, origin/master)
Author: David nin <daviidnini@gmail.com>
Date: Sun May 5 20:38:14 2019 +0100

    All works correctly, i have to do de desing of footer and home page and i finish the project

commit 2757541b0730c50df0ad890fd4802b3d53ad1a7c
Author: David nin <daviidnini@gmail.com>
Date: Fri May 3 16:45:34 2019 +0100

    Change the desing of all web, finished the desing of cart

commit 4aea3a323eb9c32d68b5a7120b63c802a72d32c4
Author: David nin <daviidnini@gmail.com>
Date: Sat Apr 27 17:43:16 2019 +0100

    Checking error of page Reserved

commit 126d6190ab178ceda4c1334feb56909dfff9c510
Author: David nin <daviidnini@gmail.com>
Date: Fri Apr 26 14:30:23 2019 +0100

    Commit secure

commit dcd85abe26b22d9d9375d5121163b103df82f76
Author: David nin <daviidnini@gmail.com>
Date: Fri Apr 19 12:01:12 2019 +0100

    Contact, not-found and profile have css

commit 840f924e619f8ebba0b2c98753ed8fa6864eb6af
Author: David nin <daviidnini@gmail.com>
Date: Wed Apr 17 17:39:31 2019 +0100

    Created a coments for our web, user can comment her opinion about us, then, aplying css on home, navbar, footer, login and profile

commit d9c6e34e8769502747b9668ac8902b10d81ef8bd
Author: David nin <daviidnini@gmail.com>
Date: Mon Apr 15 18:12:40 2019 +0100

    Created in the bbdd one thing that is comments

commit 707ea8fcb40d317ddc58cd9469cb8989e5fbfe68
Author: David nin <daviidnini@gmail.com>
Date: Mon Apr 15 15:33:31 2019 +0100

    Continue doing css and change many things

commit 6546671fda88a9cb98a8b0959aca268c7d3e555
Author: David nin <daviidnini@gmail.com>
Date: Sat Apr 13 15:40:47 2019 +0100

    save commit

commit 0d8b6fc00e0f37e87050c797c2c3bba0d20b529
Author: davud <daviidnini@gmail.com>
Date: Sat Feb 16 20:23:24 2019 +0100

    Doit user modification and error CORS finally solved :)

commit 65d42c17ad00e1bb3f4b3367ccc4f5fa852a23d3
Author: davud <daviidnini@gmail.com>
Date: Fri Feb 15 17:48:14 2019 +0100

    Tu madre

commit 9c4061fb258291a267aed3003730080cb13157f5 (origin/develop)
Author: davud <daviidnini@gmail.com>
Date: Sun Feb 10 17:43:57 2019 +0100

    Solucionated almost all of the errors of home-page and login-page

commit a789574c7148391bd62b8c8caf4d42615e23aac0
Author: davud <daviidnini@gmail.com>
Date: Thu Feb 7 16:14:49 2019 +0100

    The change are correct, now i have the login in another html page and now, i can do the home page and her css correctly

commit 938123c73c6520dc4607e8a3948369d6534elf63
Author: davud <daviidnini@gmail.com>
Date: Wed Feb 6 17:17:32 2019 +0100

    Save before a important change, the login change the name of her component, app to login, app its free now

commit f1a9cf2762220ae705ed6bac0987794f83ad7ba
Author: davud <daviidnini@gmail.com>
Date: Sat Feb 2 15:39:41 2019 +0100

    Doing the css of home page, home page doit but not functionally

commit feab78e4a955e0db11124b40c02d3bdc00b6fd16
Author: davud <daviidnini@gmail.com>
Date: Sat Feb 2 13:11:59 2019 +0100

    Finished API Node and doing the home page in angular

commit fd7cac2c74fc5853bc6825ad499c51c40541d96e
Author: davud <daviidnini@gmail.com>
Date: Fri Dec 28 16:11:40 2018 +0100

    Thinking that i finished the back-end

commit 06ae2de3bdblbf38aec140428cc17c40099b427
Author: davud <daviidnini@gmail.com>
Date: Mon Dec 24 16:45:05 2018 +0100

    Initial commit with API of back almost finished
  
```

11.5 Cuerpo de la web

[Nuestra Carta](#) [Reservar](#) [Contacto](#)

Nuestra Carta

A continuación, podrán averiguar la nueva experiencia gastronómica que ofrece nuestro icónico restaurante Chumy Churry. Podrá degustar desde la comida más típica mediterránea como las delicias mexicanas siempre elaboradas y preparadas por nuestros expertos cocineros tanto en gastronomía mediterránea como en gastronomía mexicana.

Pica Pica	Ensaladas
Pan de Coca	Enrollados de tortita
Biquinis Classicos	Biquini Tostado
Pan de Flauta	Xapata Redondo

Depende de la selección de la barra de navegación, se mostrará un contenido u otro. Como se puede observar en la captura de arriba, al clicar sobre nuestra carta, se nos mostrara una introducción hacia la carta del chumi churri aparte de la carta (escondida en estas cartas). En la captura de abajo se puede observar como si elegimos la sección de contacto, se nos mostrara información sobre el restaurante.

[Reservar](#) [Contacto](#)

Contacto

Av. de Francesc Macià, 39-41
08208 Sabadell, Barcelona

chumychurry@gmail.com

617 15 32 98

Lunes a domingo
11:30 - 01:00 h.

11.6 Registro / Inicio Sesión

```
function saveUser(req,res){
  var user = new User();
  var params = req.body;
  console.log(params)
  user.name= params.name;
  user.lastname= params.lastname;
  user.email= params.email;
  user.role= 'ROLE-USER';
  user.table='null';

  if(params.password){
    bcrypt.hash(params.password,null,null,function(err,hash){
      user.password=hash;
      if(user.name != null && user.lastname != null && user.email != null){
        user.save((err,userStored) => {
          if(err){
            res.status(500).send({message: 'Error al guardar el usuario'});
          }else{
            if(!userStored){
              res.status(404).send({message: 'No se ha registrado el usuario'});
            }else{
              // ...
            }
          }
        });
      }
    });
  }
}
```

En la captura siguiente, podemos observar cómo se encripta la contraseña una vez le llega a esta función.

```
function loginUser(req,res){
  var params = req.body;

  var email = params.email;
  var password = params.password;

  User.findOne({email: email.toLowerCase()},{err,user}=>{
    if(err){
      res.status(500).send({message: 'Error en la petición'});
    }else{
      if(!user){
        res.status(404).send({message: 'El usuario no existe'});
      }else{
        bcrypt.compare(password, user.password, function(err,check){
          if(check){
            if(params.getHash){
              res.status(200).send({
                token: jwt.createToken(user),
                user: user
              });
            }else{
              res.status(200).send({user});
            }
          }
        });
      }
    }
  });
}
```

Aquí podemos observar como una vez el usuario intenta identificarse, antes de darle una respuesta, desencriptamos la password y la comparamos con la introducida

11.7 Comentar

```
function getComments(req, resp) {
  var comments = {};
  console.log(resp.status);
  Comment.find(function (err, comment) {
    comments[comment._id] = comment;

    if (err) {
      resp.status(500).send({ message: "Error en la peticion" });
    } else {
      if (!comments) {
        resp.status(404).send({ message: "No hay Mesas!!" });
      } else {
        resp.status(200).send({ comments });
      }
    }
  });
}
```

Funcion para obtener todos los comentarios que luego se mostraran en la seccion de *el restaurante*. Aquí lo que se llama es una llamada a mongoDB para obtener todos los comentarios

11.8 TripAdvisor



```
<iframe
  src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d2985.6127221705347!2d
  .0948663152952434!3d41.55598187924902!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0
  x12a494e5e973ae33%3A0x51246798c81ba32a!2sChimi+Churri+Sabadell!15e0!3m2!1ses!2ses!4v1556900085975!5m2!1ses!2ses"
  id="mapa" height="450" frameborder="0" style="border:0" allowfullscreen></iframe>
```

RESULTADO- IFRAME

SCRIPT

```
<script async
  src="https://www.js-cache.com/wejs?wtype=cswritereviewlg&uniq=219&locationId=4063527&lang=es&lang=es&display_version=2"
  data-loadtrk onload="this.loadtrk=true"></script>
```


11.9 Error Reservas

```
ngOnInit() {
  this.cargarCalendario()
}

//Funcion para cargar el datepicker sin necesidad de recargar pagina
cargarCalendario() {
  var dateToday = new Date();
  $(function () {
    $('#datepicker').datepicker({
      numberOfMonths: 1, showButtonPanel: true, minDate: dateToday, closeText: 'Cerrar',
      prevText: '< Ant',
      nextText: 'Sig >',
      currentText: 'Hoy',
      monthNames: ['Enero', 'Febrero', 'Marzo', 'Abril', 'Mayo', 'Junio', 'Julio', 'Agosto', 'Septiembre', 'Octubre', 'Noviembre', 'Diciembre'],
      monthNamesShort: ['Ene', 'Feb', 'Mar', 'Abr', 'May', 'Jun', 'Jul', 'Ago', 'Sep', 'Oct', 'Nov', 'Dic'],
      dayNames: ['Domingo', 'Lunes', 'Martes', 'Miércoles', 'Jueves', 'Viernes', 'Sábado'],
      dayNamesShort: ['Dom', 'Lun', 'Mar', 'Mié', 'Juv', 'Vie', 'Sáb'],
      dayNamesMin: ['Do', 'Lu', 'Ma', 'Mi', 'Ju', 'Vi', 'Sá'],
      weekHeader: 'Sm',
      dateFormat: 'dd/mm/yy',
      firstDay: 1,
      isRTL: false,
      showMonthAfterYear: false,
      yearSuffix: ''
    });
    $('#datepicker').datepicker("option", "dateFormat", "yy-mm-dd");
    $('#datepicker').datepicker("option", "showAnim", "slideDown");
    $('[type="text"]').change(function () {
      var date = $(this).datepicker("getDate");
      console.log("Funcion de script: " + date);
      localStorage.setItem('fecha', JSON.stringify(date));
    });
  });
}
```

Aquí se puede ver finalmente como ha quedado el metodo de generar el datapicker, con todas sus configuraciones.

```
146 //si el usuario no mete ninguna fecha, se le assignara el dia actual automaticamente
147 for (var i = 0, length = radios.length; i < length; i++) {
148   if (radios[i].checked) {
149     this.turno = radios[i].value;
150     break;
151   }
152 }
153 if (!this.turno) {
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 2: node

ERROR in src/app/components/reserva/reserva.component.ts(148,21): error TS2339: Property 'checked' does not exist on type 'HTMLElement'.
src/app/components/reserva/reserva.component.ts(149,32): error TS2339: Property 'value' does not exist on type 'HTMLElement'.

Esta es parte del método de obtener el turno escogido que peta y sus líneas de error.

11.10 Error Identificación

```
for (var i = 0, length = this.usuarios.length; i < length; i++) {
  if (this.usuarios[i].email == this.user_register.email) {
    this.usuarioExiste = "El registro no se ha podido realizar debido a que el email que a introducido ya existe."
  }
  console.log("hace el bucle")
}
```

Con esta pequeña funcion compruebo si el email que ha introducido el usuario existe o esta disponible.

11.11 RoadMap

SPRINT	SUBTAREAS	FECHA INICIO	FECHA FIN
SPRINT 1 INVESTIGACION EN NUEVAS TECNOLOGIAS	Investigación React Investigación Node/node Express Investigación Mongo Investigación surge / heroku	10-11-2018	02-12-2018
SPRINT 2 PREPARACIÓN DE LA INFRAESTRUCTURA DE LA APLICACIÓN WEB	Instalación y configuración de servidor Instalación y configuración SGBD Instalación entorno de desarrollo Configuración Github Configuración Trello	03-12-2018	09-12-2018
SPRINT 3 EMPEZAR LA BASE DE LA APLICACIÓN WEB	Empezar a programar Back-End y poder estructurar la base, el cual es todo aquel punto rojo del punto 1. Release Product Backlog.	10-12-2018	06-01-2019
SPRINT 4 DIBUJAR PLANO RESTAURANTE CON CANVAS VIA JAVASCRIPT	Dibujar y poder manipular al detalle todo el mapeado del restaurante	07-01-2019	20-01-2019
SPRINT 5 DESARROLLO FRONT-END	Dejar terminado el desarrollo del front-end para poder comprobar que todo funcione	21-01-2019	10-02-2019
SPRINT 6 TESTING	Testear que todo el back-end y el front-end funcione correctamente	11-02-2019	24-02-2019
SPRINT 7 DESARROLLO CSS	Dejar enmaquetada y bonita la pagina web para el cliente	25-02-2019	03-03-2019
SPRINT8 REALIZAR TAREAS ADICIONALE	Una vez tenga la pagina web sencilla , comenzare a introducir programas adicionales. Aquí me refiero a poder hacer todo aquello en verde del Release Product Baclog	04-03-2019	17-03-2019
SPRINT 9 TESTING FINAL	Reparar errores , mejorar web	Resto del	Tiempo