

Universidad de Zaragoza
Sistemas de información
Proyecto de prácticas - Grupo B04, Viernes 12-14

Práctica 2

APPETIZR

848884 -Herce Cruz, Daniel
816787 -Villagrasa Labrador,Alain
844665 -Cauhé Viñao, Lucas



Índice

1. Resumen.....	3
2. Funcionalidades diseñadas.....	4
3. Análisis del modelo de negocio.....	6
4. Herramientas utilizadas.....	7
5. Recursos a utilizar.....	8
6. Storyboard.....	9
7. Bibliografía.....	11

1. Resumen

Para esta práctica se van a analizar y diseñar 2 aspectos cruciales para la aplicación como son el prototipo de la interfaz web junto con su mapa de navegación mediante el uso de un “Storyboard”, y la infraestructura que se pretende implantar con la que desarrollar la aplicación.

Primero se determinarán cuáles se consideran las funcionalidades fundamentales del proyecto mediante una tabla de requisitos aplicando el método MoSCoW. Los requisitos se han definido siguiendo la filosofía planteada en la práctica anterior, en la que se explicaban de forma más informal las necesidades que iba a intentar cubrir nuestro proyecto, pero ahora de forma más precisa y concreta, con un enfoque más formal.

Estos requisitos harán que se pueda llevar a cabo el modelo de negocio descrito, que especifica de forma resumida cuál va a ser la forma principal y las secundarias de generar beneficios con nuestro sistema de información. Este sólido modelo proporciona una primera aproximación al desarrollo futuro del proyecto que se podría llevar a cabo por una empresa real.

A continuación se comentan las diferentes tecnologías que vamos a utilizar para implementar el sistema, tanto en la capa vista como en el backend y la base de datos.

Finalmente, se presenta un primer prototipo de la interfaz de usuario mediante un storyboard, con los hilos de navegación principales que se podrán encontrar en la aplicación web una vez esté terminada. Se comenta además la plataforma utilizada para desarrollarla, intentando aplicar los principios de diseño de interfaces para conseguir una interfaz sencilla, agradable para el usuario, pero sobre todo usable y accesible.

2. Funcionalidades diseñadas

Presentamos a continuación más en detalle las funcionalidades que va a ofrecer nuestro sistema, así como algunas que se implementarán en función el tiempo disponible, y otras que no se van a implementar en este proyecto, pero que en un futuro se podrían añadir al sistema y mejorarlo o aumentar su valor como producto. Esto se visualiza mediante el método MoSCoW en la última columna.

Nº	Requisito	Prioridad
F1	El sistema debe permitir al propietario de un restaurante crear posts con el contenido que considere.	M
F2	El sistema debe ofrecer foros para los usuarios según las distintas categorías de restaurantes.	M
F3	El sistema debe permitir a los usuarios comentar en estos foros, así como reaccionar positivamente o negativamente a los comentarios de otros usuarios.	M
F4	El sistema debe permitir añadir, modificar y consultar la información de un restaurante en concreto, como su localización, horario y menú.	M
F5	El sistema debe permitir, si el propietario del restaurante lo desea, realizar pedidos a través de la aplicación <i>EINAEATS</i> , desarrollada por el equipo B05.	M
F6	El sistema debe permitir a los propietario de restaurantes consultar métricas básicas sobre su página: cuántas visitas ha recibido, cuantos pedidos, etc	M
F7	El sistema debe permitir la eliminación de posts previamente creados.	M
F8	El sistema debe permitir a los usuarios hacer comentarios sobre los restaurantes y platos concretos de un restaurante.	M
F9	El sistema debe permitir consultar los comentarios sobre los restaurantes y platos concretos de un restaurante.	M
F10	El sistema debe permitir buscar un restaurante por su nombre o categoría.	M



F11	El sistema debe recomendar a los usuarios de tipo restaurante que les pueden interesar en función de sus consultas, comentarios y pedidos anteriores.	C
F12	El sistema debe permitir la consulta de estadísticas avanzadas sobre el restaurante, mostrando gráficas y recomendaciones al propietario del restaurante sobre cómo mejorar sus métricas.	C
F13	El sistema debe adaptar todo su contenido (restaurantes, foros, etc) en función de la localización del usuario.	W
F14	El sistema debe permitir explorar los restaurantes cercanos a la ubicación del usuario.	W
NF1	El sistema requiere conexión a internet en todo momento.	-
NF2	El sistema debe cumplir con la Ley de Protección de Datos y el resto de los requisitos legales pertinentes.	-
NF3	El sistema debe ser simple, intuitivo y accesible.	-
NF4	El sistema debe ser compatible con los navegadores web más utilizados (Chrome, Firefox, Safari)	-
NF5	La aplicación debe ser eficiente en términos de uso de recursos y tiempos de respuesta	-
NF6	El sistema debe garantizar la consistencia y durabilidad de la información entre todos los usuarios que acceden a ella, sean usuarios de la aplicación o agentes externos	-

3. Análisis del modelo de negocio

El problema que resuelve nuestra aplicación es evitar toda la complejidad asociada a la consulta de opiniones e información real sobre un establecimiento de restauración. Para ello evitamos que el usuario tenga que visitar distintos sitios web con información muy heterogénea al aunar las principales funcionalidades para comentar sobre un restaurante concreto o un tema gastronómico específico en un único sitio. De esta forma el usuario dispondrá de una forma de buscar restaurantes cerca de su ubicación, ver los comentarios de un restaurante o un plato específico y la posibilidad de ver recomendaciones que otros usuarios hagan de forma rápida y sencilla.

Para implementarlo, se va a desarrollar una aplicación web que en un futuro sería posible extender a una aplicación móvil, llegando a más usuarios y añadiendo algunas funcionalidades comentadas en el apartado anterior.

La aplicación dispone de varias vías para ser rentable, siendo la monetización del tráfico de la aplicación nuestra principal fuente de ingresos. En un principio se ha planteado la implementación de los siguientes métodos de monetización:

Afiliación con EinaEats: se cobrará un porcentaje por cada compra realizada en su aplicación por un usuario proveniente de la nuestra.

Restaurante patrocinado: se cobrará a los restaurantes que quieran aparecer en la sección de restaurante patrocinado, dicha sección aporta gran visibilidad al mismo ya que aparece con un gran tamaño al principio de la página de explorar, donde aparece toda su competencia.

Anuncios: se insertarán anuncios en los foros de discusión de manera que entre los mensajes de los usuarios aparezcan anuncios de artículos de cocina u otros restaurantes que se quieran promocionar en nuestra aplicación. Esto se hará siguiendo un equilibrio entre el contenido que generan los usuarios y el publicitario para asegurar que no se sobrecarga un espacio inicialmente pensado para el usuario con publicidad.

De esta forma, conseguimos monetizar el tráfico en nuestra aplicación sin tener que recurrir a modos de monetización más agresivos como un pago por registro en la aplicación o un pago por usar alguna funcionalidad. Sin embargo sí que se contempla la posibilidad de que en un futuro los restaurantes puedan tener un plan de pago que les dé acceso a más estadísticas sobre su negocio.



4. Herramientas utilizadas

Para el desarrollo del prototipo se ha utilizado la plataforma online Figma, que además nos permite exportar los diseños (aunque con algunas dificultades y con poca precisión) directamente a HTML y CSS. Se han utilizado algunos recursos disponibles de forma libre en el portal comunitario de la propia plataforma, que se han ido modificando y adaptando para conseguir el diseño deseado. Aunque es una tarea que lleva más trabajo, consideramos que tener un primer prototipo más cercano a la interfaz real hará en un futuro que el desarrollo sea más sencillo.

Por otra parte, se ha procurado seguir los principios de diseño de interfaces como pueden ser las 8 Reglas de Oro de Ben Shneiderman, las 10 Heurísticas de Nielsen o los Principios de Diseño Universal, intentando conseguir una interfaz de usuario que sea fácil de aprender, accesible y sobre todo que sea fácil de usar.

5. Recursos a utilizar

Para implementar la aplicación web, haremos uso de distintas tecnologías comunes y ampliamente utilizadas para este tipo de aplicaciones. Para brindar una mejor experiencia al usuario y facilitar el desarrollo, dividimos el sistema en el frontend (interfaz de usuario, la parte con la que realmente se interactúa) y backend (encargada de la carga de trabajo, comunicación con la base de datos, etc).

En el frontend haremos uso de HTML 5 Y CSS 3, así como algunos elementos que permitan incluir funcionalidades típicas de la Web 2.0.

Como herramienta de desarrollo se han barajado varias posibilidades: en primer lugar se pensó en emplear alguno de los frameworks utilizados actualmente de Javascript, aunque se desechó dado su extraño comportamiento en algunas situaciones. El siguiente considerado fue Django pero dado que su uso está enfocado a aplicaciones con manipulación intensiva de datos y la curva de aprendizaje es pronunciada, finalmente hemos optado por Ruby on Rails. Consideramos que al ser un proyecto *pequeño*, este framework nos ayudará con sus plantillas preexistentes, y los módulos que ofrece, además de facilitar la comunicación con la base de datos.

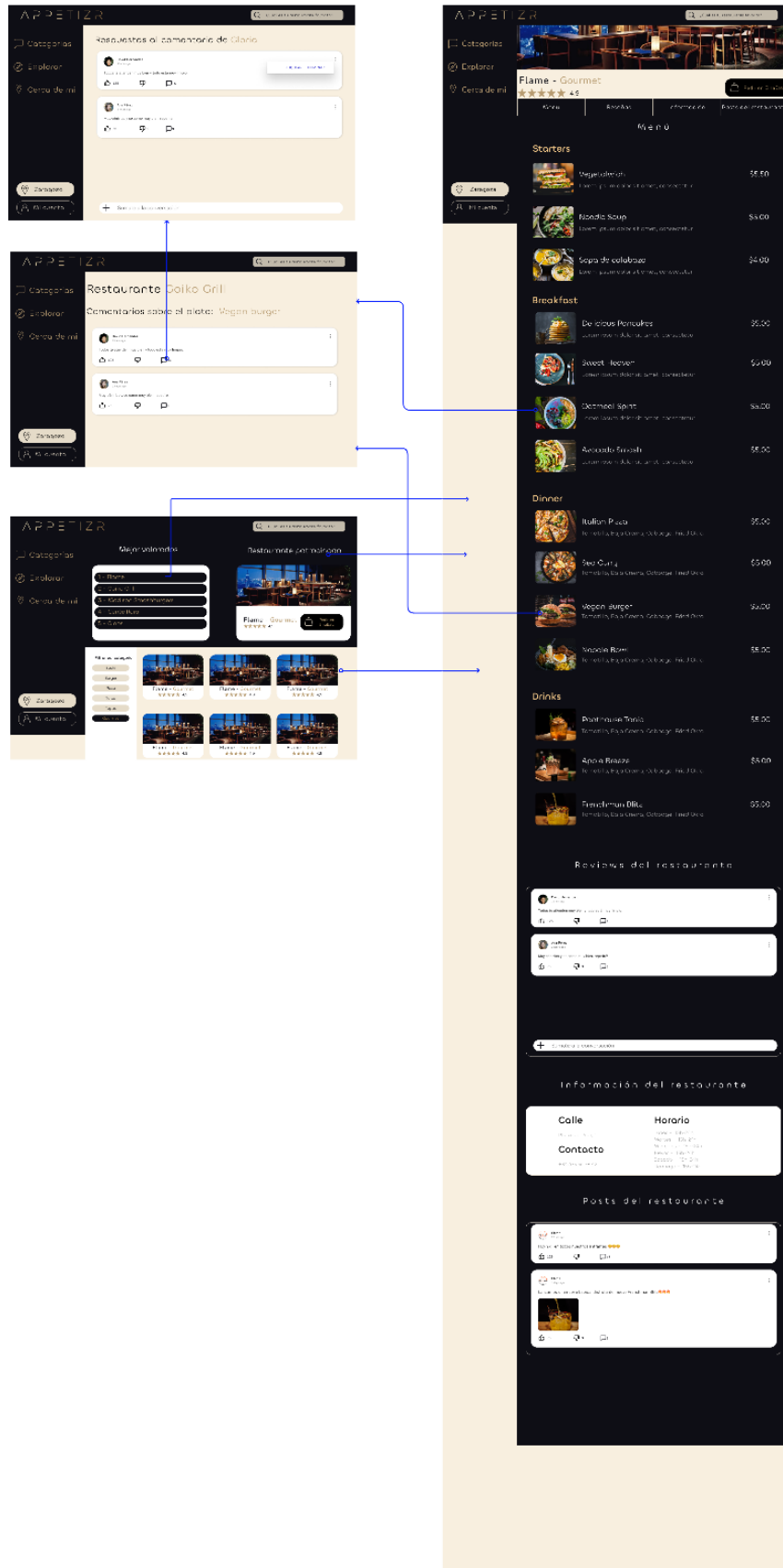
En cuanto a la base de datos, se ha elegido por su sencillez, facilidad de uso y por recomendación de los profesores de la asignatura, PostgreSQL.

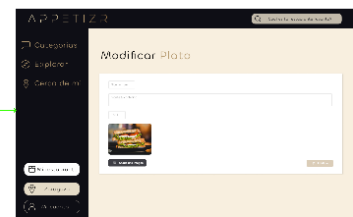
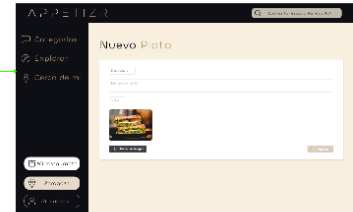
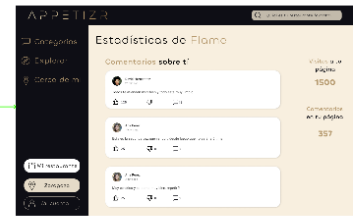
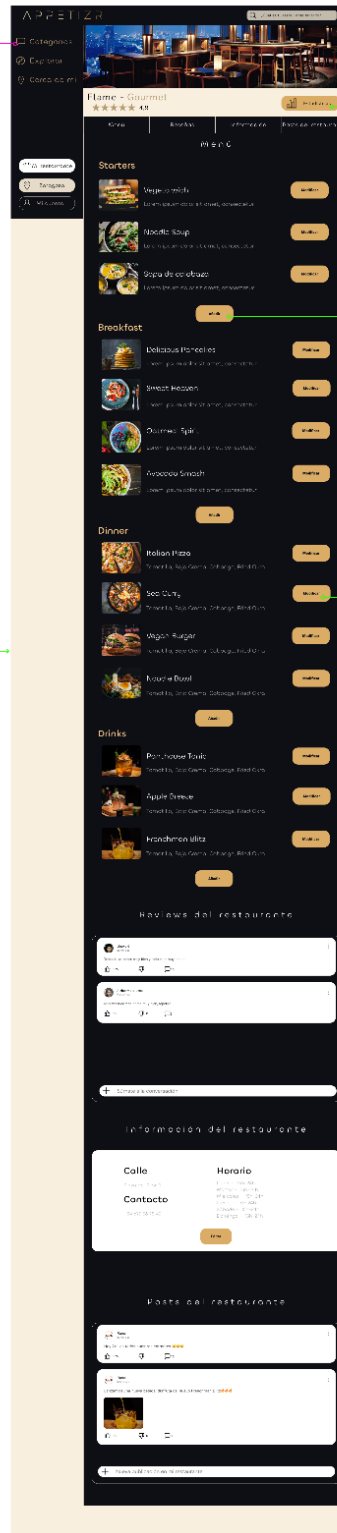
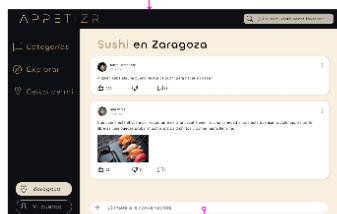
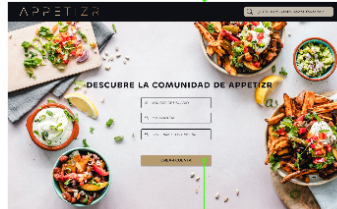
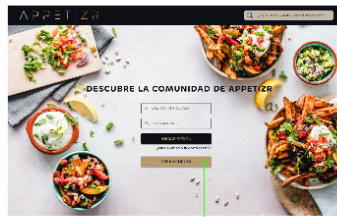
Para lanzar y probar toda la infraestructura, se hará uso de Docker, donde se mantendrán tanto la aplicación como la base de datos.

Para servir la página web, se empleará Apache ya que ofrece una rápida instalación (sobretudo con Docker) y uso para publicar los contenidos estáticos de nuestra aplicación web.

Por otra parte, internamente utilizamos git & github para el control de versiones y desarrollo de la aplicación en paralelo, facilitando el trabajo en grupo. Esta plataforma utilizada prácticamente en toda la industria del desarrollo software nos proporcionará las herramientas necesarias para poder desarrollar sin tener que preocuparnos de problemas de versiones o coordinación.

6. Storyboard







7. Bibliografía

- *Aglowiditsolutions*, [en línea] [consulta: 1 / 10 / 2023] Disponible en:
<https://aglowiditsolutions.com/blog/ruby-on-rails-vs-django/>
- *Scaler*, [en línea] [consulta: 1 / 10 / 2023] Disponible en:
<https://www.scaler.com/topics/ruby-on-rails-vs-django/>