

Victoria Almeida Calderón

**INDICE**

**1. PRESENTACIÓN**

**1.1. Estado del arte**

**1.2. Objetivo y alcance previstos**

**2. ESTUDIO DE VIABILIDAD**

**2.1. Introducción**

**2.2. Objeto**

**2.2.1. Situación del estudio**

**2.2.2. Perfil de los usuarios del proyecto**

**2.2.3. Objetivos**

**2.3. Recursos**

**2.3.1. Recursos de desarrollo**

**2.3.2. Planificación inicial**

**2.4. Conclusiones**

**3. ANÁLISIS DEL PROYECTO**

**3.1 Vistas de casos de uso**

**4. DISEÑO**

**4.1. Introducción**

**4.1.1. Selección del entorno de desarrollo**

**4.1.2. Selección de base de datos**

**4.1.3. Diagrama**

**4.2. Mi aplicación**

**4.3Líneas de investigación futuras.**

**4.4 Bibliografía**

Proyecto

**1. PRESENTACIÓN**

Este proyecto se trata de una web informativa de establecimientos de Ayamonte, en él los dueños de distintos locales podrán subir la información que ellos deseen para que así los usuarios de la web estén informados.

En este proyecto existe 3 roles de usuarios, administrador, dueño de establecimiento y usuario local, cada rol tiene su tarea como comentaremos posteriormente, pero el rol con más importancia en está web es el de dueño de establecimiento.

Inaya como tema general podemos decir que es una web informativa, pero si entramos en más profundidad, Inaya es una web publicitaría que mostrará la información de los establecimientos de los dueños de los locales,esto sera lo que proporcionará información a la web.

Como conclusión, Inaya es fundamentalmente una web que se encargará de publicitar los establecimientos de Ayamonte, mostrando la información que el dueño de dicho establecimiento desee, teniendo una pequeña configuración individual por cada categoría.

En Ayamonte sufrimos cambios constantes y no tenemos conocimiento de ello, abren y cierran locales constantemente, hay actividades y no somos informados, por ello elegí este proyecto donde en una única plataforma se nos podrá informar de todo esto. Mi motivación ha sido el estar informada de los cambios mencionados anteriormente y de igual manera informar a los demás usuarios que disfruten la plataforma.

Con lo explicado anteriormente la desinformación desde el punto de vista de usuario normal es negativa a nivel personal, ya sea porque se ha perdido una actividad que le gustaba, porque ha abierto un bar de su agrado, etc. Pero y ¿desde el punto de vista del propietario de un negocio? Ellos se llevan la peor parte ya que la desinformación de un negocio lleva a la poca clientela y por consiguiente a pérdidas económicas, lo que puede ocasionar el cierre del mismo.

Ayamonte es una ciudad turística con muchísimos negocios, por ello he hecho la web, para ayudar a los dueños de los negocios de ayamonte y a las personas que quieran estar informadas.

**1.1. Estado del arte**

Durante el proyecto he descubierto que hay varias páginas de información de Ayamonte, pero todas ellas pertenecen al ayuntamiento e informan sobre los distintos procesos que hay en este o distintas actividades a nivel local, estas webs aunque no tienen la misma finalidad que la mía son fundamentales para Ayamonte por lo que se han mencionado en el pie de página de mi web puesto que aunque la temática sea distinta la finalidad es la misma, Informar.

Estas webs informativas del Ayuntamiento llevarán la gestión de los procesos burocráticos del Municipio, ya sea el empadronamiento, la gestión de multas, etc.

**1.2. Objetivo y alcance previstos**

Mi principal objetivo es la configuración de establecimientos, esto permitirá que dependiendo de la categoría el establecimiento tenga una pequeña configuración personalizada. Los dueños de los establecimientos son los que rellenan esta información haciendo así la web un poco más dinámica e individual.

Aunque mi principal objetivo es la configuración de establecimientos, otros objetivos son la valoración de los usuarios logueados, estos podrán valorar el establecimiento de 1 a 5 estrellas, las configuración del administrador y la mensajería individualizada por rol.

**2. ESTUDIO DE VIABILIDAD**

**2.1. Introducción**

El objetivo del estudio de viabilidad es el análisis de un conjunto concreto de necesidades para proponer una solución a corto plazo, que tenga en cuenta restricciones económicas, técnicas, legales y operativas.

El estudio de viabilidad se orientará a la especificación de requisitos ya que costos, riesgos, problemas legales, etc., están fuera de este proyecto.

Durante el desarrollo de este capítulo se analizará el alcance de la necesidad planteada y se identifican las restricciones relativas a la sincronización con otros procesos, que puedan interferir en la planificación y futura puesta a punto del sistema objeto de estudio. Esta información se recoge en el apartado de requisitos.

**2.2. Objeto**

**2.2.1. Situación del estudio**

Después de haber hecho un estudio sobre mi aplicación, estas son las necesidades que debe cubrir:

| Información general y dinámica en el inicio | Apartado de mensajería | Estadísticas del dueño |
| --- | --- | --- |
| Estadísticas del administrador | Panel del administrador | Panel del dueño |
| Valoraciones de usuarios | Perfil | Apartado de categorías y establecimientos |

**2.2.2. Perfil de los usuarios del proyecto**

Como se explicará posteriormente, existen distintos tipos de usuarios, para los cuales el sistema ofrecerá un conjunto de funcionalidades específicas. Independientemente del tipo de usuario, el sistema será desarrollado con el objetivo de que sea fácil de utilizar, intuitivo y que no se requiere conocimiento previo por parte de los usuarios, para que sea accesible a cualquier tipo de persona, tenga conocimientos de las nuevas tecnologías o no.

**2.2.3. Objetivos**

A. Objetivo General

Desarrollar una aplicación informativa de ayamonte que permita a los dueños subir promocionar sus establecimientos y a los usuarios normales valorar dichos establecimientos.

B. Objetivos específicos

* Información general y dinámica en el inicio
* Apartado de mensajería
* Estadísticas del dueño
* Estadísticas del administrador
* Panel del administrador
* Panel del dueño
* Valoraciones de usuarios
* Perfil
* Apartado de categorías y establecimientos

**2.3. Recursos**

**2.3.1. Recursos de desarrollo**

Requisitos mínimos para el desarrollo de la aplicación.

Hardware:

* Sistema operativo de 64 bits, procesador basado en x64, AMD Ryzen 5 3400G with Radeon Vega Graphics 3.70 GHz, 16Gb memoria RAM, 3.5TB de disco duro
* Conexión a internet: 7.5Mb.

Software:

* Servidor Web: Apache.
* Entorno de desarrollo: Visual Studio Code.
* Base de datos: MySQL.
* Editor de base de datos: phpMyAdmin.
* Programación web: PHP, HTML, jQuery,js.
* Sistema Operativo: Windows 11/Windows 10
* Navegador: Internet Explore/Google Chrome
* Documentación del proyecto: word google

En el capítulo 4, se explicarán las tecnologías mencionadas en la lista previa

**2.3.2. Planificación inicial**

Comenzamos tomando una idea y la plasmamos brevemente en un anteproyecto que será aprobado o no a juicio de nuestros tutores, continuaremos creando un modelo entidad relación de la idea ya aprobada esta parte tomará algo más de tiempo con relación a lo anterior expuesto y con esta parte además iniciamos nuestro proyecto.

Tras haber llegado a una conclusión en la bd, comenzamos a planificar las distintas tareas que tendrá la aplicación y con esto iniciamos a codificar, para ello realizamos un estudio del lenguaje y de la distinta documentación que necesitemos para llevarlo a cabo.

Finalmente, una vez desarrollada la aplicación se llevarán a cabo las distintas pruebas para que por último se lleve al cierre del proyecto y a la defensa del mismo.

**2.4. Conclusiones**

Después de haber analizado el problema, ver qué soluciones existen en la actualidad exterior para solucionar este problema, podemos llegar a las siguientes conclusiones:

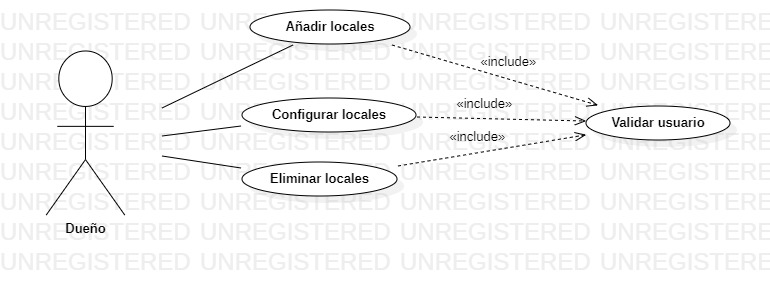
* No existe prácticamente en el mercado un software que se adapte a todas las necesidades buscadas.
* Los recursos para el desarrollo e implementación del software no son difíciles de conseguir, excepto las pantallas táctiles para la cual se propone como alternativa la utilización de ratones.
* Existe una programación la cual ayuda para llevar un seguimiento de las mismas. El modelo de desarrollo es idóneo para este tipo de proyectos.

Una vez expuestas las conclusiones, la última conclusión que se podría extraer es que este proyecto es “VIABLE”

**3. ANÁLISIS DEL PROYECTO**

**3.1 Vistas de casos de uso**

****

****

Explicación de los diagramas:

* Primer diagrama, diagrama del administrador: El administrador llevará a cabo la gestión de categorías, formatos, secciones y redes sociales.
* Segundo diagrama, diagrama del dueño: El dueño de un establecimiento llevará a cabo la gestión de establecimientos y dentro de está existirá un apartado de configuración que editará los datos de un establecimiento, añadirá imágenes al mismo, mostrará publicaciones y se llevará a cabo una configuración individual por categorias donde se podrá decidir que mostrar.
* Tercer diagrama, diagrama del usuario: El usuario podrá valorar los establecimientos.

**4. DISEÑO**

**4.1. Introducción**

En este capítulo, se tratará el diseño del sistema, es decir, se describirán las estructuras de la aplicación y con que se ha desarrollado.

**4.1.1. Selección del entorno de desarrollo**

A continuación se van a describirán aspectos importantes para el diseño del sistema.

* Entorno de desarrollo

Visual Studio Code: Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux. Viene con soporte integrado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes y tiempos de ejecución (como C++, C#, Java, Python, PHP, Go, .NET).

* Servidor web

Apache: es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.11 y la noción de sitio virtual. Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

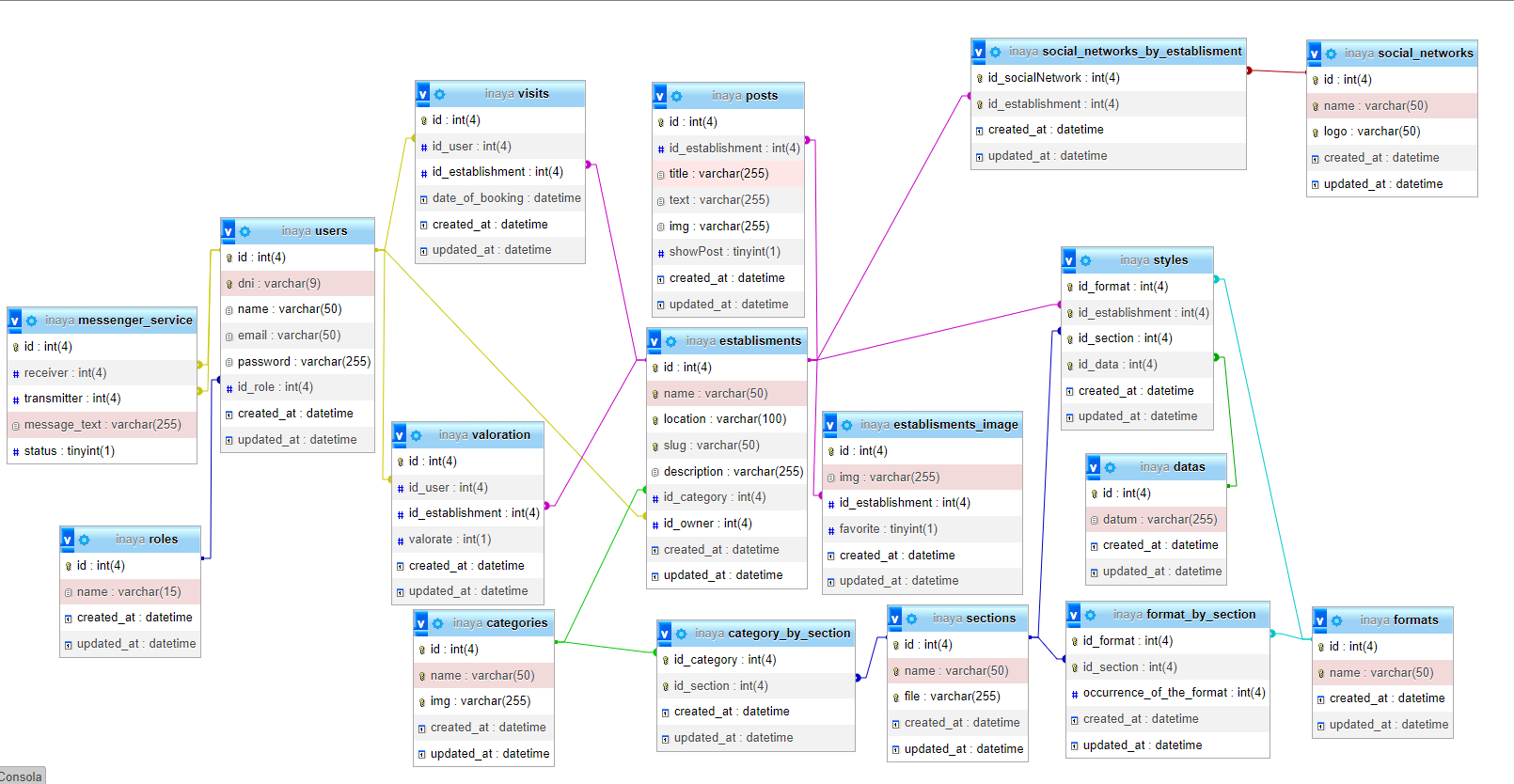
**4.1.2. Selección de base de datos**

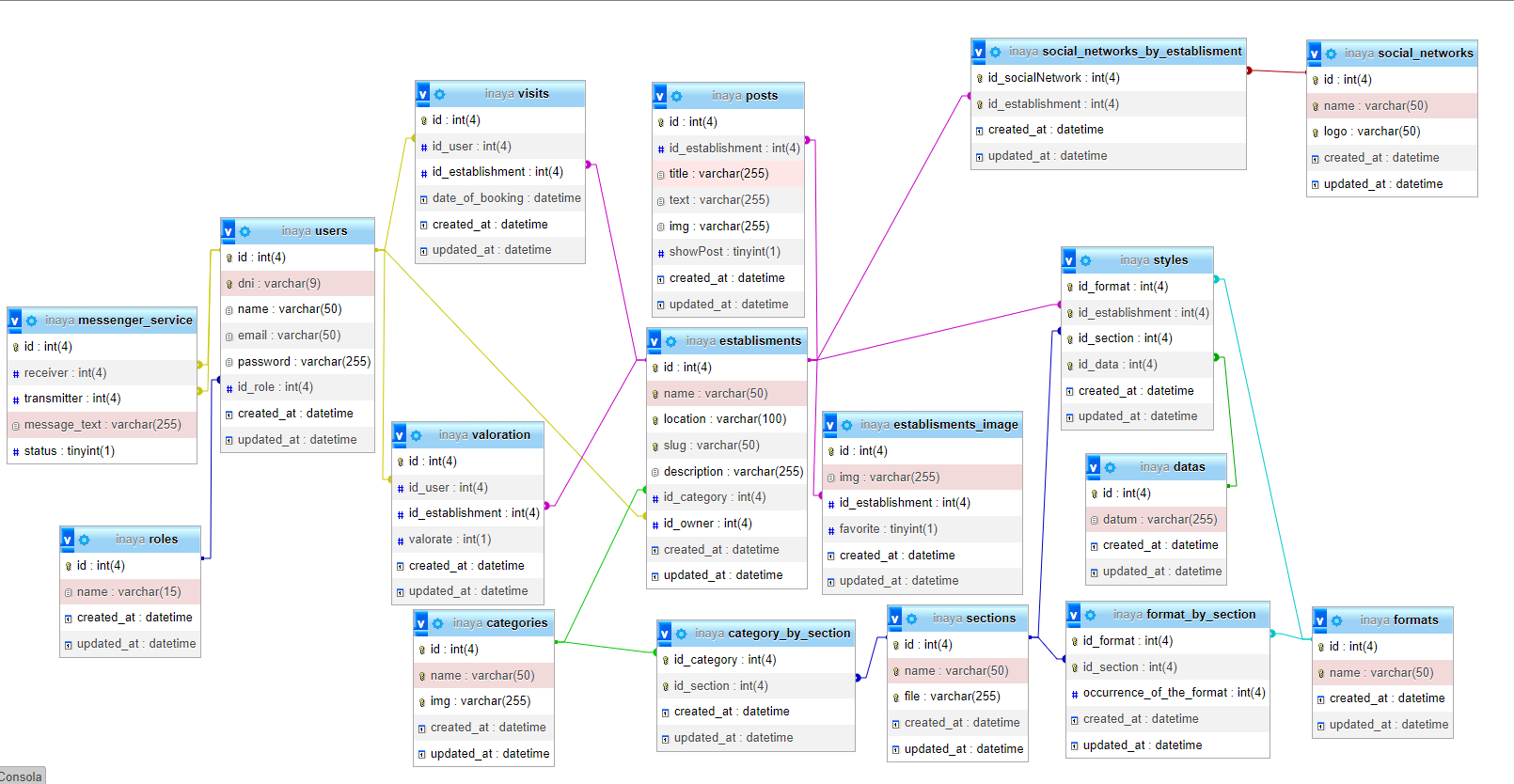
* Base de datos

MySQL: es un sistema de gestión de base de datos relacional, multihilo y multiusuario. Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

* Editor de la base de datos

phpMyAdmin: es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web. Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos. Se encuentra disponible bajo la licencia GPL.



**4.1.3. Diagrama**

Explicación del diagrama:

El diagrama está compuesto por distintas tablas unidas unas a otras, ahora desarrollaré los campos de cada tabla iniciando su explicación por la/las relacion/es de la misma.   
Tablas:

* roles: La tabla roles está relacionada a la tabla users con el campo id\_role que tiene la tabla users. Esta tabla está compuesta por los campos:
  + id
  + name
  + created\_at
  + updated\_at
* users: La tabla users está relacionada a las tablas role con el campo id\_role, con ella misma formando una nueva tabla, mensajería, cuyos campos relacionados son el receiver y el transmitter que se encuentran en la tabla messenger\_service, por último también está relacionada con la tabla establishment 3 veces, en una se enlaza con el campo id\_owner que se encuentra en la tabla establisments, en otra relación se forma la tabla valoration que se relacionarán con esta por el campo id\_user, en la última relación se forma la tabla visits que se relacionarán con esta por el campo id\_user. Esta tabla está compuesta por los campos:
  + id
  + dni
  + name
  + email
  + password
  + id\_role
  + created\_at
  + updated\_at
* messenger\_service: es una tabla que se forma por la relación de N:M de la tabla users con ella misma.
  + id
  + receiver
  + transmitter
  + message\_text
  + status
* valoration: es la tabla de N:M que se forma en la unión de la tabla users y establisments.
  + id
  + id\_user
  + id\_establishment
  + valorate
  + created\_at
  + updated\_at
* visits: es la tabla de N:M que se forma en la unión de la tabla users y establisments.
  + id
  + id\_user
  + id\_establishment
  + date\_of\_booking
  + created\_at
  + updated\_at
* categories: La tabla categories está relacionada a la tabla establisments con el campo id\_category que tiene la tabla establisments, además tambien esta relacionada a la tabla secciones con el campo id\_category que tiene esta. Esta tabla está compuesta por los campos:
  + id
  + name
  + img
  + created\_at
  + updated\_at
* posts: La tabla post esta relacionada con la tabla establisments por el campo id\_establishment que tiene la tabla post. Esta tabla está compuesta por los campos:
  + id
  + id\_establishment
  + title
  + text
  + img
  + showPost
  + created\_at
  + updated\_at
* establisments\_image: esta tabla está relacionada con la tabla establisments por el campo id\_establishment que tiene la tabla establisments\_image. Esta tabla está compuesta por los campos:
  + id
  + img
  + id\_establishments
  + favorite
  + created\_at
  + updated\_at
* sections: la tabla section está relacionada por el campo id\_section con, categories y forma una tabla llamada category\_by\_section, con la tabla styles, con la tabla formats y forma una tabla format\_by\_section. Esta tabla está compuesta por los campos:
  + id
  + name
  + file
  + created\_at
  + updated\_at
* establisments: la tabla establishment está relacionada con la tabla categories por el campo id\_category de la tabla establisments, está relacionada con la tabla visits mencionada anteriormente, por la tabla valoration también mencionada anteriormente, con la tabla users a través del campo id\_owner que definirá quien es el dueño, esta relacionada con la tabla post que mencioné antes, además de todas estas relaciones que se explicaron anteriormente también está relacionada con la tabla social\_network con el campo id\_establishment, esta relación forma una tabla de N:M, con styles por el campo id\_establishment y con la tabla establisments\_image por el campo id\_establishment. Esta tabla está compuesta por los campos:
  + id
  + name
  + location
  + slug
  + description
  + id\_category
  + id\_owner
  + created\_at
  + updated\_at
* category\_by\_section: Esta tabla se forma de la unión de la tabla sections y categories. Esta tabla está compuesta por:
  + id\_category
  + id\_section
  + created\_at
  + updated\_at
* social\_networks: esta tabla está relacionada con la tabla establisments y forman una nueva tabla llamada social\_networks\_by\_establisment que contra el id de social\_networks como el campo id\_socialNetwork. Esta tabla está compuesta por los campos:
  + id
  + name
  + logo
  + created\_at
  + updated\_at
* social\_networks\_by\_establisment: es la tabla formada por la relación de N:M entre social\_networks y establisments. Esta tabla está compuesta por los campos:
  + id\_socialNetwork
  + id\_establishment
  + created\_at
  + updated\_at
* format\_by\_section: es la tabla formada por la relación de N:M entre formats y sections. Esta tabla está compuesta por los campos:
  + id\_format
  + id\_section
  + ocurrence\_of\_the\_format
  + created\_at
  + updated\_at
* formats: la tabla format está relacionada con la tabla section formando una nueva tabla llamada format\_by\_section que se relaciona a formats por el campo id\_format, además también estará relacionada con la tabla styles por el campo id\_format. Esta tabla está compuesta por los campos:
  + id
  + name
  + created\_at
  + updated\_at
* datas: la tabla datas está relacionada con la tabla styles por el campo id\_data. Esta tabla está compuesta por los campos:
  + id
  + datum
  + created\_at
  + updated\_at
* styles: Esta tabla es la más compleja, puesto que se trata de la formación de una tabla por la relación entre otras 4, las tablas relacionadas son formats, datas, establisments y sections, todas ellas relacionadas por su id como comente anteriormente, esta tabla tiene como primary\_key los id de las cuatro tablas mencionadas anteriormente. Esta tabla está compuesta por los campos:
  + id\_format
  + id\_section
  + id\_data
  + id\_establishment
  + created\_at
  + updated\_at

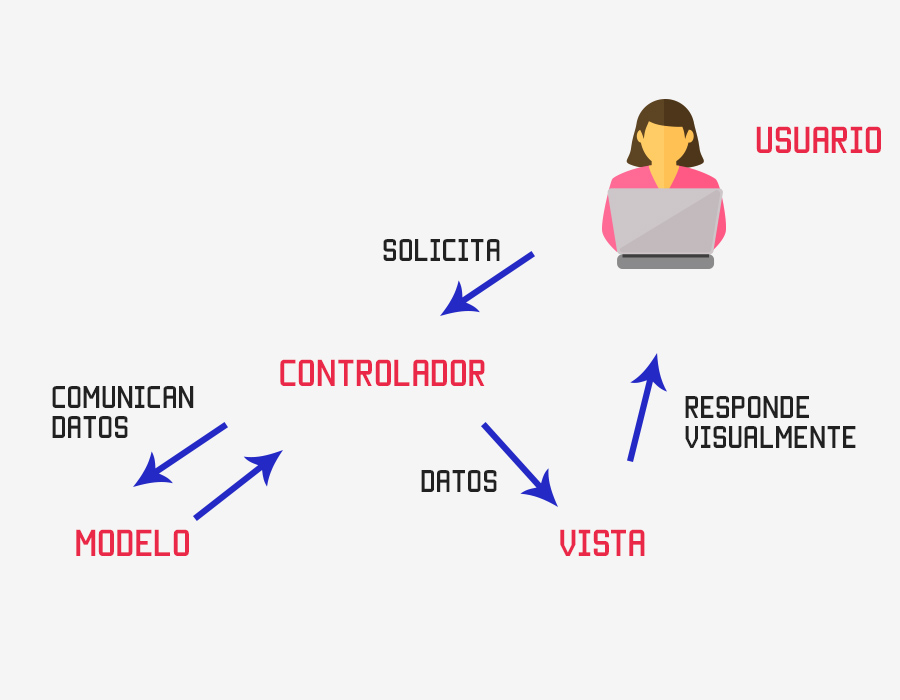
**4.2. Mi aplicación**

La aplicación se ha desarrollado sobre distintos lenguajes de programación que otorgan la capacidad y flexibilidad oportunas.

En mayor parte se ha desarrollado en PHP, donde se encuentra el motor principal de la aplicación. Por otra parte, se ha implementado en Java Script el control de datos y la limpieza de la URL.

Para finalizar, se ha utilizado CSS (Cascading Style Sheets) para dar el diseño físico de la aplicación.

Se ha utilizado el patrón de diseño de software MVC (Modelo Vista Controlador), que separa los datos en una aplicación, la interfaz de usuario, y la lógica de control entre capas distintas.



**4.3 Líneas de investigación futuras.**

En un futuro me gustaría implementar vistas distintas para cada tipo de establecimiento, realizar una revisión de código (Cuando pasa el tiempo y continuas programando te das cuenta de algunas mejoras de código).

Me gustaría seguir aprendiendo PHP y sus frameworks como laravel o Symfony.

Por ultimo, me gustaría en un futuro añadir al proyecto una selección de idiomas y una customización total del establecimiento mas cómoda y dinámica para el dueño del mismo.

**4.4 Bibliografía**

Durante el proceso de desarrollo he visitado la documentación de:

* chartJ: https://www.chartjs.org/docs/latest/
* mapBox:https://docs.mapbox.com/mapbox-gl-js/api/
* PHP : https://www.php.net/docs.php
* Stackoverflow: https://stackoverflow.com/
* heroku: https://id.heroku.com
* gitHub: https://github.com/
* W3School: https://www.w3schools.com/
* fontawesome: https://fontawesome.com/v4/
* TableVanilla: https://vanillaframework.io/docs/base/tables
* Bootstrap: https://getbootstrap.com/
* sweetAlert2: https://sweetalert2.github.io/