



INAYA

Documentación del proyecto



Memoria del proyecto de Desarrollo de aplicaciones
web realizada por:

Victoria Almeida Calderón y dirigido por Isabel
Jiménez Cumbreiras

13 DE JUNIO DE 2022

IES LA ARBOLEDA

Resumen

Este proyecto tiene como objetivo desarrollar una aplicación en formato web capaz de gestionar un blog informativo sobre eventos de Ayamonte, además de promocionar a locales de este municipio. El proyecto contiene las facilidades de publicar cualquier contenido tanto de los locales como de los eventos de Ayamonte. Nuestro objetivo consistirá en darle todas la facilidades al usuario para la creación de su propio contenido, así como la actualización y eliminación del mismo, promocionar al cliente y dar una entrada agradable a cualquier usuario.

INAYA: INFORMACION DE AYAMONTE

1. PRESENTACIÓN	5
Resumen	5
Estado del arte.....	5
Objetivo y alcance previstos.....	6
Justificación	¡Error! Marcador no definido.
2. Estudio de viabilidad.....	6
2.1 Introducción	6
2.2 Objeto	6
○ Objeto	6
○ Perfil de los usuarios del proyecto	7
○ Objetivos	7
2.3 Recursos	8
○ Recursos de desarrollo	8
2.3.2 Recursos de implementación.....	9
2.4 Planificación del proyecto	9
2.4.1 Planificación inicial	9
2.4.2 Modelo de desarrollo	10
2.5 Conclusiones	11
3. Analisis del proyecto.....	12
3.1 Requerimientos funcionales	12
3.1.1 Visión general	12

INAYA: INFORMACION DE AYAMONTE

3.1.2	Requisitos funcionales de usuarios.....	12
3.1.3	Vistas de casos de uso.....	13
3.2	Requerimientos no funcionales.....	13
4.	Diseño	14
4.1	Introducción	14
4.1.1	Selección del entorno de desarrollo.....	14
4.1.2	Selección de base de datos	15
4.1.3	Diagrama de Entidad-Relación	17
4.2	Estructura de la base de datos.....	17
4.2.1	Contenido de las tablas de la base de datos	17
4.3	Arquitectura de la aplicación.....	22
5.	Implementación	22
5.1	Introducción	22
5.2	Codificación de las diferentes capas de la aplicación.....	23
5.2.1	Interfaces.....	¡Error! Marcador no definido.
5.2.1.1	Entorno de usuarios.....	¡Error! Marcador no definido.
5.2.2	Motor de la aplicación	23
5.2.3	Capa de datos.....	24
6.	PRUEBAS	25
6.1	Pruebas de compatibilidad.....	25
6.2	Pruebas de seguridad.....	27

INAYA: INFORMACION DE AYAMONTE

6.3	Pruebas de integración de servicio	27
6.4	Pruebas según el tipo de usuario.....	¡Error! Marcador no definido.
6.4.1	Administrador.....	¡Error! Marcador no definido.
6.4.2	Cliente	¡Error! Marcador no definido.
6.4.3	Todos los usuarios	¡Error! Marcador no definido.
7.	Conclusiones y resultados	27
7.1	Consecución de objetivos	27
7.2	Líneas de ampliación	27
7.3	Valoración personal de la experiencia	27
8.	Bibliografía.....	28
8.1	Programación	28
8.1.1	Paginas web.....	28
9.	Como conclusion	29

1. PRESENTACIÓN

Resumen

En la situación actual la ausencia de páginas web de información sobre Ayamonte hace que los distintos eventos y/o establecimientos de este no sean visitados debido a la desinformación de los residentes del mismo municipio y turistas. Esto provoca en la mayoría de los casos cierres de establecimientos y ausencia de personas en los distintos eventos.

La necesidad en algunos casos imperativa de promocionar sus establecimientos con la escasez de recursos se vuelve una tarea compleja para los dueños de los establecimientos de Ayamonte, provocando así la casi inexistencia de clientes y el prematuro cierre del establecimiento. Normalmente los motivos de estos cierres es la desinformación del bar/restaurante/pub/etc. un ejemplo de desinformación de los establecimientos sería la ubicación del establecimiento, la relación precio/calidad e incluso la existencia de estos, siendo la más común la desinformación de la relación del precio y la calidad, en muchos casos estos establecimientos poseen buen servicio y precios económicos, pero no se dan a conocer.

Todo esto provoca un fracaso tanto para los dueños de los locales como para el propio municipio que pierde un establecimiento. En el caso de los eventos la desinformación de fechas o incluso del mismo evento, conlleva pérdidas económicas al municipio de Ayamonte que invierte en estos eventos. En muchos casos se llevan acabo eventos que no han sido informados o han sido poco promocionados y por lo tanto pierden audiencia.

Por ello he decidido implementar una aplicación que informe a turistas y ciudadanos de Ayamonte sobre estos establecimientos y eventos. Los dueños de los establecimientos podrán promocionar sus establecimientos e incluso recibir reservas de clientes si así lo desean. De esta forma los establecimientos tendrán un lugar donde promocionarse y los usuarios tienen las facilidades de reservar en dichos establecimientos y reunir información sobre los distintos eventos del municipio, además el usuario también podrá subir su propio contenido en el apartado de cada evento volviendo la web más dinámica y la experiencia del usuario más atractiva.

Estado del arte

Una web de información debe ser clara y con una interfaz fácil de comprender para los usuarios, además una web en la cualquier usuario puede subir su propio contenido tiene que tener un control exhaustivo de la información y evitar el ruido visual para que la web sea legible, aparte de esto como cualquier usuario puede publicar información la interfaz aun tenga o no conocimientos informáticos la interfaz debe ser practica y cómoda para una buena experiencia de cualquier usuario de la aplicación.

Las necesidades más básicas de esta web son: implementar la gestión de categorías, la gestión de establecimientos y la configuración de estos. (*Aun hay más necesidades en la web, pero esto sería lo más básico*).

No hay aplicaciones de este tipo, porque, aunque existen páginas de Ayamonte información estas se encuentran abandonada y con un contenido muy atrasado.

Objetivo y alcance previstos

Gestionar establecimientos de Ayamonte y configurarlos. Para esto se desarrollará una pagina web que permita gestionar la información de estos, como de los usuarios registrados y el contenido que van a subir. Vamos a conseguir la participación de todos los tipos de usuario haciendo mas dinámica la web, ya que siempre habrá usuarios que deseen publicar su propio contenido y por esto no se quedará abandonada.

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

○ Objeto

El proyecto esta pensado para dar información sobre los establecimientos del municipio de Ayamonte que por el momento no tiene ninguna página que proponga esto.

Después de haber visto las paginas desactualizadas he observado las necesidades de estas webs, que a continuación expondré:

- Configuración de la web

INAYA: INFORMACION DE AYAMONTE

- Gestion de categorias
- Gestión de establecimientos
- Configuración de los establecimientos proponiendo individualidad
- No hay ninguna página que proponga esto.

○ Perfil de los usuarios del proyecto

Como se explicará en el capítulo 3, existen 2 tipos de usuarios los cuales poseerán cualidades específicas. Independientemente del tipo de usuario ellos necesitarán una interfaz sencilla y limpia que además sea intuitiva para que los usuarios que no tengan conocimientos previos puedan tener una buena experiencia. Por ello se ha propuesto la inserción de distintos métodos para que sea mas sencillo a simple vista.

○ Objetivos

A. Objetivo general

Gestionar establecimientos de Ayamonte y configurar la web de cada uno de estos de manera individual, dando esta manera mas opciones de estilos al dueño del local y a su vez una grata experiencia al usuario. Transformar la clásica pagina de información estática en un blog de información dinámica publicada volviendo esta pagina en una experiencia mas cercana al usuario.

La individualidad que se busca por cada local es el objetivo primordial.

B. Objetivos específicos

En este apartado quiero destacar que llamaré “habitantes” a todo ciudadano temporal o no de este municipio, es decir habitantes son tanto turistas como ciudadanos.

- Mejorar la experiencia de los dueños de los establecimientos los cuales pueden editar el estilo de su web a su gusto,
- Disminuir la desinformación de los distintos establecimientos y que estos tengan un mayor acercamiento a los “habitantes”.
- Hacer la experiencia del usuario agradable.

2.3 Recursos

- Recursos de desarrollo

Requisitos mínimos para el desarrollo de la aplicación:

- Hardware:

- PC:

Especificaciones del dispositivo

Nombre del dispositivo	DESKTOP-MP92HVU
Procesador	AMD Ryzen 5 3400G with Radeon Vega Graphics 3.70 GHz
RAM instalada	16,0 GB
Identificador de dispositivo	ECDD4274-389B-48A1-9B38-4415E32DE05F
Id. del producto	00329-90000-00001-AA037
Tipo de sistema	Sistema operativo de 64 bits, procesador basado en x64
Lápiz y entrada táctil	La entrada táctil o manuscrita no está disponible para esta pantalla

Copiar

Cambiar el nombre de este equipo

- Software:

- Servidor web: Apache
 - Entorno de desarrollo: Visual Studio Code
 - Base de datos: Mysql
 - Editor de base de datos: phpmyadmin y mysql workbench
 - Programación web: PHP, HTML, jquery, javascript
 - Sistema Operativo:

Especificaciones de Windows

Edición	Windows 10 Enterprise N
Versión	21H1
Instalado el	22/02/2021
Compilación del sistema operativo	19043.1706
Experiencia	Windows Feature Experience Pack 120.2212.4170.0

- Navegador: Google Chrome.
 - Documentación del proyecto: Word y pdf
 - Diagrama de caso de uso: Realizado con StarUML
 - Diagramas de BD: Realizado con DIA y obtenido por phpmyadmin (también mostrare la visualización por workbench puesto que es un programa que no hemos usado en clase)

INAYA: INFORMACION DE AYAMONTE

En el apartado 4 se hablarán las tecnologías habladas en esta lista.

2.3.2 Recursos de implementación

Requisitos mínimos para la implementación de la aplicación

- Servidor
 - Servidor WAMP (Windows, Apache, MYSQL, PHP), 1 Gb memoria RAM, 1 GB de disco duro.
- Software de los usuarios
 - Navegador: Google Chrome.

2.4 Planificación del proyecto

2.4.1 Planificación inicial

Numero	Tarea	Descripción
1	Inicio del proyecto: asignación y matriculación del proyecto	Asignación y matriculación del proyecto elegido.
2	Planificación	Llevar a cabo tareas del proyecto en el tiempo establecido.
3	Análisis de la aplicación	Llevar a cabo una lista de requisitos.
4	Estudio del lenguaje	Estudio del los lenguajes usados en la aplicación web
5	Diseño de la aplicación	Diseño de las distintas partes que conforman la aplicación.

INAYA: INFORMACION DE AYAMONTE

6	Desarrollo de la aplicación	Desarrollar la aplicación una vez realizado las fases de análisis y diseño.
7	Pruebas	Realización de pruebas del sistema: <ul style="list-style-type: none">♣ Compatibilidad♣ Seguridad♣ Integración del servicio♣ Según el tipo de usuario
8	Implementación	Instalación y configuración del sistema en un ordenador
9	Generación de documentos (memoria del proyecto)	Generación de la memoria del proyecto, detallando los pasos seguidos.
10	Cierre del proyecto	Comprobación de que el proyecto está correctamente concluido.
11	Defensa del proyecto	Defensa del proyecto delante de un tribunal.

2.4.2 Modelo de desarrollo

Se ha elegido el modelo de desarrollo evolutivo, que es una metodología de desarrollo de software muy relacionada con, pero claramente distinta de, desarrollo por prototipos. El énfasis está puesto sobre la importancia de obtener un sistema de producción flexible y expandible. Así, si los requerimientos cambian durante el desarrollo del sistema, entonces con un mínimo de esfuerzo y tiempo se puede desarrollar un sistema de trabajo flexible. La diferencia fundamental entre desarrollo evolutivo y prototipos de software es que el desarrollo evolutivo busca reemplazar el viejo sistema con uno nuevo que tendría la

propiedad de satisfacer los nuevos requerimientos lo más rápido posible. En contraste, prototipos usa un enfoque iterativo sólo para determinar los requerimientos organizacionales. Por lo tanto el tiempo tomado entre cada iteración es mucho más importante para el desarrollo evolutivo.

El desarrollo evolutivo asume que los requerimientos de un proyecto están sujetos a cambios continuos, por lo cual es necesario definir una estrategia de desarrollo que refleje esta situación.

La idea entonces de la metodología de desarrollo evolutivo es estar liberando constantemente una nueva versión del sistema que sea completamente funcional; así, cada sistema producto de las iteraciones sucesivas del método tendría incorporado los nuevos requerimientos que ha sido posible identificar y que no estarían considerados en la anterior versión.

Así, las etapas del desarrollo evolutivo tienen por objetivo extender los incrementos de un producto de software operacional, en las direcciones determinadas por la evolución de la experiencia operacional. El método evolutivo tiene la gran ventaja de reconocer la existencia de una constante de cambios en los requerimientos y, desde esta premisa, propone una solución, la cual es válida para la solución de ese problema pero que no resolvería la inquietud original.

2.5 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. ANALISIS DEL PROYECTO

3.1 Requerimientos funcionales

3.1.1 Visión general

Como ya se ha comentado, este proyecto tiene como objetivo desarrollar una aplicación en formato web capaz de informar a los usuarios y que estos puedan subir su propio contenido a la web.

Nuestro objetivo es una web dinámica hecha los dueños de los establecimientos de Ayamonte para que no quede en abandono, y cuyo objetivo sería informar a todos los usuarios estén o no registrados de los establecimientos de Ayamonte.

En las páginas a las que puede acceder cualquier usuario, se llevará a cabo un desarrollo de información, mientras que a las áreas privadas todos los usuarios podrán gestionar el contenido establecido al tipo de usuario que es.

Existen 3 tipos de usuario:

- Administrador: Gestión de categorías, visualización de usuarios y establecimientos, y gestión del home.
- Cliente (Dueños de los establecimientos de Ayamonte) : Gestión de establecimientos y configuración de estos

3.1.2 Requisitos funcionales de usuarios

A continuación se presentan los requisitos funcionales de cada tipo de usuario, con fin de detallar los roles o capacidades de cada uno de ellos en el proyecto. Es decir hablaremos del apartado del menú del administrador, lo que podrá ver en ese apartado

- Administrador:
 - Visualizar cliente
 - El administrador es capaz de ver a todos los clientes registrados en la app
 - CRUD de categorías
 - En este CRUD se elegirán las secciones que habrá por cada categoría, así cuando se registre un nuevo establecimiento al elegir la categoría estas aparecerán para el establecimiento que elija esta, también se puede modificar pero no se podrá eliminar.
 - Modificar el Home
 - El administrador podrá cambiar varios apartados del home, para que de esta manera sea mas dinamico.

Acciones que no puede hacer el administrador.

INAYA: INFORMACION DE AYAMONTE

- CRUD de establecimientos
 - Configurar establecimientos
- Propietario
 - CRUD de establecimientos
 - El cliente puede dar de alta sus establecimientos y que estos aparezcan en el apartado correspondiente. También pueden actualizar y borrar los establecimientos que consideren.
 - Configuración de establecimientos
 - La configuración de establecimientos se llevará a cabo por los dueños de los mismos, esta configuración se compone de una nueva plantilla únicamente para los establecimientos, esta a su vez se dividirá en secciones que el mismo dueño podrá manipular a su gusto modificando así texto e imagen que originalmente era estática a una más dinámica puesto que todas estas modificaciones se llevarán a cabo a través de consultas.

Me gustaría añadir además, que cada categoría tendrá una sección por defecto elegida por el administrador, por lo tanto al llevar a cabo la creación o modificación de los establecimientos las secciones disponibles serán las que estén seleccionadas de cada categoría por el administrador, y a su vez el dueño del establecimiento puede elegir qué secciones mostrar y cuáles no dentro de las disponibles.

3.1.3 Vistas de casos de uso

3.2 Requerimientos no funcionales

Los siguientes requerimientos no funcionales no forman parte de la funcionalidad del sistema, pero son importantes ya que son restricciones que debe cumplir el sistema. A continuación, se muestran los requerimientos no funcionales de este sistema:

- Control de acceso de cada dueño:
 - Que los dueño accedan a sus apartados y no puedan entrar a alguno que no sea equivalente con su tipo. EJ: Un dueño no puede acceder al CRUD de categorías.
- La interfaz tiene que ser sencilla para que pueda entrar cualquier usuario.
 - Los colores de esta aplicación mejoran la accesibilidad de está.
- El sistema tiene que ser inmediato puesto que se trata de facilitar información al usuario.
- Los formularios tienen sus respectivas validaciones y mensajes de error.

4. DISEÑO

4.1 Introducción

Los siguientes requerimientos no funcionales no forman parte de la funcionalidad del sistema, pero son importantes ya que son restricciones que debe cumplir el sistema. A continuación, se muestran los requerimientos no funcionales de este sistema:

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 es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.

- 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

- MySQL



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.

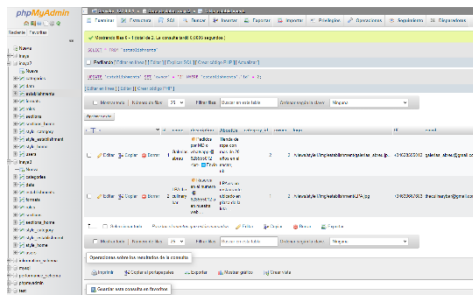
- Editor de la base de datos

He usado dos editores de base de datos para mayor comodidad.

Para comenzar he usado el conocido phpmyadmin que aparece en el propio panel de xampp como parte de la administración de mysql.

phpMyAdmin es una herramienta de software libre escrita en PHP, destinada a manejar la administración de MySQL en la Web. phpMyAdmin admite una amplia gama de operaciones en MySQL y MariaDB. Las operaciones de uso frecuente (administración de bases de datos, tablas, columnas, relaciones, índices, usuarios, permisos, etc.) se pueden realizar a través de la interfaz de usuario, mientras aún tiene la capacidad de ejecutar directamente cualquier instrucción SQL.

El proyecto phpMyAdmin es miembro de Software Freedom Conservancy. SFC es una organización sin fines de lucro que ayuda a promover, mejorar, desarrollar y defender proyectos de software libre, libre y de código abierto (FLOSS).



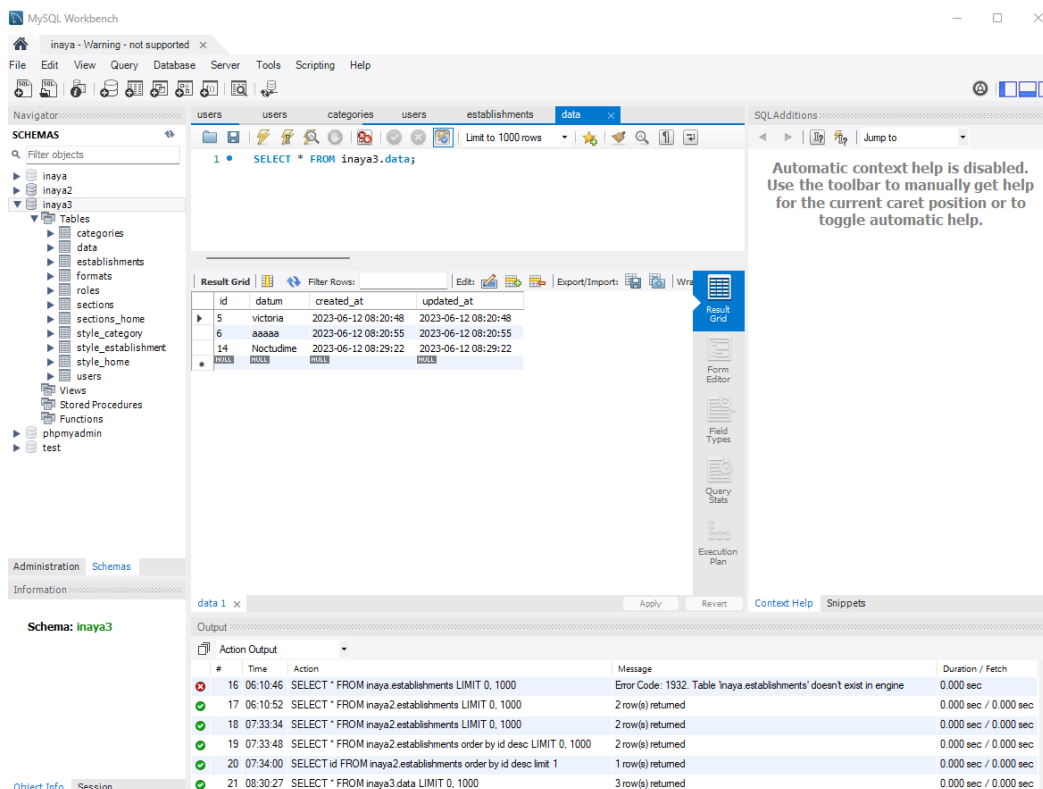
A continuación habremos de workbench.

INAYA: INFORMACION DE AYAMONTE

Workbench es un editor de base de datos muy conveniente, como este editor nos ha dado en clase voy a exponerlo un poquito más que los anteriores que ya se han dado en clase.

Workbench al iniciar te permite conectar con distintos puertos de bases de datos, esto te permite desarrollar aplicaciones con distintos puertos y a su vez manejar distintos sistemas de gestión de base de datos siendo los popularmente conocidos MySQL, MariaDB, PostgreSQL, MongoDB. Una vez iniciada la conexión con el puerto correspondiente al sistema de base de datos podemos acceder a todas las bases de datos que contiene ese propio puerto, dentro del cual podemos manejarlo como siempre.

MySQL Workbench ofrece herramientas visuales para crear, ejecutar y optimizar consultas SQL. El editor de SQL proporciona resaltado de sintaxis en color, autocompletado, reutilización de fragmentos de SQL e historial de ejecución de SQL. El Panel de conexiones de bases de datos permite a los desarrolladores administrar fácilmente las conexiones de bases de datos estándar, incluido MySQL Fabric. El Explorador de objetos proporciona acceso instantáneo al esquema y los objetos de la base de datos.



4.1.3 Diagrama de Entidad-Relación

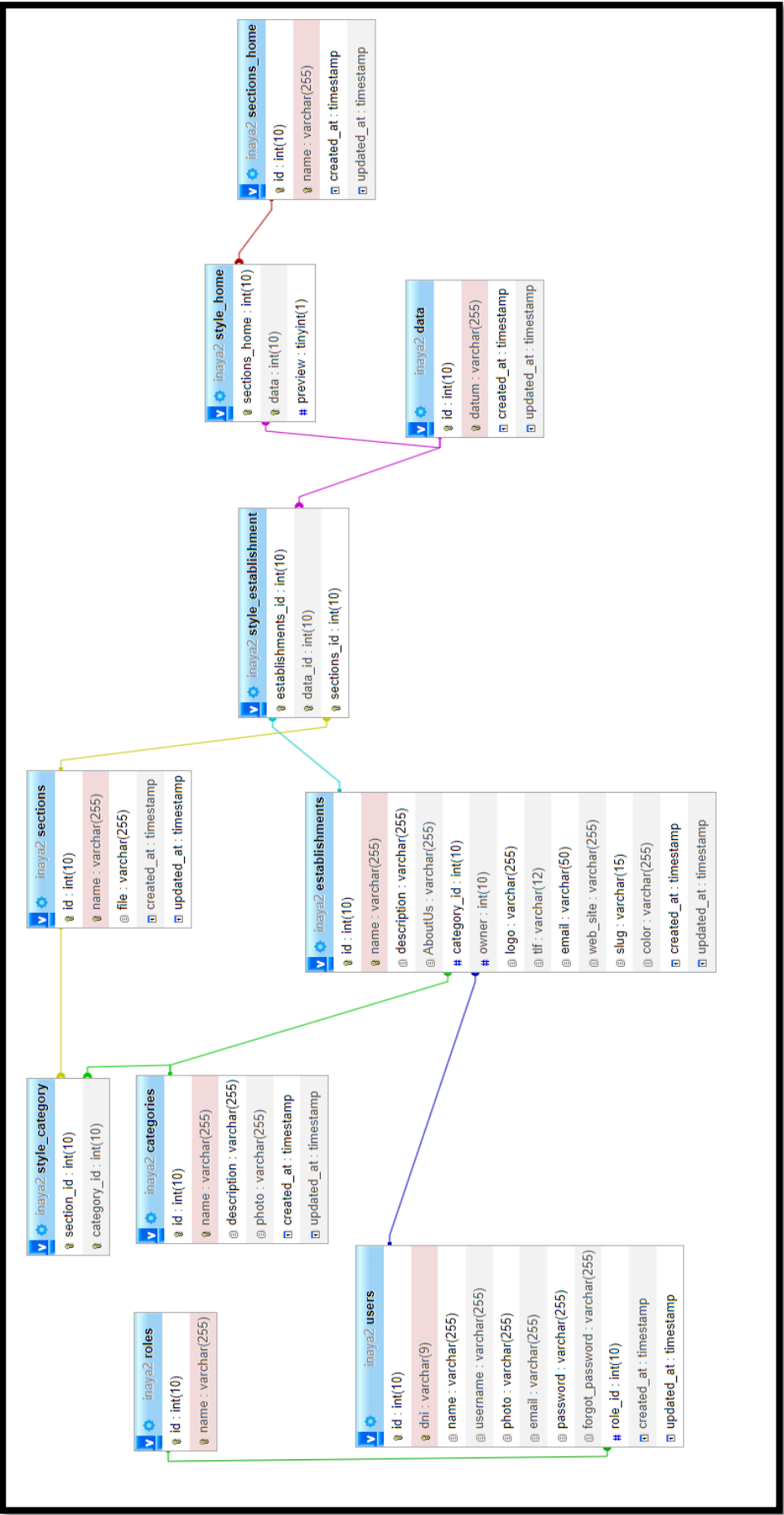
Este modelo representa a la realidad a través de un esquema gráfico empleando la terminología de entidades, que son objetos que existen y son los elementos principales que se identifican en el problema a resolver con el diagrama y se distinguen de otros por sus características particulares denominadas atributos, el enlace que se rige la unión de las entidades está representada por la relación del modelo.

4.2 Estructura de la base de datos

A partir de las entidades y sus interrelaciones, la base de datos constará de 14 tablas. Estas tablas guardaran la información pertinente a la web.

4.2.1 Contenido de las tablas de la base de datos

Se pasa a detallar el contenido principal de cada una de ellas, el motivo de las relaciones y los tipos de datos más significativos.



INAYA: INFORMACION DE AYAMONTE

Tabla – “users”

Es el nombre de la tabla donde se guardaran la información de todos los usuarios.

Detalle de los campos:

- Id: identificador unico autonumerico.
- DNI: identificador unico secundario alfanumerico.
- Name: Nombre del usuario. Es un varchar.
- Username: Usuario del perfil. Es un varchar.
- Email: Email del usuario. Es un varchar.
- Photo: Tipo timestamp que se encarga de verificar el email.
- Password: Contraseña del usuario encryptada. Es un varchar.
- **Forgot_password**:Codigo que se adjuntara en el email.
- Role_id: Tipo de usuario, enum que tiene 3 valores: admin, user, customer.
- Created_at: Cuando se creo el nuevo usuario. Es un timestamp.
- Updated_at: Ultima actualización del campo. Es un timestamp.

Tabla – “categories”

categories es el nombre de la tabla que guarda información sobre las categorias.

Detalle de los campos:

- Id: identificador unico autonumerico.
- name: nombre de la categoria.
- Description: descripcion de la categoria.
- Photo: foto de la categoria.
- Created_at: Cuando se creo el campo. Es un timestamp.
- Updated_at: Ultima actualización del campo. Es un timestamp.

Tabla – “Roles”

Roles es el nombre de la tabla que guarda información de los roles de los usuarios.

Detalle de los campos:

- Id: identificador unico autonumerico.
- Name: Nombre del rol. Varchar.
- Created_at: Cuando se creo el campo. Es un timestamp.
- Updated_at: Ultima actualización del campo. Es un timestamp.

Tabla – “Establishment”

Establishment es el nombre de la tabla que guarda información de los establecimientos publicados.

Detalle de los campos:

- Id: identificador unico autonumerico.
- Name: nombre del establecimiento. Es un Varchar.
- Description: descripcion del establecimiento.
- aboutUs: descripcion mas larga del establecimiento.
- Category_id: clave externa de categoria.
- Owner: propietario del establecimiento.
- Logo: logo del estableciminet.
- Tlf: Telefono de contacto. Es un Varchar.
- Email: email de contacto del establecimiento.
- Web_site: pagina principal del establecimiento.
- Slug: cadena corta para acceder a un establecimiento a travez de la URL.
- Color: color de ciertas secciones de la plantilla.
- Created_at: Cuando se creo el campo. Es un timestamp.
- Updated_at: Ultima actualización del campo. Es un timestamp.

Tabla – “sections”

Aquí se guardaran las secciones de las que esta dividida la plantilla de los establecimientos que como he mencionado anteriormente, es distinta a la plantilla de la web.

Detalle de los campos:

- Id: identificador unico autonumerico.
- Name: Nombre del seccion. Varchar.
- Created_at: Cuando se creo el campo. Es un timestamp.
- Updated_at: Ultima actualización del campo. Es un timestamp.

Tabla – “sections_home”

Aquí se guardaran las secciones de las que esta dividida la parte del Home.

Detalle de los campos:

INAYA: INFORMACION DE AYAMONTE

- Id: identificador unico autonumerico.
- Image: Imagen de la publicación. Es un Varchar.
- Comment: Comentario de la publicacion. Es un Varchar.
- event_id: Id del evento. Es un Bigint.
- user_id: Id del usuario. Es un Bigint.
- Created_at: Cuando se creo el campo. Es un timestamp.
- Updated_at: Ultima actualización del campo. Es un timestamp.

Tabla – “style_category”

Es una tabla de N:M entre las secciones y las categorias, esta se usara para la funcion de agregar las secciones por categoria.

Detalle de los campos:

- section_id: Id del evento. Es un Bigint.
- category_id: Id del usuario. Es un Bigint.
- Created_at: Cuando se creo el campo. Es un timestamp.
- Updated_at: Ultima actualización del campo. Es un timestamp.

Tabla – “style_establishment”

Es una ternaria en la que se unen los datos para anadir informacion a la plantilla.

Detalle de los campos:

- Establishment_id: identificador del establecimiento.
- data_id: Id de los datos.
- sections_id: Id de las secciones.
- Created_at: Cuando se creo el campo. Es un timestamp.
- Updated_at: Ultima actualización del campo. Es un timestamp.
- Updated_at: Ultima actualización del campo. Es un timestamp.

Tabla – “style_home”

Estilos del home, en esta parte se guardara la relacion de N:M entre sections_home y data

Detalle de los campos:

- Sections_home: id de las secciones del home.
- Data: id de los datos

INAYA: INFORMACION DE AYAMONTE

- Preview: booleano que mostrara si el campo es para la prevista o no.
- Created_at: Cuando se creo el campo. Es un timestamp.
- Updated_at: Ultima actualización del campo. Es un timestamp.

Tabla – “data”

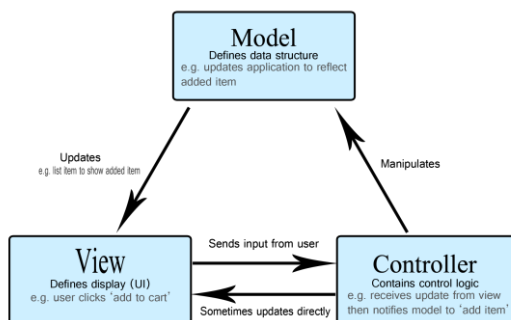
Las publicaciones de los establecimientos

Detalle de los campos:

- Id: identificador unico autonumerico.
- Datum: Contiene los datos. Es un Varchar.
- Created_at: Cuando se creo el campo. Es un timestamp.
- Updated_at: Ultima actualización del campo. Es un timestamp.
- Updated_at: Ultima actualización del campo. Es un timestamp.

4.3 Arquitectura de la aplicación

La aplicación utiliza una arquitectura MVC.



5. IMPLEMENTACIÓN

5.1 Introducción

Se han implementado algunas de las funciones anteriores como

- Administrador: Gestión de categorías y configuración del home.
- Cliente: Gestión de establecimientos y configuración de establecimientos.

5.2 Codificación de las diferentes capas de la aplicación

A continuación se mostrarán las capturas de pantalla más importantes del sistema, correspondientes a los diferentes usuarios del sistema

5.2.1 Motor de la 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 concretamente mediante modelo vista controlador

Por otra parte, se ha implementado en JavaScript para el uso de distintos plugin y atacar distintas API's. Para hacer modificaciones en el entorno cliente hemos usado JQUERY.

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

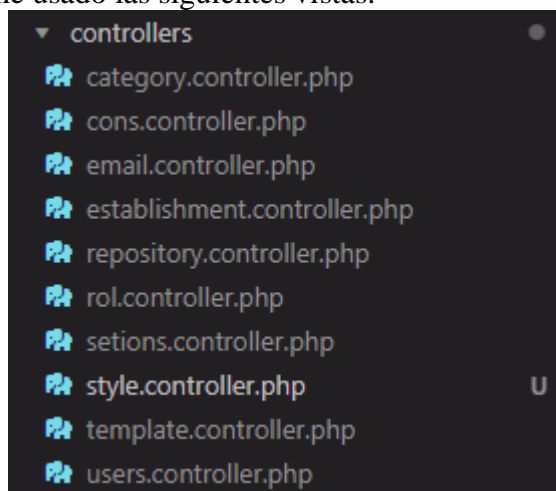
Se ha usado el patrón de diseño MVC el cual se caracteriza por dividir las conexiones entre el servidor y el usuario:

- Controladores: Se encuentran dentro de la carpeta /controllers. Estos archivos llaman a los distintos modelos para obtener datos que después muestran en las vistas.

En mi aplicación he usado

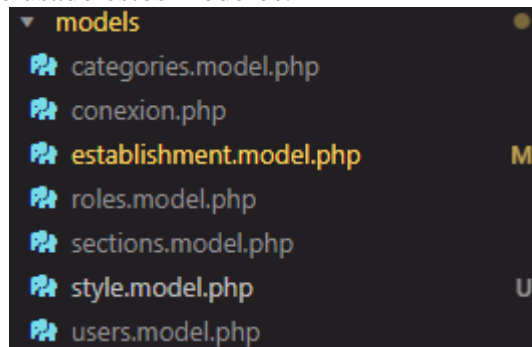
- Vistas: Se encuentran dentro de la carpeta Resources/views. Estos contienen el HTML y CSS que se mostraran por el lado cliente.

En mi aplicación he usado las siguientes vistas:



- Modelos: Se encuentran dentro de la carpeta /models. Estos están relacionados con la tabla pertinente de la BD, la relación entre tablas se lleva a cabo en ellos y también la creación de campos calculados.

En mi aplicación he usado estos modelos:



- Archivo const.controller.php: en este archivo encontraremos una constante que devolverá las rutas permitidas.

```
1 <?php
2 class ControllerConst{
3     public function getRoutes()
4     {
5         $routes = [
6             "public" => ["home", "categories", "establishments"],
7             "login" => ["login", "register", "forgotPassword", "code"],
8             "private" => ["menu", "logout", "preview"]
9         ];
10        return $routes;
11    }
12 }
```

5.2.2 Capa de datos

La aplicación se ha desarrollado con una base de datos en MySQL, según el diseño original previsto. Así, toda la información queda almacenada en sus respectivas tablas.

La aplicación se ha realizado con consultas hechas a través de modelos.

Para garantizar una mayor seguridad los datos pasan a las vistas desde el controlador como una variable, de esta manera los datos permanecen seguros sin necesidad de sesiones ni cookies, todo es a través de la tabla. El único dato que prevalece en una sesión es el usuario que esta logueado.

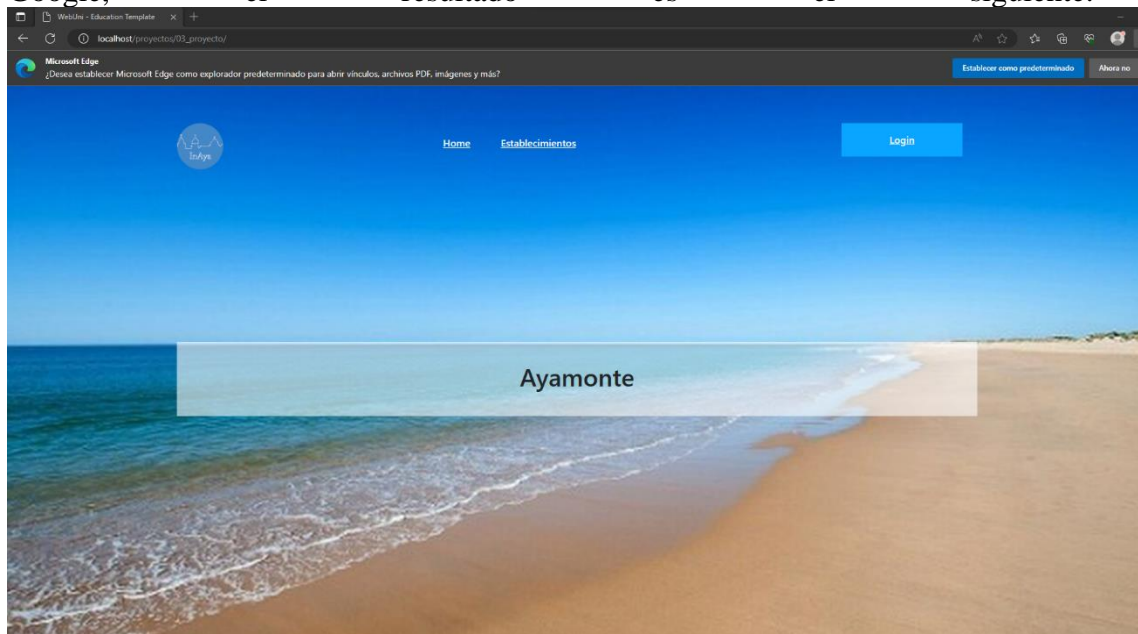
6. PRUEBAS

Se han efectuado varias pruebas para comprobar la solidez y robustez de la aplicación. Como se puede ver a continuación, se han efectuado pruebas de compatibilidad, de seguridad y de integración de servicio o funcionamiento.

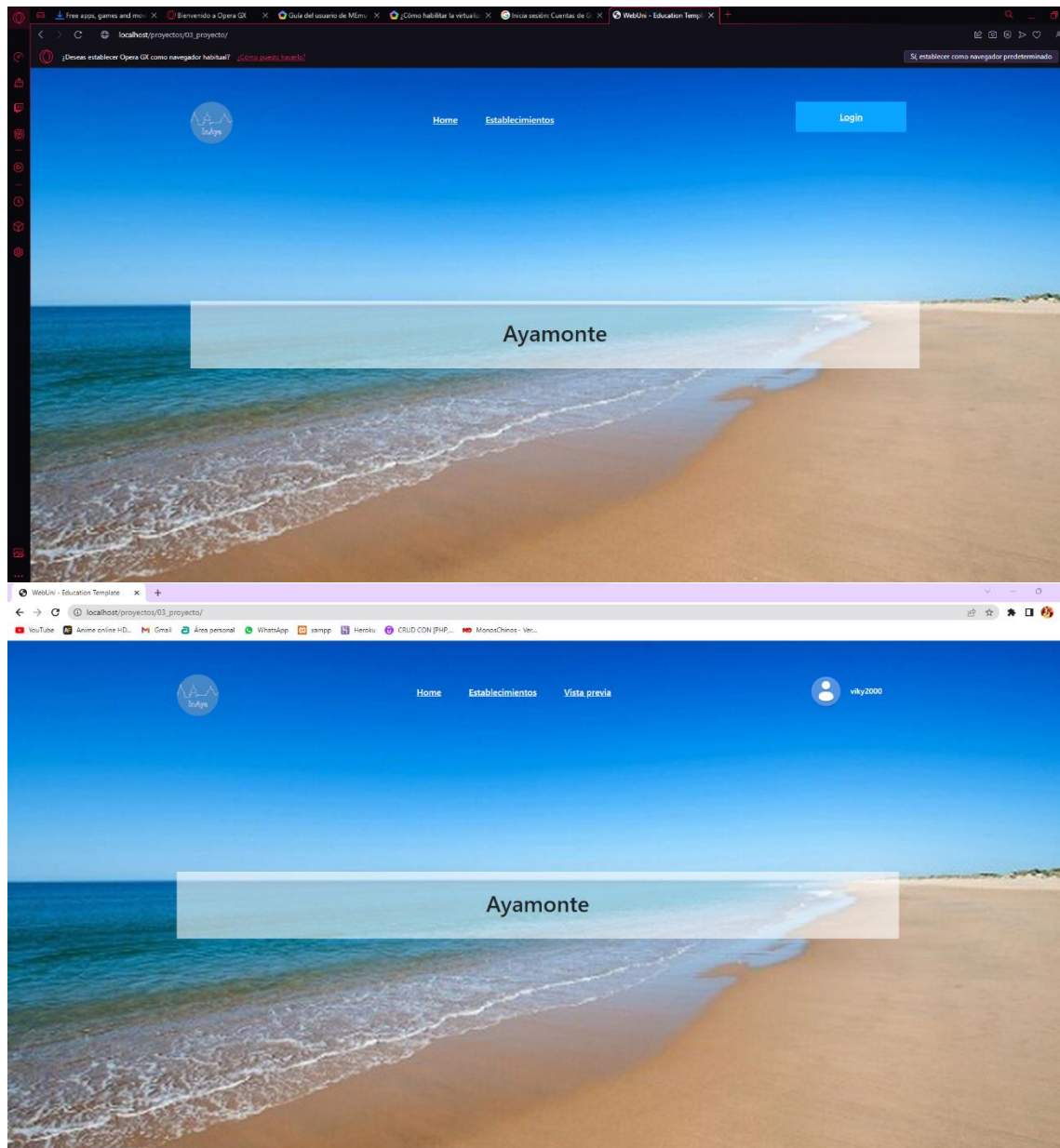
6.1 Pruebas de compatibilidad

Las pruebas de compatibilidad ayudan a determinar si el producto funcionará correctamente con otro software en el entorno de operación esperado.

Se han realizado pruebas de funcionamiento en internet Microsoft ,operaGX y Google, el resultado es el siguiente:



INAYA: INFORMACION DE AYAMONTE



Como podemos apreciar es el mismo resultado, la primera imagen muestra como se veria en microsoft, la segunda el Opera GX que es un navegador distinto de los comunes ya que esta desarrollado para gamers y en la tercera en internet explorer.

He probado el funcionamiento en ambos navegadores y el resultado ha sido satisfactorio.

El tamaño de la pantalla varia un poco en los navegadores, pero no supone ningun problema.

6.2 Pruebas de seguridad

Cada usuario puede acceder a su área en cuestión, un cliente no puede acceder a el área del administrador, y viceversa, los roles de los usuarios están estrictamente controlados por los middleware.

Cada usuario solo podrá acceder a aquellas zonas publicas o privadas en las que tenga permisos.

6.3 Pruebas de integración de servicio

En la sección anterior, sección 6.1: Pruebas de compatibilidad, se probó el funcionamiento de la aplicación, para todas sus opciones y en diferentes navegadores. Por esto se puede concluir que, el funcionamiento de la aplicación, ha sido correcta soportando todos los tests de pruebas realizados.

7. CONCLUSIONES Y RESULTADOS

7.1 Consecución de objetivos

En este trabajo, se han propuesto como objetivo Gestionar eventos y establecimientos de Ayamonte, publicar información sobre estos eventos de forma individual. Transformar la clásica pagina de información estática en un blog de información dinámica publicada por todo tipos de usuarios volviendo esta pagina en una experiencia mas cercana al usuario y que además tenga la impresión de ser una red social de esta manera los usuarios estarán mas activos. Todo esto es para hacer una web mas atractiva al usuario y que obtenga información.

De estos objetivos se puede afirmar que se han concluido exitosamente todos ellos.

7.2 Líneas de ampliación

La aplicación desarrollada es ampliable en diferentes aspectos según necesidades posteriores al uso. Se citan algunas líneas posibles de ampliación:

- Mejorar el aspecto estético de la aplicación web.
- Generar una factura para los clientes, añadir un filtro en locales, etc.

7.3 Valoración personal de la experiencia

Mi valoración del proyecto es que he puesto en practica todo lo que he dado en el ciclo y mejorado en la implementación de plugins, puesto que es algo que no se me daba bien. Respecto al contenido:

INAYA: INFORMACION DE AYAMONTE

- php he hecho el proyecto en base
- he usado bd para obtener los datos correctos de las tablas
- javascript he usado para obtener datos de campos o añadirle esos datos. De JQUERY lo he usado para varias cosas, la primera en apreciar es el slider del principio.
- De HTML y CSS he modificado la plantilla inicial.
- Chartjs para obtener gráficos.
- Datatables para hacer las tablas dinámicas.

Como resumen, esta aplicación aunque se vea simple y sencilla que es lo que pretendía lograr, tiene bastante funcionalidad interna. El hecho de que se vea tan sencilla es para que toda clase de usuario la vean muy intuitiva y simple.

Para finalizar me gustaría decir que he disfrutado aprendiendo como usar API's e implementar plugin, he disfrutado el haber estado investigando cada uno y su funcionamiento, quizás lo que no me ha gustado tanto ha sido las veces que no encontraba una solución inmediata y por lo tanto tardaba días en hayarla.

8. BIBLIOGRAFÍA

A continuación se exponen las diversas fuentes de bibliografía que han sido necesarias para la realización del proyecto. Durante el mismo se, han hecho dos tipos de consultas bibliográficas:

8.1 Programación

8.1.1 Paginas web

- VISUAL STUDIO CODE Me he ayudado de plugins como: Tabnine, Auto close tag, bracket pair colorizer, HTTP/s and relative link checker, PHP inteliphense, etc.
- CHARTJS para la implementación de gráficos
- STACKOVERFLOW para las dudas
- BOOTSTRAP para el diseño
- W3SCHOOLS para las dudas
- DATATABLES para la implementacion de tablas dinámicas y la inserción de plugins en las mismas
- CDN para la insercion de distintos links como jquery

9. COMO CONCLUSION

Esta aplicación se ha llevado a cabo por Victoria Almeida Calderón como proyecto de fin de grado.

Cumple con los requisitos mencionados anteriormente.

PROYECTO FINAL DE DESARROLLO DE APLICACIONES WEB: INAYA, una web de información de Ayamonte.

Un saludo.