**Resumen**

En este proyecto se desarrolla una aplicación web de adopciones y ayudas para una asociación protectora de animales de Corrales, Huelva llamada Apac. Esta web permitirá tramitar todas las funcionalidades propias de una asociación.

TABLA DE CONTENIDOS

**1 Introducción**

En este proyecto se desarrolla una aplicación web para el trámite de las funcionalidades propias de una asociación protectora de animales centrándose principalmente en facilitar el trabajo de los trabajadores de dicha asociación.

La aplicación contendrá una base de datos de animales y de peticiones de usuarios para colaborar de una u otra forma con dicha asociación. Permitiendo así poder trabajar mediante internet si necesidad de tener que gastar tiempo en desplazamientos y papeleo a la vieja usanza.

* 1. **Objetivos**

El principal objetivo de este trabajo de fin de grado es aplicar, ampliar y consolidar la gran cantidad de conocimientos que se han ido adquiriendo durante todo el proceso formativo.

Otro de los objetivos es facilitar el trabajo de esta asociación permitiéndoles gastar menos tiempo en su trabajo. Haciendo más sencillo el papeleo que tramitan diariamente.

* 1. **Contexto**

Actualmente el dinero que manejan la mayoría de las asociaciones defensoras de los animales es limitado, esto es debido a que gran parte por no decir su totalidad es invertido en vacunas y medicinas para intentar salvar la vida de los animales.

Por lo tanto, mi proyecto lo que intenta es ayudar con el trabajo que diariamente estas personas lidian. De esta forma ellos tendrán más tiempo para dedicar a los animales y necesitarán menos para contactar con los posibles interesados en ayudar a la asociación ya sea donando, adoptando, ayudando como casa de acogida…

Para lograr alcanzar los objetivos para la aplicación web, se ha decidido trabajar con formularios para la inserción de datos por parte de los usuarios, cruds para la gestión de datos para los trabajadores y varias funcionalidades más como exportar en pdf, Excel etc..

* 1. **Estructura del documento**

El documento está estructurado siguiendo las siguientes etapas de desarrollo:

**Especificación de requisitos:** Se describe detalladamente el software a desarrollar.

**Análisis:** Se describe el comportamiento esperado del software y su interacción con los usuarios y/u otros sistemas.

**Diseño:** Se explica la arquitectura que se ha utilizado en la aplicación.

**Desarrollo:** Se comentan las tecnologías y herramientas utilizadas.

**Pruebas:** Se describen las pruebas que se han utilizado para asegurar el correcto funcionamiento del proyecto.

1. **Especificación de requisitos**
   1. **Introducción**
   2. **Propósito**

La especificación de Requisitos Software (ERS) tiene como propósito el establecer el conjunto de funcionalidades de la aplicación web, la definición y el resumen de cada una de estas.

La ERS pretenda dar una visión completa de las funcionalidades y requisitos de la aplicación, siendo el punto de partida del desarrollo de la aplicación web.

La ERS se ha realizado siguiendo el estándar IEEE Std. 830-1998, el cual describe el contenido de las calidades de una buena especificación de requisitos de software.

**2.1.2 Ámbito**

La aplicación web que se va a desarrollar tiene como principal objetivo facilitar a los trabajadores de Apac la gestión de sus servicios y los datos de los usuarios interesados en ayudar con ellos.

Para facilitar la gestión de los datos con los que trabajan se les proporcionará Crud’s y herramientas que hagan que el control de los datos sean más sencillos y fáciles de manejar. Ver, editar, añadir o eliminar peticiones de adopciones o cualquier otro tipo de petición será de fácil manejo para los trabajadores de la asociación y con ello gestionarán todo el trabajo de una manera más óptima y mas organizada.

A su vez los usuarios interesados en ayudar de alguna forma a la asociación no tendrán que recurrir a las redes sociales o a las llamadas de teléfono para poder solicitar un servicio de la asociación. Simplemente mediante el uso de formularios podrá solicitar el servicio deseado mediante un simple formulario. En el caso de las adopciones por ejemplo no tendrá que desplazarse hasta la asociación para poder ver el catálogo de animales que buscan hogar, de esta forma se evitará en pequeñas medidas las reuniones con los trabajadores y será un incentivo mayor para estos usuarios el poder ver y solicitar todos los servicios desde la comodidad de su casa o desde donde sea que accedan. (Servicios como las adopciones requerirán ir físicamente en el momento de la adopción a la asociación, pero al solicitarla por internet facilitará la toma de datos ya que no habrá que realizarla de forma manual y solo será necesario imprimir dichos datos).

**2.1.3 Visión general del documento**

A continuación, se realizará una descripción general de la aplicación describiendo los factores y requisitos a cumplir.

**2.2Descripción General**

**2.2.1 Perspectiva del producto**

La aplicación que se va a desarrollar es totalmente independiente, no está relacionada con ninguna otra aplicación.

En un futuro sería posible que se integrase junto a una aplicación móvil para poder añadir alguna funcionalidad interesante como podría ser geolocalización para poder mandar la ubicación de camadas de animales callejeros.

**2.2.2 Funciones del producto**

La aplicación ofertará distintas funcionalidades dependiendo del tipo de usuario. Los usuarios no registrados (cualquier interesado en solicitar un servicio) y los registrados (los trabajadores de la asociación).

**Los usuarios no registrados podrán:**

* Solicitar una adopción.
* Ofertar ser socio.
* Ofertar ser cada de acogida.
* Ofertar ayudar en las labores de voluntariado.
* Ingresar una donación mediante paypal.

**Los usuarios registrados podrán hacer todo lo que hacen los usuarios no registrados y además podrán:**

* Añadir, editar y eliminar registros de los animales disponibles para adoptar.
* Editar las solicitudes y ofertas.
* Eliminar las solicitudes y ofertas.

**2.2.3 Características del usuario**

La aplicación tendrá como principales usuarios a las personas interesadas en adoptar o ayudar a la asociación.

El nivel informático de dichos usuarios se estimará como usuarios básicos para ello se intentará mostrar mediante avisos visuales los errores que ellos puedan cometer para que el uso de la aplicación sea sencillo para todo tipo de usuarios indiferentemente de su nivel informático.

**2.2.4 Restricciones**

Para acceder a la aplicación será necesario un dispositivo con acceso a internet, esto será indiferente del tipo de dispositivo utilizado ya que la aplicación contará con un diseño adaptativo a base de Bootstrap. Esto permitirá que la web sea operativa en cualquier tamaño de dispositivo desde móviles a ordenadores de sobremesa que manejen grandes resoluciones.

Las contraseñas de los trabajadores estarán encriptadas ya que al ser ellos los que tienen el control total de la aplicación sus claves son un punto vulnerable que no debe poder filtrarse a usuarios que no sean trabajadores.

Para poder solicitar cualquier servicio de la aplicación será necesario ser mayor de edad y disponer de una cuenta de correo ya que estos últimos serán utilizados para prevenir que el usuario que está solicitando los servicios tiene interés en ellos y no está intentando almacenar información falsa en el servidor para fastidiar el desarrollo de las actividades de la asociación.

**2.2.5 Supuestos y dependencias**

Es necesario para el funcionamiento de la aplicación tener conexión a internet, un navegador web desde el que acceder, una base de datos donde almacenar los datos y peticiones y un servidor web donde almacenar los ficheros de la aplicación web.

Es posible que con el paso de los años los navegadores dejen de ser compatibles con las tecnologías utilizas y esta aplicación deba evolucionar a tecnologías más modernas.

La principal dependencia es el acceso a internet no siendo está demasiado preocupante en los tiempos en los que estamos.

**2.2.6 Requisitos futuros**

En un futuro sería posible realizar mejoras a la aplicación algunos ejemplos serian:

* Rediseñar la aplicación con un framework más potente y seguro.
* Añadir funcionalidades como plugins de redes sociales para centralizar la inserción de datos desde la web a las distintas redes de forma automática de ser posible.
  1. **Requisitos específicos**
     1. **Requisitos de interfaces externos**

La interfaz de la web debe ser sencilla, conteniendo el menor número de hipervínculos y la mejor representación visual posible para hacerla atractiva y sencilla de manejar para todos los públicos.

* + 1. **Requisitos funcionales**

A continuación de se detallan los requisitos funcionales para los dos tipos de usuarios. Como los usuarios registrados (trabajadores) pueden además de todas sus funciones realizar las funciones de los usuarios no registrados no vamos a repetir dos veces las funciones que son comunes.

**Usuarios no Registrados:**

|  |  |
| --- | --- |
| RF-001 | Solicitar una adopción |
| Propósito | El usuario puede solicitar o proponer su participación en los distintos servicios: adopciones, casas de acogida, socios o voluntariados. |
| Entrada | Comprobación de un correo electrónico propio |
| Proceso | El formulario pide el formulario del servicio solicitado ingresando su email.  Se valida si el usuario forma parte de los usuarios baneados en el sistema mediante la comprobación en una tabla de la base de datos llamada BlackList. |
| Respuesta | Dependiendo de si el email esta o no en la BlackList se le permitirá al usuario a acceder al formulario o no. Mediante una url encriptada que se le enviará por correo.  Si el usuario no está en la lista se le dará un formulario que apuntará a ingresar los datos, pero si el usuario está baneado en el sistema se le redirigirá a la ventana de adopciones y se le mostrará un mensaje de error. De esta forma nos evitamos que el usuario crea que esta baneado e intente volver a molestar con otro email. Ya que creerá que hay algún error en la web. |

|  |  |
| --- | --- |
| RF-002 | Donar |
| Propósito | El usuario puede ingresar una cantidad de dinero que crea oportuna para así ayudar a la asociación a cubrir gastos de vacunas, medicamentos… |
| Entrada | El usuario accederá al plugin de paypal alojado en la web, el cual tendrá asociado el paypal de la asociación. |
| Proceso | Mediante el uso de dicho plugin el usuario se logeará con su cuenta de paypal y accederá a ingresar a la cuenta de la asociación el dinero que cree oportuno. |
| Respuesta | El plugin comprobará el ingreso y realizará la transferencia. |

**Usuarios registrados:**

|  |  |
| --- | --- |
| RF-003 | Gestionar los animales en adopción |
| Propósito | El trabajador podrá gestionar mediante Crud’s todos los datos necesarios para su trabajo. |
| Entrada | El trabajador atacará estas gestiones mediante la librería Grocery Cryd la cual permite de una forma simple y visual la inserción, edición y eliminación de datos. |
| Proceso | Mediante un sencillo crud con iconos representativos se podrá insertar, editar, eliminar y visualizar los animales, peticiones (adopciones, socios) y ofertas de casas de acogida y voluntariado. |
| Respuesta | La librería hará la operación pertinente. |

* + 1. **Atributos del sistema**
       1. **Mantenimiento**

La aplicación se diseñara para que sea fácil realizar su mantenimiento, por lo que la arquitectura se basará en el Modelo, vista controlador para diferenciar todas las funcionalidades y hacer así mucho más sencillo entender el código.

* + - 1. **Portabilidad**

La aplicación deberá ser totalmente operativa en diferentes sistemas operativos, simplemente siendo necesario el uso de navegadores.

* + - 1. **Seguridad**

La aplicación deberá comprender el uso de sesiones para limitar así el uso de las acciones avanzadas de gestión únicamente para los trabajadores. Cualquier usuario no registrado no tendrá acceso a los sistemas de gestión y solo podrá interactuar con las funciones en las que tiene acceso.

1. **Análisis**

En este capítulo se desarrollará el análisis inicial sobre la aplicación a desarrollar describiendo funcionalidad y estructura.

* 1. **Introducción**

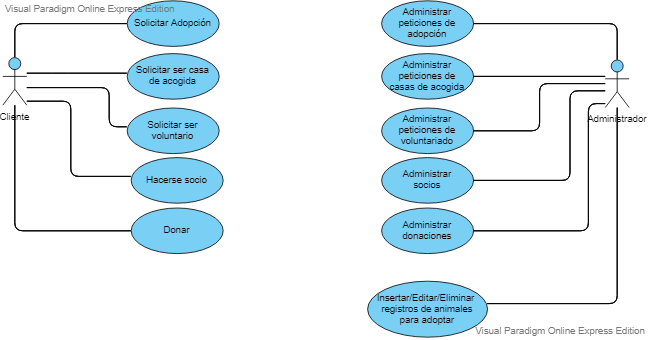
Este análisis describirá las funcionalidades de la aplicación mediante diagramas UML. Los cuales son visualmente representativos y están estandarizados bajo la normativa ISO/IEC 19505-1 e ISO/IEC 19505-2.

Utilizaremos diagramas UML como:

* Diagramas de casos de uso
  1. **Casos de uso**
     1. **Actores**

Los actores serán los dos tipos diferentes de usuarios: No registrados y registrados.

Los usuarios registrados podrán realizar sus acciones y además todas las de los usuarios no registrados.



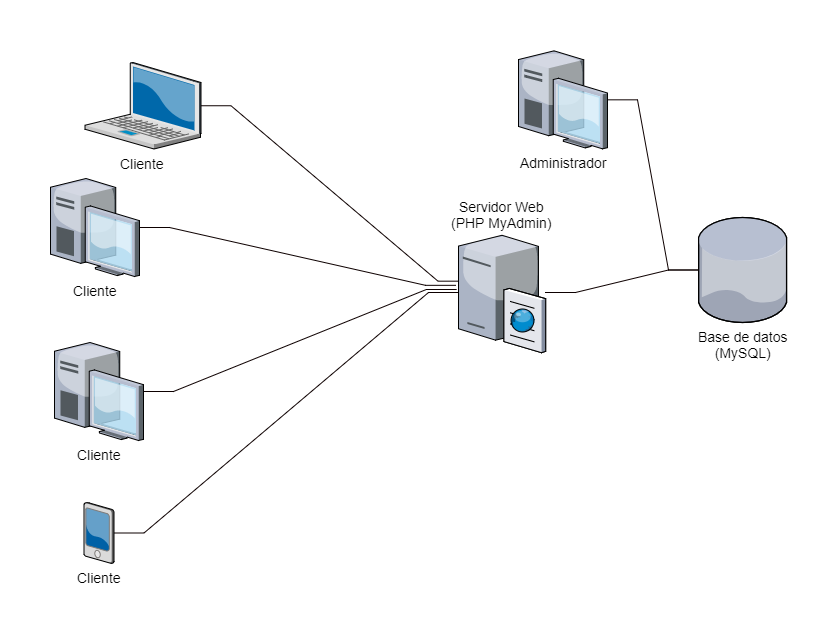
1. **Diseño**

En este capítulo se presentara el diseño separándolo en arquitectura física y la arquitectura de software.

* 1. **Introducción**

Para el diseño de la aplicación se ha optado por el patrón MVC que consta de tres componentes, el modelo, la vista y el controlador.

Este patrón de diseño se fundamente en la separación del código de los tres componentes que acabamos de mencionar permitiendo así que se pueda separar la lógica de negocio, los datos de la aplicación y la interfaz de usuario. Además también se basa en la idea de permitir la reutilización del código y la separación de conceptos facilitando así el desarrollo y mantenimiento de la aplicación.



* 1. **Arquitectura física**

Como se puede observar en la imagen de arriba tenemos tres componentes diferentes en la arquitectura física de nuestra aplicación.

* Los clientes: Usuarios no registrados y Registrados (administradores).
* Servidor web.
* Base de datos.

Los clientes mandan peticiones al servidor el cual consulta a la base de datos, esta devuelve la información solicitada al servidor y este responde al cliente mostrándole los datos solicitados. Con este sistema de arquitectura en ningún momento el cliente accede directamente al servidor ya que esto sería una vulnerabilidad crítica en el sistema.

* 1. **Arquitectura de software**

La arquitectura de nuestro proyecto toma como base el patrón MVC, el cual se caracteriza entre otras cosas por separar los [datos](https://es.wikipedia.org/wiki/Datos) y la [lógica de negocio](https://es.wikipedia.org/wiki/L%C3%B3gica_de_negocio) de la aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres [componentes](https://es.wikipedia.org/wiki/Componentes) distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.[1](https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador#cite_note-1)​[2](https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador#cite_note-burbeck-2)​ Este patrón de [arquitectura de software](https://es.wikipedia.org/wiki/Arquitectura_de_software) se basa en las ideas de [reutilización de código](https://es.wikipedia.org/wiki/Reutilizaci%C3%B3n_de_c%C3%B3digo) y la [separación de conceptos](https://es.wikipedia.org/wiki/Separaci%C3%B3n_de_conceptos), características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

* + 1. **Capa de persistencia**

En esta capa se almacenan los datos de la aplicación, permitiendo realizar las operaciones necesarias para la comunicación entre el servidor web y el servidor de base de datos.

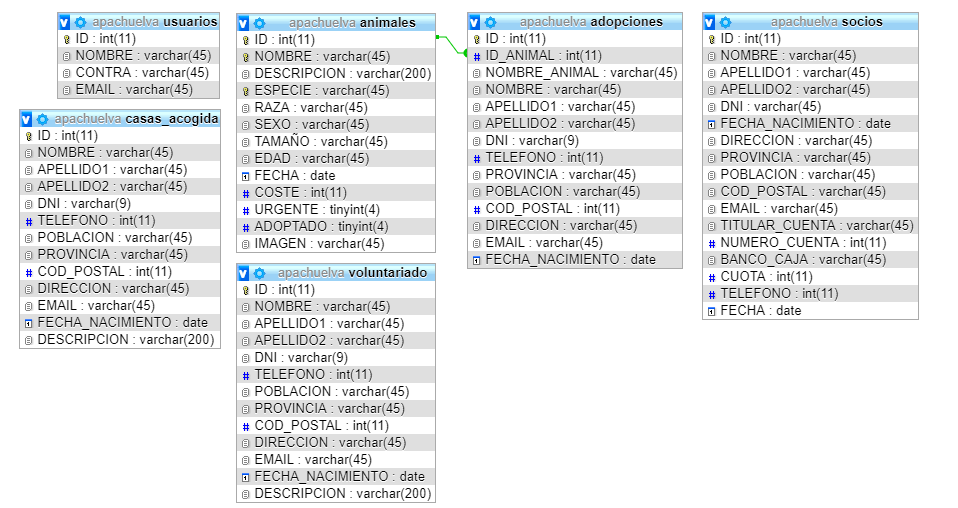
* + - 1. **Modelo**

Los modelos son los encargados de realizar las consultas a la base de datos.

Codeigniter viene preparado con una serie de métodos que tienen como ventaja la movilidad de nuestra aplicación ante cualquier cambio de tipo de base de datos, ya que si cambiamos por ejemplo de MySQL a Oracle no tendremos que cambiar las consultas y será el mismo Framework el que se encargue de definir el formato de la consulta.

* + - 1. **Modelo entidad relación de la BBDD**

Para el almacenaje de datos en nuestra aplicación utilizaremos el siguiente sistema de tablas:



* + - 1. **Descripción detalla de tablas**

**Tabla animales**

|  |  |  |
| --- | --- | --- |
| NOMBRE COLUMNA | TIPO | DESCRIPCION |
| ID | INT | Identificador del animal |
| NOMBRE | VARCHAR | Nombre del animal |
| DESCRIPCION | VARCHAR | Descripción sobre estado o características del animal |
| ESPECIE | VARCHAR | Especie a la que pertenece el animal |
| RAZA | VARCHAR | Raza a la que pertenece el animal |
| SEXO | VARCHAR | Sexo del animal |
| TAMAÑO | VARCHAR | Tamaño del animal, se expresa en string debido a que no se pondrán medidas, se usarán estimaciones(pequeño, mediano…) |
| EDAD | VARCHAR | Edad del animal, se expresa en varchar ya que el cliente lo cree oportuno por la comodidad de añadir si son días, meses o años. |
| FECHA | DATE | Fecha en la que se recogió el animal |
| COSTE | INT | Coste que supone adoptar al animal aquí se refleja el dinero que ha costado las vacunas y medicinas que el animal ha necesitado |
| URGENTE | TINYINT | Boolean que servirá para priorizar la visión de estos animales en el catálogo de adopciones |
| ADOPTADO | TINYINT | Boolean que permitirá la visualización o no del animal en el catálogo |
| IMAGEN | VARCHAR | Nombre del archivo de la imagen del animal |

**Tabla adopciones**

|  |  |  |
| --- | --- | --- |
| NOMBRE COLUMNA | TIPO | DESCRIPCION |
| ID | INT | Identificador de la adopción |
| ID\_ANIMAL | INT | Identificador del animal de la adopción |
| NOMBRE\_ANIMAL | VARCHAR | Nombre del animal de la adopción |
| NOMBRE | VARCHAR | Nombre de la persona adoptante |
| APELLIDO1 | VARCHAR | Primer apellido de la persona adoptante |
| APELLIDO2 | VARCHAR | Segundo apellido de la persona adoptante |
| DNI | VARCHAR | Dni de la persona adoptante |
| TELEFONO | INT | Teléfono de la persona adoptante |
| PROVINCIA | VARCHAR | Provincia donde reside la persona adoptante |
| POBLACION | VARCHAR | Población donde reside la persona adoptante |
| COD\_POSTAL | INT | Código postal donde reside la persona adoptante |
| DIRECCION | VARCHAR | Dirección donde reside la persona adoptante |
| EMAIL | VARCHAR | Email de la persona adoptante |
| FECHA\_NACIMIENTO | DATE | Fecha de nacimiento de la persona adoptante |

**Tabla casas\_acogida**

|  |  |  |
| --- | --- | --- |
| NOMBRE COLUMNA | TIPO | DESCRIPCION |
| ID | INT | Identificador de la adopción |
| NOMBRE | VARCHAR | Nombre de la persona adoptante |
| APELLIDO1 | VARCHAR | Primer apellido de la persona adoptante |
| APELLIDO2 | VARCHAR | Segundo apellido de la persona adoptante |
| DNI | VARCHAR | Dni de la persona adoptante |
| TELEFONO | INT | Teléfono de la persona adoptante |
| PROVINCIA | VARCHAR | Provincia donde reside la persona adoptante |
| POBLACION | VARCHAR | Población donde reside la persona adoptante |
| COD\_POSTAL | INT | Código postal donde reside la persona adoptante |
| DIRECCION | VARCHAR | Dirección donde reside la persona adoptante |
| EMAIL | VARCHAR | Email de la persona adoptante |
| FECHA\_NACIMIENTO | DATE | Fecha de nacimiento de la persona adoptante |
| DESCRIPCION | VARCHAR | Descripción sobre como es el ambiente en su hogar, gracias a este dato podemos evitar que por ejemplo adopte un gato que sea muy asustadizo de otros animales si ya vive con algún otro como un perro |

**Tabla voluntariado**

|  |  |  |
| --- | --- | --- |
| NOMBRE COLUMNA | TIPO | DESCRIPCION |
| ID | INT | Identificador de la adopción |
| NOMBRE | VARCHAR | Nombre de la persona adoptante |
| APELLIDO1 | VARCHAR | Primer apellido de la persona adoptante |
| APELLIDO2 | VARCHAR | Segundo apellido de la persona adoptante |
| DNI | VARCHAR | Dni de la persona adoptante |
| TELEFONO | INT | Teléfono de la persona adoptante |
| PROVINCIA | VARCHAR | Provincia donde reside la persona adoptante |
| POBLACION | VARCHAR | Población donde reside la persona adoptante |
| COD\_POSTAL | INT | Código postal donde reside la persona adoptante |
| DIRECCION | VARCHAR | Dirección donde reside la persona adoptante |
| EMAIL | VARCHAR | Email de la persona adoptante |
| FECHA\_NACIMIENTO | DATE | Fecha de nacimiento de la persona adoptante |
| DESCRIPCION | VARCHAR | Descripción sobre las ayudas que podría facilitar a la asociación participando como socio |

**Tabla socios**

|  |  |  |
| --- | --- | --- |
| NOMBRE COLUMNA | TIPO | DESCRIPCION |
| ID | INT | Identificador de la adopción |
| NOMBRE | VARCHAR | Nombre de la persona adoptante |
| APELLIDO1 | VARCHAR | Primer apellido de la persona adoptante |
| APELLIDO2 | VARCHAR | Segundo apellido de la persona adoptante |
| DNI | VARCHAR | Dni de la persona adoptante |
| TELEFONO | INT | Teléfono de la persona adoptante |
| PROVINCIA | VARCHAR | Provincia donde reside la persona adoptante |
| POBLACION | VARCHAR | Población donde reside la persona adoptante |
| COD\_POSTAL | INT | Código postal donde reside la persona adoptante |
| DIRECCION | VARCHAR | Dirección donde reside la persona adoptante |
| EMAIL | VARCHAR | Email de la persona adoptante |
| FECHA\_NACIMIENTO | DATE | Fecha de nacimiento de la persona adoptante |
| NUMERO\_CUENTA | INT | Número de la cuenta en la que se cargará la cuota mensual |
| BANCO\_CAJA | VARCHAR | Nombre del banco o caja en la que se retirará el dinero |
| CUOTA | INT | Cuota mensual que quiere pagar la persona que desea ser socio, tendrá un mínimo establecido |

**Tabla usuarios**

|  |  |  |
| --- | --- | --- |
| NOMBRE COLUMNA | TIPO | DESCRIPCION |
| ID | INT | Identificador del usuario |
| NOMBRE | VARCHAR | Nombre del usuario |
| CONTRA | VARCHAR | Contraseña del usuario |
| EMAIL | VARCHAR | Email del usuario |

* + 1. **Capa de presentación**

En esta capa es donde se permite al usuario interactuar con el sistema. Se recogen la información de los usuarios y se comunican con la capa lógica. Una vez la capa lógica procesa la información esta genera presentaciones visuales mediante vistas.

* + 1. **Vista**

La vista es la que presenta la información del modelo que nos proporciona el controlador. Estas contienen código HTML con el que se presenta la información al usuario de forma visual.

* + 1. **Capa de lógica**

Esta capa procesa la información que el usuario maneja comunicándosela a los modelos y viceversa. Es la encargada de la comunicación entre vistas y modelos.

**4.3.3.1 Controlador**

El controlador maneja las acciones de la aplicación, gestionando la lógica de negocio y sirviendo así como puente entre las vistas y los modelos.

Los usuarios mediante las vistas realizan acciones que el controlador maneja, este puede gestionar esta información el mismo o delegando al modelo para hacer peticiones a la base de datos por ejemplo. Como su nombre indica controla el flujo de acciones de la aplicación.

1. **Desarrollo**

A continuación desarrollaremos las tecnologías y herramientas con las que hemos contado para la elaboración de la aplicación web.

* 1. **Tecnologías** 
     1. **FrontEnd**

Es la parte de un programa o dispositivo a la que un usuario puede acceder directamente. Son todas las tecnologías de diseño y desarrollo web que corren en el navegador y que se encargan de la interactividad con los usuarios.

* + - 1. **HTML**

HTML, sigla en inglés de HyperText Markup Language (lenguaje de marcas de hipertexto), es un lenguaje de marcado utilizado para el desarrollo de páginas web, para crear y representar visualmente la página. Determina el contenido de la página web, pero no su funcionalidad.

El lenguaje HTML es un estándar a cargo de la World Wide Web Consortium (W3C), organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web.

* + - 1. **CSS**

Siglas en inglés de Cascading Style Sheets, en español "Hojas de estilo en cascada", es un lenguaje de [diseño gráfico](https://es.wikipedia.org/wiki/Dise%C3%B1o_gr%C3%A1fico) para definir y crear la presentación de un documento estructurado escrito en un [lenguaje de marcado](https://es.wikipedia.org/wiki/Lenguaje_de_marcado).[2](https://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada#cite_note-2)​ Es muy usado para establecer el diseño visual de los documentos web

* + - 1. **JavaScript**

Es un [lenguaje de programación](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) [interpretado](https://es.wikipedia.org/wiki/Int%C3%A9rprete_(inform%C3%A1tica)), dialecto del estándar [ECMAScript](https://es.wikipedia.org/wiki/ECMAScript" \o "ECMAScript). Se define como [orientado a objetos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos),[3](https://es.wikipedia.org/wiki/JavaScript#cite_note-3)​ [basado en prototipos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_basada_en_prototipos), [imperativo](https://es.wikipedia.org/w/index.php?title=Programaci%C3%B3n_imperativa&action=edit&redlink=1), débilmente tipado y dinámico.

Se utiliza principalmente en su forma del [lado del cliente (*client-side*)](https://es.wikipedia.org/wiki/Lado_del_cliente), implementado como parte de un [navegador web](https://es.wikipedia.org/wiki/Navegador_web) permitiendo mejoras en la [interfaz de usuario](https://es.wikipedia.org/wiki/Interfaz_de_usuario) y [páginas web](https://es.wikipedia.org/wiki/P%C3%A1gina_web) dinámicas

* + - * 1. **Jquery** Es una [biblioteca](https://es.wikipedia.org/wiki/Biblioteca_(inform%C3%A1tica)) [multiplataforma](https://es.wikipedia.org/wiki/Multiplataforma) de [JavaScript](https://es.wikipedia.org/wiki/JavaScript" \o "JavaScript), creada inicialmente por [John Resig](https://es.wikipedia.org/wiki/John_Resig), que permite simplificar la manera de interactuar con los documentos [HTML](https://es.wikipedia.org/wiki/HTML), manipular el árbol [DOM](https://es.wikipedia.org/wiki/Document_Object_Model), manejar eventos, desarrollar animaciones y agregar interacción con la técnica [AJAX](https://es.wikipedia.org/wiki/AJAX) a [páginas web](https://es.wikipedia.org/wiki/P%C3%A1gina_web).
        2. **SweetAlerts**

Es un plugin de Jquery que reemplaza las antiestéticas alerts para crear alertar mediante ventanas modales con un diseño mucho más cuidado.

La hemos utilizado en la aplicación para hacer más visual las acciones de los usuarios.

* + 1. **BackEnd**

Backend es la capa de acceso a datos de un software o cualquier dispositivo, que no es directamente accesible por los usuarios, además contiene la lógica de la aplicación que maneja dichos datos. El Backend también accede al servidor, que es una aplicación especializada que entiende la forma como el navegador solicita cosas.

* + - 1. **PHP**

Acrónimo recursivo en inglés de PHP: Hypertext Preprocessor (preprocesador de hipertexto), es un [lenguaje de programación](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) [de propósito general](https://es.wikipedia.org/w/index.php?title=Lenguaje_de_programaci%C3%B3n_de_prop%C3%B3sito_general&action=edit&redlink=1) de [código del lado del servidor](https://es.wikipedia.org/wiki/Script_del_lado_del_servidor) originalmente diseñado para el [desarrollo web](https://es.wikipedia.org/wiki/Desarrollo_web) de [contenido dinámico](https://es.wikipedia.org/wiki/Contenido_din%C3%A1mico). Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en un documento [HTML](https://es.wikipedia.org/wiki/HTML) en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera el HTML resultante.

* + - 1. **Codeigniter**

Codeigniter es un potente framwork de PHP muy liviano, construido para desarrolladores que necesitan un kit de herramientas simple y elegante para crear aplicaciones web completas.

Una de las características por las que Codeigniter es una elección sencilla para el desarrollo de la aplicación es que gran parte de la configuración de CodeIgniter se realiza por convención, por ejemplo poner modelos en una carpeta de "modelos". Todavía hay varias opciones de configuración disponibles, a través de scripts en la carpeta "config". Esto nos permite no tener que invertir mucho tiempo de investigación sobre el funcionamiento del framework ya que al ser un framework sencillo nos facilita enormemente el trabajo de configuración.

* + - 1. **MySQL**

 Es un [sistema de gestión de bases de datos](https://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_bases_de_datos) [relacional](https://es.wikipedia.org/wiki/Modelo_relacional) desarrollado bajo licencia dual: [Licencia pública general](https://es.wikipedia.org/wiki/Licencia_P%C3%BAblica_General)/[Licencia comercial](https://es.wikipedia.org/wiki/Software_propietario) por [Oracle Corporation](https://es.wikipedia.org/wiki/Oracle_Corporation) y está considerada como la base datos de [código abierto](https://es.wikipedia.org/wiki/Open_source) más popular del mundo,[1](https://es.wikipedia.org/wiki/MySQL#cite_note-1)​[2](https://es.wikipedia.org/wiki/MySQL#cite_note-2)​ y una de las más populares en general junto a [Oracle](https://es.wikipedia.org/wiki/Oracle_Database) y [Microsoft SQL Server](https://es.wikipedia.org/wiki/Microsoft_SQL_Server), sobre todo para entornos de [desarrollo web](https://es.wikipedia.org/wiki/Desarrollo_web).

* 1. **Herramientas utilizadas**
     1. **PhpMyAdmin**

Para administrar las bases de datos he utilizado principalmente phpMyAdmin a través de su interfaz web.