



PROYECTO DE DESARROLLO DE APLICACIONES WEB

CURSO2021/2022

Gestor empresarial

Alumno/a:

José Manuel CerraVega

Tutor/a:

Rodrigo Serran Vives

AFAFORMACIÓN CONTINUA

CICLO FORMATIVO DE DESARROLLO DE APLICACIONES WEB

ÍNDICE

Tabla de contenido

PROYECTO DE DESARROLLO DE APLICACIONES WEB.....	1
ÍNDICE.....	2
1. INTRODUCCIÓN.....	4
2. ESPECIFICACIÓN DE REQUISITOS.....	6
2.1. Requisitos Específicos.....	6
2.1.1. Requerimientos funcionales	6
2.1.2. Requerimientos de interfaces externas	7
2.1.2.1. Interfaces de usuario.....	7
2.1.3. Requerimientos de rendimiento	12
2.1.4. Obligaciones de diseño.....	13
2.1.5. Atributos.....	14
3. ANÁLISIS.	17
3.1. Diagrama entidad/relación	17
3.2. Modelo relacional	19
3.3. Diagrama de clases	20
3.4. Diagrama de casos de uso	21
4. DISEÑO.....	22
4.1. Introducción.....	22
4.2. Capa de presentación: Vistas.....	23
4.3. Capa de Negocio o Lógica de la Aplicación: Controladores.....	27
4.4. Capa de Persistencia o Datos: Modelos	28
5. IMPLEMENTACIÓN	28
5.1. Tecnologías utilizadas en el desarrollo del proyecto	28
5.2. Descripción del Proyecto	30
5.2.1. Capa de Presentación.	30
5.2.2. Capa de Negocio o Lógica de la Aplicación.....	31
5.2.3. Capa de Persistencia o de Datos.	31
5.3. Plan de Marketing.....	32



6.	EVALUACIÓN	32
6.1.	Validaciones de páginas de Estilo.....	32
6.2.	Validación de Enlaces.....	35
6.3.	Validación de Navegadores.....	36
7.	CONCLUSIÓN	37
8.	BIBLIOGRAFIA.....	39

1. INTRODUCCIÓN.

1.1. Presentación y Objetivos.

El proyecto presentado a continuación consiste en el desarrollo de un sitio web formado por varias páginas y que a través de formularios permita subir archivos a una base de datos, dando la posibilidad de visualizarlos, descargarlos y eliminarlos a través de un sistema protegido mediante usuario y contraseña.

Por ello se podría decir que el objetivo general es crear una página web funcional en la que el administrador pueda gestionar el contenido de la base de datos de su negocio.

Mientras que los objetivos específicos son los siguientes:

- Crear un sistema para proteger una página con usuario y contraseña
- Integrar correctamente instrucciones SQL dentro de un programa PHP.
- Evitar la inyección SQL y el XSS

1.2. Contexto.

El presente proyecto está basado en la premisa de que un cliente ha encargado el desarrollo del mismo.

En el hipotético caso de que dicha aplicación salga a la venta para público, en un principio el potencial mercado serían las denominadas PYMES, aquellas empresas que cuentan con entre 1 y 249 trabajadores.

Teniendo en cuenta las características de dicha empresas se ha decidido que sería en ese mercado donde mejor aceptación podría tener dicha aplicación.

Según los últimos datos ofrecidos por la Dirección General de Industria y de la Pequeña y Mediana Empresa, correspondientes al mes de Noviembre (2021), el número de pymes en España era de 2.933.435, siendo 1.317.423 correspondiente a micro empresas, 158.393 correspondiente a las pequeñas empresas y 25.412 correspondiente a las medianas empresas.

Sector	Nº de Empresas
Agrario	285.782
Industria	174.872
Construcción	332.076
Servicios	2.140.705
Total	2.933.435

Tabla 1 Datos sobre las pymes

Como se puede ver en Tabla 1, en España existen un total de 2.933.435 empresas, y aunque en un principio la aplicación no contaría con las características necesarias para adaptarse a cada una de ellas, si tenemos en cuenta los posibles añadidos y mejoras que se podrían realizar en la aplicación, se obtiene un gran nicho de mercado de donde poder obtener potenciales clientes.

2. ESPECIFICACIÓN DE REQUISITOS.

2.1. Requisitos Específicos.

Teniendo en cuenta los objetivos especificados anteriormente se han definido una serie de requisitos indispensables para el correcto desarrollo de la aplicación.

2.1.1. Requerimientos funcionales

Según Michael Arias (2005) los requerimientos funcionales definen las funciones que el sistema será capaz de realizar.

Si aplicamos dicha definición a las características del proyecto actual, obtenemos los siguientes requisitos:

- Se pedirá usuario y contraseña para iniciar sesión.
- La contraseña estará cifrada.
- Cada vez que se agregue, modifique o elimine un registro se avisara si el procedimiento fue un éxito o no.
- Habrá varios tipos de usuarios, siendo el administrador el de mayor rango.

- Las diferentes páginas y opciones serán visibles según los permisos.
- Con respecto a la base de datos:
 - Se podrá añadir nuevos archivos
 - Se podrá eliminar los existentes
 - Se podrá descargar los existentes
 - Se podrá dar de alta o baja a nuevos usuarios

2.1.2. Requerimientos de interfaces externas

- En cuanto a la interfaz de usuario, esta deberá ser intuitiva y fácil de usar.
- Los elementos de dicha interfaz variaran según el tipo de usuario y el rol que desempeña.
- Al tratarse de una aplicación web los usuarios deberán acceder a ella a través de un navegador con conexión a internet, para ello la aplicación deberá estar alojada en un hosting que permita su funcionamiento sin interrupciones y riesgos de seguridad.

2.1.2.1. Interfaces de usuario

Podemos definir una interfaz de usuario como el medio a través del cual el usuario puede interactuar con la aplicación.

Según Carlos Marrero Expósito (2006) hablamos de interfaz de usuario, para referirnos de forma genérica al espacio que media la relación de un sujeto y un ordenador o sistema interactivo

Dicha interfaz debe cumplir una serie de requisitos:

- Simple
- Clara
- Coherente
- Eficiente

Actualmente acceder a la web se ha convertido en una tarea habitual en nuestras vidas, no solo los podemos acceder a través de dispositivos tradicionales como pueden ser un ordenador, una tableta o un smartphone, cada vez más dispositivo son capaces de acceder.

Por ello a la hora de diseñar e implementar nuestra aplicación web debemos tener en cuenta a los navegadores web y el diseño responsive.

Según Danny Murillo (2017) un navegador web es un software que permite acceder a la web, interpretando la información y visualizando los sitios web.

También debemos mencionar que no es lo mismo un navegador web que un buscador, hay que diferenciarlos. Como se menciona anteriormente un navegador nos permite acceder a la web, mientras que un buscador nos permite buscar la información que nos interesa.

Una vez aclarado esto podemos afirmar que existen un gran número de navegadores web, cada uno cuenta con características que hay que tener en cuenta a la hora de diseñar nuestra aplicación.

Según la web de statista los 3 navegadores más utilizados en el 2021 han sido Chrome, safari y Firefox, acumulando el 86,69% de la cuota de mercado, respondiendo el 63.59% a Chrome, 19.46 a safari y el 3.64 a Firefox.

En las siguientes tablas podemos observar algunas de las características de los diferentes navegadores.

FUNCIONES	Chrome	Safari	Firefox
Sincronización en la nube	Sí	Sí	Sí
Gestor de descargas	Sí	Sí	Sí
Navegación privada	Sí	Sí	Sí
Modo de pantalla completa	Sí	Sí	Sí
Pestañas verticales	Sí	No	Sí
Extensiones personalizadas	Sí	Sí	Sí

Nota. De A. Mora ,2022, Comparativa de navegadores web.

Figura 2 Comparativa navegadores web

COMPATIBILIDAD	Chrome	Safari	Firefox
Windows	Sí	No	Sí
macOS	Sí	Sí	Sí
Linux	Sí	No	Sí
Android	Sí	No	Sí
iOS	Sí	Sí	Sí

Nota. De A. Mora ,2022, Comparativa de navegadores web.

Como se puede observar en las figuras 1 y 2, el navegador que se encuentra más limitado en cuanto a funciones y compatibilidad es safari, si tenemos eso en cuenta, en el caso del proyecto actual, se ha decidido desarrollar la aplicación orientada principalmente al

navegador Google Chrome, para tomar dicha decisión también se ha considerado que es el navegador más utilizado.

Como se ha mencionado anteriormente, también debemos tener en cuenta el diseño responsive.

Según Marcos y González (2013) el diseño responsive comprende una serie de técnicas y pautas de diseño que permiten adaptar sitios web al entorno de navegación, comprendiendo el entorno de navegación como los diferentes dispositivos por medio de los cuales los usuarios pueden acceder y navegar en internet.

Este tipo de diseño nos ofrece una serie de ventajas:

- Mejora el posicionamiento SEO
- Mejora la experiencia de usuario
- Evita duplicidad de contenido
- Reduce costes.

2.1.3. Requerimientos de rendimiento

De cara al rendimiento se deberán establecer una serie de requisitos y márgenes de error que no deberán ser sobrepasados por la aplicación.

Por ejemplo:

- El sistema debe tener una alta disponibilidad cuando un usuario intente acceder.

Como se ha mencionado anteriormente, la web estará dirigida a las PYMES, sin embargo, basamos en una empresa que cuente con unos 250 trabajadores y tenemos en cuenta que se espera que el 50% de los usuarios use el sistema, es decir, se dejan 125 posibles usuarios.

Teniendo en cuenta el tamaño y tipo de aplicación de cara a la comercialización se recomienda utilizar un VPS.

A la hora de escoger este servicio se ha tenido en cuenta que mediante el uso de un VPS se obtienen las siguientes ventajas:

- Recursos dedicados.
- Mayor seguridad
- Control total
- Mejor escalabilidad

2.1.4. Obligaciones de diseño

Como se ha dicho anteriormente la aplicación deberá ser intuitiva y fácil de usar, pero sin dejar de lado el diseño.

El diseño de la aplicación deberá ser elegante teniendo un aspecto lo más profesional posible, facilitando su uso en la medida de lo posible, esto es debido a que si

tenemos en cuenta que según un informe de la Organización Mundial de la Salud (OMS) y el Banco Mundial el 15% de la población mundial tiene alguna discapacidad, o lo que es lo mismo, 15 de cada 100 personas podrían encontrar dificultades para acceder al sitio web.

Dicho esto, la aplicación deberá de respetar una serie de estándares de accesibilidad de la W3C, por ejemplo:

- Pauta 3.1 Legible: Hacer que los contenidos textuales resulten legibles y comprensibles.
- Pauta 2.4 Navegable: Proporcionar medios para ayudar a los usuarios a navegar, encontrar contenido y determinar dónde se encuentran.

2.1.5. Atributos

Según Mendoza y Moreno (2011) la seguridad informática consiste en mantener protegida la información de un sistema informático, además afirman que la seguridad está basada en tres principios:

- Confidencialidad
- Integridad
- Disponibilidad

Teniendo en cuenta que en este proyecto se emplearán una serie de formularios para subir información a una base de datos, los posibles ataques serían de inyección de

código sql y de Cross-site scripting.

Según Chicaiza, Ponce y Velásquez (2020) un ataque de inyección SQL se produce mediante la inserción de una consulta SQL a través de los datos de entrada del cliente de la aplicación.

En caso de que dicho ataque sea exitoso, el atacante podría leer y modificar la información de la base de datos, llegando en algunos casos a ejecutar operaciones de administración.

Por otro lado, según José Enrique Miramontes (2011) los ataques de Cross site scripting ocurren cuando una aplicación recibe y muestra datos en el navegador sin antes haberlos validado o codificado apropiadamente.

Existen dos tipos de ataques cross site scripting:

- Reflejado: el código malicioso no se almacena en la web, sino que envía dentro de un enlace para que la víctima pinche en él.
- Persistente: El código se queda almacenado en el servidor y afecta a cualquier usuario que entre en él. Un buen ejemplo de este tipo de ataque sería en un foro, el código quedaría almacenado en la entrada, afectando a cualquier usuario que la visualice.

Este tipo de ataques podrían producir:

- Redirecciones a sitios dañinos
- Instalar programas malignos



- Robo de sesiones de usuario
- Robo de información sensible

En el caso del proyecto actual se establecerán una serie de mecanismos que permitan evitar dichos ataques, también se deberá garantizar la privacidad de la aplicación mediante el acceso por usuario y contraseña.

3. ANÁLISIS.

3.1. Diagrama entidad/relación

En el apartado siguiente se encuentran representados en varios tipos de diagramas los diferentes elementos que componen la aplicación.

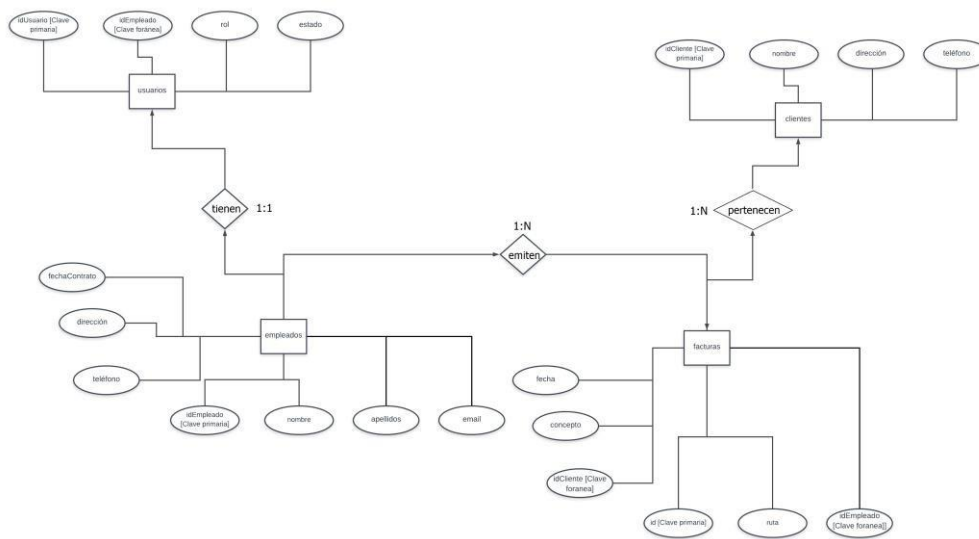


Figura3 DiagramaEntidad/Relación

Un diagrama entidad-relaciones un tipo de diagrama que ilustra a las personas, objetos y conceptos cómo entidades, así como su relación dentro del sistema. Mediante su uso se simplifica la tarea de diseñar y representar una base de datos.

Como se puede observar en la figura 3, en él se representan los atributos y relaciones de aquellas entidades que forman parte de la base de datos y que cuentan con algún tipo de relación:

- Usuarios: Relacionada con la tabla empleados, en ella se guardarán los datos de acceso a la aplicación, teniendo una relación 1:1, es decir, solo puede existir un usuario por empleado.
- Empleados: Relacionada con la tabla Usuarios, en ella se guardarán los datos de los empleados.
- Clientes: En ella se guardaran los datos de los clientes de empresa, teniendo una relación 1: N, es decir que un cliente puede contar con muchas facturas.
- Facturas: En ella se guardaran los datos de las facturas emitidas por la empresa, está relacionada con la tabla clientes.

3.2. Modelo relacional

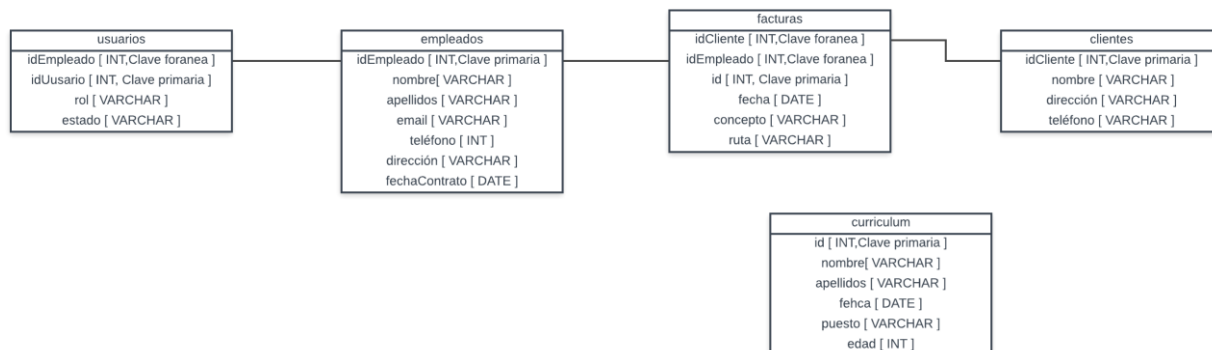


Figura4Modelorelacional

El modelo relacional permite representar una base de datos mediante el uso de tablas relacionadas.

Como se pueden ver en la figura 8 las tablas de nuestra base de datos se encuentran relacionadas por claves foráneas que hacen referencia a una columna en otra tabla, dicha columna debe ser la clave primaria en la tabla referenciada.

También como se puede observar en la figura 4, encontramos una tabla denominada “curriculum” que no guarda relación con el resto de las tablas, esto es debido a que en ella guardarán los datos y el curriculum de los solicitantes a un puesto en la empresa.

3.3. Diagrama de clases

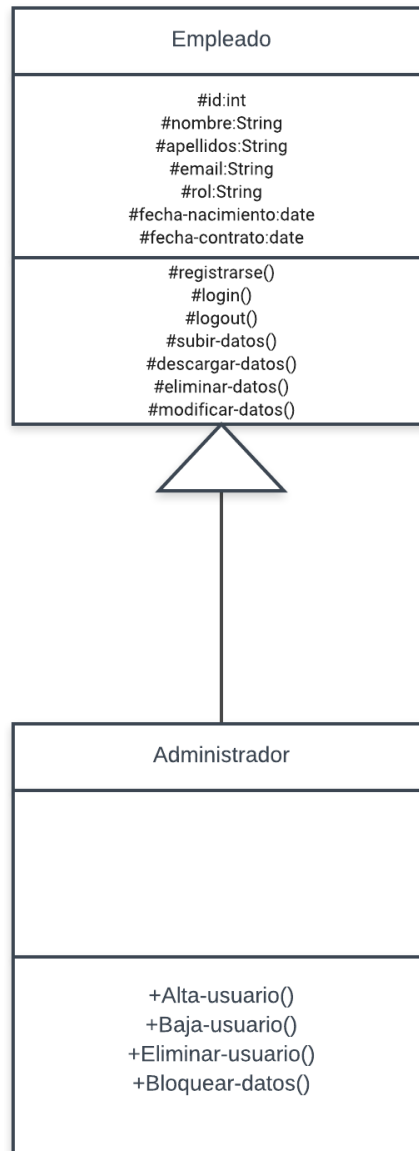


Figura5Diagramadeclasses

Un diagrama de clases permite diseñar la estructura de un sistema modelando sus

clases, atributos, operaciones y relaciones entre los objetos. Tomando la figura 5 como ejemplo, podemos observar una serie de elementos:

- + Especifica que dicho elemento cuenta con una visibilidad de tipo público.
- # Especifica que dicho elemento cuenta con una visibilidad de tipo protegido.
- \triangle Especifica que existe una relación de herencia entre los objetos.

3.4. Diagrama de casos de uso

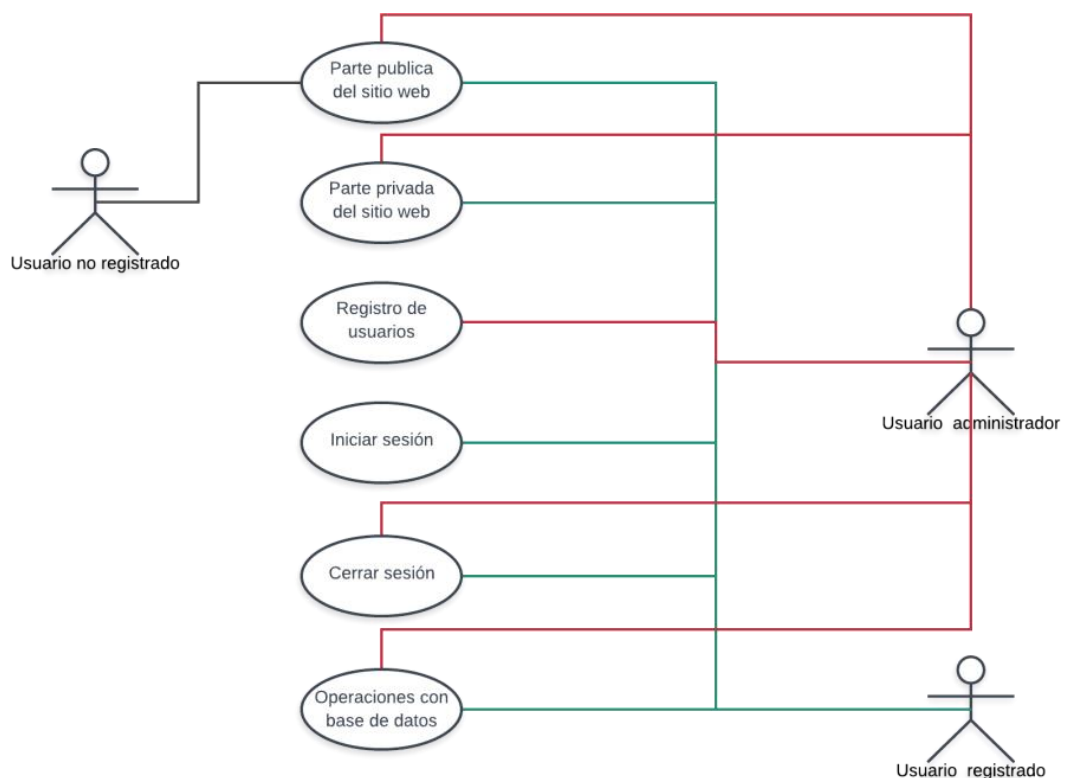


Figura6 Diagramadecasosdeuso

Un diagrama de casos de uso permite presentar los actores externos que interactúan con el sistema, mostrando de forma visual las funciones de los diferentes usuarios.

Tomando como referencia el diagrama mostrado en la figura 6, podemos diferenciar tres tipos de actores:

- Usuario no registrado: Solo puede realizar una acción
- Usuario registrado: Puede realizar una serie de acciones en función de sus permisos
- Usuario administrador: Puede realizar cualquier función dentro del sistema.

4. DISEÑO

4.1.Introducción.

El un patrón Modelo Vista Controlador fue introducido por primera vez por Trygve Reenskaug en Smalltalk-76, siendo diseñado con el objetivo reducir el esfuerzo necesario para crear y mantener un sistema.

Según Trygve M. H. Reenskaug, el patron modelo-vista-controlador fue creado con el objetivo de dar a los usuarios control sobre su información vista desde múltiples perspectivas(*WelcometothepagesofTrygve M. H. Reenskaug, s. f.*)

Para ello el MVC separa la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones, esto es conseguido utilizando tres componentes distintos que son el modelo, la vista y el controlador.

Según Eugenia Bahit (2011) el funcionamiento de la modelo vista controlador es el siguiente:

- El usuario realiza una petición
- El controlador captura el evento
- Hace la llamada al modelo/modelos
- El modelo será el encargado de interactuar con la base de datos
- El controlador recibe la información y la envía a la vista
- La vista, procesa esta información

4.2.Capa de presentación: Vistas

Las vistas definen cómo se deben mostrar los datos de la aplicación, es decir, es donde se encontrarán todos los elementos de la interfaz de usuario.

En el caso del proyecto actual, las vistas son un conjunto de páginas HTML con contenido dinámico sobre el cual los usuarios registrados pueden realizar operaciones,

dentro de esta categoría también podemos agrupar a las hojas de estilo css y los ficheros Java Script.

Se ha desarrollado un boceto de las vistas de la aplicación empleando la herramienta figma.

A continuación, se procederá a explicar cada una de las partes:

- Figura 7: Login.

Se trata de la primera zona que visualizaremos al acceder, en ella los usuarios introducirán sus credenciales para ingresar a la aplicación web.

- Figura 8: Tareas

En ella el usuario podrá visualizar diferentes elementos relacionados con su trabajo.

- Figura 9: Perfil

En esta zona el usuario podrá editar los diferentes elementos que corresponden a su perfil de usuario.

- Figura 10: Usuarios

En este apartado se podrán visualizar los diferentes usuarios dados de alta en el sistema, así como la información relacionada con los mismos.

- Figura 11: Archivos

En este apartado se podrán visualizar los datos relacionados con los archivos, tanto los curriculum como de las facturas emitidas.

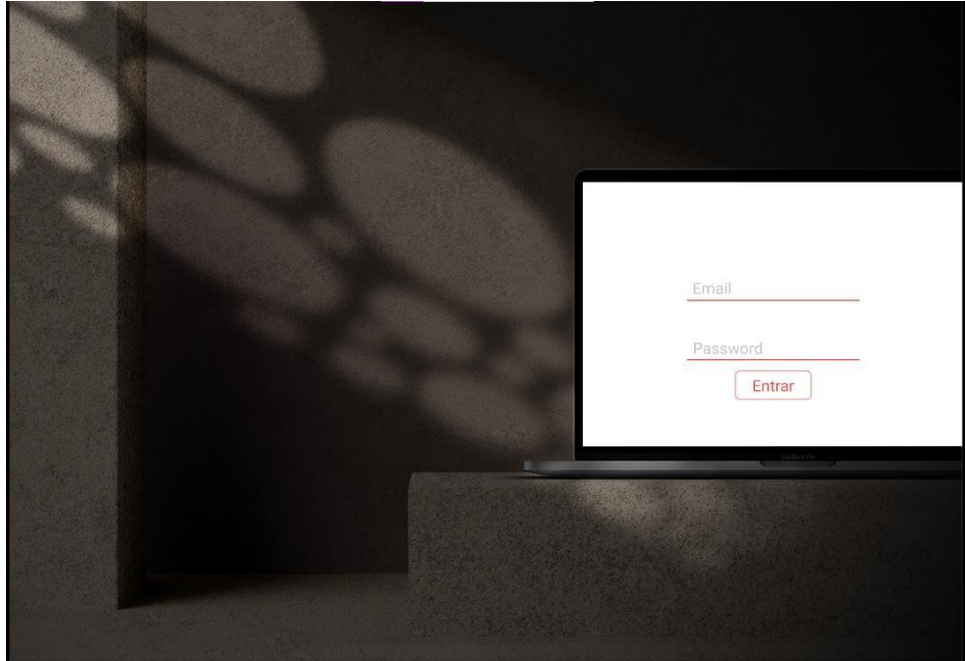


Figura 7

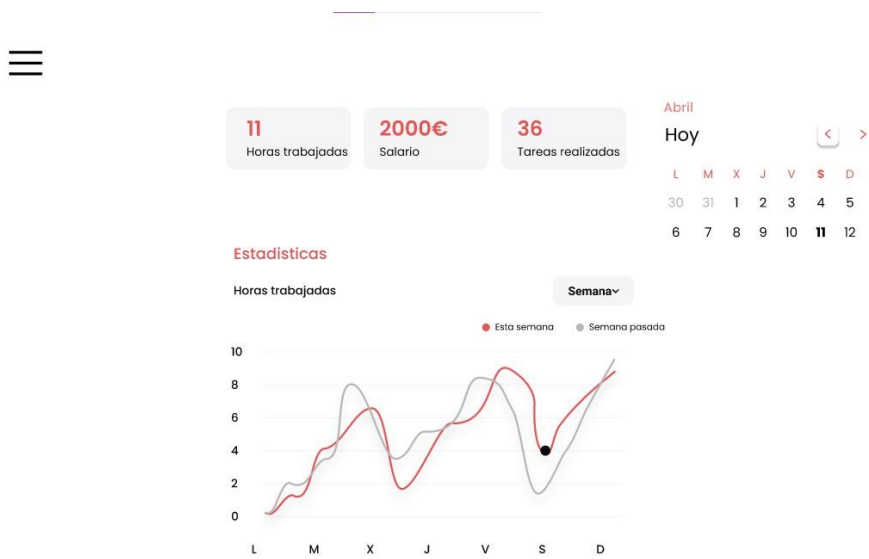




Figura 8



- Perfil
- Tareas
- Inventario
- Usuarios
- Archivos
- Salir

Imagen  **Subir**
JPG,PNG, Tamaño máximo 20KB

Email

Nombre

Número de telefono

Contraseña actual

Nueva contraseña

Confirmar

Figura 9









NOMBRE	EMAIL	DIRECCIÓN	TELEFONO	ULTIMA CONEXIÓN	ROL
 Ekaterina Tankova	ekaterina.tankova@devias.io	Parkersburg, West Virginia, USA	304-428-3097	12/04/2019	12/04/2019
 Ekaterina Tankova	ekaterina.tankova@devias.io	Parkersburg, West Virginia, USA	304-428-3097	12/04/2019	12/04/2019
 Ekaterina Tankova	ekaterina.tankova@devias.io	Parkersburg, West Virginia, USA	304-428-3097	12/04/2019	12/04/2019
 Ekaterina Tankova	ekaterina.tankova@devias.io	Parkersburg, West Virginia, USA	304-428-3097	12/04/2019	12/04/2019
 Ekaterina Tankova	ekaterina.tankova@devias.io	Parkersburg, West Virginia, USA	304-428-3097	12/04/2019	12/04/2019
 Ekaterina Tankova	ekaterina.tankova@devias.io	Parkersburg, West Virginia, USA	304-428-3097	12/04/2019	12/04/2019

Figura 10

NOMBRE DE ARCHIVO	ENTIDAD EMISORA	ENTIDAD RECEPTORA	FECHA	VALOR TOTAL	
Lorem ipsum dolor sit	Lorem ipsum dolor sit	Lorem ipsum dolor sit	Lorem ipsum dolor sit	Lorem ipsum dolor sit	 
Lorem ipsum dolor sit	Lorem ipsum dolor sit	Lorem ipsum dolor sit	Lorem ipsum dolor sit	Lorem ipsum dolor sit	 
Lorem ipsum dolor sit	Lorem ipsum dolor sit	Lorem ipsum dolor sit	Lorem ipsum dolor sit	Lorem ipsum dolor sit	 
Lorem ipsum dolor sit	Lorem ipsum dolor sit	Lorem ipsum dolor sit	Lorem ipsum dolor sit	Lorem ipsum dolor sit	 
Lorem ipsum dolor sit	Lorem ipsum dolor sit	Lorem ipsum dolor sit	Lorem ipsum dolor sit	Lorem ipsum dolor sit	 
Lorem ipsum dolor sit	Lorem ipsum dolor sit	Lorem ipsum dolor sit	Lorem ipsum dolor sit	Lorem ipsum dolor sit	 

Figura 11

También se puede visualizar de forma online a través del siguiente enlace: <https://www.figma.com/file/uDuQrPfl6IJZPy7ExxwHSI/Untitled>

4.3. Capa de Negocio o Lógica de la Aplicación: Controladores

Los controladores son los encargados de manejar y responder a las solicitudes del usuario, procesando su información e interactuando con el modelo, es decir, actúan como intermediarios entre el sistema y el usuario.

Si tomamos como ejemplo el proyecto actual, el controlador sería el encargado de controlar los formularios de entrada y los botones que nos pueden permitir crear o eliminar elementos de la base de datos.

4.4. Capa de Persistencia o Datos: Modelos

El modelo es donde se guarda toda la lógica del negocio de una aplicación, suele ser independiente de cómo se desea que se recojan los resultados o cómo se presenten.

Por ejemplo, en el caso del proyecto actual podremos encontrar que en el modelo existirán las clases como Factura o Cliente.

En cuanto a la aplicación del modelo MVC en el proyecto actual, se desarrollarán una serie de vistas a través de las cuales se podrá interactuar con la aplicación, en función de cada vista se crearán los controladores y modelos necesarios para un mejor funcionamiento y organización de la misma.

5. IMPLEMENTACIÓN

5.1. Tecnologías utilizadas en el desarrollo del proyecto

Antes de abordar las tecnologías utilizadas debemos dejar claros los conceptos de framework y base de datos.

Un framework es un esquema o marco de trabajo que permite agilizar y simplificar el desarrollo de un proyecto, es decir, una guía para la creación de nuestra aplicación.

Por otro lado una base de datos se puede definir como un conjunto de información relacionada con el mismo contexto, organizada para facilitar el acceso y la gestión.

Para el desarrollo de este proyecto se van a utilizar las siguientes tecnologías:



Laravel

Se trata de un framework PHP de código abierto que nos permite desarrollar aplicaciones en este lenguaje.

Se ha decidido utilizarlo por las siguientes razones:

- Cuenta con una documentación muy completa, además de una amplia comunidad de desarrolladores.
- Sigue el patrón Modelo-Vista-Controlador(MVC)
- Permite actualizar y migrar una base de de forma rápida y sencilla.
- Facilidad de mantenimiento y escalabilidad.

Bootstrap

Bootstrap es un framework de código abierto que facilita un desarrollo sencillo y rápido. Permite crear interfaces elegantes y compatibles con cualquier dispositivo.

Entre las ventajas que ofrece podemos remarcar las siguientes:

- Cuenta con una documentación completa
- Es de código abierto.
- Facilita el diseño responsive.
- Compatible con la mayor parte de navegadores.
- Cuenta con una serie de componentes que facilitan el desarrollo.

Se trata de un sistema de gestión de bases de datos relacionales de código abierto.

Cuenta con una serie de ventajas:

- Es rápida.
- Fácil de usar.
- Soporta bases de datos de gran tamaño

5.2.Descripción del Proyecto

Anteriormente se han definido las diferentes capas que componen el modelo MVC, a continuación como se adaptan al proyecto actual.

5.2.1. Capa de Presentación.

Compuesta por las vistas que los usuarios podrán visualizar en función de sus permisos, un total de 10:

Vistas principales:

- Facturas
- Clientes
- Curriculum
- Usuarios
- Personal
- Estadísticas

Vistas secundarias:

- Editar
- Editar Personal
- Editar Cliente
- Perfil

5.2.2. Capa de Negocio o Lógica de la Aplicación.

En este apartado se recogerán los diferentes controladores que contendrán las funciones en las que se basará el proyecto.

Un total de 5 controladores:

- cvController
- facturasController
- clientesController
- userController
- empleadosController

5.2.3. Capa de Persistencia o de Datos.

Compuesta por los modelos que formarán parte del proyecto, en este caso se creará un modelo por cada entidad de la base de datos.

Un total de 5:

- Clientes

- Facturas
- Empleados
- Usuarios
- curriculum

5.3. Plan de Marketing

Como se ha mencionado anteriormente, el presente proyecto se basa en la premisa de que el proyecto se desarrolla para un cliente en específico, por lo que no se ha necesitado la elaboración de un plan de Marketing.

6. EVALUACIÓN

El presente proyecto será sometido a una serie de evaluaciones para garantizar que cumple con una serie de parámetros de calidad.

6.1. Validaciones de páginas de Estilo.

La mayor parte de las páginas web están escritas en HTML, un lenguaje que de marcado que utiliza etiquetas para estructurar el contenido de la web.

Para mejorar el diseño y presentación de dicha estructura se utilizan las hojas de estilo o CSS.

Mediante el uso de estas hojas de estilo podemos definir como luce la página web, personalizando aspectos como los colores, tipo de letra...

Para validar los estilos del proyecto actual se utilizará el servicio de validación de CSS de la W3C, un software libre creado para ayudar a los diseñadores y desarrolladores

web a validar sus estilos.

Esta herramienta está disponible en <https://jigsaw.w3.org/css-validator/>.

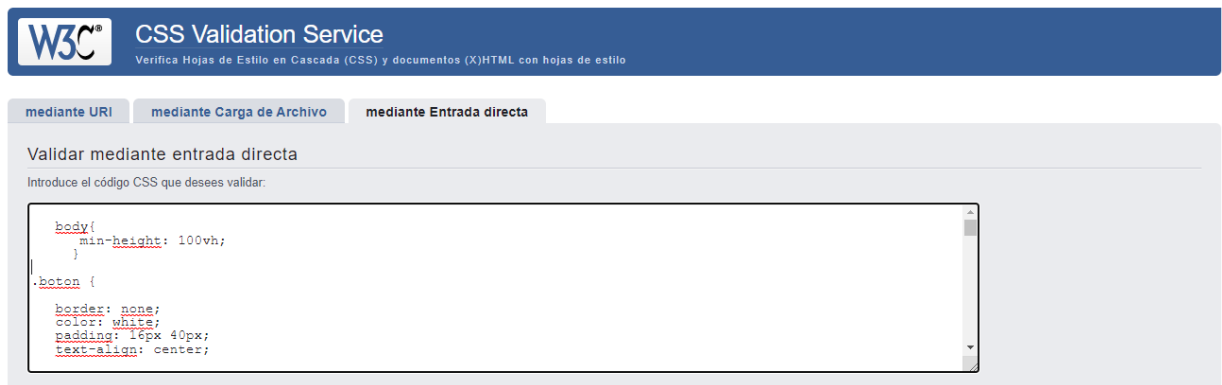


Figura 12

Como podemos ver en la Figura 12, podemos validar nuestros estilos de 3 formas diferentes:

- Mediante URI

Si utilizamos esta opción tendremos que introducir la url del sitio web que deseamos validar.

Como el proyecto actual está desarrollado en local es imposible utilizar esta opción, para comprobar su funcionamiento se ha utilizado la url <https://afa-formacion.com/quienes-somos>.

Resultados del Validador CSS del W3C para <https://afa-formacion.com/quienes-somos> (CSS versión 3 + SVG)

Disculpas! Hemos encontrado las siguientes errores (17)		
URI : https://afa-formacion.com/quienes-somos		
20	Property <code>background-repeat-y</code> doesn't exist. The closest matching property name is <code>background-repeat</code> : <code>no-repeat</code>	
URI : https://afa-formacion.com/wp-content/plugins/divi-essential/styles/style.min.css?ver=1.0.0		
1	Faltan valores o no se reconocen los valores : <code>cubic-bezier(0.075,0.135,0.135,0.15,0.86)</code>	
1	Faltan valores o no se reconocen los valores : <code>cubic-bezier(0.075,0.135,0.135,0.15,0.86)</code>	
1	<code>0.35</code> no es un valor de <code>transition</code> : <code>all 0.35 ease-in-out</code>	
1	<code>0</code> no es un valor de <code>transition-delay</code> : <code>0</code>	
1	<code>0</code> no es un valor de <code>transition-delay</code> : <code>0</code>	
1	<code>0</code> no es un valor de <code>transition-delay</code> : <code>0</code>	
1	<code>0</code> no es un valor de <code>transition-delay</code> : <code>0</code>	
1	<code>0</code> no es un valor de <code>transition-delay</code> : <code>0,0s</code>	
1	<code>0</code> no es un valor de <code>transition-delay</code> : <code>0</code>	
1	Propiedad no válida : <code>font-weight 400px</code> no es un valor de <code>font-weight</code> : <code>400px</code>	
1	Propiedad no válida : <code>visibility none</code> no es un valor de <code>visibility</code> : <code>none</code>	
1	La propiedad <code>text-stroke</code> no existe : <code>10px</code>	

Figura 13

Como se puede observar en la figura 13, al validar la página nos muestra los errores cometidos en las hojas de estilo, así como la hoja en la que se producen.

- Mediante carga de archivo

En este caso debemos subir nuestras hojas de estilo para la validación.

- Mediante entrada directa

En caso de utilizar este método, simplemente debemos copiar y pegar nuestro código CSS en el validador.

Resultados del Validador CSS del W3C para TextArea (CSS versión 3 + SVG)

¡Enhorabuena! No error encontrado.

¡Este documento es [CSS versión 3 + SVG](#) válido!

Puede mostrar este icono en cualquier página que valide para que los usuarios vean que se ha preocupado por crear una página Web interoperable. A continuación se encuentra el XHTML que puede usar para añadir el icono a su página Web:



Figura 14

Como se puede ver en la figura 14 la aplicación nos indica que los estilos introducidos son correctos, dándonos también la opción de mostrar un icono en nuestras páginas web que indica que los estilos validados son correctos.

6.2. Validación de Enlaces.

Para comprobar que los enlaces mostrados en el proyecto actual son seguros, se empleara la herramienta Dr. Link Check, disponible en <https://www.drlinkcheck.com/>.

Por ejemplo, se va comprobar el funcionamiento de la herramienta utilizando el enlace:

- <https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/js/bootstrap.bundle.min.js>

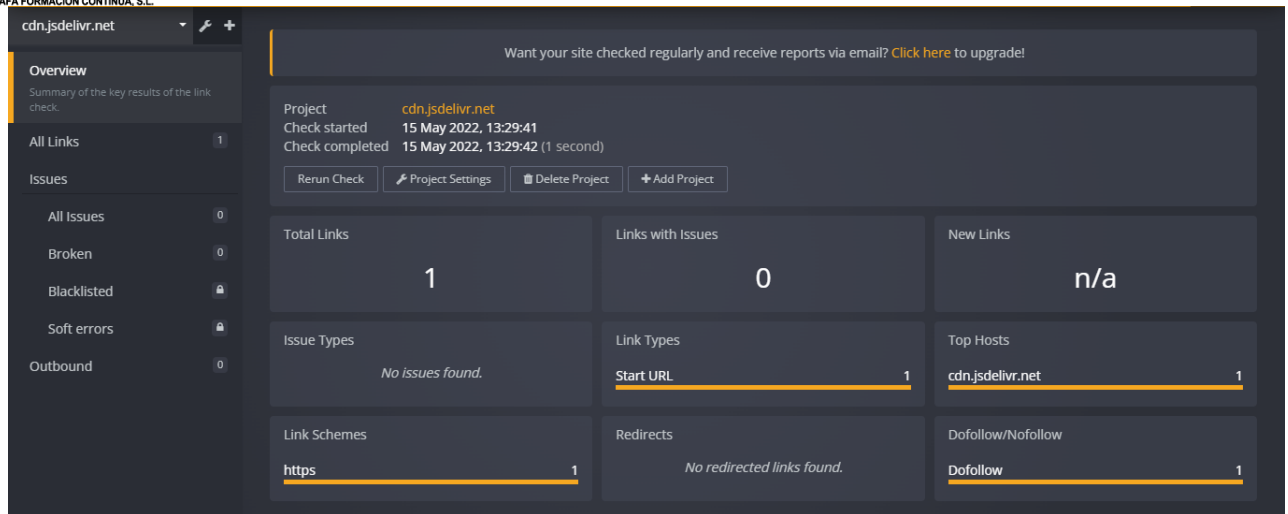


Figura 15

Como se puede ver en la figura 15, el enlace introducido no cuenta con ningún tipo de vulnerabilidad.

6.3. Validación de Navegadores.

A la hora de desarrollar cualquier tipo de aplicación web, uno de los mayores problemas es saber si las tecnologías que vamos a utilizar son compatibles con los distintos navegadores.

Para comprobar esto se va a utilizar la herramienta caniuse, disponible en <https://caniuse.com/>.

Esta herramienta nos permitirá buscar y comprobar el grado de compatibilidad de una tecnología con los diferentes navegadores.

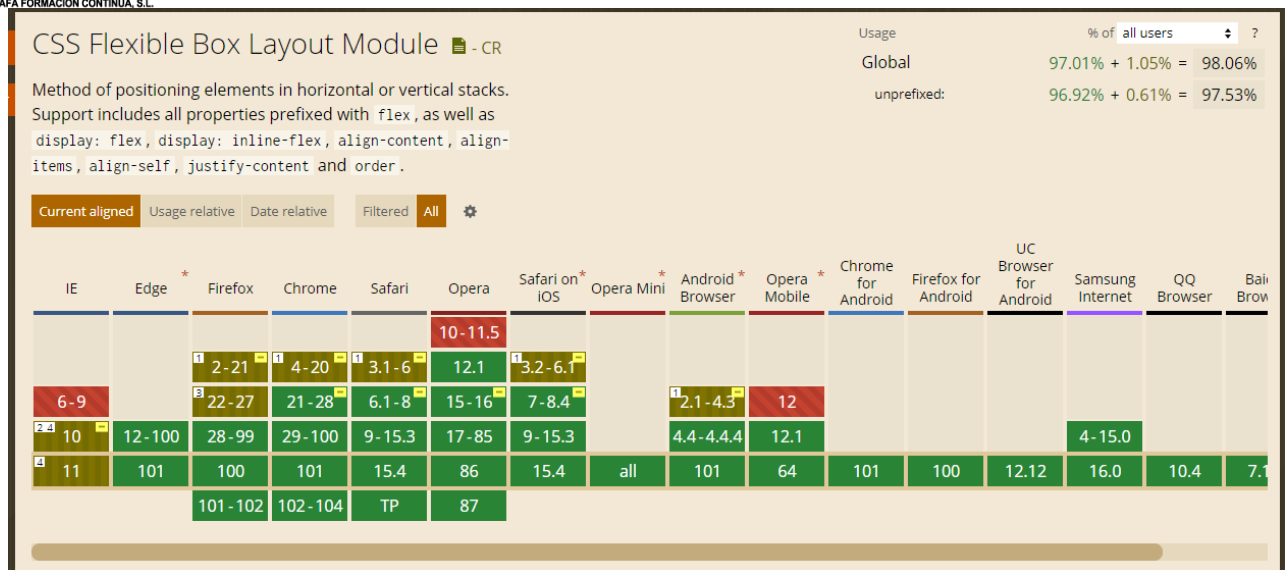


Figura 16

Para mostrar el funcionamiento de la herramienta se ha comprobado la tecnología Flexbox, como se puede observar en la figura 16, se muestran los niveles de compatibilidad con el navegador y con sus diferentes versiones.

7. CONCLUSIÓN

En este último apartado se abordaran las conclusiones obtenidas durante el proceso de desarrollo.

Se han cumplido todos los objetivos especificados al principio del desarrollo del proyecto, los principales:

- Proteger pagina con usuario y contraseña
- Incluir sentencias sql en php
- Proteger la aplicación contra ataques de inyección de sql y xss

Ha sido posible cumplir todos ellos gracias al framework utilizado para el desarrollo, que ha proporcionado las funciones necesarias ello.

En cuanto al desarrollo técnico, fue complicado ejecutar todas las ideas que se habían esbozado antes del inicio de desarrollo, siendo de suma importancia una primera fase de análisis y diseño, a través de la cual se decidió que ideas desarrollar y cuales descartar, todo ello teniendo la cuenta el tiempo disponible para el desarrollo.

Además gracias al presente proyecto se ha aprendido a desarrollar un proyecto de principio a fin y se han ampliado conocimientos sobre las tecnologías utilizadas.

En cuanto a las posibles ampliaciones se pueden contemplar varias:

- Por un lado añadir una capa extra de seguridad en el momento del despliegue de la aplicación
- Por otro lado añadir nuevas funcionalidades como:
 - Dar la posibilidad de crear las facturas desde la propia aplicación.
 - Añadir un sistema para asignar tareas a los usuarios.
 - La posibilidad de personalizar el aspecto de la interfaz por parte del propio administrador.

8. BIBLIOGRAFIA

Ministerio de Industria, Comercio y Turismo ,Dirección General de Industria y de la Pequeña y Mediana Empresa. (2021). Cifras PYME. Datos noviembre 2021.Gobierno de España.

<http://www.ipyme.org/es->

[ES/ApWeb/EstadisticasPYME/Documents/CifrasPYME-noviembre2021.pdf](http://www.ipyme.org/es-ES/ApWeb/EstadisticasPYME/Documents/CifrasPYME-noviembre2021.pdf)

Mundial,B.(2011).Informe mundial sobre la discapacidad 2011.

Welcome to the pages of Trygve M. H. Reenskaug. (s. f.). folk.universitetetioslo.no. Recuperado 1 de mayo de 2022, de <https://folk.universitetetioslo.no/trygver/>

Carrera Guanoluisa, J. G. (2014). Análisis comparativo de la productividad entre los patrones de diseño Modelo Vista Controlador (MVC) y Modelo Vista Presentador (MVP) aplicado al desarrollo del Sistema Nómina de Empleados y Rol de Pagos de la Distribuidora Soria CA (Bachelor'sthesis).

González, D., & Marcos, M. C. (2013). Responsive web design: diseño multidispositivo para mejorar la experiencia de usuario. *BiD: textos universitaris de biblioteconomia i documentació*. 2013; 31.

Ranking de los navegadores de Internet con mayor cuota de mercado mensual entre enero de 2016 y enero de 2021. (2021, febrero). <https://es.statista.com/>. Recuperado

9 de mayo de 2022, de <https://es.statista.com/estadisticas/600249/cuota-de-mercado-mensual-de-los-principales-navegadores-de-internet/>

Mora, A. (2022, 22 febrero). *Comparativa de navegadores web* [Tabla].
www.pcworld.es. <https://www.pcworld.es/mejores-productos/internet/mejores-navegadores-web-3672988/>

Chaves, M. A. (2005). La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. *InterSedes: Revista de las Sedes Regionales*, 6(10), 1-13.

Expósito, C. M. (2006). *Interfaz Gráfica de Usuario. Aproximación semiótica y cognitiva*, 1.

Mendoza, M. A., & Moreno Patiño, P. J. (2011). Desarrollo de una propuesta metodológica para determinar la seguridad en una aplicación web.

Chicaiza, G., Ponce, L., & Campos, G. V. (2020). Inyección de SQL, caso de estudio OWASP. Sangolquí, SF.

Miramontes Medrano, J. E. (2011). Análisis de seguridad web para prevenir cross-site scripting (XSS) en aplicaciones PHP.

Murillo, D. (2017). Navegadores web. *El Tecnológico*, 27(1), 13-14.

Bahit, E. (2011). Poo y mvc en php. *El paradigma de la Programación*.