

AGRADECIMIENTOS

Quiero agradecer principalmente al Tecnológico de Estudios Superior de Chalco y a mis asesores por brindarme su apoyo y observaciones necesarias durante el proceso de residencias profesionales, cuyo esfuerzo se ve reflejado en el aprendizaje que he tenido durante este proceso. De igual manera, agradezco a mis padres por el esfuerzo que hicieron para que yo pudiera culminar mis estudios. Les estoy muy agradecido por su apoyo y por animarme a seguir adelante cuando se me presentaban dificultades. Su compromiso y su amor han hecho que me sienta increíblemente afortunado de tenerlos a mi lado. Agradezco el tiempo, el esfuerzo y los recursos que han invertido en mi educación. Estoy muy agradecido por todo lo que me han dado.

Por otro lado, quiero expresar mi gratitud a mis compañeros de trabajo por la colaboración y la ayuda que me han brindado para completar esta etapa de mi formación profesional de la mejor manera. Finalmente, quiero agradecer a las empresas donde he realizado mis prácticas por ofrecerme la oportunidad de poner en práctica mis conocimientos y aprender mucho más.

RESUMEN

El resumen de este proyecto consiste en la realización de una intranet que sea capaz de llevar la administración de los recursos de la empresa estrategia Tecnológica. Esta intranet servirá como una herramienta para mejorar la comunicación y la colaboración entre los empleados. En esta aplicación se corrigieron varios errores, ya sea corrigiendo errores lógicos y de diseño, además de agregar nuevas vistas con su respectivo controlador, la cual interactúa con un modelo que representa una entidad de la base de datos.

La finalidad de esta aplicación radica en poder llevar el control y gestión de información para la empresa, como usuarios, vacantes, requisiciones, control de horas, etc. Esto permitirá una mejor organización y administración de la información.

Esta aplicación se realizó utilizando la plataforma de desarrollo ASP .NET Framework, en donde se trabajó mediante el patrón de diseño MVC (Modelo, Vista, Controlador) con el lenguaje de programación orientado a objetos C#. Algunas de las herramientas y tecnologías utilizadas para el desarrollo de esta aplicación incluyen: HTML, CSS, JavaScript, ASP.NET MVC, vistas Razor, ADO .NET para acceso a los datos, Procedimientos almacenados para acceder a la información de la Base de Datos, jQuery, Bootstrap y como Sistema Gestor de Base de Datos se utilizó SQL Server.

Esta aplicación redujo los procesos administrativos de la empresa, logrando ahorrar tiempo y recursos, lo que aumentó la productividad general. Al simplificar el flujo de trabajo y los procesos administrativos, se mejoró la eficiencia, la precisión y la seguridad. Esto permitió a la empresa obtener mejores resultados en menos tiempo y con menos recursos. La aplicación también ayudó a reducir el número de errores y aumentar la satisfacción de los procesos administrativos. Esto se logró mediante la automatización de los procesos, lo que permitió a la empresa ahorrar tiempo y recursos, mejorando su productividad.

ÍNDICE

ÍNDICE DE FIGURAS

INTRODUCCIÓN

El proyecto que se llevó a cabo fue el desarrollo de funcionalidades en un sistema web en el que se pueda llevar a cabo la administración de información de la empresa Estrategia Tecnológica, ya sea la información de usuarios, vacantes, registros de control de horas, etc., con la finalidad de reducir y administrar mejor los recursos para el manejo de la información, evitando tareas repetitivas en la parte administrativa, asegurando la integridad, disponibilidad y seguridad de la información.

El desarrollo se hizo con la plataforma de desarrollo ASP .NET Framework mediante el patrón de diseño Modelo, Vista, Controlador o por sus siglas en inglés Model, View, Controller, el cual permite representar las entidades de la base de datos mediante una capa de datos (Modelo), la cual interactúa con la capa de presentación (Vista) mediante el modelo de negocio (Controlador). Gracias a este modelo, se administra de manera óptima los recursos del sistema al momento de realizar peticiones a la base de datos, la cual está hecha en SQL Server.

C) Una descripción breve de cada una de las partes del informe desde la justificación hasta las conclusiones y recomendaciones.

El modelo relacional y los procedimientos almacenados de la base de datos es de carácter confidencial para la empresa, al igual que el código empleado, tanto en los modelos, vistas y controladores. Asimismo, durante el desarrollo no se tenía permiso realizar alteraciones a la estructura de la base de datos, ya sea crear o eliminar tablas o campos, alterar o agregar columnas. Tampoco se tenía acceso a toda la base de datos, ya que sugerían autenticación como administrador.

DESCRIPCIÓN DE LA EMPRESA U ORGANIZACIÓN Y DEL PUESTO O ÁREA DEL TRABAJO EL ESTUDIANTE

En el área de programación de la empresa llamada Estrategia Tecnológica S. A. de C. V. se requiere el desarrollo de vistas que permitan llevar a cabo la administración de recursos, como personal, corporativos, control de horas, etc., ya que se permite el ahorro de tiempo y recursos.

Datos generales de la empresa:

- Nombre: Estrategia Tecnológica S. A. de C. V.
- Giro: Servicio.
- Dirección: Nicolas San Juan 1132 Int. D; Colonia Del Valle Centro, Alcaldía, Benito Juárez.
- Teléfono: 5530989130
- Página web: <http://www.estrategiatec.com.mx/index.html>
- Correo electrónico: service.contact@estrategiatec.com.mx

La empresa tiene por objetivo brindar a nuestros clientes personal especializado para optimizar su operación laboral, Con el fin de responder a las necesidades de crecimiento y soporte tecnológico.

Estrategia Tecnológica asigna recursos humanos con habilidades técnicas, administrativas y de control de proyectos orientados al desarrollo de soluciones que automaticen aquellas funciones requeridas por su organización.

Desde 1996 Estrategia Tecnológica ha integrado capital humano en TI en diversas e importantes empresas nacionales para las siguientes especializaciones:

- Desarrollo y Mantenimiento de Sistemas Informáticos.
- Inteligencia de Negocios (BI) y Data Warehouse.
- Minería de datos.
- Procesos ETL.
- Soporte técnico a redes y Bases de datos.
- Mesas de ayuda (Help Desk - Soporte Técnico en Sitio).

PROBLEMAS A RESOLVER

La administración de la información de la empresa debe asegurar la disponibilidad, integridad y seguridad, por lo que hacer uso de herramientas tecnológicas como una aplicación web para manejar de manera adecuada esta información es importante. Actualmente, la empresa cuenta con una herramienta para llevar a cabo la gestión de la información, siendo esta desarrollada en .NET Framework y SQL Server, pero necesita desarrollo de nuevas vistas y controladores que administren mejor la información y mantenimiento correctivo y preventivo.

Como no se contaba con mucha experiencia en el manejo de estas herramientas de desarrollo, el residente se vio en la necesidad de llevar a cabo capacitaciones en diversos cursos en estas y otras herramientas que le ayudaron en el desarrollo de la aplicación. Asimismo, se necesitó llevar a cabo la depuración del código para saber el funcionamiento y la lógica que se manejaba, analizando los procedimientos almacenados, la Base de datos, las vistas, controladores y modelos, de tal manera que se encuentren los errores que se deben corregir, permitiendo implementar nuevas vistas a la aplicación.

A continuación, se muestra el cronograma de actividades señalando el periodo de realización y la descripción de la actividad.

ACTIVIDADES	PERIODO SEPTIEMBRE 2022 - FEBREO 2023																			
	SEPTIEMBRE				OCTUBRE				NOVIEMBRE				DICIEMBRE				ENERO			
	1RA SEM	2DA SEM	3RA SEM	4TA SEM	1RA SEM	2DA SEM	3RA SEM	4TA SEM	1RA SEM	2DA SEM	3RA SEM	4TA SEM	1RA SEM	2DA SEM	3RA SEM	4TA SEM	1RA SEM	2DA SEM	3RA SEM	4TA SEM
CAPACITACIONES EN .NET Y SQL SERVER																				
DEPURACIÓN Y ANALIZAR BASE DE DATOS																				
CREAR CONTROLADOR PARA GESTIÓN DE USUARIOS																				
CREAR VISTA PARA GESTIÓN DE USUARIOS																				
ADMINISTRAR INFORMACIÓN DE CORPORATIVOS A TRAVÉS DE UN CONTROLADOR																				
DISEÑO DE VISTA PARA APARTADO DE CORPORATIVOS																				
CORRECCIÓN DE BUGS Y DISEÑO DE VISTA Y CONTROLADOR DE VACANTES																				
CORRECCIÓN DE DISEÑO DE APLICACIÓN WEB																				

Capacitaciones en .NET y SQL Server: En esta actividad se llevó a cabo capacitaciones tanto en .NET Framework y .NET CORE. También se capacitó en SQL Server.

Depuración y analizar base de datos: Al comprender el funcionamiento de la plataforma .NET Framework, se requiere comenzar con la depuración o análisis de las vistas, modelos y controladores de cada uno de los componentes de la aplicación, analizando las peticiones que se hacen en los modelos, vistas y controladores.

Crear controlador para gestión de usuarios: Se necesita implementar un controlador que permita consultar la base de datos para obtener la información de los usuarios, para posteriormente mostrar dicha información en una vista.

Crear vista para gestión de usuarios: Es necesario construir una vista con la información que el controlador obtuvo al consultar la base de datos. Se requiere el uso de Frameworks y librerías, tales como Bootstrap y JQuery para mostrar mejor la información.

Administrar información de corporativos a través de un controlador: Se debe crear un controlador que acceda a la base de datos y obtenga la información de los nombres y tareas de las corporaciones.

Diseño de vista para apartado de corporativos: Es necesario crear la vista de corporativos una vez que se obtenga la información de la base de datos a través de su respectivo controlador, mostrando Bootstrap y JQuery un conjunto de componentes para que los usuarios manejen la información.

Corrección de bugs y diseño de vista y controlador de vacantes: La vista de vacantes tiene un error que muestra un elemento SELECT de HTML que no tiene relevancia, ya que, sin importar la opción que se elija, muestra la misma información, por lo que se requiere eliminar ese componente modificando tanto la vista como el controlador.

Corrección de diseño de aplicación web: Se debe hacer un diseño responsivo de la aplicación, ya que solamente en resoluciones altas se ve bien, mientras en resoluciones bajas como en celulares o tabletas el diseño tiene errores.

OBJETIVOS

GENERAL

Llevar a cabo la administración y gestión de recursos de la información a través de una aplicación web en la plataforma ASP .NET Framework.

ESPECÍFICOS

- Llevar a cabo capacitaciones en SQL Server y la plataforma de desarrollo ASP .NET.
- Desarrollar las vistas y controladores para administrar los usuarios.
- Administrar la información de los corporativos que colaboran con la empresa a través de una vista.
- Depurar el código para corregir errores o bugs en la vista y controlador de las vacantes.

JUSTIFICACIÓN

Actualmente las empresas requieren que la información sea gestionada de manera segura, íntegra y confidencial, por lo que recurren a tecnologías que permitan a los administradores cumplir con este objetivo. De esta manera, las empresas evitan el uso de herramientas como Excel que no son capaces de manejar grandes cantidades de información, las cuales pueden tener errores por un mal manejo.

La plataforma de desarrollo .NET Framework es útil para desarrollar aplicaciones y servicios web, ya que proporciona un conjunto de herramientas y bibliotecas que los desarrolladores pueden utilizar para crear ricas aplicaciones web y servicios basados en la web. Estas herramientas incluyen ASP.NET, AJAX, LINQ, ADO.NET y muchas otras. Estas herramientas permiten a los desarrolladores crear aplicaciones web avanzadas con una gran cantidad de funcionalidad. Además, .NET Framework proporciona una plataforma para la interoperabilidad entre aplicaciones y servicios web. Esto permite a los desarrolladores crear aplicaciones y servicios que se comuniquen entre sí para compartir información y recursos.

La mejora de la administración de la información se lograría mediante la implementación de interfaces intuitivas que permitan a los usuarios interactuar con la aplicación de manera sencilla. Esto se lograría mediante la creación de nuevas vistas en la capa de presentación, que permitan al usuario controlar los datos almacenados en la base de datos y realizar operaciones como la búsqueda.

Un buen enfoque para mejorar la administración de la información de la empresa es implementar una aplicación web basada en el modelo de arquitectura de tres capas. Esta arquitectura está compuesta por tres capas, cada una con una responsabilidad específica: capa de presentación (interfaz de usuario), capa lógica (controladores y lógica de negocio) y capa de datos (base de datos).

Por lo tanto, se recomienda la implementación de una solución de desarrollo moderna, como MVC (Modelo-Vista-Controlador), la cual es una arquitectura que separa los componentes del software en tres capas: modelo, vista y controlador. Esta arquitectura permite una mayor escalabilidad, mantenimiento y seguridad de los datos almacenados en la aplicación.

MARCO TEÓRICO

1. CONCEPTOS BÁSICOS

1.1. INTERNET

La Internet es una red informática de alcance mundial que interactúan entre sí para lograr la conexión entre un cliente y un servidor a través de principalmente del protocolo TCP/IP, el cual, al momento de enviar la información a un destino, este transforma la información en paquetes, y, una vez llegan al destino, se vuelven a transformar en información para que el que reciba la información pueda interpretarla.

El licenciado en computación Marco Antonio Zamora Lucio de la Universidad Autónoma del Estado de Hidalgo define al Internet de la siguiente manera: “Internet es una red integrada por miles de redes y computadoras interconectadas en todo el mundo mediante cables y señales de telecomunicaciones, que utilizan una tecnología común para la transferencia de datos.” (Zamora, 2014, p. 4).

1.2. NAVEGADOR WEB

Los navegadores web o buscadores consisten en un software que permite al usuario conectarse a través de un hyperlink o hipervínculo que lo redireccionará a cualquier sitio web, página o portal que está en la Internet, siempre y cuando tenga los permisos para acceder. Algunos navegadores pueden ser Chrome, Microsoft Edge, Opera, Brave, Mozilla FireFox, etc.

El Autor Marco Antonio Zamora Lucio da a entender qué son los navegadores web de la siguiente forma:

La información en la Web está disponible mediante páginas web. Estas páginas están escritas internamente en lenguaje HTML. Para transformar ese lenguaje en páginas Web visibles hace falta un programa, a estos programas se les llama navegadores o browsers (en inglés). Son programas complejos que realizan muchas funciones, pero desde sus inicios han sido gratuitos y se pueden descargar de la Web. (Zamora, 2014, p. 4).

1.3. SISTEMA DE INFORMACIÓN

Consiste en una colección de elementos o componentes que interactúan entre sí para llegar a una finalidad o cumplir con un objetivo en común, siguiendo una metodología ya establecida para cumplir dicha finalidad. Tienen como objetivo principal satisfacer las necesidades de los clientes o de una organización a través de recursos tecnológicos como computadoras, software, hardware, etc.

La licenciada Liseth Francesca Alvarado Rosado de la UNIVERSIDAD NACIONAL DE EDUCACIÓN Enrique Guzmán y Valle define a los sistemas de información de la siguiente manera:

Se considera con este nombre a toda agrupación de componentes o partes relacionadas y organizadas que se relacionan entre ellas para crear un objetivo. Se explica de la forma siguiente: los sistemas reciben energía, materia del ambiente o datos, y dan como resultado materia, energía o información. (Alvarado, 2018, p. 17).

1.4. BASE DE DATOS

Consiste en una colección ordenada de datos a través de un modelo relacional. Dichos datos se registran en columnas, las cuales están dentro de tablas y estas conforman la base de datos, en donde se definen las características de la estructura de la base de datos, como los tipos de datos, el tamaño de los datos, limitaciones, índices, etc., para que los datos almacenados puedan cumplir con las funciones de seguridad, accesibilidad y almacenamiento.

La Doctora María del Carmen Gómez Fuentes en su libro BASES DE DATOS define a una base de datos de la siguiente forma:

Colección de datos, donde los datos están lógicamente relacionados entre sí, tienen una definición y descripción comunes y están estructurados de una forma particular. Una base de datos es también un modelo del mundo real y, como tal, debe poder servir para toda una gama de usos y aplicaciones. (Gómez, 2013, p. 5).

1.4.1. SISTEMA GESTOR DE BASE DE DATOS

Un Sistema Gestor de Base de Datos (SGBD) consiste en un programa que permite el almacenamiento, lectura, modificación y eliminación de datos, en donde dichos datos se almacenan en campos, que a su vez se almacenan en tablas dentro de una base de datos.

La profesora Sandra Luz Hernández Mendoza de la Universidad Autónoma del Estado de Hidalgo define a un DBMS de la siguiente forma: “Un Sistema de Administración de Bases de Datos (DBMS) es software que permite a una organización centralizar los datos, administrarlos en forma eficiente y proveer acceso a los datos almacenados mediante programas de aplicación.” (Hernández, 2021, p. 5).

1.4.2. SERVICIOS DE UN SGBD

Un SGBD ofrece 3 servicios para poder gestionar la información. Primero está el DDL o Lenguaje de Definición de Datos, el cual crea la estructura definiendo el tipo de valor, el tamaño y las restricciones que este tendrá. Después está el DML o Lenguaje de Manipulación de Datos con el que se modifica solamente la información, dejando intacta la estructura en el que se almacena la información. Y, por último, está el DCL o Lenguaje de Control de Datos, en donde se definen los roles de usuario. Asimismo, se definen las acciones que tiene permitidas, restricciones, lugares en donde puede realizar las operaciones que tiene permitidas, etc., todo esto realizado por un gestor de base de datos.

El profesor L. Carámbula define los servicios de SQL de la siguiente forma: “Permite crear, modificar y eliminar las estructuras para almacenar los datos (Metadata). Permite definir el esquema de la B.D.” (Carámbula, 2010, p. 4).

1.4.3. PROCEDIMIENTOS ALMACENADOS

Un procedimiento almacenado es una función SQL la cual permite recibir valores como parámetros para realizar alguna operación en SQL, ya sea una consulta, actualización, inserción de datos, eliminación, validaciones de datos, control de transacciones, etc., lo cual facilita los procesos de Creación, Lectura, Actualización y Eliminación (CRUD) en SQL.

Estos procedimientos se escriben una vez y se pueden llamar desde diferentes aplicaciones. Esto ofrece una mayor velocidad de procesamiento, ya que la consulta se ejecuta en la base de datos y no en el lado de la aplicación. Esto también permite una mejor seguridad, ya que los usuarios no tendrían acceso a la información oculta en el procedimiento. Los procedimientos almacenados también permiten una mayor reutilización de código, ya que se pueden reutilizar en diferentes aplicaciones sin tener que reescribirlos cada vez.

José Juan Sánchez Hernández define a un procedimiento almacenado de la siguiente manera: “Un procedimiento almacenado es un conjunto de instrucciones SQL que se almacena asociado a una base de datos. Un procedimiento puede tener cero o muchos parámetros de entrada y cero o muchos parámetros de salida.” (Sánchez, 2022, p. 3).

1.5. DESARROLLO WEB

Consiste en la interacción entre un cliente y un servidor a través de una parte gráfica (front-end) enfocada al cliente para realizar peticiones al servidor, y la parte del servidor (back-end), en donde se reciben las peticiones del cliente para darles respuesta.

En el desarrollo web existen lenguajes que interactúan con ambas partes, tales como HTML, CSS, JavaScript, etc., los cuales interactúan con el cliente (front-end) y, por otra parte, C#, PHP, NodeJS, Python, etc., sirven para el servidor (back-end). Además de que existen frameworks o librerías que mejoran el diseño del sitio.

El autor Juan Pedro Barba Soler de la Facultad de Bellas Artes da la siguiente definición de desarrollo web:

Es la programación necesaria para la construcción del sitio web. Se divide en dos partes que pueden estar o no conectadas, la parte del cliente y la parte del servidor. En la parte del cliente estaríamos hablando de HTML y CSS, código básico para creación de páginas web, y JavaScript y DOM, para la interacción con el usuario. En la

parte del servidor se trabaja con código más complejo, como es PHP, ASP.NET, JSP, etc. (Barba, 2014, p. 8).

1.5.1. SITIO WEB

Consiste en una colección de documentos electrónicos y páginas web desarrollados en lenguajes web, como HTML, CSS, JavaScript, etc., que abordan temas en particular, los cuales se pueden localizar desde un navegador web utilizando Internet. Los sitios web trabajan con el protocolo http (Hyper Text Transfer Protocol) Protocolo de Transferencia de Hipertexto.

La Universidad de la República, Montevideo-Uruguay da la siguiente definición de un sitio web:

Los sitios Web son estructuras de información como tantas otras, con las características singulares que le aportan la hipertextualidad y su ubicación en un escenario diferente, de acceso múltiple y masivo como es el ciberespacio. Al igual que cualquier otro documento, deben ser concebidos y pensados críticamente, pero esto no es fácil tanto por lo reciente de su aparición como por la existencia de enormes facilidades para la edición y puesta en línea; es un hecho que prácticamente cualquier persona puede publicar en la Web. (García, 2002, p. 2).

1.5.2. TIPOS DE SITIO WEB

Existen varias clasificaciones de sitios web que la web ofrece para hallar información a través del protocolo HTTP mediante un navegador web. Hay tantas clasificaciones para sitios web y cada uno cuenta con un propósito que los diferencia de los demás, ya sea por su estructura, la tecnología que usa, la información que ofrece y su objetivo. Un sitio web se debe clasificar según sus objetivos, audiencia, contenidos, funcionalidades y administración.

El autor Mario Miqueles propone la siguiente clasificación:

SITIO DE ARCHIVO: Usado para preservar contenido electrónico valioso amenazado con extinción. Un ejemplo es Internet Archive,

el cual desde 1996 ha preservado billones de antiguas (y nuevas) páginas web.

SITIO WEBLOG (O BLOG): Sitio usado para registrar lecturas online o para exponer contenidos en línea con la fecha del día de ingreso; también puede incluir foros de discusión.

SITIO DE COMERCIO ELECTRÓNICO: Sitio web cuyo objetivo es la compra/venta de bienes.

SITIO DE COMUNIDAD VIRTUAL: Un sitio o portal social donde las personas con intereses similares se comunican unos con otros, normalmente por chat o foros o simples mensajes.

SITIO DE DESARROLLO: Un sitio con el propósito de proporcionar información y recursos relacionados con el desarrollo de software, diseño web, etc.

SITIO INSTITUCIONAL: Estos sitios son confeccionados por alguna entidad, con o sin fines de lucro, para darse a conocer, poner información propia y funcionan como un medio de contacto para sus clientes o miembros. Además, en caso de que sea una empresa, sirve para promocionar sus bienes y servicios.

SITIO DE DESCARGAS: Estrictamente usado para descargar contenido electrónico, como software, juegos, fondos de escritorio, entre otros tipos de archivos.

SITIO DE JUEGOS: Estrictamente usado para descargar contenido electrónico, como software, juegos, fondos de escritorio, entre otros tipos de archivos.

SITIO DE INFORMACIÓN: Como su nombre indica, los contenidos de estos sitios buscan informar a quienes lo visitan, pero esto necesariamente no se hace con fines económicos. Muchas veces son de organizaciones educativas o pertenecen al gobierno.

SITIO PERSONAL: Estos sitios son administrados por una o muy pocas personas y contienen material sobre cualquier temática.

SITIO EDUCATIVO: Estos sitios ofrecen cursos a distancia o presenciales, ofrecen información y contenidos descargables sobre distintas asignaturas y pueden estar orientados tanto a profesores como a alumnos.

SITIO WIKI: En estos sitios, son los usuarios los que suben y editan los contenidos.

SITIO SPAM: Sitio web sin contenidos de valor que ha sido creado exclusivamente para obtener beneficios y fines publicitarios, engañando o pretendiendo engañar a los motores de búsqueda.

SITIO WEB 1.0: Un sitio web estático. Un sitio donde el visitante sólo puede recorrer sus páginas sin posibilidad de interactuar con ellas.

SITIO WEB 2.0: Un sitio web interactivo. Un sitio donde el visitante puede hacer más cosas que recorrer sus páginas, en concreto, extraer información en la forma y criterios que estime oportuno y conveniente.

SITIO WEB 3.0: Un sitio web inteligente. Un sitio que reconoce al usuario y muestra una dinámica en función de sus gustos, preferencias, historial, el momento y el estado de ánimo en que se encuentre. (Miqueles, 2012, p. 6-39).

1.6. ARQUITECTURA MODELO VISTA CONTROLADOR

1.6.1. ¿QUÉ ES MVC?

La arquitectura o patrón de diseño Model View Controller (MVC) consiste en una forma de clasificar la información separándola en tres partes; primero se tiene la capa de datos que es el modelo, el cual es una representación mediante clases y objetos de una tabla de una base de datos; después se tiene la capa de

presentación que representa la vista, en donde el usuario hará interacciones y peticiones con el backend o la base de datos de la aplicación a través de una interfaz gráfica; finalmente está la lógica de negocios que es el controlador, el cual se encarga de comunicar la vista con el modelo, de tal manera que obtendrá toda la información que haya sido pedida por el usuario a través de la vista para posteriormente realizar alguna tarea con dicha información.

La finalidad de este patrón de diseño consiste en simplificar las tareas de implementación de aplicaciones de acuerdo con la información que soliciten los usuarios a través de peticiones mediante una interfaz gráfica o vista hacia el backend y los datos a devolver o desplegar con el controlador. De esta manera, se separan los datos, la interfaz de usuario y la lógica de una aplicación.

El ingeniero Luis Peña de la Universidad Alejandro de Humboldt de la República Bolivariana De Venezuela define de la siguiente forma esta arquitectura:

El Model-View-Controller o Modelo–Vista–Controlador (MVC) es un patrón de diseño de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos de forma que las modificaciones al componente de la vista, o a cualquier parte del sistema puedan ser hechas con un mínimo impacto en el componente del modelo de datos o en los otros componentes del sistema. Este patrón cumple perfectamente el cometido de modularizar un sistema. (Piña, 2016, p. 4).

1.6.1.1. MODELO

Es la parte que define los datos y reglas de negocio, ya que su principal funcionalidad es la representación de una base de datos mediante clases y propiedades, permitiendo que la aplicación pueda comunicarse con la base de datos accediendo a cada una de sus propiedades, ya sea a los campos de una tabla, vistas de SQL, procedimientos almacenados (stored procedures), etc. El modelo define la estructura de datos de la aplicación almacenando la información y los datos.

1.6.1.2. VISTA

La vista es aquella parte que se encarga de mostrar una interfaz gráfica de gráfica donde el usuario interactúa con la aplicación. En la vista se muestra la información del modelo al usuario a través de una representación visual de los datos que el usuario ve e interactúa.

En el caso de una aplicación web, la vista consiste en una página HTML que el navegador es capaz de desplegar, en donde se muestra dinámicamente el contenido o los datos del modelo en donde los usuarios pueden realizar operaciones sobre dicha información. La finalidad de la vista consiste en convertir los datos del modelo para que tengan un significado coherente para el usuario y este los pueda interpretar fácilmente.

En ASP .NET las vistas se realizan en archivos .cshtml, en donde, además de contar con etiquetas de HTML5, se puede hacer uso de Razor, que consiste en una sintaxis de marcado que permite incrustar código de C#.

1.6.1.3. CONTROLADOR

Esta parte es la capa que se encarga de administrar y gestionar las peticiones que realiza el usuario a través de las vistas hacia el modelo, en donde maneja las entradas que recibe del usuario a través de eventos para posteriormente procesarlas accediendo a los datos del modelo y mostrar el resultado al usuario a través de una vista.

En esta parte se maneja la lógica de negocio, la cual es la responsable del procesamiento y comportamiento de acuerdo con las peticiones que el usuario realiza a través de las vistas para acceder a los datos del modelo. De esta manera, el controlador realiza la construcción de un modelo apropiado, para después enviárselo a la vista, en donde el usuario, a través de la petición que realizó, logra visualizar apropiadamente los datos que solicitó.

El controlador es capaz de acceder al modelo y puede actualizarlo o modificarlo de acuerdo con las acciones que los usuarios soliciten a través de la vista.

1.6.2. VENTAJAS

A continuación, se muestran algunas ventajas de implementar la arquitectura MVC en las aplicaciones que se desarrollen:

1. Ofrece una estructura clara para la administración de la lógica de negocio, la interacción con la interfaz de usuario y los datos.
2. Mejora la reutilización del código, ya que la lógica de negocios, el acceso a los datos y la interfaz de usuario están separados.
3. Reduce el acoplamiento entre componentes, haciendo que el código sea más mantenible.
4. Reduce la cantidad de código en cada componente, ya que la lógica se divide en varias capas.
5. Mejora la seguridad, ya que el controlador aprovecha la lógica de negocios para proteger los datos.
6. Mejora el rendimiento, ya que el controlador puede limitar el acceso de los usuarios a los datos.
7. Facilita la implementación de arquitecturas distribuidas, ya que los controladores pueden ser implementados en múltiples computadoras.
8. Ofrece una arquitectura que es compatible con la mayoría de los lenguajes de programación.

1.6.3. DESVENTAJAS

Las desventajas radican principalmente en la estructura del código más que en el rendimiento de la aplicación, ya que en proyectos grandes se suele tener cierta complejidad al momento de estructurar los componentes y clases, ya que, al estar separado en diferentes módulos, resulta confuso saber qué tarea se encarga cada componente, acción o clase. A continuación, se da a conocer algunas desventajas de esta arquitectura de diseño.

1. Puede ser difícil de mantener y extender a medida que se agreguen más funcionalidades a la aplicación.
2. Puede ser difícil de entender para los desarrolladores que no estén acostumbrados a trabajar con esta arquitectura.
3. Puede resultar en un código desordenado debido al número de clases y métodos involucrados.
4. La separación entre la vista, el modelo y el controlador puede ser confusa para los desarrolladores principiantes.
5. Si no se usan herramientas de construcción, puede resultar en una cantidad significativa de código repetitivo que debe escribirse para cada controlador y vista.
6. Si el modelo se vuelve demasiado complejo, puede resultar en un gran número de peticiones HTTP al servidor, lo que puede causar una reducción en el rendimiento.

1.6.4. ¿CUÁNDO USAR MVC?

Cuando se tienen aplicaciones que requieren de una gran complejidad, ya sea que será una aplicación que se encargará de resolver diferentes tareas y ejecutar varios procesos al mismo tiempo, es recomendable separar la aplicación en diferentes procesos que se encarguen de una sola tarea, para después unificar todo en un solo componente. El patrón de diseño MVC cumple con esas características, debido a que es capaz de separar el código de una aplicación en componentes reutilizables, con el objetivo de mejorar el mantenimiento y la escalabilidad.

MVC es una excelente opción para aplicaciones web modernas, ya que facilita la separación de los datos de la interfaz de usuario, lo que permite una mayor flexibilidad y escalabilidad. Esto también hace que sea más fácil de mantener, ya que cada componente puede ser actualizado o cambiado por separado sin afectar al resto de la aplicación. Además, MVC es una excelente opción cuando se trata de aplicaciones web basadas en bases de datos, ya que permite una estructura consistente para la recuperación y el almacenamiento de datos.

De esta manera, la aplicación se separa en estas tres partes, por lo que la aplicación se vuelve más escalable y flexible, permitiendo cambios en cada una de ellas sin afectar el desempeño de la aplicación en su conjunto.

1.7. C#

1.7.1. ¿QUÉ ES?

Es un lenguaje de programación multiparadigma desarrollado por Microsoft en el año 2000 como parte de su plataforma .NET. Está diseñado para proporcionar una sintaxis que sea natural para los programadores C++ y para ofrecer una mayor productividad con un lenguaje orientado a objetos. C# es un lenguaje de programación multiparadigma que admite programación orientada a objetos, programación funcional, programación declarativa y programación imperativa. Está basado en la sintaxis de C y Java, y es uno de los lenguajes más populares del mundo. C# es usado en una amplia variedad de aplicaciones, desde aplicaciones de escritorio hasta aplicaciones web, desarrollo de juegos y aplicaciones móviles.

El sitio oficial de Microsoft da a conocer una definición de C# a través de su documentación oficial:

C# es un lenguaje de programación orientado a componentes, orientado a objetos. C# proporciona construcciones de lenguaje para admitir directamente estos conceptos, por lo que se trata de un lenguaje natural en el que crear y usar componentes de software. Desde su origen, C# ha agregado características para admitir nuevas cargas de trabajo y prácticas de diseño de software emergentes. En el fondo, C# es un lenguaje orientado a objetos.

Defina los tipos y su comportamiento. (Microsoft, 2022, p. 1).

1.7.2. CARACTERÍSTICAS

1. Lenguaje de programación moderno: C# es un lenguaje de programación moderno, orientado a objetos, con una sintaxis simple y clara, diseñado para permitir a los desarrolladores crear aplicaciones robustas y eficientes.

2. Lenguaje de programación orientado a objetos: C# es un lenguaje de programación orientado a objetos, lo que significa que los programas escritos en C# están estructurados usando clases, propiedades, métodos y objetos. Esto permite a los programadores crear código más robusto y reutilizable que se puede aplicar a muchos problemas diferentes.

3. Tipado estático: C# es un lenguaje de programación tipado estático, lo que significa que los tipos de datos de los valores se verifican en tiempo de compilación. Esto ayuda a evitar errores de ejecución al asegurar que los programas funcionen de la manera prevista.

4. Garbage Collection: C# utiliza un recolector de basura que ayuda a los desarrolladores a reducir el tiempo de desarrollo y aumentar la eficiencia al liberar automáticamente la memoria usada por los objetos no utilizados. Esto evita tener que escribir código para liberar manualmente la memoria.

5. Sintaxis de alto nivel: C# está escrito en un lenguaje de alto nivel, lo que significa que es mucho más fácil de leer y entender que los lenguajes de programación de bajo nivel como C y Assembly. Esto significa que los programadores pueden escribir programas mucho más rápido y tener menos errores de programación.

6. Compatible con .NET: C# es uno de los principales lenguajes utilizados para desarrollar aplicaciones en el entorno de Microsoft .NET Framework. Esto permite a los desarrolladores aprovechar el enorme conjunto de herramientas y recursos de programación de .NET.

7. Seguridad: C# es un lenguaje de programación seguro, lo que significa que es difícil para los programadores escribir código que permita a los usuarios hacer cosas que no deben hacer. Esto significa que los programadores pueden crear programas seguros que no sean vulnerables a virus, malware y otros ataques.

8. Simplicidad: C# es relativamente fácil de aprender para los programadores principiantes, ya que es un lenguaje de alto nivel que tiene una sintaxis clara y fácil de entender. Esto significa que los programadores pueden comenzar a escribir código en poco tiempo sin tener que leer mucha documentación.

9. Soporte para Windows y Linux: C# es compatible con Windows y Linux, lo que permite a los desarrolladores desarrollar aplicaciones para una amplia variedad de plataformas.

1.7.3. VENTAJAS

1. Lenguaje de alto nivel: C # es un lenguaje de alto nivel, lo que significa que es fácil de leer y entender para los programadores. Esto hace que sea un lenguaje ideal para principiantes que están aprendiendo a programar.

2. Sintaxis similar a la del lenguaje C: Si ya conoces el lenguaje C, entonces aprender C # será mucho más fácil para ti. C # utiliza mucha sintaxis similar a la del lenguaje C, por lo que no tendrás que aprender un lenguaje completamente nuevo para poder programar en C #.

3. Compatibilidad multiplataforma: C # se ejecuta en la plataforma .NET, que es compatible con muchas plataformas. Esto significa que puedes usar C # para desarrollar aplicaciones para Windows, Linux, MacOS, iOS y Android.

4. Seguridad: C # es un lenguaje seguro, ya que permite controlar el acceso a los datos y a los recursos. Esto significa que los usuarios no pueden acceder a datos o recursos que no estén autorizados para acceder.

5. Herramientas de desarrollo intuitivas: Microsoft ha creado una serie de herramientas de desarrollo de C # que permiten a los desarrolladores crear aplicaciones de forma fácil y rápida. Estas herramientas hacen que sea mucho más fácil para los principiantes aprender a programar en C #.

1.7.4. DESVENTAJAS

1. La curva de aprendizaje es un poco más pronunciada que en otros lenguajes de programación. Esto significa que los programadores principiantes tienen que invertir un poco más de tiempo para aprender C#.

2. C# es un lenguaje de propiedad privada, lo que significa que no hay soporte técnico gratuito para problemas técnicos. Si un programador necesita ayuda, debe recurrir a un sitio de terceros o una comunidad de programación para obtener respuestas.

3. C# no es un lenguaje multiplataforma. Esto significa que los programas creados con C# sólo funcionan en Windows. Si un programador desea crear una aplicación para un sistema operativo diferente, debe recurrir a un lenguaje diferente.

4. A diferencia de otros lenguajes de programación, C# no ofrece una solución de almacenamiento de datos nativa. Esto significa que los programadores tienen que recurrir a una tercera biblioteca para implementar un almacenamiento de datos.

2. .NET

2.1. DEFINICIÓN

Consiste en una plataforma de desarrollo de Microsoft, la cual contiene las herramientas necesarias para realizar el desarrollo de aplicaciones web, móviles, Machine Learning, Cloud Computing, aplicaciones de escritorio, etc., trabajando con el lenguaje de programación C#. Utiliza la arquitectura Modelo Vista Controlador (MVC) la cual separa la lógica de la aplicación de la interfaz de usuario, permitiendo una mejor reutilización de código y una mejor separación de preocupaciones. Esto permite a los desarrolladores crear aplicaciones web escalables, seguras y eficientes.

ASP.NET también ofrece una amplia gama de características para facilitar el desarrollo de aplicaciones web, incluyendo herramientas de desarrollo, APIs y servicios, bibliotecas de código abierto, soporte para la creación de APIs y una plataforma para la construcción de aplicaciones web. Además, ASP.NET admite la integración con otros lenguajes de programación como Java, PHP y Ruby on Rails. Esto permite a los desarrolladores crear aplicaciones web híbridas, utilizando herramientas y lenguajes con los que estén familiarizados.

La autora del título Capítulo I. Introducción a la Tecnología .NET Mónica Cecilia Gallegos Varela da a conocer la siguiente definición de .NET: “.NET es una infraestructura para desarrollar aplicaciones Windows y Web dentro de los entornos Microsoft a través de un conjunto de herramientas, superiores a las ya conocidas.” (Gallegos, 2015, p. 2).

2.2. .NET FRAMEWORK

2.2.1. CONCEPTO

.NET Framework es un marco de trabajo o framework de código abierto desarrollado por Microsoft. Está diseñado para ayudar a los desarrolladores de software a crear aplicaciones de forma más rápida y fácil. Está basado en la tecnología .NET y contiene una variedad de herramientas y bibliotecas que pueden ser utilizadas para crear aplicaciones en línea y aplicaciones de escritorio. El .NET Framework también ofrece un marco de trabajo para la creación de servicios web y aplicaciones de Windows. El .NET Framework es compatible con una variedad de lenguajes de programación, incluidos C#, F# y Visual Basic.

.NET Framework posee un conjunto de bibliotecas y tecnologías que permiten a los desarrolladores crear aplicaciones de software de alta calidad para Windows, servidores y dispositivos móviles. Estas tecnologías incluyen el lenguaje de programación .NET, ASP.NET, Windows Forms, Windows Presentation Foundation, Windows Communication Foundation, Windows Workflow Foundation y Entity Framework..NET Framework también proporciona un entorno de ejecución para la ejecución de aplicaciones .NET, así como una biblioteca de clases de .NET que contiene una amplia variedad de clases y componentes preconstruidos que se pueden utilizar en la creación de aplicaciones.

El investigador Jesé Alvarado de la Universidad del Vale de Orizaba da a conocer una definición de .NET Framework a través de su artículo Módulo1: Introducción al lenguaje C#:

El .NET Framework es la plataforma de desarrollo de código administrado de Microsoft. Está formado por una serie de herramientas y librerías con las que se pueden crear todo tipo de aplicaciones, desde las tradicionales aplicaciones de escritorio hasta aplicaciones para XBOX pasando por desarrollo Web, desarrollo para el Windows Store y Windows Phone así como aplicaciones de servidor con WCF.

El .NET Framework está compuesto por un conjunto de tecnologías que forman una parte importante de la plataforma .NET. Constituye una infraestructura de programación para construir, distribuir y ejecutar aplicaciones y servicios para la plataforma .NET. (Alvarado, 2018, p. 1).

2.2.2. CARACTERÍSTICAS

1. Es orientado a objetos: .NET Framework utiliza objetos para facilitar la programación orientada a objetos, lo que significa que es fácil crear aplicaciones con un diseño de objetos.
2. Tiene una amplia variedad de lenguajes: .NET Framework admite una amplia variedad de lenguajes de programación, como C#, Visual Basic, F#, entre otros.
3. Es seguro: .NET Framework cuenta con un sistema de seguridad integrado para ayudar a proteger las aplicaciones de virus y otros tipos de malware.
4. Es escalable: .NET Framework es escalable, lo que significa que se pueden crear aplicaciones que se ejecuten en diferentes dispositivos sin tener que cambiar el código.
5. Tiene una gran colección de bibliotecas: .NET Framework cuenta con una gran colección de bibliotecas que facilitan la programación y permiten a los desarrolladores crear aplicaciones de calidad.

2.2.3. VENTAJAS

1. Gran velocidad y rendimiento: .NET Framework está diseñado para ofrecer una gran velocidad y rendimiento a los usuarios. Está construido sobre la infraestructura CLR (Common Language Runtime) que ofrece una ejecución más rápida de las aplicaciones.
2. Lenguajes de programación: .NET Framework admite una variedad de lenguajes de programación. Esto significa que los programadores pueden escribir código en un lenguaje de programación que se elija. Estos incluyen lenguajes tales como C#, Visual Basic, F#, y otros.

3. Amigable con los dispositivos: .NET Framework es compatible con una variedad de dispositivos, como computadoras de escritorio, dispositivos móviles, etc. Esto significa que se pueden crear aplicaciones que se ejecuten en diferentes dispositivos sin tener que reescribir el código.

4. Seguridad: .NET Framework ofrece una seguridad robusta para proteger las aplicaciones. Esto incluye la autenticación, el control de acceso, los permisos, el control de código, etc. Esto garantiza que nadie pueda acceder a las aplicaciones sin autorización.

2.2.4. DESVENTAJAS

1. Alto costo de licencia: Microsoft cobra una tarifa anual por el uso de su tecnología de desarrollo de aplicaciones .NET. Esto puede ser una desventaja para algunas empresas que buscan mantener los costos bajos.

2. Requiere una plataforma específica: .NET Framework solo se puede ejecutar en plataformas Windows, lo que limita la portabilidad de las aplicaciones desarrolladas con esta tecnología.

3. Tiempos de ejecución lentos: El entorno de ejecución .NET es un poco más lento que otros entornos, como Java, lo que puede afectar el rendimiento de la aplicación.

4. Dificultad para aprender: El lenguaje de programación asociado a .NET Framework, C#, es un lenguaje de programación relativamente difícil de aprender. Esto significa que los desarrolladores a veces deben dedicar una cantidad significativa de tiempo para aprender a usar .NET Framework.

2.2.5. ASP .NET FRAMEWORK

ASP.NET Framework es un marco de desarrollo de código abierto para aplicaciones web desarrollado por Microsoft que permite a los desarrolladores crear sitios web dinámicos, servicios web y aplicaciones web. Utiliza la programación orientada a objetos para permitir a los desarrolladores crear contenido web interactivo, aplicaciones y servicios basados en la web para cualquier dispositivo. ASP.NET también ofrece una plataforma de desarrollo web segura y fiable para aplicaciones empresariales, como la gestión de contenido y la gestión de procesos de negocio.

La plataforma ASP.NET se puede usar para desarrollar aplicaciones web para Windows, Linux y Mac OS X.

Está compuesto por una biblioteca de clases y un conjunto de herramientas de desarrollo de software que pueden ser usadas para crear aplicaciones robustas y escalables. El marco ofrece una infraestructura para la creación de aplicaciones y servicios web que se pueden utilizar para crear aplicaciones en la Web, servicios web, aplicaciones de escritorio y aplicaciones móviles. ASP .NET Framework es un entorno de desarrollo de aplicaciones (IDE) que puede ser utilizado para desarrollar aplicaciones basadas en la web y servicios web. Está diseñado para permitir a los desarrolladores crear aplicaciones web dinámicas y escalables de forma rápida y sencilla. El marco de trabajo admite varios lenguajes de programación, incluyendo C#, Visual Basic .NET, JavaScript, F # y muchos más.

El ingeniero en computación y Miembro sénior de ACM y miembro de IEEE Computer Society Fernando Berzal da a conocer la siguiente definición en su libro Desarrollo Profesional de Aplicaciones Web con ASP .NET:

ASP.NET es el nombre con el que se conoce la parte de la plataforma .NET que permite el desarrollo y ejecución tanto de aplicaciones web como de servicios web. Igual que sucedía en ASP, ASP.NET se ejecuta en el servidor. En ASP.NET, no obstante, las aplicaciones web se suelen desarrollar utilizando formularios web, que están diseñados para hacer la creación de aplicaciones web tan sencilla como la programación en Visual Basic (.NET, claro está). (Berzal, 2005, p. 31).

2.3. .NET CORE

2.3.1. CONCEPTO

A

2.3.2. CARACTERÍSTICAS

A

2.3.3. VENTAJAS

A

2.3.4. DESVENTAJAS

A

2.3.5. ASP .NET CORE

A

2.4. .NET FRAMEWORK Y .NET CORE

A

2.5. VISTAS RAZOR

A

3. SQL SERVER

HISTORIA

3.1. DEFINICIÓN

A

3.2. CARACTERÍSTICAS

A

4. VISUAL STUDIO

HISTORIA

4.1. DEFINICIÓN

A

4.2. CARACTERÍSTICAS

A

5. LIBRERÍA Y FRAMEWORK

5.1. LIBRERÍA

A

5.1.1. CARACTERÍSTICAS

A

5.1.2. JQUERY

A

5.2. FRAMEWORK

A

5.2.1. CARACTERÍSTICAS

A

5.2.2. BOOTSTRAP

A

**PROCEDIMIENTO Y DESCRIPCIÓN DE LAS ACTIVIDADES
REALIZADAS**

A

RESULTADOS

A

**CONCLUSIONES DE PROYECTO, RECOMENDACIONES Y
EXPERIENCIA PERSONAL PROFESIONAL ADQUIRIDA**

A

RECOMENDACIONES Y EXPERIENCIA PROFESIONAL

A

COMPETENCIAS DESARROLLADAS Y/O APLICADAS

A

FUENTES DE INFORMACIÓN

Alvarado, J. (2018). Módulo1: Introducción al lenguaje C#: *¿Qué es .NET*

Framework? Recuperado de
file:///C:/Users/berna/Downloads/Introduccion_a_C_Que_es_el_NET_Framework.pdf

Alvarado, L. F. (2018). SISTEMAS DE INFORMACIÓN Concepto de Sistemas de información, naturaleza, fundamentos y Principios, tipos de sistemas de información, enfoques de sistemas, perspectivas y aplicaciones: *¿En qué consiste un sistema de información?* Recuperado de
<https://repositorio.une.edu.pe/bitstream/handle/UNE/3984/MONOGRAF%C3%8DA%20-%20ALVARADO%20ROSADO.pdf?sequence=1&isAllowed=y>

Berzal, F. (2005). Desarrollo Profesional de Aplicaciones Web con ASP.NET: *Definición de ASP .NET.* Recuperado de
<https://elvex.ugr.es/decsai/csharp/pdf/web/web-book-b5.pdf>

Barba, J. P. (2014). DISEÑO Y DESARROLLO WEB: *¿En qué consiste y partes con las que interactúa?* Recuperado de
https://riunet.upv.es/bitstream/handle/10251/49757/MEMORIA_Barba%2520Soler%252C%2520Juan%2520Pedro.pdf?sequence=1

Carámbula, L. (2010). Sistema de Base de Datos II: *Sub-lenguajes de SQL.* Recuperado de <http://www.carambula.net/wp-content/uploads/2010/08/DDL-Lenguaje-de-Definici%C3%B3n-de-Datos.pdf>

Gallegos, M. C. (2015). INTRODUCCIÓN A LA TECNOLOGÍA .NET: *Definición de .NET*. Recuperado de

<http://repositorio.utn.edu.ec/bitstream/123456789/1116/1/04%20ISC%20064%20CAPITULO%20I.pdf>

García, A. (2002). LOS SITIOS WEB COMO ESTRUCTURA DE INFORMACIÓN: *Definición de un sitio web*. Recuperado de <http://eprints.rclis.org/5491/1/B12-02.pdf>

Gómez, M. C. (2013). BASES DE DATOS: *Definición de una base de datos*. (1ra. Ed.). Cuajimalpa, Ciudad de México. Editorial Casa abierta al tiempo.

Hernández, S. L. (2021). Diseño de base de datos: *Definición de un DBMS*. Recuperado de <https://repository.uaeh.edu.mx/bitstream/bitstream/handle/123456789/19861/disenio-base-datos.pdf?sequence=1&isAllowed=y>

Microsoft. (2022). Paseo por el lenguaje C#: *Breve definición del lenguaje de programación C#*. Recuperado de <https://learn.microsoft.com/es-es/dotnet/csharp/tour-of-csharp/>

Miqueles, M. (2012). Clasificación de sitios web: *Tipos de sitios web existentes*. Recuperado de <https://www.mario.cl/clases/udla/disenio-web-ux-2/clases/clasificacion-de-sitios-web.pdf>

Piña, L. (2016). MODELO VISTA-CONTROLADOR (MVC): *Definición de la arquitectura MVC*. Recuperado de

<https://espejodeantagno.files.wordpress.com/2016/04/modelo-vista-controlador-mvc.pdf>

Sánchez, J. J. (2022). Unidad 12. Triggers, procedimientos y funciones en MySQL:

Definición de un Stored Procedure en SQL. Recuperado de
<https://josejuansanchez.org/bd/unidad-12-teoria/index.pdf>

Zamora, M. A. (2014). Internet: *Definición de internet y navegador web.* Recuperado de

https://www.uaeh.edu.mx/docencia/P_Presentaciones/prepa3/Presentaciones_Enero_Junio_2014/Definicion%20de%20Internet.pdf

ANEXOS

A

