

# TECNOLÓGICO DE ESTUDIOS SUPERIORES DE CHALCO

## INFORME DE PROYECTO

“DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN DE  
RECURSOS HUMANOS EN LA EMPRESA ESTRATEGIA  
TECNOLÓGICA”.

desarrollado en:  
Estrategia Tecnológica, S. A. de C. V.

que presenta:  
Bernardo Gabriel Mejía Pérez

Grupo:  
“6951”

DOCENTE: M. en DTI. Raúl Romero Castro.

Chalco, México a 01 de marzo del 2023.

## **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, cuya dedicación se ve reflejada en el aprendizaje que he tenido durante este proceso. Asimismo, quiero agradecer a la empresa Estrategia Tecnológica donde he realizado mis prácticas por ofrecerme la oportunidad de poner en práctica mis conocimientos y aprender mucho más.

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 empeño 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.

## 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

<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>DESCRIPCIÓN DE LA EMPRESA U ORGANIZACIÓN Y DEL PUESTO O ÁREA DEL TRABAJO EL ESTUDIANTE .....</b>	<b>2</b>
<b>PROBLEMAS A RESOLVER .....</b>	<b>3</b>
<b>OBJETIVOS .....</b>	<b>5</b>
<b>JUSTIFICACIÓN .....</b>	<b>6</b>
<b>MARCO TEÓRICO .....</b>	<b>7</b>
<b>1. CONCEPTOS BÁSICOS.....</b>	<b>7</b>
1.1. INTERNET.....	7
1.2. NAVEGADOR WEB.....	7
1.3. SISTEMA DE INFORMACIÓN.....	8
1.4. BASE DE DATOS.....	8
1.4.1. SISTEMA GESTOR DE BASE DE DATOS.....	9
1.4.2. SERVICIOS DE UN SGBD .....	9
1.4.3. PROCEDIMIENTOS ALMACENADOS .....	10
1.5. DESARROLLO WEB .....	10
1.5.1. SITIO WEB.....	11
1.5.2. TIPOS DE SITIO WEB.....	12
1.6. ARQUITECTURA MODELO VISTA CONTROLADOR.....	14
1.6.1. ¿QUÉ ES MVC? .....	14
1.6.1.1. MODELO .....	15
1.6.1.2. VISTA .....	15
1.6.1.3. CONTROLADOR .....	16
1.6.2. VENTAJAS .....	16
1.6.3. DESVENTAJAS.....	17

1.6.4. ¿CUÁNDO USAR MVC? .....	18
1.7. C# .....	18
1.7.1. ¿QUÉ ES? .....	18
1.7.2. CARACTERÍSTICAS .....	19
1.7.3. VENTAJAS .....	21
1.7.4. DESVENTAJAS.....	21
2. .NET .....	22
2.1. DEFINICIÓN .....	22
2.2. .NET FRAMEWORK .....	23
2.2.1. CONCEPTO .....	23
2.2.2. CARACTERÍSTICAS .....	24
2.2.3. VENTAJAS .....	24
2.2.4. DESVENTAJAS.....	25
2.2.5. ASP .NET FRAMEWORK.....	26
2.3. .NET CORE .....	27
2.3.1. CONCEPTO .....	27
2.3.2. CARACTERÍSTICAS .....	28
2.3.3. VENTAJAS .....	29
2.3.4. DESVENTAJAS.....	29
2.3.5. ASP .NET CORE .....	30
2.4. .NET FRAMEWORK Y .NET CORE.....	31
2.5. VISTAS RAZOR.....	33
3. SQL SERVER .....	34
3.1. DEFINICIÓN .....	34
3.2. CARACTERÍSTICAS .....	34

4. VISUAL STUDIO.....	35
4.1. DEFINICIÓN .....	35
4.2. CARACTERÍSTICAS .....	36
5. LIBRERÍA Y FRAMEWORK.....	37
5.1. LIBRERÍA .....	37
5.1.1. JQUERY .....	37
5.2. FRAMEWORK .....	38
5.2.1. BOOTSTRAP .....	39
<b>PROCEDIMIENTO Y DESCRIPCIÓN DE LAS ACTIVIDADES REALIZADAS ...</b>	<b>41</b>
<b>RESULTADOS .....</b>	<b>53</b>
<b>CONCLUSIONES DE PROYECTO, RECOMENDACIONES Y EXPERIENCIA</b>	
<b>PERSONAL PROFESIONAL ADQUIRIDA .....</b>	<b>56</b>
<b>COMPETENCIAS DESARROLLADAS Y/O APLICADAS.....</b>	<b>57</b>
<b>FUENTES DE INFORMACIÓN .....</b>	<b>59</b>
<b>ANEXOS.....</b>	<b>63</b>

## ÍNDICE DE FIGURAS

Figura 1. Cronograma de actividades.....	3
Figura 2.0. Bases de datos y SGDB.....	8
Figura 2.1. Desarrollo web y tecnologías.....	11
Figura 2.2. Arquitectura MVC.....	14
Figura 2.3. Logo de C#.....	19
Figura 2.4. .NET Framework.....	23
Figura 2.5. ASP .NET Framework.....	26
Figura 2.6. Logo de .NET Core. ....	27
Figura 2.7. ASP .NET Core. ....	31
Figura 2.8. Comparativa entre .NET Framework y .NET Core.....	32
Figura 2.9. SQL Server.....	34
Figura 2.10. Visual Studio.....	35
Figura 2.11. Librería JQuery.....	37
Figura 2.12. Framework de Bootstrap.....	39
Figura 3.0. Vista principal de alta de usuarios. ....	41
Figura 3.1. Vista de los detalles del usuario seleccionado. ....	41
Figura 3.2. Vista para las plazas para los usuarios.....	42
Figura 3.3. Vista del componente de seguimiento del usuario. ....	42
Figura 3.4. Código de la vista del formulario de usuarios. ....	43
Figura 3.5. Formulario para registrar usuarios. ....	43
Figura 3.6. Creación de un select de HTML con Razor. ....	43
Figura 3.7. Select de HTML para información de los jefes directos del usuario.....	44

Figura 3.8. Función de JQuery para componente de calendario del formulario.....	44
Figura 3.9. Función de JQuery para validaciones de RFC. ....	45
Figura 3.10. Función de JQuery para validar CURP. ....	45
Figura 3.11. Validaciones con JQuery para los campos restantes. ....	45
Figura 3.12. Función de JQuery para guardar los valores del formulario. ....	46
Figura 3.13. Vista principal de los corporativos. ....	46
Figura 3.14. Modelo de corporativos.....	47
Figura 3.15. Creación del formulario de corporativos en la vista.....	47
Figura 3.16. Utilización de una vista parcial de los corporativos.....	47
Figura 3.17. Controlador de corporativos para obtener la información.....	48
Figura 3.18. Listado de los corporativos con Razor.....	48
Figura 3.19. Utilización de librería de JQuery para información de la tabla.....	49
Figura 3.20. Consulta de información de un corporativo. ....	49
Figura 3.21. Error de la vista de vacantes. ....	50
Figura 3.22. Solución en la vista. ....	50
Figura 3.23. Solución en el controlador. ....	50
Figura 3.24. Muestra del resultado final del error. ....	51
Figura 3.25. Muestra de error responsivo de la aplicación.....	51
Figura 3.26. Muestra de solución y resultado de error responsivo.....	51
Figura 3.27. Uso de Bootstrap para solución.....	52
Figura 4.0. Resultado final de la vista de candidatos.....	53
Figura 4.1. Resultado final de la vista de usuarios activos.....	53
Figura 4.2. Vista de vacantes con error solucionado.....	54



Figura 4.3. Vista final de corporativos.....	54
Figura 4.4. Vista final de corporativos con sus respectivas consultas de proyectos.....	55
Figura 4.5. Barra de navegación responsiva.....	55
Figura 5.0. Modelo de candidatos.....	63
Figura 5.1. Modelo para consultar personal.....	63
Figura 5.2. Modelo para candidatos.....	64
Figura 5.3. Modelo para los usuarios.....	64

## INTRODUCCIÓN

El proyecto que se realizó 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.

En el documento se encuentra diversa información sobre el proceso, herramientas utilizadas, así como su definición, problemáticas que se resolvieron, experiencia y conocimientos adquiridos entre otros elementos que fueron de gran importancia para el crecimiento profesional del estudiante.

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](mailto: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																				

Figura 1. Cronograma de actividades.

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 recepcionan 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.



Figura 2.0. Bases de datos y SGDB.

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).



Figura 2.1. Desarrollo web y tecnologías.

#### 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.

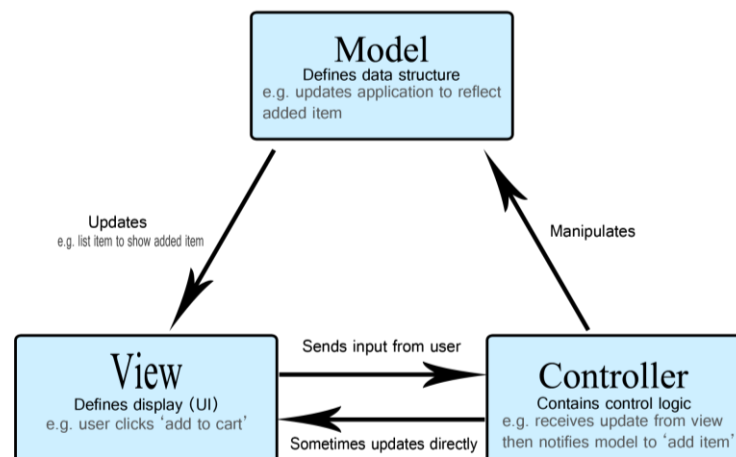


Figura 2.2. Arquitectura MVC.

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 donde el usuario interactúa con la aplicación. En la vista se muestra la información del modelo al usuario a través de una representación de los datos que el usuario es capaz de visualizar.

En el caso de una aplicación web, la vista consiste en una página HTML que el navegador puede 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).

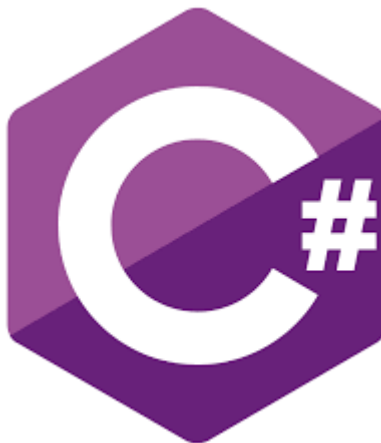


Figura 2.3. Logo de C#.

### 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. .NET Framework también ofrece un marco de trabajo para la creación de servicios web y aplicaciones de Windows. .NET Framework es compatible con una variedad de lenguajes de programación, incluidos C#, F# y Visual Basic.



Figura 2.4. .NET Framework.

.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#:



.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.

.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.

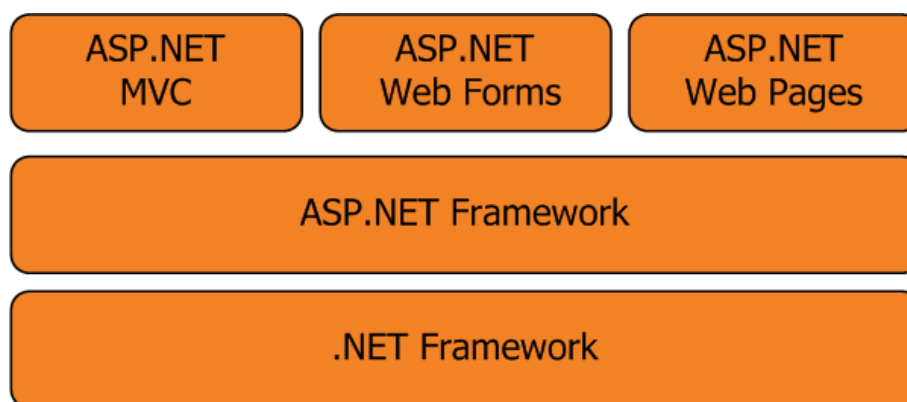


Figura 2.5. ASP .NET Framework.

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

.NET Core es una plataforma de aplicaciones de código abierto de Microsoft, que puede usarse para crear aplicaciones para Windows, Linux y macOS. Está diseñado para ser escalable y modular, para que los desarrolladores puedan crear y mantener aplicaciones de forma eficiente. .NET Core proporciona una variedad de herramientas para desarrolladores, incluidas bibliotecas, métodos de compilación, lenguajes de programación y herramientas de línea de comandos. Estas herramientas se pueden usar para crear aplicaciones para la nube, aplicaciones de escritorio y aplicaciones móviles.

.NET Core es una versión libre y de código abierto del Framework .NET de Microsoft. Está diseñado para ser rápido, flexible y moderno. Ofrece herramientas para desarrollar aplicaciones web, servicios de nube, aplicaciones de consola y aplicaciones móviles. Está optimizado para la nube, lo que significa que es ideal para desarrollar aplicaciones escalables para el cloud computing.



Figura 2.6. Logo de .NET Core.

.NET Core es una versión reducida y más moderna de la plataforma .NET Framework. Está diseñado para ser utilizado en entornos empresariales y también para la creación de aplicaciones web, aplicaciones móviles y servicios en la nube. .NET Core también ofrece un conjunto completo de herramientas de línea de comandos, incluidas herramientas de desarrollo, herramientas de pruebas unitarias y herramientas de depuración.

La autora Yazmín Nabor López da la siguiente definición de .NET Core en su libro Introducción a ASP.NET Core:

.NET Core es una implementación modular, multiplataforma y de código abierto d.NET Standard diseñada para que sea portátil entre plataformas a fin de permitir la reutilización del código al máximo y su uso compartido. Contiene muchas de las mismas APIs que contiene .NET Framework (pero .NET Core es un conjunto más pequeño) e incluye componentes de tiempo de ejecución (runtime), framework, compilador y herramientas que soportan una variedad de sistemas operativos y procesadores. La implementación de .NET Core fue orientada principalmente para las cargas de trabajo de ASP.NET Core, pero también por la necesidad y el deseo de tener una implementación más moderna. (Nabor, 2018, p. 8).

### 2.3.2. CARACTERÍSTICAS

1. Es multiplataforma: .NET Core es una plataforma multiplataforma que se ejecuta en Windows, MacOS y Linux.
2. Es de código abierto: .NET Core es de código abierto y está disponible en GitHub.
3. Es ligero y modular: .NET Core es ligero y modular, lo que significa que no está ligado a un sistema operativo y está diseñado para que los desarrolladores puedan escoger y descargar solo los componentes que necesitan para sus aplicaciones.
4. Es eficiente en términos de memoria: .NET Core usa una memoria mucho más eficiente que otros lenguajes, lo que permite a los desarrolladores crear aplicaciones más pequeñas y rápidas.

5. Soporta múltiples lenguajes: .NET Core admite varios lenguajes de programación, incluidos C#, F#, Visual
6. Posee herramientas de desarrollo, herramientas de pruebas unitarias y herramientas de depuración.

### 2.3.3. VENTAJAS

1. Plataforma multiplataforma: es una plataforma multiplataforma, lo que significa que puede ejecutarse en Windows, MacOS y Linux. Esto significa que el código puede compilarse para cualquiera de estos sistemas operativos, lo que hace que sea más fácil para los desarrolladores crear aplicaciones compatibles con varias plataformas.
2. Rendimiento mejorado: está optimizado para proporcionar un mejor rendimiento y estabilidad que su predecesor, .NET Framework. Esto se debe a que está construido a partir de cero para aprovechar los nuevos requisitos de infraestructura moderna.
3. Compatibilidad con la nube: es compatible con la nube, lo que significa que los desarrolladores pueden construir aplicaciones que se ejecuten en varias plataformas de nube, como Azure y Amazon Web Services. Esto ayuda a los desarrolladores a aprovechar la escalabilidad, el rendimiento y la seguridad ofrecidos por la nube, sin tener que configurar y administrar sus propios servidores.
4. Mejoras de seguridad: ofrece un mayor nivel de seguridad que su predecesor, .NET Framework. Esto se debe a que está construido a partir de cero para aprovechar las mejoras de seguridad de la infraestructura moderna.
5. Flexibilidad: es una plataforma de código abierto, lo que significa que los desarrolladores tienen una mayor flexibilidad para personalizar su código y construir aplicaciones únicas. Esto también significa que los desarrolladores no están limitados a la configuración predeterminada de .NET.

### 2.3.4. DESVENTAJAS

1. Menor cantidad de bibliotecas: tiene una librería de clases mucho más pequeña que la de .NET Framework. Esto significa que hay menos clases y métodos disponibles para usar.

2. Menor compatibilidad con los sistemas operativos: es compatible con Windows, macOS y Linux, pero no es compatible con el sistema operativo móvil como iOS y Android.
3. Falta de soporte para aplicaciones de escritorio: no es compatible con aplicaciones de escritorio, a diferencia de .NET Framework, que sí lo es.
4. La interface de usuario no es tan robusta: no tiene la misma interfaz de usuario que .NET Framework, que es mucho más robusta. Esto significa que hay una menor cantidad de controles de usuario disponibles para usar.
5. No es compatible con algunas herramientas de terceros: hay algunas herramientas de terceros que no son compatibles con .NET Core, como NuGet y Visual Studio. Esto significa que los desarrolladores no pueden usar estas herramientas para desarrollar aplicaciones .NET Core.
6. .Net Core aún no es compatible con todas las tecnologías de Microsoft. Aunque .NET Core es una versión mejorada de .NET Framework, no ofrece todas las características que se encuentran en .NET Framework.

#### 2.3.5. ASP .NET CORE

ASP.NET Core es una plataforma de código abierto web moderno y ligeramente orientado a la nube que se puede usar para crear aplicaciones web, servicios web y APIs web. Se puede ejecutar en Windows, macOS y Linux. Está diseñado para permitir que los desarrolladores creen aplicaciones de alta calidad con la máxima flexibilidad y seguridad. Está basado en la tecnología .NET Core, que ofrece la ventaja de permitir a los desarrolladores usar una variedad de lenguajes como C#, Visual Basic, F#, Python y JavaScript para crear aplicaciones web. ASP.NET Core también ofrece una gran cantidad de características y herramientas para ayudar a los desarrolladores a crear aplicaciones web eficaces.

Comparte algunos de los mismos componentes de .NET Framework. ASP .NET Core ofrece una variedad de características para ayudar a los desarrolladores a crear aplicaciones web modernas, incluidos enrutamiento, autenticación, soporte para contenedores, administración de la aplicación, autorización y mucho más.

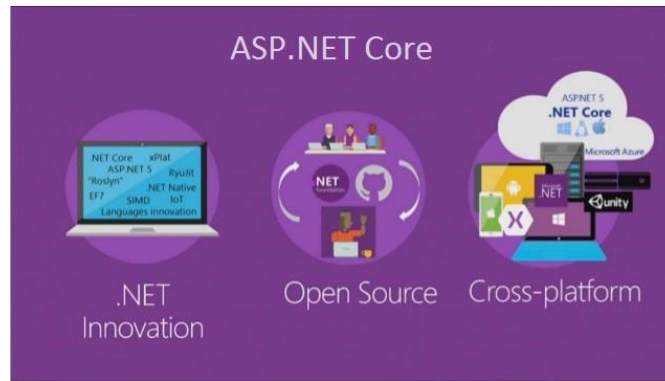


Figura 2.7. ASP .NET Core.

El sitio oficial de Microsoft define en su documentación a ASP .NET CORE de la siguiente manera: “Es la versión de código abierto de ASP.NET, que se ejecuta en macOS, Linux y Windows. ASP.NET Core se lanzó por primera vez en 2016 y es un rediseño de las versiones anteriores de ASP.NET solo para Windows.” (Windows, 2023, p. 1).

#### 2.4. .NET FRAMEWORK Y .NET CORE

.NET Framework es una plataforma de desarrollo de software desarrollada por Microsoft que se ejecuta principalmente en Windows. Está dirigido a desarrolladores de aplicaciones para crear aplicaciones de Windows y servidores web para la plataforma Windows. Por otra parte, .NET Core Es una plataforma de desarrollo de software de código abierto desarrollada por Microsoft. Está diseñado para ser una plataforma multiplataforma para desarrollar aplicaciones móviles, de escritorio y web para Windows, macOS, Linux y otros sistemas operativos.

.NET Framework es una plataforma de desarrollo de aplicaciones para Windows que se lanzó por primera vez en 2002. Está compuesto por una variedad de componentes, incluyendo la Common Language Runtime (CLR) y una biblioteca de clases que comprende el espacio de nombres de la Biblioteca de clases de .NET.

.NET Core es una versión de código abierto de .NET, que se lanzó en 2016. Está diseñado para ser una versión más ligera y modular de .NET, con el objetivo de ser multiplataforma, y se puede ejecutar en Windows, macOS y Linux. A diferencia d.NET Framework, .NET Core ofrece una arquitectura de componentes compilados de forma



nativa, lo que permite que la aplicación se ejecute más rápidamente. Esto también significa que .NET Core se puede ejecutar en múltiples sistemas operativos, lo que no es posible con .NET Framework.

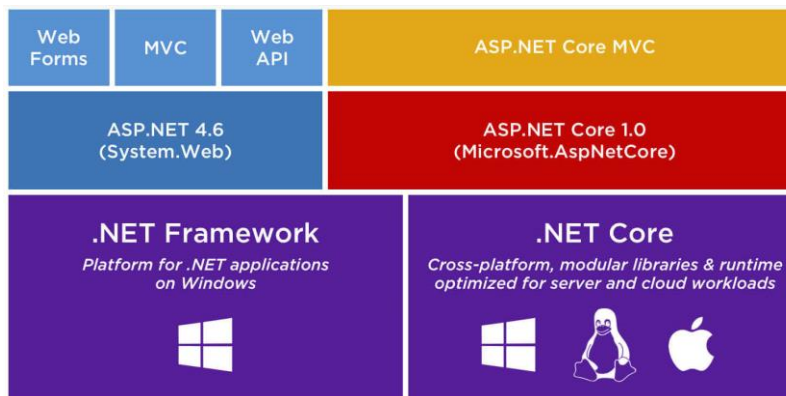


Figura 2.8. Comparativa entre .NET Framework y .NET Core.

En resumen, .NET Core es una versión de código abierto de .NET que ofrece una arquitectura de componentes compilados de forma nativa y es multiplataforma, mientras que .NET Framework es una plataforma de desarrollo de aplicaciones para Windows que ofrece una variedad de componentes, incluyendo la Common Language Runtime (CLR) y una biblioteca de clases.

1. .NET Framework es un entorno de ejecución de aplicaciones de Windows desarrollado por Microsoft que se ejecuta en una máquina virtual. Está diseñado para ejecutar aplicaciones escritas en lenguajes diseñados para la plataforma .NET, como Visual Basic y C#. .NET Core, por otro lado, es una versión moderna y multiplataforma de .NET Framework que se ejecuta en Windows, Linux y MacOS.
2. .NET Framework se actualiza de forma independiente de Windows; .NET Core, por el contrario, se actualiza con Windows Update.
3. .NET Framework no es compatible con la plataforma de código abierto; .NET Core, por el contrario, es de código abierto.
4. .NET Framework es una plataforma de desarrollo de aplicaciones para Windows; .NET Core, por otro lado, es una plataforma de desarrollo de aplicaciones multiplataforma.

5. .NET Framework no es compatible con el desarrollo de aplicaciones móviles; .NET Core, por el contrario, es compatible con el desarrollo de aplicaciones móviles.

## 2.5. VISTAS RAZOR

Razor es un lenguaje de marcado basado en la sintaxis de C# desarrollado por Microsoft, el cual se utiliza para desarrollar páginas web con Visual Studio para usar con ASP.NET. Está diseñado para proporcionar una sintaxis concisa y clara para combinar código HTML con el lenguaje de programación C#. Las vistas Razor permiten a los desarrolladores crear contenido dinámico y personalizado para los sitios web. Esto se logra mediante el uso de etiquetas HTML y C# para crear contenido dinámico. Las vistas Razor también permiten a los desarrolladores usar lenguaje de programación C# para generar código HTML sin escribir muchas líneas de código. Esto permite a los desarrolladores crear contenido dinámico para los sitios web de una manera más eficiente.

El ingeniero ESIEA y director de programas en Kantar Brice-Arnaud Guérin define a las vistas Razor en su libro ASP.NET con C# en Visual Studio 2017 de la siguiente manera:

Como alternativa a los Web Forms .aspx, Microsoft provee Razor, un sistema de representación de páginas HTML. Tiene como característica aligerar y optimizar la representación simplificando la sintaxis utilizada para describir las páginas. Razor está accesible, especialmente, en sistemas MVC que no disponen de componentes ricos como los Web Forms pero que, en contrapartida, evitan el uso de campos de caché ViewState, muy voluminosos. (Guérin, 2018, p. 245).

Las vistas Razor son una forma de escribir marcado HTML en un archivo con extensión .cshtml. Esta sintaxis se basa en la sintaxis ASP.NET y se puede usar para crear la interfaz de usuario de una aplicación. Las vistas Razor permiten realizar cualquier código C# en él, como bucles, condiciones y otras cosas. Las vistas Razor se usan comúnmente en las aplicaciones web basadas en ASP.NET.

Razor se usa para proporcionar una forma limpia y sencilla de escribir código en vistas ASP.NET. Los desarrolladores pueden escribir código C# directamente en la vista, lo que facilita la creación de páginas dinámicas. Esto evita la necesidad de escribir código HTML y luego escribir código C# en una clase separada para procesar los datos. La sintaxis de Razor es una mezcla de HTML y C#, lo que significa que los desarrolladores pueden escribir código en una forma más limpia y concisa. Esto también significa que los desarrolladores pueden escribir código HTML y C# en la misma vista, lo que simplifica el trabajo.

### 3. SQL SERVER

#### 3.1. DEFINICIÓN

SQL Server es una base de datos relacional de Microsoft que se usa para almacenar y recuperar información. Está diseñado para proporcionar un entorno de base de datos de alto rendimiento, seguridad y confiabilidad. Proporciona una amplia variedad de herramientas para la creación, administración y mantenimiento de bases de datos. También proporciona características como la replicación, la partición de datos, la recuperación de desastres, etc. para ayudar a los desarrolladores a construir y administrar bases de datos.



Figura 2.9. SQL Server.

#### 3.2. CARACTERÍSTICAS

Jose Santamaría da a conocer las siguientes características de SQL Server:

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.

- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos. (Santamaría, 2015, p. 1).

## 4. VISUAL STUDIO

### 4.1. DEFINICIÓN

Visual Studio es Un entorno de desarrollo integrado (IDE en inglés) creado por Microsoft para el desarrollo de aplicaciones de software para Microsoft Windows, así como aplicaciones web, servicios web, aplicaciones móviles, aplicaciones de escritorio, etc. Es el líder en el mercado de herramientas de desarrollo de software para la plataforma Windows. Microsoft Visual Studio incluye varias características para ayudar a los desarrolladores a crear aplicaciones de software de forma más rápida y eficiente, como herramientas de depuración, herramientas de compilación, herramientas de pruebas, herramientas de documentación, etc. Además, Visual Studio proporciona una amplia variedad de lenguajes de programación, como C#, Visual Basic, C++, F#, Python, JavaScript, TypeScript, etc.

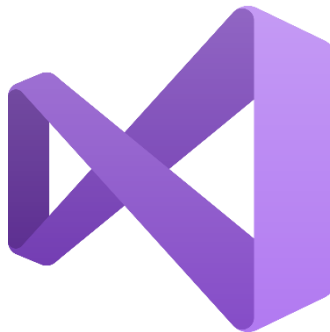


Figura 2.10. Visual Studio.

Está diseñado para ayudar a los desarrolladores de software a crear aplicaciones para la plataforma Microsoft, como Windows, Windows Mobile, Xbox 360, Windows Server, Internet Explorer, Microsoft Office y otros productos Microsoft. El IDE proporciona herramientas para diseñar, desarrollar, probar y depurar aplicaciones para varias plataformas. Además, el entorno ofrece herramientas para llevar a cabo la gestión de código fuente, el control de versiones, la administración de equipos remotos, la creación de documentación y la integración con otros sistemas de Microsoft.

La Licenciada con especialidad en Matemática e Informática Livias Segura, Nila de la Escuela Profesional de Matemática e Informática de la Universidad Nacional de Educación Enrique Guzmán y Valle da su definición de Visual Studio a través de su artículo para la obtención del Título Profesional de Licenciado en Educación: “Visual Studio es un conjunto de herramientas de desarrollo de software basadas en componentes y otras tecnologías para crear aplicaciones potentes y de alto rendimiento.” (Livias, 2018, p. 19).

#### 4.2. CARACTERÍSTICAS

1. IntelliSense: IntelliSense es una característica de Visual Studio que proporciona sugerencias de código para ayudar a los desarrolladores a escribir código más rápido.
2. Depuración y análisis: Visual Studio proporciona herramientas para depurar rápidamente el código y encontrar errores. También hay herramientas de análisis que ayudan a los desarrolladores a comprender mejor el código y mejorar la calidad.
3. Refactorización: Refactorizar el código es un proceso para simplificar y mejorar el código sin cambiar la funcionalidad. Visual Studio proporciona herramientas para refactorizar fácilmente el código y mejorar su calidad.
4. Plantillas de proyecto: Visual Studio ofrece plantillas de proyecto para crear aplicaciones de forma rápida. Estas plantillas contienen código de muestra para ayudar a los desarrolladores a obtener una idea general de cómo funciona un proyecto.
5. Herramientas de administración de código: Visual Studio permite a los desarrolladores administrar fácilmente su código. Se pueden descargar versiones

anteriores del código, comparar versiones, ver el historial de cambios, etc. Estas herramientas ayudan a los desarrolladores a administrar mejor el código.

## 5. LIBRERÍA Y FRAMEWORK

### 5.1. LIBRERÍA

Consiste en el conjunto de funciones o clases reutilizables que se utilizan para realizar alguna tarea en específico, permitiendo a los desarrolladores escribir programas de manera más eficiente y con menos esfuerzo. Las librerías contienen un conjunto de herramientas pueden ser usadas para agregar funcionalidad a un producto software. Estas herramientas contienen una colección de funciones y clases predefinidas que se pueden usar para realizar diversas tareas. Las librerías suelen ser menos complejas que los Frameworks y suelen requerir menos tiempo de configuración.

Paúl Andrés Jiménez Álvarez de la Universidad Tecnológica de Pereira da la siguiente definición: “Las librerías son un concepto antiguo. Estas no son más que una colección de métodos y clases de utilidad que el código escrito por un desarrollador puede llamar para obtener determinada funcionalidad.” (Jiménez, 2008, p. 19).

#### 5.1.1. JQUERY

jQuery es una biblioteca de JavaScript que se usa para simplificar la manera en que los desarrolladores interactúan con HTML, CSS y AJAX. Proporciona una variedad de métodos útiles para realizar tareas como animar elementos, manipular el contenido de una página web, implementar eventos, realizar solicitudes AJAX, etc. Esta biblioteca también reduce la cantidad de líneas de código necesarias para realizar ciertas tareas comunes, como la selección de elementos del DOM, el manejo de eventos, la animación y la manipulación de contenido HTML.



Figura 2.11. Librería JQuery.

La investigadora Johanna Fernanda Rengifo Briñez describe de la siguiente manera a JQuery: “Es una biblioteca de JavaScript rápida y concisa que simplifica el manejo de eventos, animación, y las interacciones Ajax para el desarrollo web rápido.” (Rengifo, 2011, p. 16).

## 5.2. FRAMEWORK

Un framework de desarrollo web es un conjunto de herramientas y tecnologías que se usan para crear aplicaciones web. Estos frameworks proporcionan una estructura para el desarrollo de aplicaciones web, incluyendo un conjunto de herramientas para la creación de interfaces de usuario, bases de datos, APIs y otros componentes. Contienen las herramientas y librerías para ayudar a los desarrolladores a crear aplicaciones robustas más rápido. Estos Frameworks o marcos trabajo proporcionan una estructura para desarrollar y organizar el código, permitiendo que el desarrollador se concentre en la lógica y no en la construcción de todos los componentes desde cero.

La desarrolladora Técnico Gerente de Cuentas Johanna Fernanda Rengifo Briñez de la Universidad Tecnológica De Pereira de la Facultad De Ingenierías Eléctrica, Electrónica, Física Y ciencias De La Computación da la siguiente definición de un Framework:

La palabra inglesa "framework" define como plataforma, entorno o marco de trabajo. También se puede definir como un conjunto estandarizado de conceptos, 28 prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar.

En el desarrollo de software, un framework es una estructura conceptual y tecnológica de soporte, en la cual otro desarrollo de software puede reutilizar este código. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto. (Rengifo, 2011, p. 27, 28).

### 5.2.1. BOOTSTRAP

Bootstrap es un Framework de código abierto para desarrollar sitios web y aplicaciones móviles con HTML, CSS y JavaScript. Fue creado por el equipo de desarrollo de Twitter y lanzado en agosto de 2011 como una herramienta para facilitar el desarrollo de sitios web con HTML, CSS y JavaScript. Bootstrap se basa en la idea de utilizar plantillas HTML y CSS para crear una estructura básica para la aplicación web. Esta estructura se compone de un conjunto de clases y etiquetas HTML predefinidas, que se pueden personalizar fácilmente. Esto permite a los desarrolladores crear interfaces de usuario modernas y responsivas sin tener que escribir cada línea de código desde cero. Asimismo, Bootstrap proporciona una gran cantidad de herramientas de diseño, como plantillas, temas, formularios, botones y cuadros de diálogo.



Figura 2.12. Framework de Bootstrap.

Además, Bootstrap proporciona una amplia colección de herramientas de diseño, como plantillas de diseño, componentes de interfaz de usuario, extensiones de JavaScript, herramientas de gráficos y más. Estas herramientas aceleran significativamente el proceso de desarrollo y permiten a los desarrolladores centrarse en la funcionalidad de la aplicación, en lugar de desperdiciar tiempo en la programación de la interfaz de usuario.

La empresa española acens technologies S.L. define a Bootstrap de la siguiente manera:

Se trata de un framework que ofrece la posibilidad de crear un sitio web totalmente responsive mediante el uso de librerías CSS. En estas librerías, nos podemos encontrar un gran número elementos ya desarrollados y listos para ser utilizados como pueden ser botones,



menús, cuadros e incluso un amplio listado de tipografías. (acens, 2016, p. 2).

Estas herramientas hacen que sea fácil para los desarrolladores crear sitios web y aplicaciones modernas y atractivas. Además, Bootstrap es compatible con los navegadores modernos y se puede utilizar en cualquier dispositivo, desde PC's de escritorio hasta teléfonos inteligentes. Esta flexibilidad hace que Bootstrap sea una excelente opción para los desarrolladores web que buscan crear sitios web.

# PROCEDIMIENTO Y DESCRIPCIÓN DE LAS ACTIVIDADES REALIZADAS

## DESARROLLO DE VISTA Y CONTROLADOR PARA ADMINISTRAR USUARIOS

Primero se desarrollaron las vistas en donde los usuarios pueden darse de alta, agregando su respectiva en formularios hechos en archivos cshtml y usando Bootstrap para el diseño. El cshtml utiliza un modelo el cual contiene la estructura de la tabla de la base de datos, que a su vez interactúa con el controlador.

The screenshot shows a web application interface for managing users. At the top, there's a navigation bar with links like 'Vacantes Activas', 'Candidatos', 'Requisiciones', 'Evaluaciones en Línea', 'Reporte Diario', and 'Administración'. The main heading is 'Usuarios'. Below it, there are two tabs: 'Altas' (selected) and 'Consulta'. The 'Altas de Usuarios' form includes fields for 'Candidato:', 'Nombre(s):', 'Email:', 'UserId:', 'CURP:', 'RFC:', 'Perfil:' (a dropdown menu), 'Fecha Ingreso:' (with a date picker set to 02/09/2023), and 'Jefe Directo:' (a dropdown menu). At the bottom of the form are 'Guardar' and 'Regresar' buttons. The footer indicates 'Estrategia Tecnológica © 2023'.

Figura 3.0. Vista principal de alta de usuarios.

También permite visualizar de manera individual la información de cada usuario, viendo información básica, las plazas en las que este puede postularse y seguimiento, en donde se pueden dar observaciones.

The screenshot displays the detailed view of a user's profile. At the top, there's a header with a 'Regresar a candidatos' button, the user's ID '26474', the name 'BERNARDO GABRIEL MEJIA PEREZ', the date '30/agosto/22', and an 'Abrir Captura' button. Below this, there are three tabs: 'Currículo' (selected), 'Plazas', and 'Seguimiento'. The 'Currículo' tab shows personal data: 'Contactado por <No Proporcionado>', 'Puesto solicitado SERVICIO SOCIAL', 'Nacionalidad Mexicana', 'Estado civil <Sin Proporcionar>', 'Fecha de nacimiento 30/agosto/2022' (with 'Edad 0 años'), 'Entrevista para el día 30/agosto/2022', 'Psicométrico [0] + [0] + [0] = [0]', 'Localización' (address, phone, email), and 'Educación' (2019 / 2022, Ingeniería Informática, TESCHA - TEC. EST. SUP. CHALCO, Licenciatura CURSANDO). The 'Plazas' and 'Seguimiento' tabs show application status and interview details.

Figura 3.1. Vista de los detalles del usuario seleccionado.

ESTRATEGIA TECNOLÓGICA

Vacantes Activas

Candidatos

Requisiciones

Evaluaciones en Línea

Reporte Diario

Hola, gmejia@estrategiatec.com.mx

Salir

Regresar a candidatos

26474

BERNARDO GABRIEL MEJIA PEREZ

30/agosto/22

Abrir Captura

Curriculo

Plazas

Seguimiento

Requisitos técnicos

Requisitos adicionales

Plazas que el candidato puede cubrir (Requisitos Técnicos)

Requisición	Puesto	Rango de percepciones	Cuota	Esquema	Zona	Inicio prob.	Duración
Herramientas: A=Administración, C=Configuración, O=Operación, P=Programación [Niveles entre 1 y 5 - Dominio en %]							
AMS20221111	DESARROLLADOR NET	\$20,000.00 - \$25,000.00	\$25,000.00	NÓMINA	CUICUILCO	05 Dec 2022	1 meses
<input type="checkbox"/> Sel							
<input type="checkbox"/> Des							
Match: 0	Herramientas con las que cumple:			Herramientas que le hacen falta:			
Prioridad: 0							
Vacantes: 0/1	Actividades:	Se busca programador net, gusto por la programación, sobresaliente en estudios y/o cursos. Capacidad de adaptación. Trabajo presencial					
AMS20221113	JAVA FULL STACK DEVELOPER- SENIOR	\$70,000.00 - \$81,000.00	\$81,000.00	NÓMINA+ PRESTACIONES SUPERIORES	HO	07 Dec 2022	12 meses
<input type="checkbox"/> Sel							
<input type="checkbox"/> Des							
Match: 0	Herramientas con las que cumple:			Herramientas que le hacen falta:			
Prioridad: 0							

Figura 3.2. Vista para las plazas para los usuarios.

ESTRATEGIA TECNOLÓGICA

Vacantes Activas

Candidatos

Requisiciones

Evaluaciones en Línea

Reporte Diario

Hola, gmejia@estrategiatec.com.mx

Salir

Regresar a candidatos

26474

BERNARDO GABRIEL MEJIA PEREZ

30/agosto/22

Abrir Captura

Curriculo

Plazas

Seguimiento

Seguimiento

☐ Comentario positivo
 ☐ Comentario negativo
 ☒ Observación general

Agregar

Estrategia Tecnológica © 2023

Figura 3.3. Vista del componente de seguimiento del usuario.

La manera en que se hizo el formulario de usuarios fue de la siguiente manera, en donde se utilizó Razor para añadir un TextBox para mostrar la información del modelo y controlador, y en los demás elementos del formulario se usaron etiquetas HTML.

```

<br />
<div class="container">
  <div class="row">
    <div class="row col-5">
      <label class="col-sm-2 col-form-label">Candidato:</label>
      <div class="col-sm-4">
        @Html.TextBox("Candidato", null, new { required = "required", @class = "txtCand form-control form-control-sm" })
        <label for="Candidato" class="text-danger" style="display:none" id="lblCandidato"></label>
      </div>
    </div>
  </div>
  <br />
  <div class="row">
    <div class="row col-5">
      <label class="col-sm-2 col-form-label">Nombre(s):</label>
      <div class="col-sm-8">
        <input type="text" class="form-control form-control-sm" id="txtNombre" onkeypress="return SoloLetras(this)" onblur="upperCase()" />
        <label for="Nombre" class="text-danger" style="display:none" id="lblErrorNombre"></label>
      </div>
    </div>
    <div class="row col-4">
      <label class="col-sm-2 col-form-label">Email:</label>
      <div class="col-sm-9">
        <input type="text" class="form-control form-control-sm" id="email" />
        <span id="emailOK" class="text-danger"></span>
      </div>
    </div>
  </div>

```

Figura 3.4. Código de la vista del formulario de usuarios.

```

<label class="col-sm-3 col-form-label">UserId:</label>
<div class="col-sm-9"><input type="text" class="form-control form-control-sm" id="txtUserId" /></div>
</div>
<br />
<div class="row">
  <div class="row col-5">
    <label class="col-sm-2 col-form-label">CURP:</label>
    <div class="col-sm-8">
      <input type="text" class="form-control form-control-sm" oninput="CURP(this)" id="txtCURP" />
      <pre id="result"></pre>
    </div>
  </div>
  <div class="row col-4">
    <label class="col-sm-2 col-form-label">RFC:</label>
    <div class="col-sm-9">
      <input type="text" class="form-control form-control-sm" oninput="RFC(this)" id="txtRFC" />
      <pre id="resultado"></pre>
    </div>
  </div>
  <div class="row col-3">
    <label class="col-sm-3 col-form-label">Perfil:</label>
    <div class="col-sm-9">
      @Html.DropDownListFor(model => model.PerfilId, new SelectList(Model.Perfiles, "PerfilId", "Perfil_Desc"), "Seleccione...", new { id = "ddlPerfiles", @class =
    </div>
  </div>

```

Figura 3.5. Formulario para registrar usuarios.

Y en esta parte se usó de Razor la opción de agregar un DropDownListFor, el cual añade de manera dinámica un elemento select de HTML, mostrando las options de acuerdo con los parámetros que se le pasen, en este caso se trajo del controlador la información del campo PerfilId de su respectivo modelo.

```

<label class="col-sm-3 col-form-label">UserId:</label>
<div class="col-sm-9"><input type="text" class="form-control form-control-sm" id="txtUserId" /></div>
</div>
<br />
<div class="row">
  <div class="row col-5">
    <label class="col-sm-2 col-form-label">CURP:</label>
    <div class="col-sm-8">
      <input type="text" class="form-control form-control-sm" oninput="CURP(this)" id="txtCURP" />
      <pre id="result"></pre>
    </div>
  </div>
  <div class="row col-4">
    <label class="col-sm-2 col-form-label">RFC:</label>
    <div class="col-sm-9">
      <input type="text" class="form-control form-control-sm" oninput="RFC(this)" id="txtRFC" />
      <pre id="resultado"></pre>
    </div>
  </div>
  <div class="row col-3">
    <label class="col-sm-3 col-form-label">Perfil:</label>
    <div class="col-sm-9">
      @Html.DropDownListFor(model => model.PerfilId, new SelectList(Model.Perfiles, "PerfilId", "Perfil_Desc"), "Seleccione...", new { id = "ddlPerfiles", @class =
    </div>
  </div>

```

Figura 3.6. Creación de un select de HTML con Razor.

También se creó otro DropDownListFor para mostrar un select con los datos del jefe directo. También se tiene un Actionlink que funciona como una etiqueta a de HTML, que regresa a una determinada ruta.

```
<br />
<div class="row">
  <div class="row col-3">
    <label class="col-sm-6 col-form-label">Fecha Ingreso:</label>
    <div class="col-sm-6">
      <input type="text" class="form-control form-control-sm" id="Fecha_Inicio" onkeypress="fecha()" size="15" />
    </div>
  </div>
  <div class="row col-4">
    <label class="col-sm-5 col-form-label">Jefe Directo:</label>
    <div class="col-sm-6">
      @Html.DropDownListFor(model => model.UsuariosActivos.Personal_Id, new SelectList(Model.UsuariosActivos.UActivos, "Personal_Id", "Personal_UserId"), "Selecci
    </div>
  </div>
</div>
<br />
<div class="form-inline">
  <div class="row">
    <div class="form-group">
      <div class="col-sm-12">
        <input type="submit" value="Guardar" class="btn btn-success" onclick="Guardar()" />
        @Html.ActionLink("Regresar", "GetAll", "Poliza", htmlAttributes: new { @class = "btn btn-danger", autocomplete = "off" })
      </div>
    </div>
  </div>
</div>
```

Figura 3.7. Select de HTML para información de los jefes directos del usuario.

En la parte de JQuery se crearon varias funciones para obtener las fechas de manera dinámica, usando el componente de datepicker para mostrar un calendario interactivo de JQuery.

```
$(document).ready(function () {
  $("#Fecha_Inicio").datepicker({
    changeMonth: true,
    changeYear: true,
    dateFormat: "dd/mm/yy"
  });
  $("#Fecha_Salida").datepicker({
    changeMonth: true,
    changeYear: true,
    dateFormat: "dd/mm/yy"
  })
  $("#Candidato").autocomplete({
    source: function (request, response) {
      $.ajax({
        url: "../AltaUsuarios/buscarNombres",
        data: "{ 'q': '" + request.term + "' }",
        dataType: "json",
        type: "POST",
        contentType: "application/json; charset=utf-8",
        success: function (data) {
          response($.map(data, function (d) {
            return {
              label: d.Nombre_Candidato + " - " + d.ID_Candidato,
              value: d.Nombre_Candidato,
              id: d.ID_Candidato
            }
          })
        )
      }
    }
  })
})
```

Figura 3.8. Función de JQuery para componente de calendario del formulario.

También se agregaron funciones de validación de campos para el rfc y el curp, a través de expresiones regulares.

```

    }
    //Función para validar un RFC
    // Devuelve el RFC sin espacios ni guiones si es correcto
    // Devuelve false si es inválido
    // (debe estar en mayúsculas, guiones y espacios intermedios opcionales)
    function rfcValido(rfc, aceptarGenerico = true) {
        const re = /^(A-Z)([3,4])?(?:-)?(\d{2})(?:0[1-9]|1[0-2])(?:0[1-9]|[12]\d|3[01])(?:-)?(?:A-Z\d{2})?(A\d)?$/;
        var validado = rfc.match(re);

        if (!validado) //Coincide con el formato general del regex?
            return false;

        //Separar el dígito verificador del resto del RFC
        const digitoVerificador = validado.pop(),
            rfcSinDigito = validado.slice(1).join(''),
            len = rfcSinDigito.length,

        //Obtener el dígito esperado
        diccionario = "0123456789ABCDEFGHIJKLMNÑOPQRSTUVWXYZ Ñ",
            indice = len + 1;
        var suma,
            digitoEsperado;

        if (len == 12) suma = 0
        else suma = 481; //Ajuste para persona moral
    }

```

Figura 3.9. Función de JQuery para validaciones de RFC.

```

//Función para validar una CURP
function curpValida(curp) {
    var re = /^[A-Z][AEIOUX][A-Z]{2}\d{2}(?:0[1-9]|1[0-2])(?:0[1-9]|[12]\d|3[01])[HM](?:AS|B[CS]|C[CLMSH])|C[CLMSH]I[NDP]$/;
    validado = curp.match(re);

    if (!validado) //Coincide con el formato general?
        return false;

    //Validar que coincida el dígito verificador
    function digitoVerificador(curp17) {
        //Fuente https://consultas.curp.gob.mx/CurpSP/
        var diccionario = "0123456789ABCDEFGHIJKLMNÑOPQRSTUVWXYZ",
            lngSuma = 0.0,
            lngDigito = 0.0;
        for (var i = 0; i < 17; i++)
            lngSuma = lngSuma + diccionario.indexOf(curp17.charAt(i)) * (18 - i);
        lngDigito = 10 - lngSuma % 10;
        if (lngDigito == 10) return 0;
        return lngDigito;
    }

    if (validado[2] != digitoVerificador(validado[1]))
        return false;

    return true; //Validado
}

```

Figura 3.10. Función de JQuery para validar CURP.

También se tienen diferentes validaciones para los demás campos del formulario.

```

function SoloLetras(e) {
    var letra = e.onkeypress.arguments[0].key;
    if (!/^[a-zA-Z ]/.test(letra)) {
        $("#lblErrorNombre").hide();
        return true;
    }
    else {
        if (e.id == "txtNombre") {
            $("#lblErrorNombre").text('Solo se aceptan letras');
            $("#lblErrorNombre").show();
        }
        return false;
    }
}

function SoloNumeros(e) {
    var letra = e.onkeypress.arguments[0].key;
    if (!/^[0-9]/g.test(letra)) {
        $("#lblCandidato").hide();
        return true;
    }
    else {
        if (e.id == "txtCandidato") {
            $("#lblCandidato").text('Solo se aceptan numeros');
            $("#lblCandidato").show();
        }
    }
}

```

Figura 3.11. Validaciones con JQuery para los campos restantes.

```

function PrimeraLetraMayuscula(string) {
    return string.charAt(0).toUpperCase() + string.slice(1);
}
function Guardar() {
    var Usuario = {
        Nombre: $('#txtNombre').val(),
        CandidatoId: $('#Candidato').val(),
        PersonalId: 0,
        UserId: $('#txtUserId').val(),
        FechaIngreso: $('#Fecha_Inicio').val(),
        CURP: $('#txtCURP').val(),
        RFC: $('#txtRFC').val(),
        Email: $('#email').val(),
        iPerfil: $('#ddlPerfiles').val(),
        FechaSalida: $('#Fecha_Salida').val(),
        iEmpresa: 0,
        iNominaId: 0,
        iAsignadoA: $('#ddlUsuarios')
    }
    Alta(Usuario);
}

```

Figura 3.12. Función de JQuery para guardar los valores del formulario.

## DESARROLLO DE VISTA Y CONTROLADOR PARA CORPORATIVOS

La siguiente imagen se muestra la vista de corporativos, donde se lleva a cabo la administración de los proyectos en los que la empresa está involucrada.

ESTRATEGIA TECNOLÓGICA Vacantes Activas Candidatos Requisiciones Evaluaciones en Línea Reporte Diario

### Corporativos / Areas

[+ Nuevo](#)

Mostrar 10 registros Buscar:

Corporativo	Categoria	Proyectos
-APP: BEACON TECH	Propio	<a href="#">Proyectos</a>
-APP: CANDIDATOS	Propio	<a href="#">Proyectos</a>
-APP: EXAMENES LINEA	Propio	<a href="#">Proyectos</a>
-APP: EXAMS TECNICOS	Propio	<a href="#">Proyectos</a>
-APP: HMN	Propio	<a href="#">Proyectos</a>
-APP: IAVISOS	Propio	<a href="#">Proyectos</a>
-APP: PERSONAL	Propio	<a href="#">Proyectos</a>
-APP: PORTAL DOCENTE	Propio	<a href="#">Proyectos</a>
-APP: PORTAL EDUCAT.	Propio	<a href="#">Proyectos</a>
-APP: R.REGULATORIOS	Propio	<a href="#">Proyectos</a>

Mostrando registros del 1 al [Anterior](#) [1](#) [2](#) [3](#) [4](#) [5](#) ... [13](#) [Siguiente](#)

Figura 3.13. Vista principal de los corporativos.

Primero se utilizó el siguiente modelo que representa a la tabla corporativos de la base de datos, manteniendo tanto sus campos como las relaciones con otras tablas.

```

6 referencias
public partial class sp_ObtenerCorporativos_Result
{
    3 referencias
    public int Corporativo_Id { get; set; }
    4 referencias
    public string Corporativo_Dsc { get; set; }
    0 referencias
    public short Corporativo_Financiamiento { get; set; }
    0 referencias
    public byte Corporativo_Status_Id { get; set; }
    0 referencias
    public string Corporativo_Status_Dsc { get; set; }
    0 referencias
    public byte Corporativo_Categoria_Id { get; set; }
    2 referencias
    public string Corporativo_Categoria_Dsc { get; set; }
    0 referencias
    public Nullable<decimal> Corporativo_Facturacion { get; set; }
    0 referencias
    public Nullable<int> Corporativo_Retraso { get; set; }
}

```

Figura 3.14. Modelo de corporativos.

Primero se desarrolló la vista de acuerdo con modelo y al controlador, usando Razor para mostrar los datos dinámicamente.

```

<div class="card-body hide" id="divCorporativo">
    @using (Ajax.BeginForm("_Corporativos", "Corporativo", new { }, new AjaxOptions { HttpMethod = "POST", InsertionMode = InsertionMode.Replace, UpdateTargetId = "corporativo" })
    {
        <div class="form-row">
            <div class="row col-10">
                <label class="col-sm-2 col-form-label">Corporativo</label>
                <div class="col-sm-8">
                    @Html.TextBox("corporativoName", null, new { @class = "form-control form-control-sm", required = true })
                </div>
            </div>
            <div class="form-row">
                <div class="row col-10">
                    <label class="col-sm-2 col-form-label">Categoria</label>
                    <div class="col-sm-8">
                        @Html.DropDownList("categoriaId", new SelectList(ViewBag.Categorias, "Value", "Text"), "Seleccione...", new { required = true, @class = "form-control form-control-sm" })
                    </div>
                </div>
                <div class="float-right">
                    <input type="submit" value="Agregar" class="btn btn-secondary btn-sm" />
                </div>
            </div>
        </div>
    }
</div>
</div>

```

Figura 3.15. Creación del formulario de corporativos en la vista.

Para no hacer el código HTML muy grande, se utilizaron vistas parciales, de esta manera solo se requiere una parte de código la cual puede ser reutilizada en otras vistas de ser necesario.

```

</div>
<br />
@Html.Action("_Corporativos", new { })
</div>
<div class="col-6">
    <div id="proyectospartial">
    </div>
</div>
</div>
</div>

```

Figura 3.16. Utilización de una vista parcial de los corporativos.



Para la vista parcial se utilizó los datos del contexto, el cual se encarga de comunicar al controlador con la Base de datos a través del modelo, por lo que se hizo uso de un procedimiento almacenado pasándole por parámetros la información de los corporativos para que retorne la vista con dicha información.

```
0 referencias
public ActionResult _Corporativos()
{
    var corporativos = contexto.sp_ObtenerCorporativos(0, 0, 0).ToList();
    ViewBag.Corporativos = corporativos;
    return View();
}
```

Figura 3.17. Controlador de corporativos para obtener la información.

Finalmente se mostró en una tabla los datos de los corporativos seleccionados, mostrando la descripción, categoría y los proyectos de ese corporativo.

```
<table id="tableCorporativos" class="table table-striped table-sm">
  <thead class="thead-light">
    <tr>
      <th>Corporativo</th>
      <th>Categoría</th>
      <th></th>
    </tr>
  </thead>
  <tbody>
    <@foreach (var item in ViewBag.Corporativos as List<sp_ObtenerCorporativos_Result>)
    {
      <tr>
        <td>@item.Corporativo_Dsc</td>
        <td>@item.Corporativo_Categoria_Dsc</td>
        <td>@Ajax.ActionLink("Proyectos", "_Proyectos", new { id = item.Corporativo_Id, n=item.Corporativo_Dsc },
        </tr>
      }
    </tbody>
  </table>
```

Figura 3.18. Listado de los corporativos con Razor.

Para mostrar la información en una tabla dinámica, es decir, que permita ordenar los datos de forma ascendente o descendente, que dé la opción de buscar o mostrar de 10 a 100 registros, se utilizó la librería DataTable de JQuery, obteniendo primero la referencia de la tabla a través de su id para después llamar la función DataTable y pasándole por parámetros las opciones que ofrece la librería.

```

<script>
$(document).ready(function () {
    var table = $('#tableCorporativos').DataTable({
        "searching": true,
        "paging": true,
        "ordering": true,
        "info": true,
        "stateSave": true,
        "scrollCollapse": true,
        "scroller": true,
        "language": {
            "url": "/Scripts/SpanishDT.json"
        }
    });
});
</script>

```

Figura 3.19. Utilización de librería de JQuery para información de la tabla.

De este modo, se pueden visualizar la información de los proyectos de ese corporativo.

The screenshot shows a web application interface with two main sections. The left section, titled 'Corporativos / Areas', contains a table with columns 'Corporativo', 'Categoria', and 'Proyectos'. The right section, titled 'Proyectos / Grupo de Actividades: -APP: BEACON TECH', contains a table with columns 'Acronimo', 'Proyecto Desc', and 'Interno'.

Corporativo	Categoria	Proyectos
-APP: BEACON TECH	Propio	Proyectos
-APP: CANDIDATOS	Propio	Proyectos
-APP: EXAMENES LINEA	Propio	Proyectos
-APP: EXAMS TECNICOS	Propio	Proyectos
-APP: HMN	Propio	Proyectos
-APP: IAVISOS	Propio	Proyectos
-APP: PERSONAL	Propio	Proyectos
-APP: PORTAL DOCENTE	Propio	Proyectos
-APP: PORTAL EDUCAT.	Propio	Proyectos
-APP: R.REGULATORIOS	Propio	Proyectos

Acronimo	Proyecto Desc	Interno
<S/PROYECTO>	<S/PROYECTO>	no
ADMINISTRADOR WEB	Administrador para el funcionamiento del app móvil	no
APP ANDROID	APP ANDROID	yes
APP IOS	APP IOS	yes
DISEÑO	DISEÑO DE INTERFACE	yes
MARKETING	investigación de mercado	no
MARKETING	-MARKETING	yes
SEGUIMIENTO	REUNION DE SEGUIMIENTO	yes
SITIO WEB	SITIO WEB	yes

Figura 3.20. Consulta de información de un corporativo.

## CORRECCIÓN DE DISEÑO Y BUGS

En la vista de Vacantes existía un error de diseño y lógica, en donde había un select de HTML que contenía información errónea, mostrando una opción para mostrar todas las vacantes y las demás tenían el texto de Borrar, ya que no eran requeridas, pues ninguna interactuaba con el controlador ni con el modelo, solo la primera opción.



Figura 3.21. Error de la vista de vacantes.

Lo que se realizó fue modificar el controlador para que el procedimiento almacenado para obtener la información del equipo siempre seleccione la opción de <Todos>, de esta manera no tomaría en cuenta las demás opciones. También se modificó la vista eliminando ese elemento del HTML, dejando sólo las opciones del mes y de los integrantes.

```
<div class="form-inline justify-content-center">
  <label class="mx-2">Mes</label>
  @Html.DropDownListFor(model => model.PeriodoId, Model.PeriodoList, "Selecione...", htmlAttributes: new { @class = "form-control form-control-sm" })
  <label class="mx-2">Equipo</label>
  @Html.DropDownListFor(model => model.EquipoId, Model.EquipoList, "Selecione...", htmlAttributes: new { @class = "form-control form-control-sm" })
  <label class="mx-2">Integrante</label>
  @Html.DropDownListFor(model => model.IntegranteId, Model.IntegranteList, htmlAttributes: new { @class = "form-control form-control-sm" })
  <button type="submit" class="btn btn-primary btn-sm mx-3">
    <i class="fas fa-sync-alt"> Buscar</i>
  </button>
</div>
```

Figura 3.22. Solución en la vista.

```
private void PopulateEquipoList(SearchModel model)
{
    model.EquipoList = db.sp_SelectEquiposReclutadoras(0).Select(o => new SelectListItem
    {
        Text = o.Equipo_Recluta_Acrónimo,
        Value = o.Equipo_Recluta_Id.ToString()
    });
    model.EquipoId = 4;
}
```

Figura 3.23. Solución en el controlador.

De esta manera, se corrigió ese bug y ahora ya no aparecía ese SELECT, e internamente ya traía la información automáticamente del equipo sin necesidad de utilizar el formulario.

The screenshot shows the 'Vacantes Activas' section of a web application. At the top, there is a navigation bar with links: 'ESTRATEGIA TECNOLÓGICA', 'Vacantes Activas', 'Candidatos', 'Requisiciones', 'Evaluaciones en Línea', and 'Reporte Diario'. On the right, it says 'Hola, gmejia@estrategiatec.com.mx' and a 'Salir' link. Below the navigation bar, there is a search form titled 'Vacantes Activas' with a 'Mes' dropdown (showing 'Seleccione...'), an 'Integrante' dropdown, and a 'Buscar' button. Below the search form, there are checkboxes for 'Incluir: Canceladas', 'Detenidas', and 'Expiraron'. Below this is a table header with columns: 'N° Requisición', 'Interno/ Externo', 'Cliente - Perfil', 'Vacantes x Cubrir', 'Total Mensual', 'Prioridad', 'Estado', 'Cambio de estado', 'Cubiertas', 'Tipo de vacante', 'Descartes/ Enviados', 'Horas Recluta', 'Semanas de Cobro', 'Cuota x Recurso', 'Estimado en el mes', and 'Concretado'. Below the table header, it says 'Ningún dato disponible en esta tabla'.

Figura 3.24. Muestra del resultado final del error.

También había errores de maquetación, ya que, en el menú principal superior, cuando se le hacía zoom, se desbordaba toda la información, haciendo inaccesibles algunas opciones, como cerrar sesión.

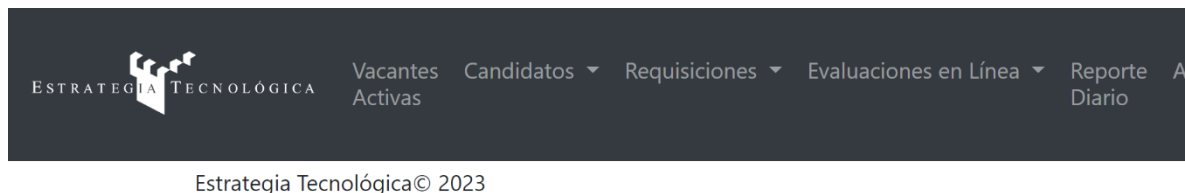


Figura 3.25. Muestra de error responsivo de la aplicación.

Por lo que se añadieron clases y estilos de Bootstrap, en donde, al tener resoluciones bajas, el menú superior ahora aparecía como un botón.

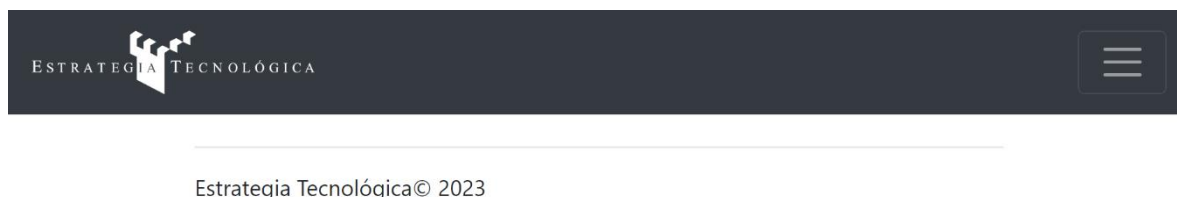


Figura 3.26. Muestra de solución y resultado de error responsivo.

Todo esto fue con clases que ofrece Bootstrap.

```
y>  
<div class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top fh-fixedHeader">  
  <a class="navbar-brand loading" href="/Home/Index">  
      
  </a>  
  <button type="button" class="navbar-toggler" data-toggle="collapse" data-target=".navbar-collapse">  
    <span class="navbar-toggler-icon"></span>  
  </button>  
</div>
```

Figura 3.27. Uso de Bootstrap para solución.

## RESULTADOS

Con las vistas desarrolladas en la intranet, se obtuvo una mejor organización en cuanto a la información, ya que antes se tenía la siguiente vista, en la que se puede buscar a cualquier candidato, obteniendo todos aquellos que en algún momento estuvieron registrados dentro de la empresa, ya sean trabajadores, becarios o de servicio social.

**Consulta de Candidatos**

Periodo: Último semestre Desde: 24/07/22 Hasta: 24/01/23 Entrevistó: Seleccione...

Ids: Nombre: Herramientas:

**Nuevo candidato**

Mostrar: 10 registros

ID	Nombre	Estatus	Actualizado	Creado	Puesto	Preferencias	Teléfono	Email	Ent Inicial	Herr Lab
26795		En Proceso (Ter Vot)	23 Jan 23	23 Jan 23	Desarrollador .NET					2
26794		En Proceso (Ter Vot)	23 Jan 23	23 Jan 23	Operador informática					2
26793	47.7	En Proceso (Ter Vot)	20 Jan 23	20 Jan 23	Desarrollador iOS					2
26792		En Proceso (Ter Vot)	18 Jan 23	18 Jan 23	SERVICIO SOCIAL					

Descarga: Doc Pdf Docx Print Print

26791		En Proceso (Ter Vot)	13 Jan 23	13 Jan 23	SERVICIO SOCIAL					
26790		En Proceso (Ter Vot)	12 Jan 23	12 Jan 23	Desarrollador .NET					3
26789		En Proceso (Ter Vot)	12 Jan 23	12 Jan 23	Analyst Support					2

Figura 4.0. Resultado final de la vista de candidatos.

Pero ahora se tiene una vista en donde solo están los empleados activos, mostrando información relevante, como el nombre, fecha de registro, etc., además en la misma vista se puede dar de alta a un nuevo usuario.

**Usuarios**

Altas

**Consulta de Usuarios**

Nombre	User Name	CURP	RFC	Email	Fecha Ingreso	Estatus
	Ricardo Guerrero				28/02/1996	<input checked="" type="checkbox"/>
	Edgar Trevedhan				28/02/1996	<input checked="" type="checkbox"/>
	David Guerrero				16/06/2002	<input checked="" type="checkbox"/>
	Daniel Osorio				18/04/2006	<input checked="" type="checkbox"/>
	Magdalena Guerrero				21/07/2008	<input checked="" type="checkbox"/>
	Susana Reyes				30/01/2009	<input checked="" type="checkbox"/>
	Anya Marin				15/06/2009	<input checked="" type="checkbox"/>
	Alejandro Moyado				15/03/2007	<input checked="" type="checkbox"/>
	Felipe Vargas				17/08/2010	<input checked="" type="checkbox"/>
	Eduardo Gaytan				26/03/2015	<input checked="" type="checkbox"/>
	Angel Cortez				13/01/2017	<input checked="" type="checkbox"/>
	Alberto Ramirez				26/03/2010	<input checked="" type="checkbox"/>
	Nallely Vera				02/05/2017	<input checked="" type="checkbox"/>

Figura 4.1. Resultado final de la vista de usuarios activos.

La vista de las vacantes se modificó para arreglar detalles visuales y de lógica, eliminando componentes innecesarios que confundían a las personas que manejaban los datos a través de esta intranet.

ESTRATEGIA TECNOLÓGICA

Vacantes Activas

Candidatos

Requisiciones

Evaluaciones en Línea

Reporte Diario

Hola, gmejia@estrategiatec.com

Vacantes Activas

Mes

Jan 23

Integrante

< Todos >

Buscar

Incluir:

☒ Cargadas

☒ Detenidas

☒ Expiraron

Plazas totales	58	Horas de reclutamiento	690.9	Concretado	\$124,250.0	Plazas externas	0
Disponibles	21	Procesos trabajados	26 de 58	Estimado en el mes	\$25,000.0	Plazas externas a cubrir en el mes	0
Objetivos del mes	1	Promedio	26.57 hrs/proceso	Canceladas, detenidas o expiradas	34	Plazas externas cubiertas en el mes	0 - 0.0%
Plazas cubiertas	3 - 300.0%	Nuevos en el mes	4				

N°	Requisición	Interno/ Externo	Cliente - Perfil	Vacantes x Cubrir				Cambio de estado	Cubiertas	Tipo de vacante	Descartes/ Enviados	Horas Recluta	Semanas de Cobro	Cuota x Recurso	Estimado en el mes
				Total	Menual	Prioridad	Estado								
<SIN ASIGNACION>															
1	AMS20221203	Interno	CTA: COGNIZANT - LEAD JAVA	1	0	Baja	Expiró	12 Jan 23	0	Normal	0/0	0.0	0.0	\$0.0	\$0.0
1	AMS20221204	Interno	CTA: COGNIZANT - PROJECT MANAGER SR	1	0	Baja	Expiró	17 Jan 23	0	Normal	0/0	0.0	0.0	\$0.0	\$0.0
158 -	BCR20221006	Interno	CTA: SCOTIABANK - ADMINISTRADOR WINDOWS (INTEL)	2	0	Baja	Activa	06 Oct 22	0	Normal	0/0	0.0	0.0	\$0.0	\$0.0
Brenda.Real				4	0				0	6/6	0.0			\$0.0	\$0.0
1	BCR20220902	Interno	CTA: BRILLIO - JAVA BACKEND	4	0	Baja	Cub. Parcial	10 Jan 23	0	Normal	6/6	0.0	0.0	\$0.0	\$0.0
Fernanda.Reynoso				1	0				0	0/0	0.0			\$0.0	\$0.0
1	INV20221003	Interno	CTA: INDRA - DESARROLLADOR ASD	1	0	Baja	Cancelada	06 Jan 23	0	Normal	0/0	0.0	0.0	\$0.0	\$0.0
Itzel.Pacheco				17	0				0	5/10	299.0			\$0.0	\$0.0
1	INGL20220904	Interno	EMP: ESTRATEGIA - DESARROLLADOR IOS	1	0	Baja	Expiró	05 Jan 23	0	Normal	0/0	43.0	0.0	\$0.0	\$0.0
1	AMS20221205	Interno	BERRYMEK - INGENIERO DE SOPORTE TECNICO GLOBAL	0	0	Baja	Cub. Total	09 Jan 23	0	Normal	0/0	10.0	0.0	\$0.0	\$0.0

Figura 4.2. Vista de vacantes con error solucionado.

Asimismo, se tenía mayor manejo de la información de los proyectos, gestionando mejor los tiempos que se dedicaban a cada uno a través de su prioridad.

ESTRATEGIA TECNOLÓGICA

Vacantes Activas

Candidatos

Requisiciones

Evaluaciones en Línea

Reporte Diario

Hola, gmejia@estrategiatec.com.mx

Salir

Corporativos / Areas

Nuevo

Mostrar 10 registros

Buscar:

Corporativo	T:	Categoría	T:	T:
-APP: BEACON TECH		Propio		Proyectos
-APP: CANDIDATOS		Propio		Proyectos
-APP: EXAMENES LINEA		Propio		Proyectos
-APP: EXAMS TECNICOS		Propio		Proyectos
-APP: HMN		Propio		Proyectos
-APP: IAVISOS		Propio		Proyectos
-APP: PERSONAL		Propio		Proyectos
-APP: PORTAL DOCENTE		Propio		Proyectos
-APP: PORTAL EDUCAT.		Propio		Proyectos
-APP: R.REGULATORIOS		Propio		Proyectos

Mostrando registros del 1 al 13

Anterior

1

2

3

4

5

...

13

Siguiente

Proyectos / Grupo de Actividades: -APP: BEACON TECH

Nuevo

Mostrar 10 registros

Buscar:

Acronimo	T:	Proyecto	Dsc	T:	Interno	T:
<S/PROYECTO>		<S/PROYECTO>				no
ADMINISTRADOR WEB		Administrador para el funcionamiento del app móvil				no
APP ANDROID		APP ANDROID				yes
APP IOS		APP IOS				yes
DISEÑO		DISEÑO DE INTERFACE				yes
MARKETING		investigación de mercado				no
MARKETING		-MARKETING				yes
SEGUIMIENTO		REUNION DE SEGUIMIENTO				yes
SITIO WEB		SITIO WEB				yes

Mostrando registros del 1 al 9 de un total de 9 registros

Anterior

1

Siguiente

Estrategia Tecnológica© 2023

Figura 4.3. Vista final de corporativos.

A esta vista también se le agregaron detalles de diseño como la librería de DataTable de JQuery para que la información se muestre mejor con un paginado que se puede modificar el número de filas que se muestran, además de ordenarlos por la columna que se desee y buscarlos a través de un input que ofrece la misma librería.

Corporativo	Categoría	
-APP: BEACON TECH	Propio	<a href="#">Proyectos</a>
-APP: CANDIDATOS	Propio	<a href="#">Proyectos</a>
-APP: EXAMENES LINEA	Propio	<a href="#">Proyectos</a>
-APP: EXAMS TECNICOS	Propio	<a href="#">Proyectos</a>
-APP: HMN	Propio	<a href="#">Proyectos</a>
-APP: IAVISOS	Propio	<a href="#">Proyectos</a>
-APP: PERSONAL	Propio	<a href="#">Proyectos</a>
-APP: PORTAL DOCENTE	Propio	<a href="#">Proyectos</a>
-APP: PORTAL EDUCAT.	Propio	<a href="#">Proyectos</a>
-APP: R.REGULATORIOS	Propio	<a href="#">Proyectos</a>

Acronimo	Proyecto Dsc	Interno
APP ANDROID	APP ANDROID	yes
APP IOS	APP IOS	yes
DISEÑO	DISEÑO DE INTERFACE	yes
MARKETING	-MARKETING	yes
SEGUIMIENTO	REUNION DE SEGUIMIENTO	yes
SITIO WEB	SITIO WEB	yes
<S/PROYECTO>	<S/PROYECTO>	no
ADMINISTRADOR WEB	Administrador para el funcionamiento del app móvil	no
MARKETING	investigación de mercado	no

Figura 4.4. Vista final de corporativos con sus respectivas consultas de proyectos.

También se usaron clases de Bootstrap para dar mejor diseño al sitio, haciéndolo responsivo en resoluciones pequeñas. Haciendo uso también de JavaScript para que Bootstrap haga que ciertos componentes mejoren el aspecto visual del sitio.

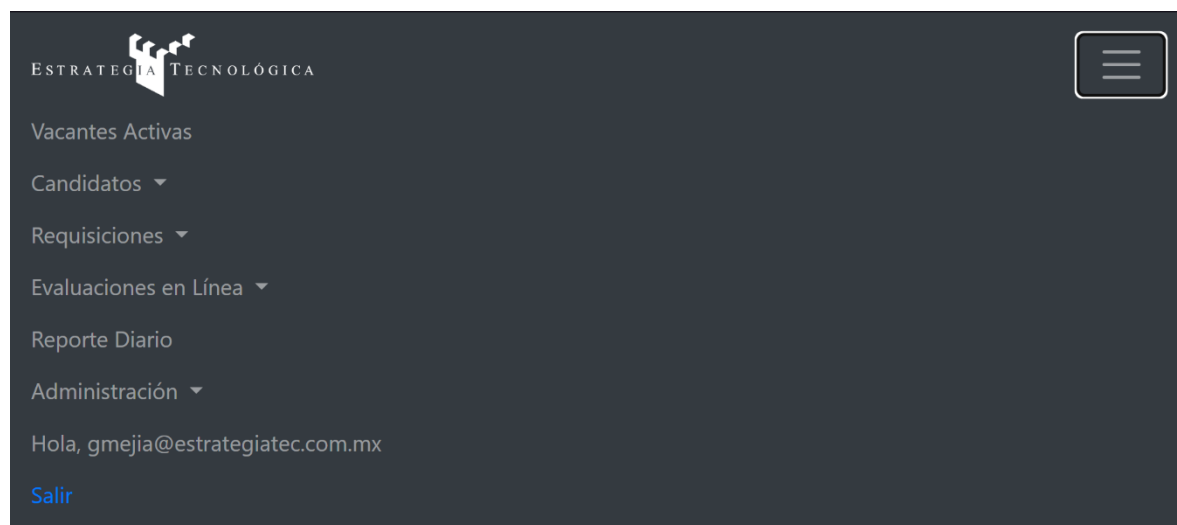


Figura 4.5. Barra de navegación responsiva.



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

Tras finalizar con el desarrollo en esta aplicación web, se determinó que existen varios patrones que pueden ser útiles a la hora de crear una aplicación, ya sea el modelo vista modelo vista, modelo vista presentador, modelo template controlador, modelo vista controlador, etc., todos poseen características que pueden ser útiles dependiendo del tamaño y alcance del proyecto, pero en este caso el modelo MVC fue muy útil, pues permitió separar la lógica de la aplicación de la interfaz, lo que permitió una mayor facilidad para mantener y desarrollar la aplicación.

Es recomendable tener un enfoque para saber a qué se desea especializarse, ya que existen varias áreas, como desarrollo web, móvil, de aplicaciones de escritorio, internet de las cosas, inteligencia artificial, Cloud Computing, realidad aumentada, etc., cada una siendo demandada con muchas oportunidades laborales. No es recomendable ser Multitask, ya que al no haber enfoque, no hay muchas oportunidades de crecimiento y no se explota del todo las capacidades de alguna tecnología.

Asimismo, la experiencia y los conocimientos en estas tecnologías fueron demasiado útiles para la formación académica del estudiante, ya que amplió su panorama para conocer más tecnologías y metodologías, permitiendo obtener mejores oportunidades laborales, pues estas tecnologías como .NET, C#, Entity Framework, etc., son bastantes demandadas ya que permiten crear aplicaciones de cualquier tipo.

## **COMPETENCIAS DESARROLLADAS Y/O APLICADAS**

### **COMPETENCIAS DESARROLLADAS Y/O APLICADAS**

Desarrollar e implementar sistemas de información con técnicas herramientas para la Web en distintas plataformas.

Tomar decisiones con base en los elementos teórico-práctico adquirido que permitan optimizar costos en soluciones informáticas bajo ambiente Web.

Desarrollar aplicaciones web que implementan acceso a datos.

Desarrollar aplicaciones web que implementan el intercambio de información a través de servicios web ya definidos.

Analiza y modela proyectos de sistemas de información aplicando el paradigma orientado a objetos.

Aplica técnicas y herramientas para el diseño del sistema de información.

Aplica los métodos de la ingeniería de software para desarrollar e implementar sistemas de información.

Aplica técnicas y herramientas para la implementación del modelo del sistema de información.

Conoce y utiliza sistemas de base de datos acordes a las necesidades del problema que atiende, considerando la optimización de los recursos de datos en el tratamiento y seguridad de la información.

### **COMPETENCIAS GENÉRICAS**

Capacidad de análisis y síntesis.

Capacidad de organizar y planificar.

Habilidad para buscar y analizar información proveniente de fuentes diversas.

Solución de problemas.

Toma de decisiones.

Trabajo en equipo.

Capacidad de aplicar los conocimientos.

Habilidades de investigación.

Capacidad de generar nuevas ideas.

Compromiso ético.

Comunicación oral y escrita.

Capacidad de aprender.

Habilidad para trabajar en forma autónoma.

Búsqueda del logro.

## FUENTES DE INFORMACIÓN

acens. (2016). Bootstrap, un framework para diseñarportales web: *Definición de Bootstrap*. Recuperado de <https://www.acens.com/wp-content/images/2016/10/bootstrap-framework-acens-wp.pdf>

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\\_Frameworkor k.pdf](file:///C:/Users/berna/Downloads/Introduccion_a_C_Que_es_el_NET_Frameworkor k.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%2520Sol er%252C%2520Juan%2520Pedro.pdf?sequence=1](https://riunet.upv.es/bitstream/handle/10251/49757/MEMORIA_Barba%2520Sol er%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 LATECNOLOGÍA .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.
- Guérin, B. A. (2018). ASP.NET con C# en Visual Studio 2017: *Las vistas Razor y su definición*. (1ra. Ed.). Cornellá de Llobregat, Barcelona, España. Editorial Ediciones ENI.
- 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/diseño-base-datos.pdf?sequence=1&isAllowed=y>
- Jiménez, A. (2008). IMPLEMENTACIÓN DE UNA LIBRERÍA QUE PREPROCESE CÓDIGO ESCRITO EN C++ SIGUIENDO EL ESTÁNDAR ISO/IEC 14882 Y

CREAR UNA HERRAMIENTA QUE HAGA USO DE ELLA: *Definición de librería.*

Recuperado de <https://core.ac.uk/download/pdf/71395648.pdf>

Livias, N. (2018). Conceptos de programación visual, fundamentos del Visual NET, elementos de MS Visual, sentencias de control, procedimientos y matrices de controles, acceso a una base de datos, asistente para aplicaciones, aplicaciones: *Definición del Entorno de desarrollo integrado Visual Studio.* Recuperado de [https://repositorio.une.edu.pe/bitstream/handle/20.500.14039/4345/M025\\_43833721M.pdf?sequence=1&isAllowed=y](https://repositorio.une.edu.pe/bitstream/handle/20.500.14039/4345/M025_43833721M.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/>

Microsoft. (2023). Arquitectura de aplicaciones de ASP.NET Core: *Definición de ASP .NET CORE.* Recuperado de <https://dotnet.microsoft.com/es-es/learn/aspnet/architecture>

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>

Nabor, Y. (2018). Introducción a ASP.NET Core; *¿Qué es .NET Core?* (1ra. Ed.). Puebla, México. Editorial Microsoft Most Valuable Professional.

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>

Rengifo, J. F. (2011). FRAMEWORKS Y HERRAMIENTAS PARA EL DESARROLLO DE APLICACIONES ORIENTADAS A DISPOSITIVOS MÓVILES: *¿En qué consiste un Framework?* Recuperado de <https://core.ac.uk/download/pdf/71396662.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>

Santamaría, J. (2015). SQL SERVER VS MySQL: *Características del SGBD SQL Server*. Recuperado de <https://iessanvicente.com/colaboraciones/sqlserver.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](https://www.uaeh.edu.mx/docencia/P_Presentaciones/prepa3/Presentaciones_Enero_Junio_2014/Definicion%20de%20Internet.pdf)

## ANEXOS

A continuación, se muestra parte de los modelos que se usaron para el acceso de los datos de la base de datos.

```
1 referencia
public sp_ObtenerCandidato_Result Candidato { get; set; }
2 referencias
public sp_ObtenerCandidatoCursos_Result Curso { get; set; }
//public List<sp_ObtenerCandidatoCursos_Result> Cursos { get; set; }
1 referencia
public IEnumerable<SelectListItem> ListCursoTipo { get; set; }
2 referencias
public sp_ObtenerCandidatoEstudios_Result Estudio { get; set; }
//public List<sp_ObtenerCandidatoEstudios_Result> Estudios { get; set; }
1 referencia
public IEnumerable<SelectListItem> ListNivelEscolar { get; set; }
1 referencia
public IEnumerable<SelectListItem> ListDoctoEscolar { get; set; }
2 referencias
public sp_ObtenerCandidatoLaboral_Result Experiencia { get; set; }
1 referencia
public IEnumerable<SelectListItem> ListTipoEmpleo { get; set; }
1 referencia
public IEnumerable<SelectListItem> ListMeses { get; set; }
1 referencia
public IEnumerable<SelectListItem> ListMeses2 { get; set; }
```

Figura 5.0. Modelo de candidatos.

```
0 referencias
public BuscarPersonalModel()
{
    PersonalList = new List<sp_PersonalActual_Result>();
    EstatusId = 0;
    Desde = DateTime.Now.AddDays(-1);
    Hasta = DateTime.Now;
}
2 referencias
public int EstatusId { get; set; }
2 referencias
public DateTime Desde { get; set; }
2 referencias
public DateTime Hasta { get; set; }
1 referencia
public string Nombre { get; set; }
1 referencia
public IEnumerable<SelectListItem> EstatusList { get; set; }
2 referencias
public List<sp_PersonalActual_Result> PersonalList { get; set; }
```

Figura 5.1. Modelo para consultar personal.



```

4 referencias
public sp_ObtenerCandidato_Result Candidato { get; set; }
3 referencias
public List<sp_ObtenerCandidatoCursos_Result> Cursos { get; set; }
1 referencia
public sp_ObtenerCandidatoCursos_Result Curso { get; set; }
0 referencias
public IEnumerable<SelectListItem> ListCursoTipo { get; set; }
3 referencias
public List<sp_ObtenerCandidatoEstudios_Result> Estudios { get; set; }
0 referencias
public sp_ObtenerCandidatoEstudios_Result Estudio { get; set; }
3 referencias
public List<sp_ObtenerCandidatoLaboral_Result> Laboral { get; set; }
3 referencias
public List<sp_ObtenerExperienciaTecnica_Result> Idiomas { get; set; }
0 referencias
public List<sp_ObtenerExperienciaTecnica_Result> IdiomasList { get; set; }
3 referencias
public List<sp_ObtenerExperienciaTecnica_Result> Tecnologia { get; set; }
3 referencias
public List<sp_PlazasxCandidato_Result> Plazas { get; set; }
3 referencias
public List<sp_ReqAdxCandidato_Result> PlazasAdicionales { get; set; }
3 referencias
public List<sp_SeguimientoCandidato_Result> Seguimiento { get; set; }
3 referencias
public IEnumerable<SelectListItem> ListCategorias { get; set; }
2 referencias

```

Figura 5.2. Modelo para candidatos.

```

public class Usuario
{
    0 referencias
    public string Nombre { get; set; }
    1 referencia
    public Nullable<int> CandidatoId { get; set; }
    0 referencias
    public Nullable<int> PersonalId { get; set; }
    1 referencia
    public string UserId { get; set; }
    1 referencia
    public Nullable<DateTime> FechaIngreso { get; set; }
    1 referencia
    public string CURP { get; set; }
    1 referencia
    public string RFC { get; set; }
    1 referencia
    public string Email { get; set; }
    1 referencia
    public Nullable<int> iPerfil { get; set; }
    0 referencias
    public Nullable<System.DateTime> FechaSalida { get; set; }
    0 referencias
    public Nullable<int> iEmpresa { get; set; }
    0 referencias
    public Nullable<int> iNominaId { get; set; }
    1 referencia
    public int iAsignadoA { get; set; }
}

```

Figura 5.3. Modelo para los usuarios.