**Universidad Hispanoamericana**

**PROGRAMACION 5**

**Proyecto de Investigación**

**Aplicación del** **patrón de desarrollo MVC en React, VUE o Flutter mediante una aplicación web/móvil**

**Estudiantes:**

**Fabian Carmona Solís**

**Danny Castro Barboza**

**Pablo Sánchez Ocampo**

**Tutor: JAIRO ALBERTO ZUÑIGA GOMEZ**

**Diciembre, 2023.**



# Resumen

A continuación, en la investigación realizada se podrán observar temas acerca de MVC (Modelo-Vista-Controlador), frameworks como, React, Vue y Flutter, cada uno de ellos con sus descripciones respectivas tanto como su integración en el MVC, Además se investigó sobre los antecedentes de esto temas y se examinaron casos de aplicaciones web/móviles que hayan sido desarrolladas ya sea con React, Vue o Flutter y hayan implementado el patrón de MVC con éxito. Todo lo anterior fue realizado mediante un estudio de la arquitectura de los frameworks que han sido mencionados anteriormente para así de esta manera lograr comprender como se realiza la integración de estos frameworks con el MVC. Para ello indagamos tutoriales, artículos y ejemplos de código que hayan sido proporcionados por la comunidad y desarrolladores de cada framework, de esta manera analizamos las aplicaciones web/móviles que hayan tenido éxito implementado el patrón MVC, que fueron desarrolladas ya sea con React, Vue o Flutter. Por ello que también estudiamos un poco sobre la estructura, arquitectura y ventajas que estos frameworks pueden tener en proyectos específicos al tener aplicado el patrón MVC, de igual manera también se describieron las similitudes que pueden llegar a tener estos frameworks al igual que las diferencias que estos pueden tener en temas tales como: arquitectura, compilación o el uso de recursos. Además, se creó una aplicación simple utilizando React e implementando el patrón MVC para así de esta manera lograr evaluar la eficiencia del desarrollo, el rendimiento y la legibilidad del código generado en la aplicación creada.

Tabla de contenido

[Resumen 2](#_Toc154151216)

[Objetivo general 4](#_Toc154151217)

[Objetivos específicos 4](#_Toc154151218)

[Justificación 5](#_Toc154151219)

[Antecedentes 6](#_Toc154151220)

[Innovación 8](#_Toc154151221)

[Desarrollo 9](#_Toc154151222)

[Conclusiones 29](#_Toc154151223)

[Recomendaciones 31](#_Toc154151224)

[Bibliografía 33](#_Toc154151225)

# Objetivo general

Explorar la aplicación del patrón de desarrollo Modelo-Vista-Controlador (MVC) en frameworks como React, Vue o Flutter para el desarrollo de aplicaciones web/móvil.

# Objetivos específicos

|  |  |
| --- | --- |
| Objetivos específicos | Actividades |
| Analizar la integración del patrón Modelo-Vista-Controlador (MVC) en los frameworks React, Vue y Flutter. | Estudiar la arquitectura de React, Vue y Flutter para comprender cómo se integra el patrón MVC o su equivalente en cada uno. Revisar la documentación oficial, tutoriales, artículos y ejemplos de código proporcionados por la comunidad y los desarrolladores de cada framework. |
| Identificar la contribución a la mejora en eficiencia del desarrollo, el rendimiento de la aplicación y la legibilidad del código. | Crear aplicación simple utilizando React implementando el patrón MVC. Evaluar la eficiencia del desarrollo, el rendimiento de la aplicación y la legibilidad del código generado. |
| Examinar casos de aplicaciones web/móvil desarrolladas con React, Vue y Flutter que hayan implementado el patrón MVC de manera exitosa. | Identificar y analizar aplicaciones web/móvil desarrolladas con React, Vue y Flutter que hayan implementado con éxito el patrón MVC. Estudiar la estructura, arquitectura y ventajas obtenidas al aplicar este patrón en proyectos específicos en cada framework. |

# Justificación

La principal razón de la investigación a realizar es descubrir cómo aplicar el patrón de desarrollo MVC en React, VUE o Flutter mediante una aplicación web/móvil. Para ello una detallada investigación será realizada, en el transcurso de dicha investigación nos daremos a la tarea de indagar sobre el contexto e historia de la tecnología relacionada con este tema, para iniciar con ello primero se investigará sobre el MVC (Modelo-Vista-Controlador) para que así de esta manera logremos dejar claro que es y para qué sirve el MVC, seguidamente, se investigaran los conceptos de React, VUE, Flutter. De este modo tendremos los conceptos principales de la investigación definidos en el área del desarrollo, de tal manera que luego haya un enfoque en el tema principal de la investigación el cual es como aplicar el Modelo-Vista-Controlador en React, VUE o Flutter mientras se averigua como realizar esta integración, también estará investigando un poco sobre el contexto histórico relacionado con el tema como fue mencionado anteriormente. Además también se investigara sobre casos de aplicaciones web/móviles desarrolladas ya sea con React, VUE o Flutter que hayan implementado el patrón de MVC de una manera exitosa, el objetivo al concluir con esta investigación es que los demás estudiantes de las carreras relacionadas con estos temas tengan claro cómo será solventada la solución del tema de esta investigación y también que adquieran los conocimientos sobre los conceptos que estarán en dicha investigación y de igual manera puedan aprender un poco sobre la historia de la tecnología relacionada a los temas de React, VUE, Flutter.

# Antecedentes

Elegir la tecnología adecuada para la creación de un proyecto determinado es un factor que no se puede dejar de lado por el simple hecho que una errónea selección puede retrasar la productividad y desarrollo de un sistema. Muchas de las aplicaciones web hoy en día es posible que encontraran dicho dilema, pero con tiempo y análisis se logra sacar a flote los proyectos que se lleguen a plantear, por eso es importante aprender de los fallos de elaboraciones pasadas y actualizar o mejorar la producción.

El artículo "Desarrollo web moderno usando ReactJS" de Sanchit Aggarwal (2018) resalta la aplicación de ReactJS en la construcción de interfaces de usuario interactivas y reutilizables. ReactJS, como parte esencial del desarrollo web, se presenta como una herramienta para construir componentes reutilizables y dinámicos destinados a la interfaz de usuario. Destaca como la capa de vista en el patrón MVC, permitiendo la creación de aplicaciones web complejas capaces de modificar sus datos sin necesidad de recargar la página. Un aspecto significativo es su empleo del DOM virtual para mejorar el rendimiento, lo cual implica comparar y actualizar únicamente los nodos necesarios en el DOM del navegador, evitando la manipulación directa de todo el árbol del DOM. La implementación de un flujo de datos unidireccional asegura la inmutabilidad de los componentes y simplifica el enlace de datos, ofreciendo una programación más intuitiva mediante el uso de JSX para la escritura de marcado y la gestión de eventos. ReactJS presenta un ciclo de vida de componentes con etapas de montaje, actualización y desmontaje, permitiendo sincronizar el DOM con los cambios de estado y propiedades. A pesar de sus ventajas, ReactJS tiene limitaciones notables, como su enfoque en la vista en lugar del controlador o modelo, lo que podría requerir el uso de herramientas adicionales para proyectos más completos.

Otro trabajo de investigación, liderado por Viviana Elizabeth Tacuri Auquilla y Oscar Vicente Escobar Guachambala (2021), se enfocó en el desarrollo de una aplicación web mediante React y una aplicación móvil con Flutter. Este proyecto formó parte de un trabajo de integración curricular para optar al grado académico de Ingeniero de Software. Durante el estudio, se exploró la relevancia de React al ofrecer una arquitectura basada en componentes reutilizables, proporcionando una gestión eficiente y escalable de datos en el contexto del Modelo-Vista-Controlador (MVC). Asimismo, se resaltó la importancia de Flutter para crear interfaces móviles altamente personalizables, garantizando un rendimiento óptimo dentro del paradigma MVC. La metodología de desarrollo aplicada se basó en Scrum, dividiendo el trabajo en 4 sprints y logrando un alto porcentaje (98.64%) de pruebas exitosas. Además, se identificaron desafíos en la difusión de información turística por parte del Gobierno Autónomo Descentralizado Parroquial Huambaló. La evaluación de la usabilidad de las aplicaciones mediante el cuestionario USE demostró calificaciones favorables (promedio de 6.63 sobre 7 para la aplicación web y 6.51 para la aplicación móvil), indicando la capacidad efectiva de las aplicaciones para ser comprendidas, aprendidas y utilizadas por los usuarios.

Continuando con la investigación de Gerson Wilfredo Zegarra Obando en (2018), titulada "Aplicación Web basada en MVC para Mejorar la Gestión de Pagos en el Instituto de Educación Superior Pedagógico Público Chimbote, Ancash", se centra en abordar las dificultades existentes en la gestión de pagos de los alumnos. El Instituto enfrentaba problemas de largos tiempos de espera, proceso manual de recibos y reportes, y limitaciones en la capacidad de atención en la tesorería, lo que generaba demoras y complicaciones en los pagos. La solución propuesta, una aplicación web basada en el modelo MVC, logró reducir significativamente el tiempo de elaboración de recibos y pagos, así como el costo humano en la generación de reportes. Esto resultó en un aumento notable en la satisfacción del personal administrativo. El proyecto demostró su viabilidad económica y técnica en un período de desarrollo razonable, ofreciendo una solución efectiva para optimizar la gestión de pagos en el Instituto.

# Innovación

Este proyecto de investigación aporta una perspectiva sobre la integración del patrón Modelo-Vista-Controlador (MVC) en el entorno de desarrollo de React. Se destaca la exploración profunda de cómo la arquitectura MVC se adapta y se implementa dentro del ecosistema de React, destacando las mejores prácticas, desafíos y beneficios al utilizar esta estructura en la construcción de aplicaciones web. Además, se proporcionan ejemplos concretos de aplicaciones funcionales que demuestran la aplicación exitosa del patrón MVC en proyectos específicos desarrollados con React, enfocándose en la separación clara de la lógica de negocio, la interfaz de usuario y el controlador, lo cual contribuye a una mejor organización del código y a una mayor escalabilidad de las aplicaciones.

# Desarrollo

|  |  |  |
| --- | --- | --- |
| Fecha | | Actividades |
| Mes | **Día** |
| Diciembre | 1 | Búsqueda de información. |
| Diciembre | 5 | Explicación de conceptos (MVC, React, Vue, Flutter). |
| Diciembre | 7 | Desarrollar explicación sobre ventajas y desventajas entre React, Vue y Flutter. |
| Diciembre | 10 | Explicar diferentes características entre React, Vue y Flutter. |
| Diciembre | 14 | Analizar diversos casos de implementación de React (se decide trabajar con React). |
| Diciembre | 16 | Redactar una de las formas de la cual se puede implementar React en MVC. |
| Diciembre | 19-21 | Creación de aplicación funcional donde se implemente React al patrón MVC. |

**MVC (Modelo-Vista-Controlador)**

Es un patrón de diseño arquitectónico de software que sirve para clasificar la información, la lógica del sistema y la interfaz que se le presenta al usuario. Este tipo de arquitectura es muy utilizado para temas de desarrollo web, sin embargo, al tener varios lenguajes para crear un sitio es muy fácil generar confusión entre cada componente si estos no son separados de una forma adecuada.

Componentes del MVC:

**Modelo**

Es el que se encarga de manipular gestionar y actualizar datos. Al utilizar las bases de datos en este componente este mismo se encargará de realizar las consultas como por ejemplo las búsquedas, filtros y actualizaciones.

**Vista**

Su objetivo es mostrarle al usuario final las pantallas, ventanas, páginas y formularios. Este componente es el encargado del frontend, cuyo caso podría ser la interfaz del usuario de una aplicación de escritorio o la visualización de una página web.

**Controlador**

Por medio de este componente el modelo y la vista se comunican, se encarga de gestionar las instrucciones recibidas de esta manera las atiende y las procesa. Solicita los datos necesarios al modelo, manipulándolos para obtener los resultados y luego entregarlos a la vista para que estos puedan ser mostrados en pantalla.

**React**

Es una de las librerías más populares de JavaScript para el desarrollo de aplicaciones móviles y web. Fue creada por Facebook y contiene una colección de fragmentos de código JavaScript reutilizables utilizados para crear interfaces de usuario. Además, con React es posible desarrollar páginas web de una manera gratuita y sencilla gracias a sus componentes reutilizables.

**Ventajas:**

* Vemos como el desarrollo es ordenado y eficiente.
* Reutiliza componentes.
* Mejora el proceso del DOM. (Document Object Model)
* Es accesible en varios motores de búsqueda.

**Desventajas:**

* Tiene la necesidad de bibliotecas adicionales para la gestión de estado y enrutamiento.
* Es limitado al manejo de vista en el patrón MVC.
* Uso de plantillas en línea y JSK puede ser complejo.
* Contiene errores en tiempo de compilación.

**VUE**

Es un framework open source de JavaScript, el cual nos permite construir interfaces de usuarios de una forma muy sencilla. Una de las características más importantes de Vue es el trabajo con componentes. Un componente Vue es un elemento el cual se encapsula código reutilizable.

**Ventajas:**

* Curva de Aprendizaje Suave
* Flexibilidad y Adaptable
* Integración Gradual
* Versatilidad en Ecosistema
* Rendimiento Eficiente

**Desventajas:**

* Menor Comunidad en Comparación con React.
* Menos Patrocinio Empresarial que React.
* Menos Herramientas y Extensiones que React.
* Estándares Menos Definidos.
* Madurez Relativa.

**Flutter**

Es un framework creado por Google que permite el desarrollo de un proyecto de programación. Permite crear una aplicación nativa con una sola base de código, puede usar un lenguaje de programación y una base de código para crear dos aplicaciones diferentes.

**Ventajas:**

* Recarga en caliente para desarrollo rápido.
* Funciona en múltiples plataformas.
* Uso de Dart con curva de aprendizaje baja.
* Gran catálogo de widgets.
* Fácil configuración del entorno de desarrollo.

**Desventajas:**

* Bibliotecas limitadas en comparación con otras tecnologías.
* Jerarquía del árbol de widgets puede volverse compleja.
* Tamaño de las aplicaciones Flutter tiende a ser más grande.

React, Vue y Flutter comparten algunas similitudes clave, pero difieren en términos de lenguaje, ecosistema, arquitectura y otros aspectos, lo que los hace adecuados para diferentes contextos y preferencias de desarrollo.

**Similitudes:**

* Enfoque de Componentes: React, Vue y Flutter comparten un enfoque centrado en componentes. Los componentes son unidades independientes y reutilizables que facilitan la construcción modular de aplicaciones.
* Curva de Aprendizaje Baja: Los tres frameworks se esfuerzan por proporcionar una curva de aprendizaje accesible para los desarrolladores. Vue, al igual que React y Flutter, adopta un enfoque intuitivo para facilitar la adopción rápida.
* Recarga en Caliente (Hot Reload): React, Vue y Flutter permiten la recarga en caliente, una característica valiosa para los desarrolladores que les permite ver los cambios inmediatamente durante el desarrollo sin tener que reiniciar la aplicación.

**Diferencias**

**Lenguaje de Programación:**

* React: Utiliza JavaScript y JSX para definir la interfaz de usuario.
* Vue: Emplea JavaScript con opciones de sintaxis más flexibles y la capacidad de usar JSX opcionalmente.
* Flutter: Utiliza Dart como su lenguaje principal.

**Ecosistema:**

* React: Es ampliamente utilizado en el desarrollo web, especialmente en aplicaciones de una sola página (SPA).
* Vue: Se centra en el desarrollo web y es conocido por su flexibilidad y adaptabilidad.
* Flutter: Se orienta principalmente al desarrollo de aplicaciones móviles, pero también se puede utilizar para aplicaciones web y de escritorio.

**Arquitectura:**

* React: Se presenta como una biblioteca para construir interfaces de usuario, dejando decisiones arquitectónicas a los desarrolladores.
* Vue: Proporciona una estructura más definida en comparación con React y sigue una arquitectura basada en componentes.
* Flutter: Es un framework completo con una arquitectura basada en widgets que proporciona una estructura coherente.

**Compilación:**

* React: Las aplicaciones React se ejecutan en navegadores y se interpretan en tiempo de ejecución.
* Vue: Puede integrarse en proyectos existentes y también puede compilarse para mejorar el rendimiento.
* Flutter: Compila a código nativo para cada plataforma, ofreciendo un rendimiento similar al de las aplicaciones nativas.

**Uso de Recursos:**

* React: Puede requerir bibliotecas adicionales para ciertas funcionalidades, como la gestión de estado y el enrutamiento.
* Vue: Proporciona un conjunto de herramientas integradas, pero también permite la integración de bibliotecas externas según sea necesario.
* Flutter: Ofrece un amplio catálogo de widgets integrados y no depende tanto de bibliotecas externas para funcionalidades básicas.

**Tamaño de Aplicación:**

* React: El tamaño de las aplicaciones React puede variar según las bibliotecas utilizadas, pero en general, se esfuerza por mantener un tamaño razonable.
* Vue: Tiende a tener un tamaño de aplicación moderado, similar a React.
* Flutter: Las aplicaciones Flutter pueden ser un poco más grandes debido a la inclusión del motor, pero ofrece un rendimiento nativo.

**Implantación de React en el patrón MVC por medio de ASP.Net Core**

Se llega a la decisión de implementar de React en el patrón MVC por medio de ASP.Net Core por medio de Visual Studio, para esto se requiere de los siguientes pasos:

1. Instalar las siguientes cargas de trabajo que va a utilizar Visual Studio: **Desarrollo de ASP.NET y web, Desarrollo en Node.js**.
2. Instalar el Node.js desde la página de internet y verificar que el npm (Node Package Manager) se instale también.
3. Al completar toda la instalación necesaria, abrir el Visual Studio, darle a crear nuevo proyecto y seleccionar ASP.NET Core con React.
4. Seleccionar la configuración deseada tanto nombre del proyecto, ruta de guardado, nombre de la solución del proyecto, versión de Framework (en este caso se utiliza el .NET 7.0), tipo de autenticación (en este caso se selecciona ninguna), configurarlo para HTTPS, y el no usar declaraciones de alto nivel.
5. Por último, esperar la configuración automática de la creación del proyecto donde se encontrará el React listo para utilizar.
6. La aplicación creada viene por defecto con un hola mundo.

En caso de seleccionar el tipo de autenticación como Individual Accounts se creará automáticamente la carpeta de Models y Data donde contiene el dbContext de la aplicación, esto se deberá configurar en relación con la necesidad de cómo se quiere la aplicación y su sentido es que se requiere de un login como verificación de quien entra y por esa razón se crea automáticamente el modelo el cual se necesita para gestionar los datos que transitan por ese login.

**Diferencias en el tipo de creación antes mencionado es el siguiente:**

1. Proyecto creado sin autenticación de Individual Accounts:

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

1. Proyecto creado con autenticación de Individual Accounts:

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

**Explicación del código creado**

**Texto

Descripción generada automáticamente**

Este es el archivo “Program.cs” es donde configuramos la aplicación web, registrando servicios en la cual podemos permitir el funcionamiento del patrón MVC donde se manejan solicitudes de HTTP, también verifica si la aplicación esta en modo desarrollo y en caso de que no lo este, se ejecuta. Dentro de su ejecución fuerza a los navegadores a utilizar HTTPS. Redirecciona las solicitudes HTTP al equivalente de HTTPS automáticamente. Permite el uso de archivos estáticos, se configura las rutas URL para que la aplicación determine como responder, establece una ruta por defecto en la cual inicia y la ruta de reversa redirecciona al índex. Donde por ultima se ejecuta la aplicación.

Captura de pantalla de computadora

Descripción generada automáticamente

La clase “WeatherForecast” es como un pequeño modelo el cual nos sirve para encapsular información relacionada al pronostico del tiempo donde va a ser mostrada en la aplicación en su momento de ejecución.

Texto

Descripción generada automáticamente

El controlador “WeatherForecastController” es de un API donde se utiliza solicitudes de GET con HTTP para obtener pronósticos de tiempo, se declara controlador de API lo cual facilita el su uso y la respuesta ante errores de manera automática e inferir parámetros en solicitudes. Se declara un array el cual sirve para determinar el tipo de clima del día. Se encuentra el método que nos va a devolver la cantidad de objetos especificada la cual va de 1 a 5, dentro de estos objetos tenemos la fecha actual que va a incrementar con la suma con relación al índice del bucle. Una variable random que genera la temperatura en Celsius entre -20 y 55 donde serán asignados a cada pronostico, y por ultimo el resumen del clima donde se asigna un valor aleatorio del array anteriormente explicado que menciona los distintos valores pero que simplemente serán asignados de manera aleatoria sin filtro alguno por lo que la temperatura y el resumen muy difícilmente van a coincidir. Por ultimo se transfiera la información en un array el cual será obtenido por medio de una solicitud GET la cual va a devolver cada uno de los datos.

Texto

Descripción generada automáticamente

En el archivo “aspnetcore-react.js” se encuentra el JavaScript de configuración al archivo “.env.development.local” el cual es un archivo de configuración utilizado en entornos de desarrollo para establecer variables de entorno locales específicas, facilitando la personalización del entorno de desarrollo sin afectar la configuración en otros entornos como pruebas, producción, etc. Regresando con el archivo “aspnetcore-react.js” dentro de toda la configuración también sirve para el HTTPS utilizando el certificado de desarrollo de ASP.NET Core en el proxy de desarrollo de webpack.

Texto

Descripción generada automáticamente

En el archivo “aspnetcore-https.js” posee JavaScript que está diseñado para configurar HTTPS para una aplicación utilizando el certificado HTTPS de ASP.NET Core. La función principal es generar y exportar un certificado HTTPS en formato “.pem” usando el comando “dotnet dev-certs” https de ASP.NET Core, en caso de no existir se ejecuta para asegurar su correcto funcionamiento. Contando el anterior archivo mencionado, se permite la activación de una comunicación segura o lo que se conoce como HTTPS dentro de una aplicación de ASP:NET Core.

Texto

Descripción generada automáticamente

Dentro de la carpeta de componentes vamos a encontrar cada elemento de las vistas, es común encontrar un archivo en HTML donde se muestre lo elementos a utilizar por cada pagina creada, pero en React se divide en componentes de JavaScript los cuales podrán poseer el mismo código pero que serán llamados en el momento requerido, lo cual permite una reutilización de código y subdivisión para trabajar de manera aislada cada componente.

Captura de pantalla de computadora

Descripción generada automáticamente

El archivo “App.js” define el componente principal de la aplicación React, configura las rutas de la aplicación utilizando react-router-dom, y utiliza un componente Layout como contenedor principal para el contenido de la aplicación. Se importan las dependencias como los componentes de react, las rutas de la aplicación que proporciona el “react-router-dom”, “AppRutes” donde se encuentran las rutas definidas de la aplicación, el componente de diseño llamado “Layout” y el css el cual da la personalización a la aplicación.

Se define un componente de clase llamado App, el cual es exportado por defecto para ser utilizado en otros módulos. Se le asigna el nombre de la app el cual va a ser el mismo al que se le puso en la creación del proyecto, en este caso “InvestigacionReact” y después el método render al cual utiliza el “Layout” como contenedor principal, en donde se encuentra “Routes” que son las rutas en las cuales se van a asignar a cada dirección enviando sus respectivas propiedades.

Captura de pantalla de computadora

Descripción generada automáticamente

El archivo “App.test.js” el cual este contenido por “App.js” como el nombre lo menciona es utilizado para pruebas y verificar su correcto funcionamiento por medio del “MemoryRouter” y un tiempo de espera, de esta forma se asesora que el componente principal se monte correctamente sin inconvenientes. “MemoryRouter” es un componente de la biblioteca “react-router-dom” que permite la simulación de enrutamiento en memoria para probar los componentes de una estructura sin afectar la barra de direcciones del navegador real.

Texto

Descripción generada automáticamente

El archivo “AppRoutes.js” se encuentra el direccionamiento con el respectivo componente donde se importan los componentes necesarios y se les asigna como una URL la cual sirve para redireccionar al contenido que ocupamos. Gracias a React esta forma de gestionar las rutas se vuelva muy ordenado, además cada ruta será guardada en un Array el cual se exporta para ser utilizado en otros archivos de la aplicación.

Texto

Descripción generada automáticamente

El archivo “custom.css” permite la personalización de la aplicación, es donde especificamos los valores que cambiaran la apariencia de la aplicación.

Texto

Descripción generada automáticamente

En el archivo “index.js” se llega a importar el Bootstrap, las bibliotecas de React, función “createRoot” de React DOM se utiliza para crear un "root" en el DOM para la aplicación, importa el enrutador “BrowserRouter” de React Router, que será utilizado para la gestión del enrutamiento en la aplicación, el componente principal de la aplicación desde el archivo “App.js” y por último importa funciones relacionadas con el registro del “service worker” y la recolección de métricas web.

El funcionamiento empieza por la recolección de la URL relativa de los componentes con “baseUrl”, el “rootElement” que es el contenedor principal donde se renderiza la aplicación y el “root” que utiliza la función anterior para la creación de este. Ya dentro del “root” se renderiza el componente “App”. Para finalizar se elimina o des registra el “service worker” e inicia la recolección de datos por medio de la función “reportWebVitals” la cual sirve para el registro de métricas de rendimiento y utilizarlas para análisis si fuera el caso.

Captura de pantalla de computadora

Descripción generada automáticamente

En el archivo “reportWebVitals.js” es donde se recupera los datos para medir y reportar métricas de rendimiento web específicas. Estas métricas pueden ser utilizadas para monitorear y mejorar el rendimiento de la aplicación web, proporcionando información valiosa sobre su rendimiento en términos de velocidad y experiencia del usuario.

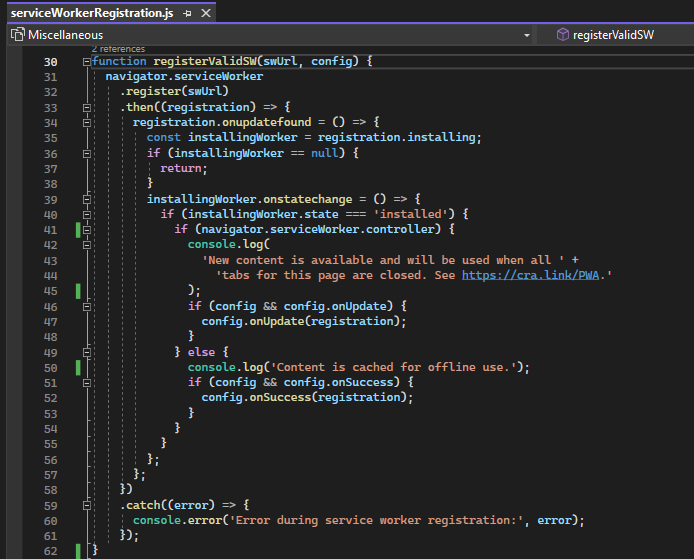
Texto

Descripción generada automáticamente

En el archivo “service-worker.js” se define estrategias de caché, enrutamiento de navegación, manejo de activos precargados y otras funcionalidades para mejorar el rendimiento y la experiencia del usuario en la aplicación web utilizando Service Workers y la librería Workbox. Dentro de este archivo se encuentra comentado el código “/\* eslint-disable no-restricted-globals \*/” y es una directiva especial para deshabilitar una regla específica del linter ESLint dentro de ese archivo o bloque de código en particular. ESLint es una herramienta de análisis estático de código para JavaScript, ayuda a mantener la calidad del código, asegurando que siga prácticas recomendadas, estilos de codificación consistentes y evitando posibles errores comunes.

Texto

Descripción generada automáticamente



Texto

Descripción generada automáticamente

El archivo “serviceWorkerRegistration.js” contiene funciones para registrar, verificar la validez, actualizar y eliminar Service Workers en la aplicación web. Estas funciones permiten manejar la lógica asociada con los Service Workers para ofrecer capacidades de cacheo y funcionalidades offline en la aplicación. Un Service Worker es un tipo especial de script de JavaScript que se ejecuta en segundo plano en el navegador web y funciona como un proxy entre la aplicación web y la red.

Texto

Descripción generada automáticamente

El archivo “setupProxy.js” configura un middleware de proxy en una aplicación Node.js para redirigir las solicitudes que coincidan con la ruta “/weatherforecast” a un servidor remoto, utilizando las opciones de configuración especificadas. Esto es útil, por ejemplo, cuando se trabaja con aplicaciones que utilizan diferentes servicios o servidores para manejar ciertas rutas específicas. Middleware se un software intermediario entre las solicitudes del cliente con la logia de una aplicación, por lo que en este archivo se crea ese intermediario.

# Conclusiones

La investigación sobre la aplicación del patrón Modelo-Vista-Controlador (MVC) en los frameworks React, Vue y Flutter para el desarrollo de aplicaciones web/móvil ha proporcionado una comprensión profunda de cómo estas tecnologías abordan la arquitectura de software y la organización del código. Los objetivos específicos trazados han sido abordados de manera sistemática, analizando la integración del patrón MVC en cada framework, evaluando la eficiencia del desarrollo y el rendimiento de la aplicación, e identificando casos exitosos de implementación del patrón en proyectos específicos.

La relevancia de esta investigación radica en la necesidad constante de optimizar el desarrollo de aplicaciones, seleccionando las tecnologías más adecuadas para cada proyecto. Se ha destacado la importancia de entender cómo el patrón MVC, fundamental en el desarrollo web, se adapta y se implementa en React, Vue y Flutter.

En cuanto a React, se ha observado su enfoque en la construcción de interfaces de usuario reactivas y reutilizables. La ventaja de la reutilización de componentes y la mejora en el rendimiento del DOM mediante el uso de un DOM virtual son aspectos destacados. Sin embargo, se señalan desventajas como la necesidad de bibliotecas adicionales para la gestión de estado y enrutamiento.

En el caso de Vue, se destaca su curva de aprendizaje suave, flexibilidad y adaptabilidad. Aunque tiene una menor comunidad en comparación con React, se presenta como una opción versátil y eficiente en rendimiento. Las desventajas incluyen una menor cantidad de herramientas y extensiones, así como estándares menos definidos en comparación con React.

Flutter, por otro lado, se presenta como un framework completo que permite el desarrollo de aplicaciones nativas con una sola base de código. Se ha subrayado su capacidad para el desarrollo rápido mediante la recarga en caliente y su funcionamiento en múltiples plataformas. Sin embargo, las bibliotecas limitadas y el tamaño potencialmente más grande de las aplicaciones Flutter se mencionan como desventajas.

Las similitudes y diferencias entre React, Vue y Flutter han sido detalladas, abordando aspectos como el enfoque de componentes, la curva de aprendizaje baja y la recarga en caliente como características compartidas. Las diferencias en lenguaje de programación, ecosistema, arquitectura, compilación, uso de recursos y tamaño de aplicación han sido analizadas para proporcionar una visión completa de cada framework.

En los antecedentes, se observa cómo la selección adecuada de tecnologías puede impactar significativamente en el éxito de un proyecto, y se presentan ejemplos de investigaciones anteriores que exploran el desarrollo con tecnologías específicas.

# Recomendaciones

Antes de seleccionar un framework, es crucial entender las necesidades y requisitos específicos del proyecto. La elección entre React, Vue y Flutter debe basarse en consideraciones como el tipo de aplicación (web, móvil o multiplataforma), las preferencias del equipo de desarrollo y la complejidad del proyecto.

Dada la importancia del patrón MVC en el desarrollo web, se recomienda que los desarrolladores adquieran un sólido conocimiento del patrón y su implementación en cada framework. Comprender cómo se manejan el modelo, la vista y el controlador en el contexto de React, Vue y Flutter es esencial para optimizar la arquitectura de las aplicaciones.

Al desarrollar con React, se debe prestar especial atención a la gestión del estado y al enrutamiento, ya que estas áreas pueden requerir bibliotecas adicionales.

Al elegir un framework, la fortaleza de la comunidad y el respaldo de la empresa son factores críticos. React cuenta con una comunidad amplia y el respaldo de Facebook, Vue se destaca por su flexibilidad y una comunidad creciente, mientras que Flutter, respaldado por Google, ofrece soluciones para el desarrollo multiplataforma.

La curva de aprendizaje suave de Vue es una ventaja para los desarrolladores, especialmente aquellos que están empezando. Sin embargo, es fundamental tener en cuenta que la elección de un framework también puede depender de la familiaridad del equipo con ciertas tecnologías.

Dada la naturaleza cambiante del desarrollo de software, se recomienda mantenerse actualizado con las nuevas versiones y características de React, Vue y Flutter. Las actualizaciones pueden introducir mejoras, correcciones de errores y nuevas herramientas que beneficiarán al desarrollo de aplicaciones.

La realización de pruebas es esencial para garantizar la robustez y la calidad del código. Evaluar la eficiencia del desarrollo, el rendimiento de la aplicación y la legibilidad del código mediante pruebas sistemáticas contribuye a la mejora continua y a la entrega de aplicaciones de alta calidad.

# Bibliografía

Aggarwal, S. (2018). Modern Web-Development using ReactJS. *International Journal of Recent Research Aspects*, *5*(1), 133–137. <http://ijrra.net/Vol5issue1/IJRRA-05-01-27.pdf>

Cristancho, F. (2022). ¿Qué es Flutter y para qué sirve? - Talently. *Talently Blog*. <https://talently.tech/blog/que-es-flutter/>

Deyimar, A. (2020). *Qué es React: definición, características y funcionamiento*. Tutoriales Hostinger. <https://www.hostinger.es/tutoriales/que-es-react>

García, E. (2019). *¿Qué es Vue.JS?* Codigofacilito.com. <https://codigofacilito.com/articulos/que-es-vue>

García, M. (2017). *MVC (Modelo-Vista-Controlador): ¿qué es y para qué sirve?* Codingornot.com. <https://codingornot.com/mvc-modelo-vista-controlador-que-es-y-para-que-sirve>

Obando, Z., & Wilfredo, G. (2018). *Aplicación web basada en MVC para mejorar la gestión de pagos de los alumnos en el Instituto de Educación Superior Pedagógico Público Chimbote, Ancash* [Universidad César Vallejo]. <https://repositorio.ucv.edu.pe/handle/20.500.12692/29064>

Tacuri Auquilla, V. E., & Escobar Guachambala, O. V. (2021). *Desarrollo de una aplicación web con React para la gestión de información turística y una aplicación móvil con Flutter para fomentar el turismo en la parroquia Huambaló* [Escuela Superior Politécnica de Chimborazo]. <http://dspace.espoch.edu.ec/handle/123456789/19274>