

Código Asignatura	13997
Carrera	Ingeniería Informática
Asignatura	Taller de Multimedios
Unidad	1
Nombre	Eduardo Andrés Díaz Ramírez
Fecha de Entrega	04/04/2023

Descripción de la solución propuesta	1
Arquitectura de software	2
Tecnología Web	3
Bibliografía	4

Link de repositorio: <https://github.com/Eduardo9870/serviApp.git>

Link de acceso publico: <https://taller2023ediaz.000webhostapp.com/serviApp/>

Link de video: https://taller2023ediaz.000webhostapp.com/serviApp/Video_Evaluaci%C3%B3n_1.mp4

Descripción de la solución propuesta

La problemática que se pretende resolver en el texto es la identificación de nuevas oportunidades de negocios mediante la evaluación del interés de posibles mercados objetivos. Para ello, la empresa LIRCAYHUB trabaja en conjunto con su equipo de marketing para realizar estudios y mediciones que permitan identificar dichas oportunidades. La empresa quiere evaluar la posibilidad de ofrecer distintos tipos de servicios a través de una aplicación web que permita a los usuarios interactuar con ella. Esta aplicación deberá ser accesible desde múltiples plataformas (web, teléfonos y tablets) y ofrecer servicios como reserva de espacios de lavado de autos, servicios de reparación de bicicletas, mantenimiento de autos y bicicletas, limpieza de hogar y muebles, y reparaciones en general.

La problemática principal que se aborda es la escasez de servicios e información de ellos, lo que dificulta el mantenimiento en buen estado de los automóviles, scooters eléctricos y bicicletas, que se han convertido en objetivos primordiales para las personas. Con la aplicación que se propone, los usuarios tendrían acceso a una amplia variedad de servicios y podrían reservarlos de forma sencilla y rápida, lo que solucionaría en gran medida este problema.

Para facilitar el hallazgo de una solución se desglosaron desde el texto los siguientes requerimientos funcionales y no funcionales:

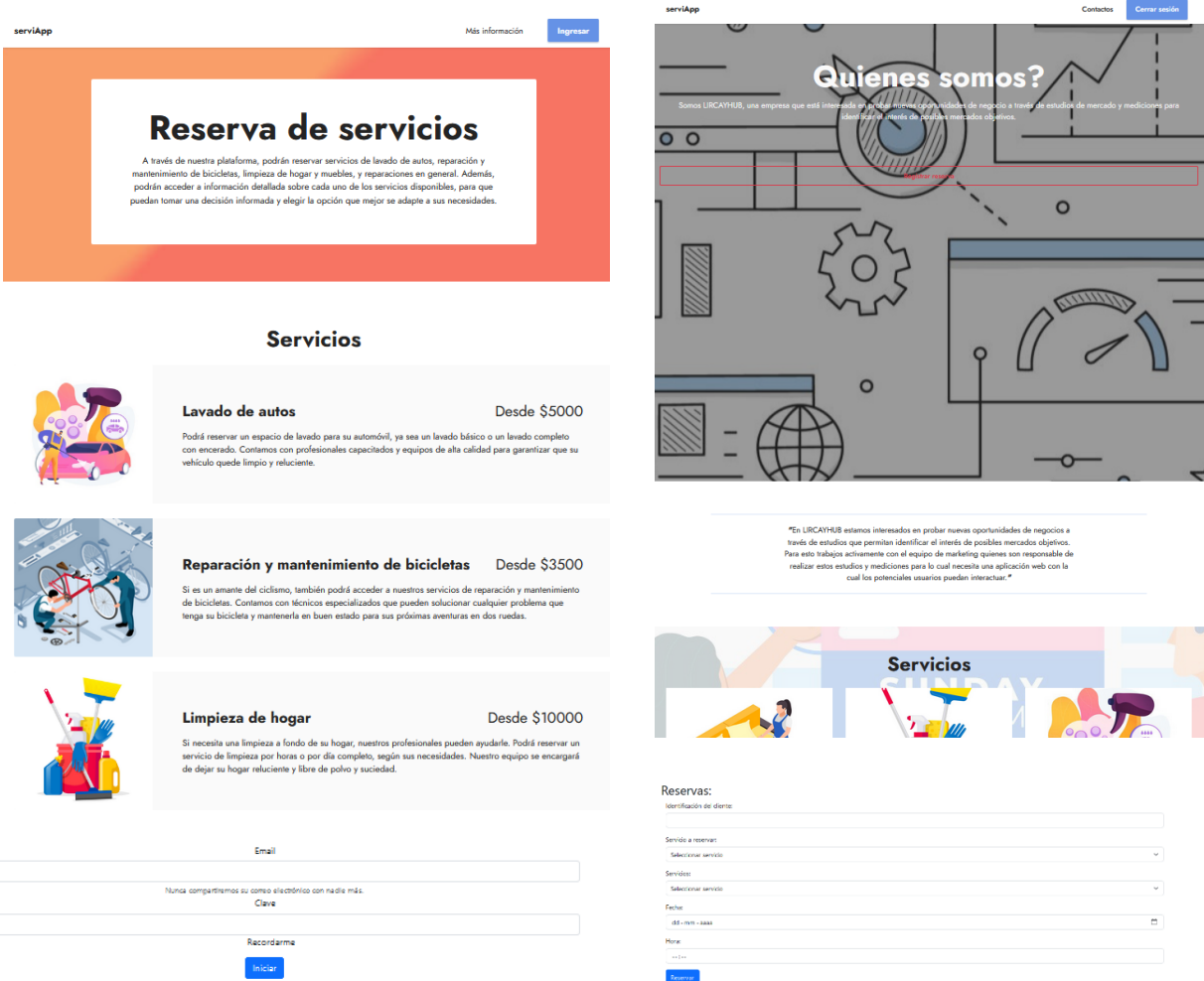
Requerimientos Funcionales:

- La aplicación web debe permitir a los usuarios potenciales interactuar con los servicios ofrecidos en la plataforma.
- La aplicación web debe permitir la reserva de servicios disponibles en la plataforma.
- La aplicación web debe permitir la creación de un proyecto, incluyendo su nombre, descripción general y las principales características del servicio.
- La aplicación web debe mostrar una lista de servicios disponibles en la plataforma.
- La aplicación web debe permitir la realización de reservas por hora o trabajo, en terreno y para la ejecución del trabajo.

Requerimientos No Funcionales:

- La aplicación web debe ser fácil de usar y navegar para los usuarios.
- La aplicación web debe ser accesible desde diferentes dispositivos y plataformas.
- La aplicación web debe tener un tiempo de respuesta rápido y un alto rendimiento.
- La aplicación web debe ser segura y confiable, protegiendo los datos personales de los usuarios y las transacciones financieras.
- La aplicación web debe ser escalable para manejar grandes volúmenes de tráfico y datos.
- La aplicación web debe ser fácil de mantener y actualizar.

Arquitectura de Software



Según la arquitectura de tres capas, las paginas realizadas en este proyecto son capas de presentación en las cuales el usuario puede visualizar la información e interactuar en la misma.

En las páginas de inicio de sesión son capas lógicas en donde se encarga de procesar la información que se recibe de la capa de presentación y tomar decisiones basadas en esa información. También se guarda la información a través de los formularios completados por los usuarios.

En el modelo MVC, la vista es la capa de presentación, el modelo es la capa de datos y el controlador es la capa lógica o de negocio. El controlador actúa como intermediario entre la vista y el modelo, procesando la información recibida de la vista y actualizando el modelo según sea necesario. La vista muestra la información al usuario y recibe sus entradas, y el modelo almacena y recupera los datos del sistema de almacenamiento permanente.

En resumen, aunque hay algunas similitudes entre la arquitectura de tres capas y el modelo MVC, cada uno tiene sus propias características y enfoques para la separación de responsabilidades en una aplicación. En el texto proporcionado, parece que se está utilizando la arquitectura de tres capas.

Tecnología Web

Para realizar este proyecto se utilizaron distintas tecnologías web, estas ayudan a los desarrolladores a realizar su trabajo de manera más simple y dinámica.

En este proyecto se utilizaron las siguientes tecnologías web:

-HTML: HTML es un lenguaje de marcado utilizado para crear páginas web. Permite estructurar el contenido y agregar elementos como texto, imágenes, enlaces, formularios y otros tipos de contenido multimedia. HTML es la base de la mayoría de los sitios web y se utiliza en combinación con CSS y JavaScript para crear páginas web interactivas.

-CSS: CSS es un lenguaje utilizado para dar estilo a páginas web. Permite controlar la presentación y el diseño de los elementos HTML, como el tamaño, el color, la fuente y la posición en la página. CSS se utiliza en combinación con HTML y JavaScript para crear páginas web visualmente atractivas e interactivas.

-Bootstrap 5: Bootstrap es un framework de diseño web que permite crear sitios web responsivos y móviles. Proporciona una serie de componentes y estilos predefinidos que se pueden utilizar para crear rápidamente páginas web visualmente atractivas y con un diseño coherente en diferentes dispositivos y pantallas.

-Moberaize: Moberaize es un framework de desarrollo web que utiliza tecnologías modernas como React y Next.js. Proporciona una serie de componentes y herramientas predefinidas que se pueden utilizar para crear sitios web y aplicaciones web de forma rápida y eficiente.

-GIT: Git es un sistema de control de versiones distribuido, diseñado para gestionar y rastrear cambios en el código fuente de proyectos de software. Permite a los desarrolladores trabajar en diferentes versiones de un proyecto de forma colaborativa, manteniendo un historial detallado de los cambios realizados en cada archivo y permitiendo la reversión a versiones anteriores si es necesario.

Además de su uso en el desarrollo de software, Git también se utiliza en otros campos, como la gestión de documentos y la colaboración en equipo.

Bibliografía

1. Richards, M. & Ford, N. (2015). Software Architecture Patterns. O'Reilly Media.
2. Martin, R. C. (2017). Clean Architecture: A Craftsman's Guide to Software Structure and Design. Prentice Hall.
3. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional.
4. Rascia, T. (s.f.). Model-View-Controller (MVC) Explained. Recuperado de <https://www.taniarascia.com/model-view-controller-mvc-explained/>.
5. Kotler, P., & Keller, K. L. (2016). Marketing management (15th ed.). Pearson Education Limited.
6. Laudon, K. C., & Traver, C. G. (2016). E-commerce 2016: business, technology, society. Pearson Education Limited.
7. Tsekouras, D., & Viasseri, A. (2019). The role of digital innovation in creating and capturing value in business models: A critical review and future directions. Journal of Business Research, 96, 136-146.
8. Chacon, S., & Straub, B. (2014). Pro Git. Apress. <https://git-scm.com/book/en/v2>.
9. Moberaize. (2021). Documentación de Moberaize. Recuperado el 2 de abril de 2023, de <https://moberaize.com/docs/>.
10. Bootstrap. (2021). Documentación de Bootstrap 5. Recuperado el 2 de abril de 2023, de <https://getbootstrap.com/docs/5.0/getting-started/introduction/>.
11. MDN Web Docs. (2021). Introducción a CSS. Recuperado el 2 de abril de 2023, de https://developer.mozilla.org/es/docs/Web/CSS/Introduccion_a_CSS.
12. MDN Web Docs. (2021). Introducción a HTML. Recuperado el 2 de abril de 2023, de https://developer.mozilla.org/es/docs/Web/HTML/Introduccion_a_HTML.