Nombre del Proyecto: Ecobeach

Plan de Producto de Software

Fecha: 12/11/2023

Identificación del Grupo: Tecnologías web

Nombre del Grupo: Team Ecobeach

Integrantes: Alejandro Duvan Correa Polo, Xavier Andrés De La Hoz Salas, Jaider José Gil Guzmán, Vanessa Paola Romero Domínguez y Froilán Daniel Vergara Romero.

Historial de Cambios

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Revisión | Fecha | Ítem | Descripción de Cambio | Responsable |
| 1.0 | 6/10/23 | Desde 1 hasta 6.2.4 | Inicio de Documentación | Todos los miembros del equipo |
| 2.0 | 12/11/23 | Desde 6.2.4 hasta 6.7.6 | Finalización de documentación | Todos los miembros del equipo |

Tabla de Contenido

[1 Proyecto de Producto de Software 5](#_Toc147429584)

[1.1 Identificación y Contexto del Problema 5](#_Toc147429585)

[1.2 Propósito, Alcance y Objetivo del Proyecto 5](#_Toc147429586)

[1.3 Suposiciones y Restricciones 6](#_Toc147429587)

[1.4 Entregables del proyecto 6](#_Toc147429588)

[1.5 Cronograma y Resumen del Presupuesto 7](#_Toc147429589)

[2 Documentos de Referencia 8](#_Toc147429590)

[3 Definiciones y Acrónimos 9](#_Toc147429591)

[4 Resumen del Trabajo Requerido 10](#_Toc147429592)

[5 Organización y Control del proyecto 12](#_Toc147429593)

[6 Proceso de Desarrollo de Software 13](#_Toc147429594)

[6.1 Entorno del Proceso de Software 13](#_Toc147429595)

[6.2 Diseño de Software 14](#_Toc147429596)

[6.2.1 Definición de Roles o Actores 14](#_Toc147429597)

[6.2.2 Análisis de Requerimientos de Software 15](#_Toc147429598)

[6.2.3 Requerimientos Funcionales 15](#_Toc147429599)

[6.2.4 Requerimientos No funcionales 16](#_Toc147429600)

[6.3 Modelado de Software 17](#_Toc147429601)

[6.3.1 Diagramas de Casos de Uso 17](#_Toc147429602)

[6.3.2 Diagramas de Secuencia 18](#_Toc147429603)

[6.3.3 Diagramas de Actividad 18](#_Toc147429604)

[6.3.4 Diagrama de Clases 18](#_Toc147429605)

[6.4 Diseño de la Interfaz 18](#_Toc147429606)

[6.4.1 Arquitectura de Información 18](#_Toc147429607)

[6.4.2 Wireframe 19](#_Toc147429608)

[6.5 Diseño de Datos 19](#_Toc147429609)

[6.5.1 Método de acceso a la base de datos 19](#_Toc147429610)

[6.5.2 Definición de las Entidades de datos 19](#_Toc147429611)

[6.5.3 Diagrama Entidad Relación 19](#_Toc147429612)

[6.5.4 Diccionario de Datos 19](#_Toc147429613)

[6.5.5 Restricciones de seguridad e integridad 20](#_Toc147429614)

[6.6 Diseño de la Arquitectura de Software 20](#_Toc147429615)

[6.6.1 Patrón de Diseño 20](#_Toc147429616)

[6.6.2 Diagrama de Componentes 20](#_Toc147429617)

[6.6.3 Diagrama de Despliegue 20](#_Toc147429618)

[6.7 Construcción e Implementación de Software 20](#_Toc147429619)

[6.7.1 Integración del sistema de software 20](#_Toc147429620)

[6.7.2 Verificación y validación del sistema de software 20](#_Toc147429621)

[6.7.3 Integración de software en Hardware 21](#_Toc147429622)

[6.7.4 Transición de software y gestión de lanzamientos 21](#_Toc147429623)

[6.7.5 Mantenimiento del software 21](#_Toc147429624)

[6.7.6 Gestión de defectos y problemas de software 21](#_Toc147429625)

[Anexos 22](#_Toc147429626)

# Proyecto de Producto de Software

## Identificación y Contexto del Problema

La creciente marea de desechos plásticos y basura que anualmente invade nuestros océanos, con una asombrosa cifra de ocho millones de toneladas, presenta un desafiante dilema ambiental. Este fenómeno, exacerbado por la falta de una gestión efectiva de residuos en tierra, impacta directamente a la vida marina, dejando a más de 100 mil animales marinos heridos o muertos cada año a causa de la ingestión accidental de plásticos y enredos en desechos flotantes. Aunque el plástico lidera la lista de contaminantes, el 90% de esta polución proviene de solo 10 grandes ríos en Asia y África, según estudios de “Environmental Science & Technology”. Además, las playas, ya amenazadas por la erosión costera inducida por el cambio climático, ven comprometidos sus hábitats naturales, como estuarios y humedales. Este contexto subraya la urgencia de una gestión integral de residuos y la necesidad crítica de abordar la contaminación marina para preservar la biodiversidad, la salud de los ecosistemas y garantizar la sostenibilidad de nuestros océanos para las generaciones venideras (Vanguardia, 2022).

## Propósito, Alcance y Objetivo del Proyecto

El propósito fundamental de este proyecto es abordar de manera efectiva la creciente amenaza de la contaminación marina, en especial en las zonas costeras, a través de la creación de una aplicación web dedicada. La cual, busca proporcionar a las personas una herramienta accesible y colaborativa para organizar y participar en eventos de limpieza costera.

Por lo tanto, tiene como finalidad ayudar a mitigar la contaminación marina, a través de una plataforma innovadora. La aplicación permitirá a los usuarios publicar, organizar y unirse a eventos de limpieza costera. Los usuarios podrán crear fácilmente sus propias iniciativas, proporcionando detalles sobre la ubicación, fecha y tipo de limpieza, y otros miembros podrán unirse a estas actividades. La plataforma facilitará la coordinación eficiente de esfuerzos para combatir la contaminación, promoviendo la participación ciudadana y concientizando sobre la importancia de la preservación de los océanos.

El proyecto se enfocará en la creación de una plataforma interactiva con el objetivo de catalizar un aumento significativo en la conciencia ambiental, evidenciado a través de la proliferación de eventos de limpieza y la participación de la comunidad en la preservación costera. Además de lograr una participación más informada, el proyecto aspira a generar un impacto palpable en la reducción de desechos marinos. Para abordar posibles limitaciones, se analizarán obstáculos relacionados con el acceso tecnológico y geográfico, implementando soluciones que garanticen la accesibilidad del aplicativo. Se reconocen desafíos inherentes, como la necesidad de asegurar la sostenibilidad financiera a largo plazo y la dependencia crucial de la participación comunitaria para lograr un impacto genuino en la mitigación de la contaminación marina. La plataforma también incorporará prácticas sostenibles en su diseño y operación, medirá su impacto a través de métricas específicas y fomentará la colaboración activa entre los miembros de la comunidad, organizaciones y autoridades locales. Además, se pondrá énfasis en la educación continua para mantener a la comunidad informada y comprometida a largo plazo en la preservación ambiental.

## Suposiciones y Restricciones

Para este proyecto, existen unos supuestos, restricciones y consideraciones que deben tenerse en cuenta, algunos de los supuestos son, el hecho de que se asume que la aplicación web, logrará una participación de la comunidad en cuanto a organización y participación de los eventos de limpieza costera.

Se parte del supuesto de que la mayoría de la comunidad objetivo tiene acceso a dispositivos tecnológicos y conectividad para utilizar la aplicación web, se supone también que la conciencia ambiental será suficiente para motivar la participación de eventos de limpieza, así como la adopción de la aplicación web.

Por otro lado, tenemos las restricciones como, el tiempo ya que la implementación de la aplicación ser realizará en fases iterativas, priorizando las características esenciales, se establecerá un plazo estricto para la entrega de la aplicación, limitando el tiempo de desarrollo para garantizar una respuesta rápida a la problemática de la contaminación marina.

En adición a lo anterior mencionado, tenemos restricciones en cuanto al costo, ya que se optará por soluciones de herramientas gratuitas siempre que sea posible, minimizando los costos asociados con licencias y desarrollo personalizado, también se buscará diseñar una arquitectura eficiente desde el principio con el fin de minimizar los gastos relacionados con el crecimiento de la aplicación o una posible expansión futura.

## Entregables del proyecto

**Plataforma de Aplicación Web:**

* Diseño con interfaz de usuario intuitiva.
* Funcionalidades para la publicación, organización y participación en eventos de limpieza costera.
* Implementación de un sistema de registro para los usuarios.
* Desarrollo de perfiles de usuario con información personal y opciones de configuración.
* Herramienta para que los usuarios creen y publiquen eventos de limpieza.

**Documentación Técnica:**

* Diagrama relacional para visualizar la estructura de la base de datos.
* Diagramas de casos de uso para representar la interacción usuario-sistema.
* Diagrama de clases para detallar la estructura de clases y relaciones.
* Y todos los Aspectos detallados sobre la arquitectura de la aplicación web.

## Cronograma y Resumen del Presupuesto

|  |  |
| --- | --- |
| CRONOGRAMA DE PROYECTO | |
| DESARROLLO | TIEMPO |
| DEFINICIÓN DE REQUERIMIENTOS Y DOCUMENTACIÓN | SEMANA 1 |
| DISEÑO DE LA APLICACIÓN WEB | SEMANA 2 |
| DESARROLLO DE LA APLICACIÓN | SEMANA 3-4 |
| PRUEBAS Y DEPURACIÓN | SEMANA 5 |
| ENTREGA DE APLICACIÓN | SEMANA 6 |

|  |  |  |
| --- | --- | --- |
| PRESUPUESTO | | |
| DESCRIPCIÓN | VALOR UNITARIO | VALOR TOTAL |
| Desarrollo de Software. |  | $5.000.000 COP |
| Servicios de alojamiento y dominio. | $400.000 COP | $2.000.000 COP |
| Remuneración para el equipo de trabajo. | $1.800.000 COP | $9.000.000 COP |
| Dispositivos para el desarrollo. | $3.000.000 COP | $6.000.000 COP |
| Gastos varios. | $500.000 COP | $3.000.000 COP |
| Soporte y Mantenimiento del aplicativo web. | $1.000.000 COP | $4.000.000 COP |
| Costos Administrativos. |  | $5.000.000 COP |
| Contingencias |  | $2.000.000 COP |
| Capacitación |  | $1.100.000 COP |
| Pruebas y Garantía de Calidad. | $400.000 COP | $1.600.000 COP |
| Base de datos | $500.000 COP | $2.000.000 COP |
| TOTAL DE PRESUPUESTO | //////////////////////////////// | $40.700.000 COP |

# Documentos de Referencia

Referencias

|  |  |  |  |
| --- | --- | --- | --- |
| Referencia | Titulo | Autor | Año |
| [R1] | Vanguardia. (2022, Septiembre 28). Playas y océanos libres de basura: ¿Cómo contribuir para evitar la contaminación? www.vanguardia.com. https://www.vanguardia.com/mundo/ola-verde/playas-y-oceanos-libres-de-basura-HB5737687#:~:text=Cada%20a%C3%B1o%20un%20promedio%20de,que%20se%20enfrentan%20los%20oc%C3%A9anos | Vanguardia | 2022 |
| [R2] | Cámara de comercio de España (2019), ¿Qué es scrum y cómo puede ayudar a tu empresa?, disponible en:  https://www.camara.es/blog/innovacion-y-competitividad/que-es-scrum-y-como-puede-ayudar-tu-empresa , 3 de septiembre de 2022. | Cámara de comercio de España | 2019 |
| [R3] | Kostiantyn Levin (2020), https://www.alamy.es/metodologia-de-marco-de-scrum-ciclo-de-vida-diagrama-plano-esquema-infografias-con-acumulacion-sprint-planificacion-revision-incremento-y-retrospectiva-image359582734.html. | Kostiantyn Levin | 2020 |
| [R4] | Proceso de vida en océanos, mares y ríos por Johanna Caizatoa. (s. f.). PPT. https://www.slideshare.net/JohannaEstefaniia/proceso-de-vida-en-ocanos-mares-y-ros-por-johanna-caizatoa-195147535 | Johanna Caizatoa | s.f |
| [R5] | KeepCoding, R. (2022, 21 marzo). ¿Qué es el product backlog en scrum? KeepCoding Bootcamps. https://keepcoding.io/blog/que-es-el-product-backlog-en-scrum/ | KeepCoding | 2022 |
| [R6] | EcoAvant.com. (s. f.). Hábitat - EcoAvant.com. https://www.ecoavant.com/tag/habitat | EcoAvant | s.f |
| [R7] | Atlassian. (s. f.). Scrum: qué es, cómo funciona y cómo empezar | Atlassian. https://www.atlassian.com/es/agile/scrum | Atlassian | s.f |
| [R8] | colaboradores de Wikipedia. (2023). Contaminación. Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Contaminaci%C3%B3n#:~:text=La%20contaminaci%C3%B3n%20es%20la%20presencia,sustancias%20nocivas%20u%20otros%20elementos | Wikipedia | 2023 |

# Definiciones y Acrónimos

**CONTAMINACION:** es la presencia de un constituyente, impureza o algún otro elemento indeseable que estropea, corrompe, infecta, inutiliza o degrada un material, cuerpo físico, entorno natural, lugar de trabajo, etc (Wikipedia, 2023).

**DESECHOS PLASTICOS:** Se genera un residuo de plástico cuando cualquier elemento fabricado a partir de este material es descartado por su poseedor al no encontrarle valor o utilidad. Una de las características más notorias de los residuos plásticos es su larga duración.

**MEDIO AMBIENTE:** Es el entorno en el que se despliega la existencia de diversos seres, facilitando sus relaciones mutuas. Este espacio alberga tantos organismos vivos.  
**VIDA MARINA:** La vida marina, vida en el mar o vida oceánica, la conforman las plantas, los animales y otros organismos que viven en el agua salada de los mares y océanos, o el agua salobre de los estuarios costeros. En un nivel fundamental, la vida marina ayuda a determinar la naturaleza misma de nuestro planeta (Johanna Caizatoa, s. f.).

**HABITAT:** En un ecosistema, el hábitat es el lugar donde vive la comunidad. El concepto de hábitat es utilizado por biólogos y ecólogos con una acepción mientras que arquitectos y urbanistas lo utilizan desde una visión antrópica (EcoAvant.com, s. f.).

**SCRUM:** Es una estructura de gestión de proyectos en el contexto de la metodología ágil que colabora en la organización y dirección de los trabajos mediante un conjunto de valores, principios y técnicas. scrum anima a los equipos a aprender a través de las experiencias, a autoorganizarse mientras aborda un problema y a reflexionar sobre sus victorias y derrotas para mejorar continuamente (Atlassian, s. f.).

**INTERAZ:** En el ámbito de la informática, se emplea una interfaz para referirse a la conexión funcional que posibilita la comunicación y el intercambio de información entre dos sistemas, programas, dispositivos o componentes de diversa índole, y esto también se puede aplicar como un caso dentro del contexto de la realidad virtual.

**SOFTWARE:** El término "software", también denominado "lógico" o "soporte lógico", se refiere al sistema formal dentro de una computadora que engloba todos los componentes lógicos esenciales que permiten la ejecución de funciones específicas, en contraposición a los elementos físicos que se conocen. como "hardware".

**PRODUCT BACKLOG:** es una lista de trabajo ordenado por prioridades para el equipo de desarrollo que se obtiene de la hoja de ruta y sus requisitos. Los elementos más importantes se muestran al principio del backlog del producto para que el equipo sepa qué hay que entregar primero (KeepCoding, 2022).

# Resumen del Trabajo Requerido

El proyecto se propone desarrollar una plataforma de aplicación web con el objetivo central de facilitar la organización y participación en eventos de limpieza costera. Se parte de la suposición fundamental de que la comunidad objetivo se involucrará activamente en la plataforma, publicando y participando en eventos de limpieza. Asimismo, se presupone que la mayoría de los usuarios potenciales tienen acceso a dispositivos tecnológicos para utilizar la plataforma. No obstante, este esfuerzo de desarrollo de software está sujeto a diversas restricciones. En primer lugar, existe una restricción de cronograma que impone la necesidad de completar el desarrollo en un período específico para maximizar su eficacia. Asimismo, se impone una restricción presupuestaria que guiará las decisiones de desarrollo y asegurará la gestión eficiente de los recursos disponibles. Además, el equipo cuenta con recursos limitados, lo que influirá en la complejidad y escala del proyecto. En términos tecnológicos, la elección de tecnologías estará sujeta a restricciones presupuestarias y a la necesidad de asegurar la accesibilidad y usabilidad para un amplio rango de usuarios.

En cuanto a los objetivos de desarrollo de software, se busca crear una plataforma intuitiva que facilite la interacción y coordinación entre los usuarios, así como la evaluación visual del impacto de los eventos de limpieza. Los productos entregables incluirán la plataforma web funcional con módulos clave, documentación técnica detallada que abarque diagramas, y documentación para garantizar la comprensión y uso eficiente de la plataforma. Este enfoque integral pretende asegurar el desarrollo exitoso de una plataforma que no solo cumpla con los objetivos técnicos, sino que también satisfaga las necesidades y expectativas de la comunidad, contribuyendo así a la preservación efectiva de las zonas costeras.

# Organización y Control del proyecto

|  |  |  |
| --- | --- | --- |
| Nombre y Apellidos | Rol en el Proyecto | Responsabilidades |
| Alejandro Correa | Gerente de proyecto | Responsable de la planificación general y supervisión del proyecto, garantía de cumplimiento de objetivos y plazos. |
| Xavier De la Hoz | Documentación técnica | Garantía de una documentación clara y comprensible y Colaboración con el equipo para describir procesos y decisiones técnicas. |
| Jaider Gil | Desarrollador Back-end | Desarrollo de la lógica del servidor y la base de datos e implementación de funcionalidades no visibles para el usuario. |
| Vanessa Romero | Desarrollador Front- end | Implementación de la interfaz de usuario y desarrollo de funcionalidades visibles para los usuarios. |
| Froilán Vergara | Evaluador/Control de Calidad | Realización de pruebas exhaustivas,  Garantía de calidad del software e  Identificación y corrección de errores. |

# Proceso de Desarrollo de Software

## Entorno del Proceso de Software

El modelo de ciclo de vida que se utilizará será una metodología de desarrollo ágil, para este caso Scrum, se aplicará de manera colaborativa con el fin de obtener el mejor desarrollo posible de nuestro proyecto por medio de ciclos cortos en donde en cada uno de ellos proporcione un incremento al producto final.

Scrum, se desarrolla por medio de ciclos temporales cortos y de duración fija, estos son llamados Sprints, la duración de estos es alrededor de 2 semanas, pero tiene la posibilidad de ampliarse si el producto que se está llevando a cabo lo requiere.

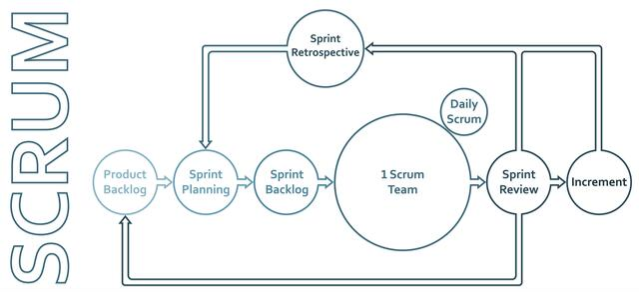
Esta metodología de desarrollo ágil de software, parte de las necesidades que son base para el plan de proyecto, esto es conocido como **Product Backlog,** para la lista de necesidades interviene el clienteo también conocido como el **Product Owner**, el cual,es quien realmente prioriza las funciones.

Dentro de todo este desarrollo, existe una persona que gestiona todos los procesos de trabajo y se asegura que todo se ejecute de la mejor forma y las entregas se cumplan en los tiempos establecidos, esta persona se conoce como el **Scrum Master,** quien, del mismo modo, resuelve todos aquellos inconvenientes que puedan surgir.

Las fases del proceso Scrum son, **planificación** que consta en la selección de requisitos y planificación, se realiza una reunión donde se presenta lo que se quiere; **Ejecución y desarrollo** se define el Sprint Backlog y se establece quien desempeñará cada tarea y para estas se suponen entregan parciales, por último, **Revisión del sprint,** cuando se terminan las iteraciones del sprint, se realiza una revisión de los requisitos dados por el product owner y se dan los respectivos cambios y se observa todas las dificultades que se presentan con el fin de darle un solución y que esta no sea un obstáculo para el próximo Sprint (Cámara de comercio de España, 2019).

Figura 1

*Ciclo de vida Scrum*



*Nota. Tomada de Metodología de marco de Scrum ciclo de vida diagrama plano esquema infografías con acumulación, sprint, planificación, revisión, incremento y retrospectiva [Fotografía], Kostiantyn Levin (2020), https://www.alamy.es/metodologia-de-marco-de-scrum-ciclo-de-vida-diagrama-plano-esquema-infografias-con-acumulacion-sprint-planificacion-revision-incremento-y-retrospectiva-image359582734.html.*

## Diseño de Software

### Definición de Roles o Actores

|  |  |
| --- | --- |
| Nombre del Rol | Descripción |
| Usuarios | Debido a la creciente preocupación por el impacto devastador de los desechos en la vida submarina y la evidente falta de una gestión efectiva para abordar esta problemática, los usuarios, quienes desempeñan un papel crucial en el funcionamiento de esta aplicación web, experimentan una creciente necesidad de contar con plataformas que faciliten la interacción entre ellos satisfaciendo el propósito fundamental de esta plataforma que es fomentar la conservación de la fauna marina. En línea con los objetivos de este proyecto, se ha incorporado la iniciativa de organizar eventos para la limpieza costera como una estrategia integral. Estos eventos no solo fortalecerán la conciencia ambiental, sino que también proporcionarán a los usuarios una plataforma tangible para contribuir activamente a la conservación marina, generando así un impacto positivo y sostenible en el entorno costero. |

### Análisis de Requerimientos de Software

Se llevarán a cabo entrevistas con stakeholders clave, incluidos miembros de la comunidad interesados en eventos de limpieza costera, organizadores de eventos, y posibles usuarios de la plataforma.

Las entrevistas permitirán una comprensión más profunda de las necesidades específicas, desafíos y expectativas de los usuarios.

### Requerimientos Funcionales

|  |  |
| --- | --- |
| Referencia | Requerimiento |
| RF01 | Para registrarse en la plataforma, debe existir un apartado de registro de cuentas  para que los usuarios puedan ingresar su información para crear una cuenta. |
| RF02 | Para iniciar sesión, el sistema debe verificar que el usuario y la contraseña ingresados sean correctos para que los usuarios puedan acceder a la plataforma después de una verificación exitosa. |
| RF03 | Para cerrar sesión, debe existir un mecanismo para borrar la sesión almacenada en el navegador para que los usuarios puedan salir de sus cuentas, garantizando la seguridad de la sesión. |
| RF04 | Para crear eventos, debe existir un apartado que permita a los usuarios crear eventos para que los mismos puedan contribuir a la plataforma generando nuevos eventos. |
| RF05 | Para listar eventos disponibles, el sistema mostrará eventos no unidos para que los usuarios vean eventos en los que aún no han participado (eventos disponibles). |
| RF06 | Para unirse a un evento, el sistema permitirá unirse a eventos existentes para que los usuarios puedan participar en eventos disponibles. |
| RF07 | Para listar eventos donde se ha unido el usuario, el sistema mostrará eventos en los que ha participado en donde se verá una lista de aquellos a los que se haya unido. |
| RF08 | Para dejar de seguir un evento, el sistema cambiará el estado del usuario con respecto a su participación en el evento para cancelar su participación al evento. |
| RF09 | Para borrar un evento creado, el sistema permitirá al usuario eliminar el evento que ha creado. |
| RF10 | Para eliminar eventos finalizado, el sistema permitirá al usuario eliminar Los eventos finalizados, se eliminarán automáticamente de la lista activa. |
| RF11 | El sistema permitirá a los usuarios listar un apartado de ajustes. |
| RF12 | El sistema permitirá a los usuarios cambiar su contraseña si lo desea. |
| RF13 | El sistema permitirá a los usuarios la actualización de los datos. |
| RF14 | El sistema permitirá al usuario generar un reporte pdf de los eventos que ha sido creado por el usuario logueado |
| RF15 | El sistema permitirá al usuario generar un reporte pdf de los eventos a los que se ha unido |

### Requerimientos No funcionales

|  |  |  |
| --- | --- | --- |
| Referencia | Tipo | Descripción |
| RFN-01 | Usabilidad | La interfaz de usuario deberá proporcionar una experiencia de navegación intuitiva para facilitar la exploración de eventos y la creación de nuevos eventos. |
| RNF-02 | Usabilidad | La aplicación debe ser diseñada de manera lógica de elementos en la pantalla, etiquetas claras para botones y menús, y flujos de navegación coherentes. |
| RNF-03 | Eficiencia | Toda funcionalidad del sistema debe responder al usuario en menos de 5 segundos. |
| RNF-04 | Eficiencia | La aplicación debe ser óptima en el uso de recursos, minimizando la carga en dispositivos móviles y tabletas, para facilitar su aprendizaje, uso y desempeño en diferentes plataformas. |
| RNF-05 | Seguridad | La aplicación debe implementar medidas de seguridad contra ataques comunes, como inyecciones SQL o ataques de fuerza bruta. |
| RNF-06 | Seguridad | La información de los usuarios, como contraseñas y datos personales, debe estar protegida mediante cifrado robusto. |
| RFN-07 | Aspectos éticos | La aplicación debe obtener el consentimiento explícito de los usuarios para el manejo de sus datos personales y respetar su privacidad. |
| RFN-08 | Aspectos éticos | Proporcionar información clara sobre cómo se utilizan los datos de los usuarios y las políticas de la aplicación en términos de ética y uso responsable. |
| RFN-09 | Aspectos Regulatorios y Legislativos | La aplicación debe cumplir con todas las leyes y regulaciones de privacidad de datos y protección al consumidor aplicables en las regiones donde opera la aplicación. |
| RFN-10 | Aspectos Regulatorios y Legislativos | Establecer políticas y mecanismos para evitar la publicación de contenido inapropiado o perjudicial en la plataforma. |

## Modelado de Software

Para modelar el diseño de nuestro sistema, realizaremos diagramas que de manera visual explicarán como funcionará el programa por ende, se demostrará de alguna manera con los diagramas como cada parte está conectada y como está relacionado directamente con los requisitos.

### Diagrama Descripción generada automáticamenteDiagramas de Casos de Uso

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Caso de Uso | Crear cuenta | | | |
| Versión | 1.0 | | Referencia | CU#01 |
| Autor | Vanessa Romero | | Fecha | 20/10/2023 |
| Actores | Usuario | | | |
| Referencias | RF01 | | | |
| Precondiciones | Ninguna | | | |
| Postcondición | Se muestra la pantalla de inicio al aplicativo web. | | | |
| Flujo Normal | | | | |
| Paso | Actor | Descripción | | |
| 1 | Sistema | Muestra el formulario de registro | | |
| 2 | Usuario | Digita sus datos personales y presiona el botón registrarse | | |
| 3 | Sistema | Verifica la información registrada | | |
| 5 | Sistema | Muestra la pantalla de inicio del aplicativo web. | | |
| Flujos alternos | Si en el formulario de registros se encuentran campos vacíos, al momento del usuario registrarse, el sistema enviará un mensaje al usuario “Campos vacíos, Verifique nuevamente”. | | | |
| Excepciones | Ninguna. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Caso de Uso | Iniciar sesión | | | |
| Versión | 1.0 | | Referencia | CU#02 |
| Autor | Xavier De La Hoz | | Fecha | 20/10/2023 |
| Actores | Usuario | | | |
| Referencias | RF02 | | | |
| Precondiciones | Crear cuenta | | | |
| Postcondición | Muestra pantalla de inicio del usuario. | | | |
| Flujo Normal | | | | |
| Paso | Actor | Descripción | | |
| 1 | Sistema | Muestra el formulario de inicio de sesión | | |
| 2 | Usuario | Digita el usuario, clave y presiona el botón ingresar | | |
| 3 | Sistema | Verifica la información registrada en la base de datos | | |
| 4 | Sistema | Muestra la pantalla de inicio al usuario | | |
| Flujos alternos | Si el nombre de usuario no está registrado en la base de datos el sistema muestra un mensaje al usuario “mensaje”  Si el nombre de usuario o la clave son incorrectas el sistema mostrará al usuario un mensaje “usuario o clave incorrecta, verifique nuevamente” | | | |
| Excepciones | El sistema puede denegar el ingreso. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Caso de Uso | Crear eventos | | | |
| Versión | 1.0 | | Referencia | CU#03 |
| Autor | Jaider Gil | | Fecha | 20/10/2023 |
| Actores | Usuario | | | |
| Referencias | MURF4 | | | |
| Precondiciones | Iniciar Sesión | | | |
| Postcondición | Muestra la pantalla de inicio con los eventos creados. | | | |
| Flujo Normal | | | | |
| Paso | Actor | Descripción | | |
| 1 | Sistema | Muestra el formulario para que el usuario pueda crear el evento. | | |
| 2 | Usuario | Digita los campos del formulario de crear eventos. | | |
| 3 | Sistema | Verifica los datos del formulario registrados | | |
| 4 | Sistema | Guarda la información del evento en la base de datos. | | |
| 5 | Sistema | Muestra la pantalla de inicio. | | |
| Flujos alternos | Si al momento de llenar el formulario, el usuario deja un campo vacío, el sistema enviará un mensaje al usuario “Campos vacíos, Verifique nuevamente”. | | | |
| Excepciones | Ninguna. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Caso de Uso | Listar eventos creados | | | |
| Versión | 1.0 | | Referencia | CU#03 |
| Autor | Jaider Gil | | Fecha | 20/10/2023 |
| Actores | Usuario | | | |
| Precondiciones | Iniciar Sesión | | | |
| Postcondición | Muestra la pantalla de inicio con los eventos creados. | | | |
| Flujo Normal | | | | |
| Paso | Actor | Descripción | | |
| 1 | Sistema | Muestra el apartado de eventos creados por el usuario. | | |
| 2 | Usuario | Elige ver la información del evento. | | |
| 3 | Sistema | Muestra un apartado con la información del evento que se creó | | |
| Flujos alternos | En caso de que el usuario quiera cancelar el evento que creo, puede hacerlo. | | | |
| Excepciones | Ninguna. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Caso de Uso | Listar eventos disponibles | | | |
| Versión | 1.0 | | Referencia | CU#04 |
| Autor | Xavier De La Hoz | | Fecha | 12/10/2023 |
| Actores | Usuario | | | |
| Referencias | RF05, RF06 | | | |
| Precondiciones | Iniciar sesión | | | |
| Postcondición | Muestra pantalla de inicio del usuario. | | | |
| Flujo Normal | | | | |
| Paso | Actor | Descripción | | |
| 1 | Sistema | Muestra todos los eventos disponibles. | | |
| 2 | Usuario | Selecciona al evento al cual quiere participar. | | |
| 3 | Sistema | Guarda el evento al que usuario se unió en la base de dato. | | |
| 4 | Sistema | Muestra la pantalla de eventos disponible | | |
| Flujos alternos | Si el usuario no le interesa ningún evento, este puede regresar a la pantalla de bienvenida de la aplicación. | | | |
| Excepciones | Ninguna. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Caso de Uso | Generar reporte pdf | | | |
| Versión | 1.0 | | Referencia | CU#05 |
| Autor | Vanessa Romero | | Fecha | 12/11/2023 |
| Actores | Usuario | | | |
| Referencias | RF14 y RF15 | | | |
| Precondiciones | Iniciar sesión | | | |
| Postcondición | Mostrar en el navegador el reporte pdf. | | | |
| Flujo Normal | | | | |
| Paso | Actor | Descripción | | |
| 1 | Sistema | Muestra botón de generar reporte pdf | | |
| 2 | Usuario | Selecciona el botón | | |
| 3 | Sistema | Genera en el navegador el respectivo reporte pdf. | | |
| Flujos alternos | No tiene ningún flujo alterno | | | |
| **Excepciones** | Ninguna | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Caso de Uso | Listar eventos unidos | | | |
| Versión | 1.0 | | Referencia | CU#05 |
| Autor | Xavier De La Hoz | | Fecha | 12/10/2023 |
| Actores | Usuario | | | |
| Referencias | RF07, RF08, RF10 | | | |
| Precondiciones | Iniciar sesión | | | |
| Postcondición | Muestra pantalla de inicio del usuario. | | | |
| Flujo Normal | | | | |
| Paso | Actor | Descripción | | |
| 1 | Sistema | Muestra todos los eventos a los que el usuario se ha unidos. | | |
| 2 | Usuario | Selecciona al evento al cual quiere cancelar su participación | | |
| 3 | Sistema | Guarda el estado(participación) del usuario de dicho evento en la base de dato. | | |
| 4 | Usuario | Elimina cualquier evento al que ha asistido o que ya finalizado | | |
| 5 | Sistema | Elimina el evento en la base de datos. | | |
| 6 | Sistema | Muestra la pantalla de eventos disponible | | |
| Flujos alternos | Si el usuario no le quiere cancelar su participación o eliminar un evento finalizado, este puede regresar a la pantalla de bienvenida de la aplicación. | | | |
| Excepciones | Ninguna. | | | |

### Diagramas de Secuencia

Diagrama

Descripción generada automáticamente

### Diagramas de Actividad

Diagrama, Esquemático

Descripción generada automáticamente

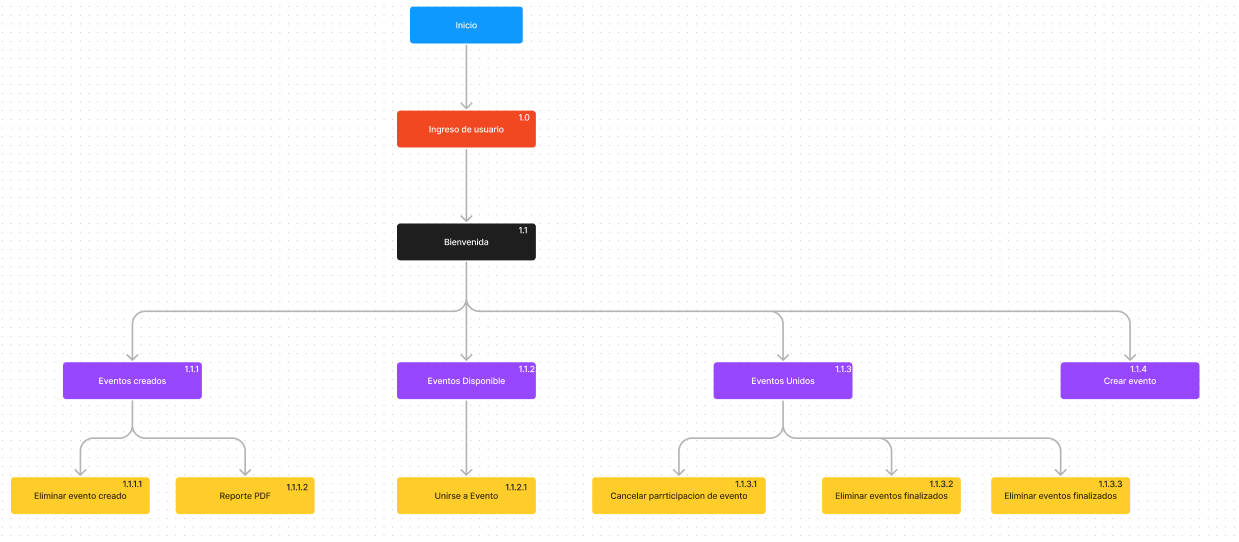
### Diagrama de Clases

Diagrama

Descripción generada automáticamente

## Diseño de la Interfaz

### Arquitectura de Información

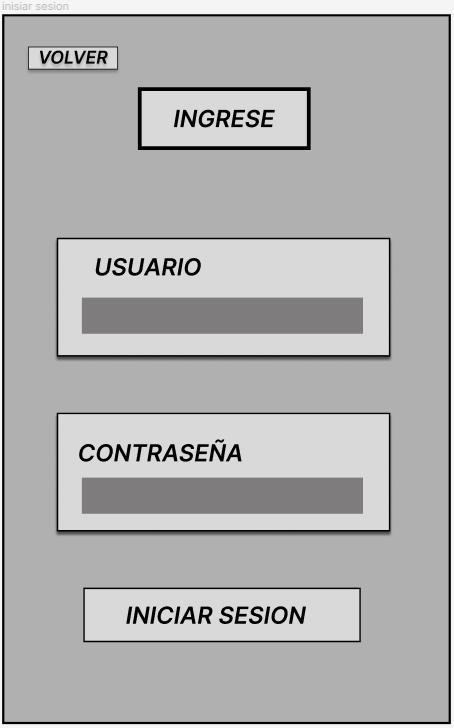


Los elementos de navegación principal son aquellos que se encuentran luego del ingreso como usuario, aquí vamos a poder encontrar directamente todas aquellas opciones disponibles para interactuar con el aplicativo web, las cuales son, **Eventos creados** permite a los usuarios tener un apartado de los eventos que el mismo creo y generar un reporte pdf de ellos, también encontrará otra opción como lo es **eventos disponibles**, la cual le permite al usuario visualizar todos los estos eventos que se encuentran disponible y elegir a en que evento puede participar, que sea de su interés. Y como última opción de navegación estos **eventos unidos** que le brinda al usuario ver a los eventos en los cuales se ha unido, y a su vez le proporciona la opción de cancelar su participación en dicho evento y eliminar los eventos que ya han caducado, además, permite generar un reporte pdf. Por último, tenemos el elemento de navegación **Crear evento** que permite que el usuario genere un nuevo evento para que los demás puedan unirse a el.

### Wireframe

Interfaz de usuario gráfica

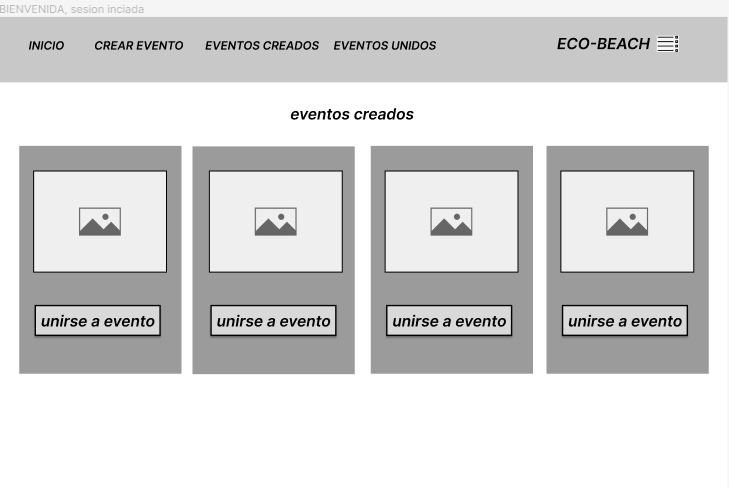
Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza bajaPrimero se cuenta con una página principal donde se encuentran dos opciones, una por si el usuario no tiene cuenta y se pueda registrar, también con una opción para iniciar sesión ya una vez obtenida la cuenta.

al seleccionar click en registrar se despliega un apartado donde deberán llenar los campos de completar información, también la presencia de un botón para registrar la información.

Al seleccionar en iniciar sesión se desplega una pantalla donde colocaran el usuario y contraseña respectiva y así ingresar al aplicativo.



Al iniciar sesión se mostrará la pantalla de bienvenida donde mostraran los eventos creados, opción para ver eventos creados por ti, la opción de crear un evento y el botón para ver los eventos a los que te has unido

Interfaz de usuario gráfica

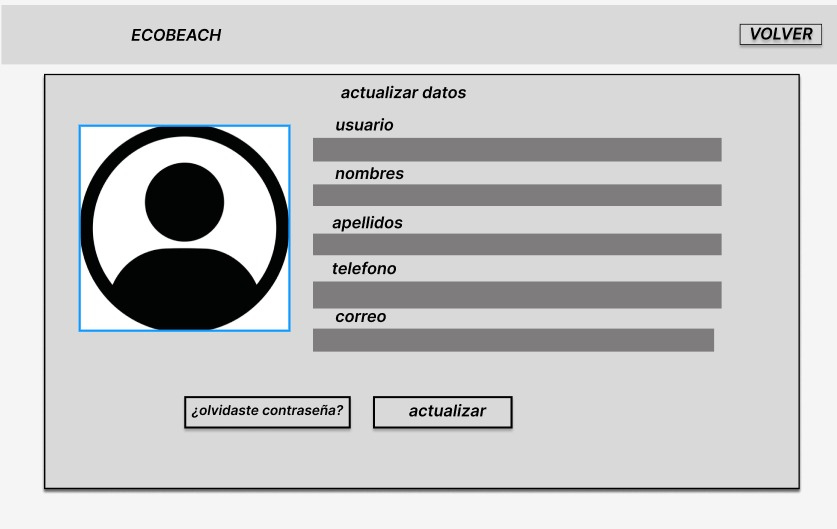
Descripción generada automáticamente

Al dar click en el botón crear evento arrojara este apartado donde podrán ingresar la información requerida para que un evento sea creado, una opción o botón publicar donde confirmaran la creación de tal evento.

Texto

Descripción generada automáticamente

Abra un apartado donde se encontrará opciones de ajustes que hacen referencia a actualización de información o cambio de contraseña y opción de cerra sesión.



Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

al dar click en la opción de eventos unidos, desplegara esta pantalla acompañada de los eventos a los que te has unido con su respectiva información y un botón para generar el archivo pdf.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

al dar click en la opción de eventos creados, desplegara esta pantalla acompañada de los eventos que has creado con su respectiva información y un botón para generar el archivo pdf.

## Diseño de Datos

### Método de acceso a la base de datos

Para este caso, el método de acceso a la base de datos se realizó de la siguiente manera: Se utilizó Angular, en el lado del cliente, el cual interactúa con el servidor Node.js que actúa como intermediario entre la interfaz de usuario y la base de datos gestionada por phpMyAdmin.

En este escenario, Node.js se conecta a la base de datos MySQL de phpMyAdmin utilizando un controlador de base de datos compatible con Node.js, como "MySQL". La conexión a la base de datos se establece mediante una cadena de conexión que contiene la información necesaria, como la dirección del servidor, el nombre de usuario y la contraseña.

### Definición de las Entidades de datos

|  |  |
| --- | --- |
| Entidad | Descripción |
| Cuenta\_Usuarios | Por medio de esta, el usuario accede al aplicativo para crear eventos y unirse a eventos. |
| Eventos\_Me | Guarda la información de los eventos creados por un usuario. |
| Enventos\_Unidos | Guarda todos los eventos a los que los usuarios se ha unido. |
| Eventos\_Creados | Guarda toda la información de dichos eventos creado por el usuario. |

### Diagrama Entidad Relación



### Diccionario de Datos

|  |  |  |
| --- | --- | --- |
| TABLA EVENTOS\_ME | | |
| NOMBRE | TIPO DE DATO | DESCRIPCIÓN |
| Ideventos\_me | ENTERO | Clave primaria. |
| cuentas\_usuarios\_idcuentas\_usuarios | ENTERO | Clave foránea de tipo entero que hace referencia a la tabla cuenta\_usuarios. |
| eventos\_creados\_ideventos\_ creados | ENTERO | Clave foránea de tipo entero que hace referencia a la tabla eventos\_creados. |

|  |  |  |
| --- | --- | --- |
| TABLA CUENTA\_USUARIOS | | |
| NOMBRE | TIPO DE DATO | DESCRIPCIÓN |
| idcuentas\_usuarios | ENTERO | Clave primaria. |
| usuario | VARCHAR | Hace referencia al nombre de usuario que será creado por el usuario que quiera crear eventos o unirse a eventos. |
| contraseña | VARCHAR | Hace referencia a la contraseña que será creado por el usuario que quiera crear eventos o unirse a eventos. |
| nombre | VARCHAR | Hace referencia al nombre ingresado por el usuario. |
| apellidos | VARCHAR | Hace referencia al apellido ingresado por el usuario. |
| teléfono | VARCHAR | Hace referencia al campo que contiene información de contacto del usuario. |
| correo | VARCHAR | Hace referencia al campo que contiene información del correo electrónico del usuario. |

|  |  |  |
| --- | --- | --- |
| TABLA EVENTOS\_UNIDOS | | |
| NOMBRE | TIPO DE DATO | DESCRIPCIÓN |
| ideventos\_unidos | ENTERO | Clave primaria. |
| estado | VARCHAR | Hace referencia al estado del evento, si se encuentra disponible, cancelado o finalizado. |
| cuentas\_usuarios\_idcuentas\_usuarios | ENTERO | Clave foránea de tipo entero que hace referencia a la tabla cuenta\_usuarios. |
| eventos\_creados\_ideventos\_creados | ENTERO | Clave foránea de tipo entero que hace referencia a la tabla eventos\_creados. |

|  |  |  |
| --- | --- | --- |
| TABLA EVENTOS\_CREADOS | | |
| NOMBRE | TIPO DE DATO | DESCRIPCIÓN |
| ideventos\_creados | ENTERO | Clave primaria. |
| nombre | VARCHAR | Hace referencia al nombre del evento ingresado que el usuarios va a crear. |
| descripción | VARCHAR | Se refiere a la descripción del evento que el usuario está por crear, los datos necesarios para los usuarios(ubicación). |
| estado | VARCHAR | Se refiere al estado del evento, si el usuario que lo creó, lo elimina o si está disponible |
| fecha\_inicio | TIMESTAMP | Hace referencia a la fecha de inicio escogida por el usuario para llevar a cabo el evento. |
| fecha\_finalizacion | TIMESTAMP | Hace referencia a la fecha escogida por el usuario en donde finalizara el evento. |

### Restricciones de seguridad e integridad

En primer lugar, se utilizan claves primarias y claves foráneas para establecer las relaciones entre las diferentes tablas. Por ejemplo, la tabla **Cuentas\_Usuario** tiene una clave primaria idcuentas\_usuarios, que se utiliza como clave foránea en la tabla **eventos\_me** para establecer la relación entre estas dos tablas.

En segundo lugar, se utilizan restricciones de integridad referencial para garantizar que no se puedan insertar o eliminar registros en las tablas si existe una relación referencial activa con otra tabla. Por ejemplo, si se intenta eliminar un registro en la tabla **eventos\_unidos** que tiene registros relacionados en la **eventos\_creados,** se generaría un error debido a la restricción de integridad referencial establecida.

En tercer lugar, se utilizan restricciones de integridad de valor para garantizar que los datos ingresados en las tablas cumplan con ciertas reglas o restricciones. Por ejemplo, la tabla **cuenta\_usuarios** tiene una restricción de integridad de valor en el campo **usuario** para asegurarse de que el valor ingresado no exceda de 20 caracteres.

## Diseño de la Arquitectura de Software

### Patrón de Diseño

El componente trasero **“backend”** de la aplicación se encuentra construido sobre un servidor NodeJS, escrito en JavaScript. Este servidor está diseñado para implementar el patrón REST (Representational State Transfer). En esencia, esto implica emplear consistentemente métodos HTTP (GET, POST, PUT, DELETE) para manipular los distintos recursos disponibles. Para desplegar este servicio, se requiere ejecutar el servidor NodeJS mediante el comando "node nombre\_servidor.js" en la ubicación donde se localiza el archivo correspondiente. El servicio se encuentra accesible a través de una ruta de despliegue, principalmente en "localhost:3000".

Por otro lado, el **Frontend**, es decir, la propia aplicación web, se ha confeccionado utilizando AngularJS, siguiendo rigurosamente el **patrón de diseño MVC (Modelo Vista Controlador)**.

**Modelo:** Este componente representa la estructura de datos de la aplicación. No debe confundirse con la base de datos que puede estar operando en segundo plano. El modelo consiste en el código que representa y gestiona los datos de la aplicación.

**Vista:** Se encarga de ofrecer la interfaz al usuario. Es decir, comprende todo lo que el usuario visualiza en pantalla y con lo que puede interactuar. Una aplicación normalmente contiene múltiples vistas, cada una representando distintas partes del modelo.

**Controlador:** Actúa como intermediario entre la vista y el modelo. Establece la comunicación entre ambos, asegurando que no haya interacciones directas entre el modelo y la vista. El controlador determina qué vista debe mostrarse en cada interacción entre el usuario y la aplicación. En un nivel más profundo, los controladores permiten, a través de programación, implementar la lógica de presentación en la aplicación AngularJS, gestionando el flujo de la parte del cliente.

El principal beneficio de emplear el **patrón MVC** radica en la capacidad para añadir o modificar aspectos de la aplicación con relativa facilidad. Este enfoque se ajusta perfectamente a una aplicación desarrollada en AngularJS, ya que utiliza la directiva ng-view para indicar en qué sección del HTML (index.html) se van a inyectar las vistas.

Este beneficio es posible gracias a la aplicación del principio de separación de preocupaciones, donde el **patrón MVC** descompone la aplicación en diversas partes, cada una encargada de manejar una preocupación específica

Otras de las grandes ventajas adicionales del uso del patrón de diseño Modelo Vista **Controlador (MVC)** es su contribución al proceso de pruebas (testing) durante el desarrollo de la aplicación. Al estar estructurada en partes claramente definidas, resulta considerablemente más sencillo identificar y abordar errores a lo largo del proceso de desarrollo.

### Diagrama de Componentes

Diagrama

Descripción generada automáticamente

### Diagrama de Despliegue

Diagrama

Descripción generada automáticamente

## Construcción e Implementación de Software

En el desarrollo de este proyecto, se han seleccionado dos tecnologías clave que desempeñan roles fundamentales tanto en el frontend como en el backend.

Hacemos uso de **ANGULAR**, comprendido como un marco de desarrollo de código abierto mantenido por Google, diseñado para facilitar la creación de aplicaciones web de una sola página (SPA). Algunas de las razones para elegir Angular incluyen:

* **Arquitectura MVVM (Model-View-ViewModel):** Angular sigue un patrón de diseño que facilita la separación de la lógica de presentación y la manipulación de datos.
* **Componentización:** La estructura de Angular se basa en componentes reutilizables, lo que simplifica la creación y el mantenimiento de la interfaz de usuario.
* **TypeScript:** Angular utiliza TypeScript como su lenguaje principal, proporcionando ventajas como el tipado estático y un desarrollo más robusto.
* **Inyección de Dependencias:** Facilita la gestión de dependencias y mejora la testabilidad del código.

Para la creación de la **API**, y por consiguiente el **CRUD**, apartado de lectura, actualización y eliminación de datos hacemos uso de **Node.js** es un entorno de ejecución de JavaScript del lado del servidor construido sobre el motor V8 de Google. Algunas razones para elegir Node.js para el desarrollo del backend son:

* **JavaScript en Ambos Lados**: Utilizando JavaScript tanto en el frontend como en el backend, se simplifica la coherencia del código y se mejora la productividad del equipo de desarrollo.
* **Event-Driven y No Bloqueante:** Node.js sigue un modelo de E/S no bloqueante, lo que mejora la eficiencia y la escalabilidad del servidor.
* **Express.js:** Se ha utilizado el marco Express.js para simplificar el desarrollo de la API RESTful, proporcionando herramientas robustas para la creación de rutas y la gestión de solicitudes HTTP.
* **Gran Comunidad y Ecosistema:** La amplia comunidad y el vasto ecosistema de módulos de Node.js facilitan la integración de diversas bibliotecas y herramientas.

### Integración del sistema de software

Para cumplir con los objetivos de costo, programación y rendimiento del sistema de software, se requiere una integración eficaz de distintos elementos o módulos de software que trabajen en conjunto de manera coordinada, es esencial considerar diferentes aspectos como:

* Servicios.
* Rendimiento.
* Middleware
* Herramientas de Monitoreo y Análisis.
* Base de datos.
* Eficiencia.
* Librerías y Frameworks.
* Accesibilidad.

### Verificación y validación del sistema de software

se llevará a cabo mediante un enfoque integral que abarque tanto los aspectos técnicos como los no técnicos, con especial atención a la seguridad y protección de la aplicación. Se implementarán diversas etapas de pruebas para garantizar la calidad y confiabilidad del sistema.

* Pruebas Unitarias

Las pruebas unitarias se centrarán en la verificación de unidades individuales de código, como componentes de Angular y funciones del backend en Node.js. Cada componente y función será sometido a pruebas exhaustivas para garantizar su correcto funcionamiento y para detectar posibles errores en una fase temprana del desarrollo.

* Pruebas de Integración

Se realizarán pruebas de integración para evaluar la interacción correcta entre los componentes del sistema, tanto en el frontend como en el backend. Esto asegurará que todas las partes del sistema se integren de manera efectiva y que la comunicación entre el cliente (Angular) y el servidor (Node.js) sea fluida y confiable.

* Pruebas de Funcionalidad

Las pruebas de funcionalidad se enfocarán en validar que todas las funciones y características especificadas en los requisitos del sistema estén implementadas correctamente. Se realizarán escenarios de prueba que abarquen diferentes casos de uso para garantizar que la aplicación cumpla con las expectativas del usuario final.

* Pruebas de Usuario Final

Se llevarán a cabo pruebas con usuarios finales reales o simulados para evaluar la usabilidad general de la aplicación. Esto incluirá la verificación de la interfaz de usuario, la facilidad de navegación y la respuesta del sistema a las acciones del usuario. Los comentarios de los usuarios finales se utilizarán para realizar mejoras finales antes del despliegue.

### Integración de software en Hardware

Gracias a su desarrollo basado en arquitectura web, este software posee la versatilidad de integrarse perfectamente con una amplia gama de hardware. Su accesibilidad a través de navegadores web permite que sea compatible con dispositivos móviles, sistemas operativos de escritorio y prácticamente cualquier tipo de navegador. Esta flexibilidad en su diseño facilita su utilización sin importar la plataforma, brindando a los usuarios la comodidad de acceder y utilizar todas sus funcionalidades desde cualquier lugar y dispositivo con conexión a internet.

### Transición de software y gestión de lanzamientos

En el actual flujo de trabajo, se lleva a cabo un riguroso control de versiones mediante copias de respaldo de códigos, junto con el uso de herramientas como Git y GitHub. Estas plataformas permiten visualizar las múltiples versiones anteriores del software, otorgando una esencial capacidad de recuperación de datos. La implementación de actualizaciones continuas para el sistema es una prioridad constante, asegurando que el servicio permanezca disponible durante y después del lanzamiento de nuevas versiones. Esta metodología garantiza tanto la estabilidad del sistema como la disponibilidad ininterrumpida del servicio, brindando a los usuarios una experiencia confiable y sin interrupciones.

### Mantenimiento del software

El mantenimiento de este software una vez salga al mercado se hará trimestralmente y abarca una serie de prácticas fundamentales para garantizar su óptimo funcionamiento. Esto implica la implementación de actualizaciones regulares tanto del software como de la seguridad, asegurando la protección constante mediante parches y certificados SSL, además de la realización de copias de seguridad periódicas para prevenir la pérdida de datos. Asimismo, se enfoca en la optimización del rendimiento a través de la mejora del código, la eficiencia en la carga y la supervisión continua para detectar y abordar cualquier fallo o caída en el sistema. El soporte técnico constante y la optimización de la experiencia del usuario son claves para brindar un aplicativo web confiable, seguro y de alta calidad.

### Gestión de defectos y problemas de software

La detección de problemas de software a veces puede estar relacionada con problemas de hardware, lo que puede impactar el funcionamiento del software.

La gestión de problemas de software implica la identificación mediante pruebas exhaustivas, revisiones de código y monitoreo en producción, seguido del registro en sistemas de seguimiento de problemas para su clasificación según gravedad y prioridad. Una vez identificados, los problemas se abordan a través de un enfoque sistemático: se inicia una investigación profunda para comprender la causa raíz y se procede con la corrección, donde el equipo de desarrollo trabaja en soluciones concretas. Estas soluciones pueden abarcar desde modificaciones en el código, parches de software o actualizaciones. Después, se llevan a cabo pruebas de validación para asegurar la efectividad de las correcciones y finalmente se implementan en el software. El seguimiento constante garantiza la verificación de la solución y el cierre del problema una vez resuelto. Posteriormente, se realiza un análisis detallado para identificar lecciones aprendidas y áreas de mejora en los procesos de desarrollo, permitiendo una mejora continua en la calidad del software y la prevención de problemas similares en el futuro.

aseguramiento de la calidad y la mejora de los procesos en el desarrollo de software, se consideran los siguientes aspectos fundamentales:

* Establecer estándares de calidad y definir métricas para evaluar el rendimiento del software.
* Realizar pruebas según el plan establecido, identificar y documentar defectos o problemas.
* Registrar, priorizar y dar seguimiento a los defectos detectados.
* Realizar revisiones periódicas del proceso de desarrollo para identificar áreas de mejora.
* Actualizar y ajustar las prácticas y estándares de calidad en función de las lecciones aprendidas.
* Utilizar técnicas como análisis FODA (Fortalezas, Oportunidades, Debilidades, Amenazas) o análisis causa raíz para identificar problemas subyacentes.
* Establecer métricas y KPIs (indicadores clave de rendimiento) para medir la efectividad de los cambios implementados.
* Recopilar datos y analizar tendencias para evaluar el impacto de las mejoras.
* Documentar los cambios implementados y comunicarlos eficazmente a todo el equipo para garantizar una comprensión y adopción adecuadas.