

UISIL
UNIVERSIDAD INTERNACIONAL SAN ISIDRO
LABRADOR

Geolocalización y monitorización en ganadería

JEFFERSON GERARDO MARIN GUEVARA

Alexis Amador Ilima

(ISB-32) PROGRAMACIÓN AVANZADA

Fecha:

04 de Abril 2024

Tabla de contenido

Capítulo 1: Introducción	3
Propósito y Estructura	3
Capítulo 2: Análisis y Diseño	3
Conceptos Fundamentales	3
Análisis de Procedimientos y Requerimientos	4
Paradigmas de Desarrollo y Lenguajes de Programación: ...	8
Capítulo 3: Ingeniería de Requerimientos	15
Definición y Recolección de Requerimientos:	15
Modelado con UML y Diccionarios de Datos:	16
Capítulo 4: Administración de Proyectos	19
Gestión del Alcance y Planificación:	18
Ejecución y Seguimiento:	22
Cierre y Gestión de Riesgos:	23
Liderazgo, Comunicación y Gestión de Equipos:	26
Gestión de Adquisiciones:	23
Capítulo 5: Diagramas y Herramientas de Modelado	29
Diagramas UML:	29
Diagramas de Base de Datos.....	33
Diccionarios de Datos	33
Herramientas de Modelado	34

Capítulo 1: Introducción

Propósito de la Guía: El proyecto se basa en una aplicación web de geolocalización y monitorización en la ganadería, ofrece primero un login se analiza muchos aspectos como el ganado el rancho y los usuarios y el propósito de la guía es proporcionar a los usuarios una visión general del proyecto, que se centra en el desarrollo de una aplicación web para la geolocalización y monitorización en la ganadería. La guía tiene como objetivo explicar los beneficios y funcionalidades de la aplicación, así como proporcionar instrucciones sobre cómo utilizarla de manera efectiva en el contexto de la ganadería. Además, la guía puede incluir información sobre las tecnologías utilizadas, los requisitos del sistema y las mejores prácticas para la implementación y el mantenimiento de la aplicación. En resumen, el propósito de la guía es servir como recurso integral para que los usuarios comprendan y aprovechen al máximo la aplicación de geolocalización y monitorización en la ganadería.

Estructura del Documento:

El documento se estructura en secciones que abarcan desde los fundamentos del análisis y diseño de sistemas hasta la gestión de proyectos y aspectos relacionados con el liderazgo y la comunicación. Se organiza en secciones que abarcan desde los fundamentos del análisis y diseño de sistemas hasta la gestión de proyectos y aspectos relacionados con la geolocalización y la monitorización.

Capítulo 2: Análisis y Diseño

Conceptos Fundamentales:

Definición de sistemas de información y su importancia.

EI LOGIN O INTERFAZ

Pose una interfaz de inicio de sección interactiva para el usuario ya sea para registrarse o iniciar a geograzin hub, también ya al ingresar el usuario podrá en su apartado de rancho ingresar su ganado y numerarlo con su respectivo nombre o raza y fecha de nacimiento y

peso actual para valorar su peso futuro y brindar una experiencia única a sus usuarios ganaderos, también es fácil de usar y muy accesible para una amplia cantidad de usuarios.

El Diseño Minimalista y Enfocado en la demanda del cliente para su utilidad:

El login y la interfaz son de un diseño simple que contiene elementos claves y algunos no tan claves los textos ordenados para mejor experiencia al usuario final. Lo que serian el inicio de sesión y registro se pueden leer de manera fácil y la tipografía que llegamos a elegir también ayuda.

Optimizaciones:

En el apartado de inicio de sesión o de entrada el acceso se facilita para que el usuario sea de ayuda con las indicaciones que hay.

REGISTRO

Registrarse o iniciar sesión:

En el registro y el inicio de sesión, esto permite a los usuarios a acceder y utilizar todas las funciones de la aplicación web, que son sumamente importantes.

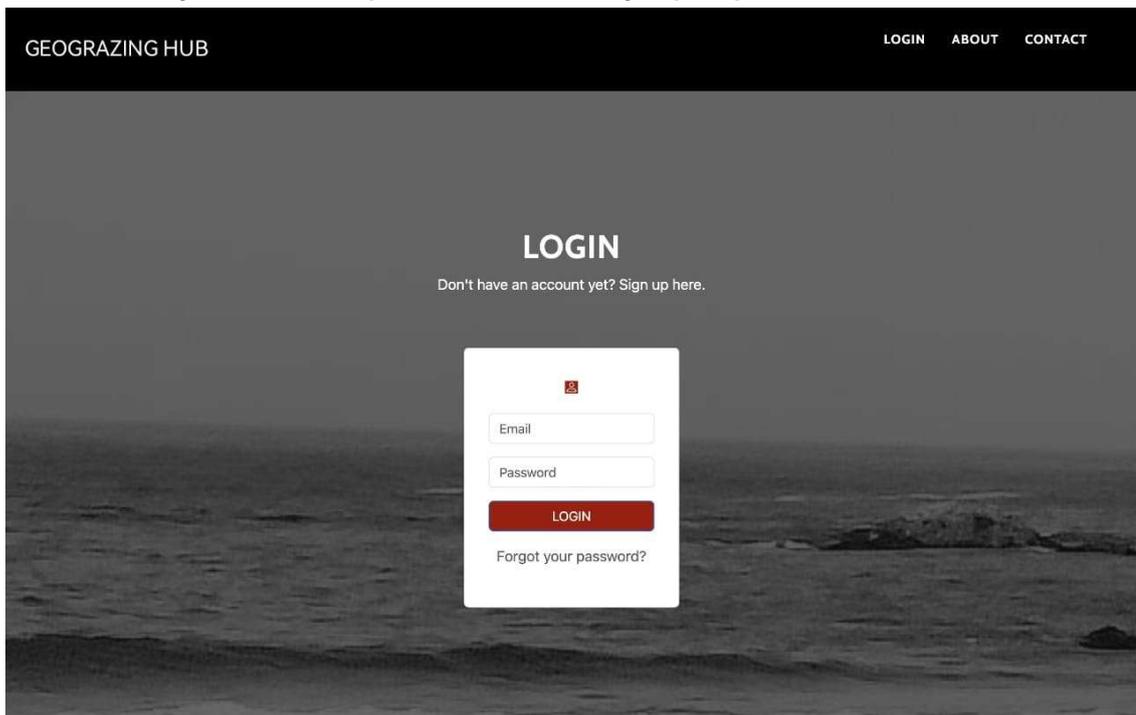
Registro: Requisitos:

Algunos requisitos para los usuarios que deben tener son su nombre, dirección de correo electrónico y contraseña segura para dar un buen registro.

Estos datos permiten a cada usuario tener su cuenta única que nadie más podrá tener y además personalizable que les da acceso a todas las funcionalidades de la plataforma.

Flujo de acción y verificación:

Después de haber concluido con el registro, el usuario se dirige a la página de inicio, donde puede ir registrando al ganado de manera individual, y da una experiencia fluida y apta para el usuario.

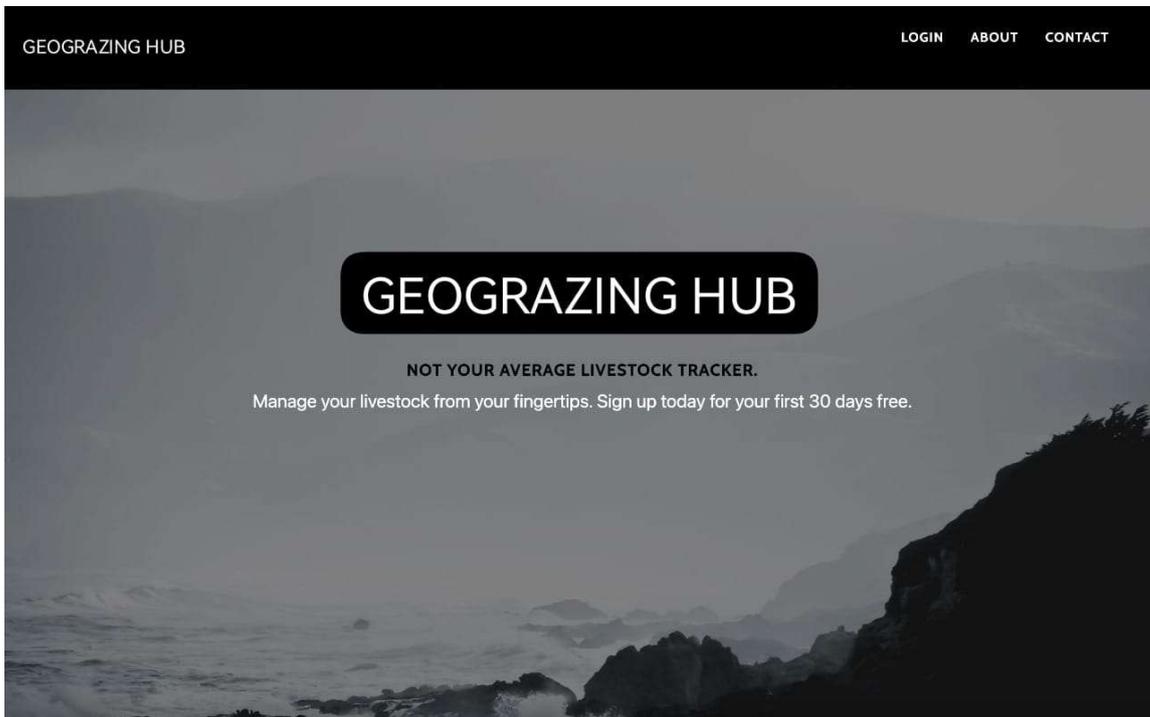


INICIO DE SESION

El inicio de sesión de usuario permite ingresar un usuario el cual ya se registro previamente de manera rápida.

INICIO

El inicio es la página principal, que principalmente es informativa donde se ofrece al cliente toda la información o usuario, se mantiene las normas de diseño, así como una interfaz interactiva y fácil de comprender para los usuarios



ABOUT

Básicamente el punto fuerte de mi plataforma es que los usuarios puedan tener mejor control en su apartado en la ganadería y el ganado en específico.

CATEGORIAS

Los usuarios tienen una manera organizada de tener también especializada de ver y mantener visualización de su ganado en sus áreas marcadas.

Principios de ingeniería del software y sus componentes.

LOGIN

REGISTRO

En el Proceso de registro comienza con la gestión de datos eficiente utilizando una base de datos MySQL. Con la tabla llamada "usuarios" en esta base de datos con columnas específicas para almacenar información importante como nombre, correo electrónico y contraseña. Para administrar los perfiles de usuario, este diseño de base de datos proporciona una estructura organizada y escalable.

INICIO

La creación de una página de inicio generalmente implica el uso de lenguajes estándar de la web, como HTML, CSS y JavaScript, que forman la base de su estructura y funcionalidad. Con estas tecnologías, se diseña un entorno atractivo y efectivo para los usuarios. El HTML proporciona la estructura jerárquica fundamental de la página, mientras que el CSS se encarga de dar estilo y formato al contenido, asegurando una presentación visualmente coherente y atractiva.

Además de la presentación, se emplean capacidades interactivas gracias a JavaScript, lo que contribuye a mejorar la usabilidad y la experiencia del usuario. Para agilizar el desarrollo y garantizar la compatibilidad con diversos dispositivos, se recurre a bibliotecas externas como Bootstrap. Estas bibliotecas ofrecen una serie de componentes predefinidos y estilos CSS que simplifican la tarea de crear diseños atractivos y responsivos, lo que permite adaptar la página a una amplia gama de dispositivos y tamaños de pantalla.

Exploración de paradigmas de desarrollo y selección de lenguajes de programación.

METODOLOGIA SIX SIGMA

En este proyecto se usa para la gestión de calidad. Generalmente se la describe como una filosofía más que como una metodología tradicional. A menudo se la combina con una metodología Lean o un marco ágil, también conocido como Six Sigma Lean y Six Sigma ágil. El objetivo principal de Six Sigma es la mejora continua de los procesos y la eliminación de fallas. Esto se logra a través de mejoras continuas por parte de expertos en la materia para definir, apoyar y controlar los procesos.

Y en la selección de lenguajes de programación se utilizan Java Script, HTML, CSS, handlebars, y fontawesome y también package.json.

La metodología Six Sigma es un enfoque sistemático y altamente estructurado para mejorar los procesos empresariales y reducir la variabilidad en la producción y entrega de productos y servicios. Se basa en el principio de buscar y eliminar defectos o errores en los procesos mediante el uso de datos y análisis estadístico, con el objetivo de alcanzar un nivel de calidad extremadamente alto.

Métodos y herramientas para el análisis de procedimientos y requerimientos.

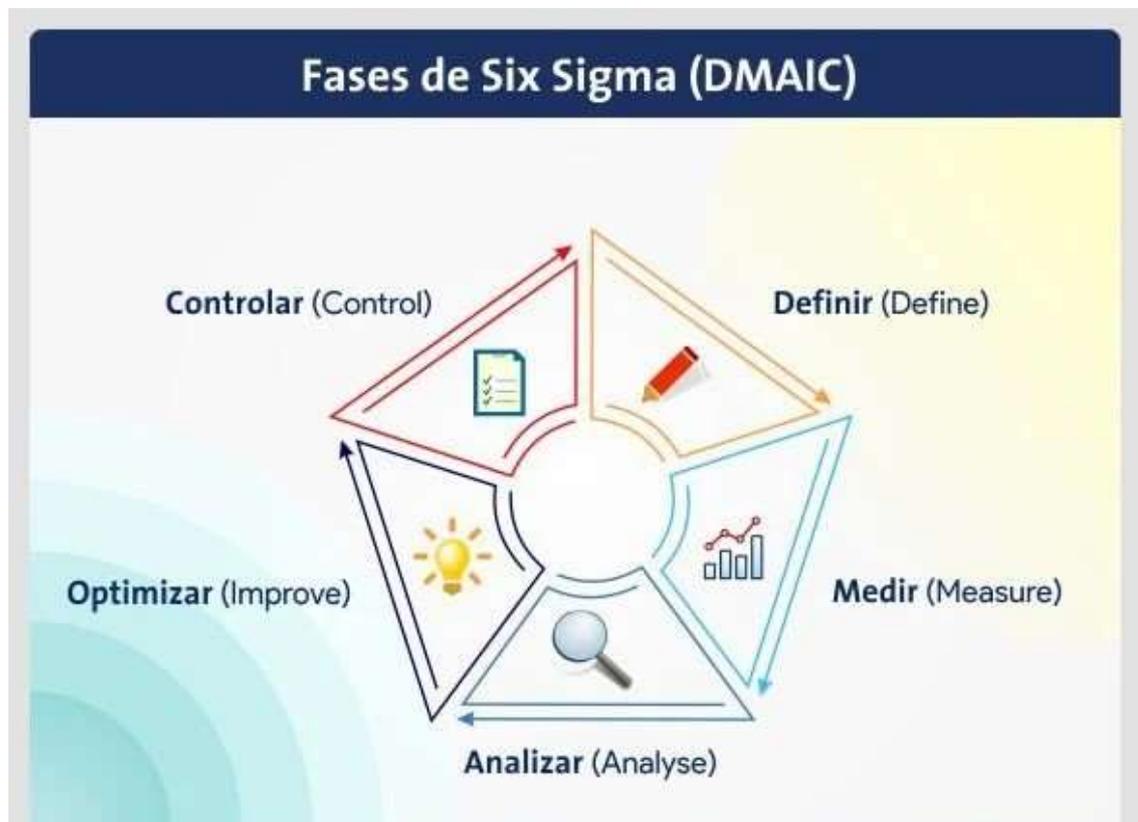
El éxito de cualquier proyecto de desarrollo de software depende en gran medida de cómo se analizan los procedimientos y requerimientos del sistema con precisión y exhaustividad. Para lograr esto, se utilizan una variedad de técnicas y recursos que permiten una evaluación completa y precisa de las necesidades del proyecto. Luego se examinan las técnicas y herramientas específicas utilizadas:

Metodología six sigma:

Para este proyecto la metodología Six Sigma es un enfoque sistemático y altamente estructurado para mejorar los procesos empresariales y reducir la variabilidad en la producción y entrega de productos y servicios. Se basa en el principio de buscar y eliminar defectos o errores en los procesos mediante el uso de datos y análisis

estadístico, con el objetivo de alcanzar un nivel de calidad extremadamente alto.

Six Sigma se enfoca en identificar y eliminar causas de variabilidad que pueden conducir a defectos en los productos o servicios ofrecidos por una organización. Utiliza un conjunto de herramientas y técnicas específicas, como el DMAIC (Definir, Medir, Analizar, Mejorar, Controlar), que proporciona un marco estructurado para la mejora continua.



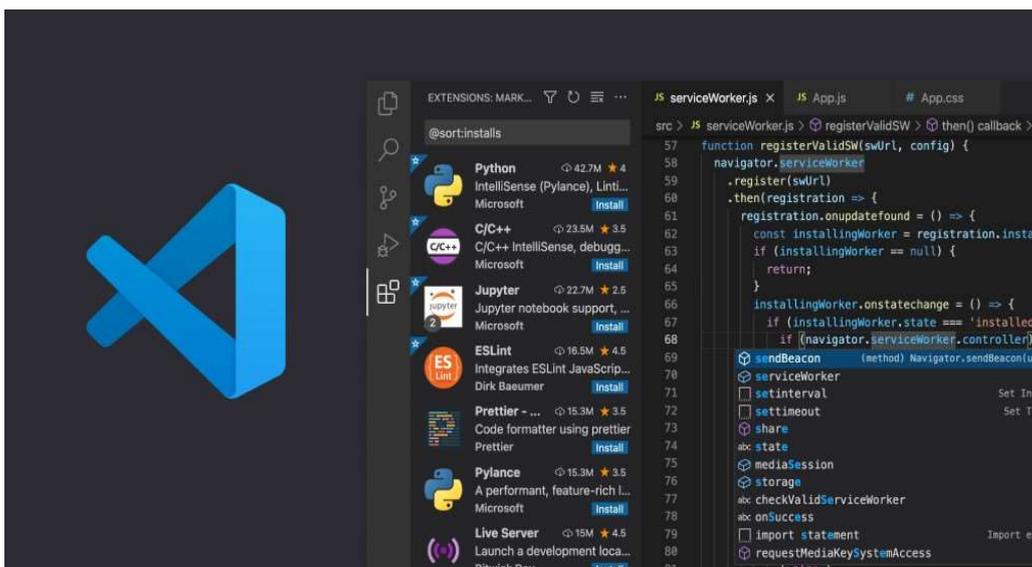
Entorno de Desarrollo Visual Studio Code:

Para este proyecto, se seleccionó Visual Studio Code como el entorno de desarrollo integrado (IDE) principal debido a su excepcional potencia y flexibilidad. Este IDE ofrece una amplia gama de características que simplifican la gestión de proyectos, la depuración y la escritura de código. Su interfaz intuitiva y personalizable proporciona un entorno de trabajo eficiente para desarrolladores de diversos niveles de experiencia.

Además de su robusto conjunto de herramientas, Visual Studio Code se destaca por su vasta comunidad de usuarios y una extensa biblioteca de extensiones. Estas extensiones permiten personalizar aún más el entorno de desarrollo para adaptarse a las necesidades específicas del proyecto, desde la integración con sistemas de control de versiones hasta la creación de flujos de trabajo automatizados.

Al aprovechar las capacidades de Visual Studio Code, el proceso de desarrollo se agiliza significativamente, lo que conduce a una mayor productividad del equipo. La interfaz intuitiva y las características avanzadas de este IDE permiten a los desarrolladores concentrarse en la creación de código de alta calidad y en la resolución de problemas de manera eficiente.

En resumen, Visual Studio Code emerge como una opción ideal para proyectos de desarrollo de software gracias a su potencia, flexibilidad y soporte comunitario. Su capacidad para simplificar el proceso de desarrollo y mejorar la productividad del equipo contribuye a una entrega rápida y efectiva del proyecto.



Gestión de bases de datos con XAMPP, MySQL: confiabilidad y eficacia

XAMPP es un entorno de desarrollo local para la gestión de bases de datos

que incluye un servidor web Apache, MySQL y PHP. MySQL es el sistema de

gestión de bases de datos relacionales preferido porque es robusto, confiable

y se utiliza ampliamente en la industria. MySQL proporciona un entorno

estable y seguro para almacenar y gestionar los datos del sistema, asegurando un rendimiento óptimo y una alta disponibilidad.



Factibilidad del Proyecto: Utilización Eficiente de Recursos:

El sistema propuesto es viable y factible, según la evaluación de factibilidad del proyecto. Esto se debe a la utilización eficiente de los recursos, donde los componentes se han seleccionado cuidadosamente para cumplir con los requisitos del proyecto. El uso de MySQL para la gestión de la base de datos y la lógica del servidor, respectivamente, garantiza una solución sencilla pero efectiva. Estas

tecnologías son ampliamente utilizadas y cuentan con una amplia base de soporte y recursos disponibles, lo que simplifica el desarrollo e implementación de sistemas.

Roles y responsabilidades del analista, diseñador y programador de sistemas.

Analista de Sistemas:

Responsable de comprender y evaluar los requisitos del proyecto, incluidas las especificaciones técnicas y las necesidades del cliente. Se encarga de identificar oportunidades y problemas para el desarrollo de software.

crea documentación detallada sobre los requisitos del sistema, que incluye diagramas de flujo, casos de uso y otras especificaciones técnicas.

Colabora estrechamente con el cliente y otros miembros del equipo para garantizar que se comprendan y se implementen de manera efectiva los requisitos del sistema.

Diseñador de Sistemas:

Responsable de comprender y evaluar los requisitos del proyecto, incluidas las especificaciones técnicas y las necesidades del cliente. Se encarga de identificar oportunidades y problemas para el desarrollo de software.

Crea documentación detallada sobre los requisitos del sistema, que incluye diagramas de flujo, casos de uso y otras especificaciones técnicas. Colabora estrechamente con el cliente y otros miembros del equipo para garantizar que se comprendan y se implementen de manera efectiva los requisitos del sistema.

Programador de Sistemas:

Responsable de la codificación, la programación y el desarrollo de software, así como de la implementación técnica del sistema.

desarrolla el código fuente del sistema de acuerdo con las especificaciones y requisitos establecidos por el diseñador y el analista.

Se encarga de la integración de varios componentes del sistema y de que funcionen juntos.

Realiza pruebas unitarias e integradoras para garantizar que el sistema funcione correctamente y corregir errores o problemas de funcionamiento.

Enfoque del Desarrollador y Administrador del Proyecto:

El profesional es responsable de realizar una variedad de tareas como desarrollador y administrador de proyectos. Es esencialmente un programador web que puede diseñar front-end y desarrollar back-end.

Debe ser capaz de aplicar conceptos de diseño para crear una interfaz de usuario

(front-end) intuitiva y atractiva.

Además, debe tener experiencia en el desarrollo back-end, como la manipulación de las bases de datos y la implementación de lógica comercial.

Además, se espera que el desarrollador y administrador del proyecto tenga la capacidad de analizar y reestructurar la página de manera eficiente y comprensible para optimizar la experiencia del usuario y el rendimiento del sistema.



Paradigmas de Desarrollo y Lenguajes de Programación:

Discusión sobre los diferentes paradigmas de desarrollo.

El enfoque del paradigma modelo de cascada, también conocido simplemente como "modelo cascada", es un método de desarrollo de software que sigue una secuencia lineal y ordenada de fases. En este enfoque, cada fase debe completarse antes de pasar a la siguiente, y los cambios realizados en una fase posterior pueden resultar costosos o difíciles de implementar.

Objetivos y tipos de lenguajes de programación.

El objetivo principal del desarrollo web es utilizar los lenguajes de programación apropiados para crear páginas y aplicaciones web de manera eficiente y lógicamente correcta. Para lograr este objetivo, se utilizan una variedad de tipos de lenguajes, cada uno de los cuales tiene sus propias características y funciones. Estos son los tipos de lenguajes utilizados en el desarrollo web y sus objetivos:

Lenguajes:

HTML

Es el lenguaje estándar utilizado para crear y diseñar páginas web. Consiste en una serie de elementos (etiquetas) que definen la estructura y el contenido de una página web. Estas etiquetas permiten al navegador web interpretar cómo se debe mostrar la información en la pantalla del usuario.

CSS

CSS, acrónimo de Cascading Style Sheets (Hojas de Estilo en Cascada), es un lenguaje utilizado para describir la presentación y el estilo visual de un documento HTML. Con CSS, se pueden controlar aspectos como el color, la tipografía, el espaciado, el diseño y otros aspectos de la apariencia de una página web. Esto permite separar el contenido de la presentación, lo que facilita el mantenimiento y la actualización de los sitios web

JAVA SCRIPT

JavaScript es un lenguaje de programación de alto nivel, interpretado y orientado a objetos que se utiliza principalmente para agregar interactividad y dinamismo a las páginas web. Además de su uso en el desarrollo web, JavaScript también se utiliza en el desarrollo de aplicaciones web, aplicaciones de servidor, aplicaciones móviles y mucho más.

MySQL

MySQL es un sistema de gestión de bases de datos relacional (RDBMS, por sus siglas en inglés) ampliamente utilizado en el desarrollo de aplicaciones web y empresariales. Permite almacenar, organizar y recuperar datos de manera eficiente y segura.

CAPÍTULO 3: INGENIERÍA DE REQUERIMIENTOS

Definición y Recolección de Requerimientos:

Importancia de los requerimientos y las dificultades asociadas.

Claridad y comprensión: Los requisitos establecen una base clara y fácilmente comprensible para el proyecto. Explican los objetivos, las funcionalidades necesarias y los criterios de éxito, lo que ayuda a alinear las expectativas del cliente, el equipo de desarrollo y otros interesados en el proyecto. A lo largo del ciclo de vida del proyecto, estos requisitos actúan como guía para la toma de decisiones, facilitando la priorización de tareas, la identificación de riesgos y la resolución de conflictos. Además, sirven como puntos de referencia para evaluar el progreso del proyecto y garantizar que se avance en la dirección correcta.

Garantía de Calidad: Los requisitos establecen los estándares de calidad que deben cumplirse para que un proyecto sea considerado exitoso. Esto abarca aspectos como la usabilidad, la seguridad, la compatibilidad con diversos dispositivos y navegadores, así como la funcionalidad y el rendimiento del software. El producto final cumple con las expectativas y necesidades del cliente al satisfacer los requisitos establecidos.

Facilita la Comunicación: Los requerimientos actúan como un punto de referencia común para todos los involucrados en el proyecto, facilitando la comunicación entre el cliente, el equipo de desarrollo y otros interesados. Proporcionan un marco común para discutir y negociar cambios, ajustes y mejoras en el proyecto.

Aunque son importantes, los requisitos pueden plantear algunos problemas:

Cambios Constantes: Los requisitos pueden cambiar a lo largo del proyecto debido a modificaciones en las necesidades del cliente, nuevos requisitos legales o tecnológicos, o una mejor comprensión del problema a resolver. El equipo de desarrollo debe adaptarse para manejar estos cambios de manera efectiva.

Ambigüedad o falta de detalle: Los requisitos pueden ser ambiguos o carecer de suficiente detalle, lo que dificulta su interpretación e

implementación. Esto puede ocasionar confusiones, retrasos en el progreso y resultados insatisfactorios para el cliente.

Conflictos entre Requerimientos: En ocasiones, los requisitos pueden entrar en conflicto entre sí, lo que hace más difícil cumplirlos simultáneamente. Es necesario mantener el equilibrio y trabajar en conjunto para encontrar soluciones que satisfagan las necesidades de todas las partes involucradas en la resolución de estos conflictos

MÍNIMO:

Requiere un procesador y un sistema operativo de 64 bits

SO *: Windows 7/8/10 (64-bit OS required)

Procesador: Intel Core i3-2300 2.0 GHz/AMD Phenom II X4 945
2.0 GHz or equivalent

Memoria: 2 GB de RAM

Almacenamiento:3GB

RECOMENDADO:

Requiere un procesador y un sistema operativo de 64 bits

SO *: Windows 7/8/10 (64-bit OS required)

Procesador: Intel Core i5 4790 3.6 GHz/AMD FX-9590 4.7 GHz
or equivalent

Memoria: 4 GB de RAM

Gráficos: NVIDIA GTX 1800 3GB/AMD Radeon R 580 4GB or
equivalent

Almacenamiento: 4 GB de espacio disponible

NAVEGADORES:

Google Chrome de 64 bits (versión 96 o posterior)

Mozilla Firefox de 64 bits (versión 100 o posterior)

Nuevo Microsoft Edge de 64 bits (basado en Chromium, versión 96 o posterior) Safari de 64 bits (versión 15.2 o posterior)

Métodos y técnicas para la recolección efectiva de requerimientos.

Los requerimientos son una parte esencial en cualquier proyecto, particularmente en el desarrollo de aplicaciones web, ya que establecen los estándares y especificaciones que orientan el proceso desde el diseño hasta la implementación. Su importancia radica en diversos aspectos fundamentales:

Claridad y comprensión: Los requerimientos proporcionan una base clara y comprensible para el proyecto, explicando objetivos, funcionalidades necesarias y criterios de éxito. Esto alinea las expectativas del cliente, el equipo de desarrollo y otros interesados.

Guía para la toma de decisiones: A lo largo del ciclo de vida del proyecto, los requerimientos sirven como guía para priorizar tareas, identificar riesgos y resolver conflictos. También son puntos de referencia para evaluar el progreso y asegurar el avance correcto.

Garantía de calidad: Establecen estándares de calidad, incluyendo usabilidad, seguridad, compatibilidad y rendimiento del software. Se asegura que el producto final cumpla con las expectativas y necesidades del cliente.

Facilita la comunicación: Los requerimientos proporcionan un punto de referencia común para todos los involucrados, facilitando la comunicación entre cliente, equipo de desarrollo y otros. Esto crea un marco para discutir cambios, ajustes y mejoras en el proyecto.

A pesar de su importancia, los requerimientos pueden presentar

algunos desafíos, tales como:

Cambios constantes: Pueden cambiar a lo largo del proyecto debido a nuevas necesidades del cliente, requisitos legales o tecnológicos, o mejor comprensión del problema. El equipo debe adaptarse para gestionar estos cambios.

Ambigüedad o falta de detalle: Los requerimientos pueden ser ambiguos o carecer de suficiente detalle, dificultando su interpretación e implementación y causando confusión y retrasos.

Conflictos entre requerimientos: En ocasiones, los requerimientos pueden entrar en conflicto, lo que dificulta cumplirlos simultáneamente. Es necesario equilibrar y encontrar soluciones que satisfagan a todas las partes involucradas.

Utilización de UML para modelar casos de uso y diagramas.

Existen múltiples razones por las cuales el empleo de diccionarios de datos y la utilización de UML (Unified Modeling Language) resultan fundamentales en el proceso de desarrollo de software:

Claridad y Comunicación Reforzadas: UML provee un lenguaje visual estandarizado que posibilita a los equipos de desarrollo transmitir los requisitos del sistema de manera clara y efectiva. Los diagramas UML, como los de casos de uso, resultan comprensibles para diversos interesados en el proyecto, incluyendo analistas, desarrolladores y clientes.

Visualización Mejorada de Requisitos: El modelado UML facilita la visualización organizada y estructurada de los requisitos del sistema. Por ejemplo, los diagramas de casos de uso exhiben la interacción de los usuarios con el sistema y las funcionalidades necesarias para satisfacer sus requerimientos. Esta representación simplifica la identificación y comprensión de los requisitos de forma más clara y

detallada.

Facilita Análisis y Diseño: UML simplifica el análisis y diseño de sistemas al permitir que los equipos de desarrollo visualicen y comprendan previamente la estructura, comportamiento e interacciones del sistema. Los diagramas UML ayudan a identificar problemas o inconsistencias en los requisitos antes de su implementación en el código, lo que contribuye a reducir los costos y riesgos asociados con cambios en etapas posteriores del desarrollo.

Documentación y Mantenimiento Eficientes: UML proporciona un método estructurado para documentar los requisitos del sistema, facilitando su comprensión y mantenimiento a lo largo del tiempo. Los desarrolladores pueden emplear diagramas UML como guías visuales para entender la estructura del sistema y el funcionamiento de sus distintos componentes.

Promueve la Estandarización: UML es un lenguaje estándar ampliamente utilizado en la industria del software. Su adopción para documentar y comunicar requisitos del sistema estandariza el proceso, lo que simplifica la colaboración entre diferentes equipos y organizaciones.

CAPÍTULO 4: ADMINISTRACIÓN DE PROYECTOS

Gestión del Alcance y Planificación:

Desarrollo de un plan integral y técnicas de estimación.

La planificación y la gestión del alcance son elementos cruciales en el desarrollo de cualquier proyecto, y en esta instancia, se han implementado medidas sustanciales para asegurar el éxito desde sus cimientos. Se ha elaborado un plan exhaustivo que aborda los principios fundamentales del proyecto y un cronograma detallado de ejecución, empleando la metodología seleccionada.

Con el objetivo de proporcionar una experiencia de usuario óptima, el desarrollo de la interfaz de usuario constituye una pieza clave en el proceso de planificación. La dedicación de aproximadamente tres semanas a esta tarea refleja nuestro compromiso con la excelencia y el cuidado meticuloso en el diseño de la aplicación. Además, se han establecido plazos concretos para garantizar que se cumplan los estándares de calidad y que la interfaz resulte intuitiva y fácil de usar.

El enfoque original ha sufrido una modificación significativa en cuanto al desarrollo de la base de datos. Se ha decidido utilizar PHP en lugar de una API para simplificar la conectividad. Esta elección se fundamenta en consideraciones de eficiencia y economía, al descubrirse que la implementación basada en PHP es más sencilla y económica.

Se ha revaluado la necesidad de bases de datos adicionales y se ha concluido que, para el alcance actual del proyecto, solo se requiere el módulo de acceso. Esta reducción del alcance no solo simplifica el desarrollo y reduce los costos, sino que también concuerda con la estrategia general del proyecto, que prioriza la funcionalidad esencial y mantiene un enfoque ágil y eficaz.

Control del alcance y ejemplos prácticos.

El control del alcance del proyecto es de suma importancia para asegurar su ejecución dentro de los límites preestablecidos y alcanzar los objetivos planteados. Esto requiere una supervisión continua del progreso del proyecto y la implementación de medidas para prevenir desviaciones significativas del alcance originalmente definido. Aquí se presentan algunos ejemplos prácticos de cómo llevar a cabo el control de alcance:

Establecimiento de límites precisos: Es esencial fijar límites claros y definir con precisión el alcance del proyecto desde el principio. Esto implica identificar qué aspectos están incluidos y cuáles están excluidos del proyecto. Por ejemplo, en el caso de un proyecto a nivel nacional, es crucial especificar que su alcance se restringe al territorio

nacional y no se extiende a otros países.

Monitoreo constante del alcance: Se debe realizar un seguimiento continuo del alcance del proyecto para identificar cualquier desviación temprana y tomar medidas correctivas de manera oportuna. Esto implica mantener una comunicación fluida entre todas las partes interesadas y utilizar herramientas adecuadas para el seguimiento del progreso.

Reevaluación periódica del alcance: A medida que avanza el proyecto, es importante realizar evaluaciones periódicas del alcance para asegurarse de que sigue siendo relevante y realista en función de las condiciones cambiantes del entorno. Esto puede implicar ajustes en el alcance para adaptarse a nuevas necesidades o circunstancias emergentes.

Adaptación proactiva ante cambios: Ante cualquier cambio en los requisitos o condiciones del proyecto, es fundamental tomar medidas proactivas para ajustar el alcance en consecuencia. Esto requiere una evaluación cuidadosa de las implicaciones de los cambios y una comunicación efectiva con todas las partes involucradas.

Ejecución y Seguimiento:

Coordinación, gestión de actividades y resolución de problemas.

La coordinación, la gestión de actividades y la resolución de problemas son fundamentales para asegurar el éxito de cualquier proyecto. Inevitablemente, surgirán dificultades y obstáculos durante su desarrollo, a pesar de una planificación minuciosa. Aquí presentamos algunos métodos útiles para afrontar estos desafíos:

Planificación robusta: Establecer una base sólida requiere una planificación detallada y exhaustiva. Esto implica identificar claramente los objetivos del proyecto, definir las tareas y actividades necesarias para alcanzarlos, asignar recursos y establecer plazos realistas. Una planificación adecuada crea un marco claro para la

coordinación y gestión de actividades.

Comunicación efectiva: La coordinación efectiva entre los miembros del equipo y las partes interesadas exige una comunicación abierta y transparente. Es esencial establecer canales claros de comunicación y programar reuniones periódicas para mantener a todos informados sobre el progreso del proyecto, identificar posibles problemas y discutir soluciones.

Seguimiento y monitoreo: Es fundamental realizar un seguimiento regular del progreso del proyecto para identificar cualquier problema o desviación a medida que avanza. Esto implica vigilar el avance de las actividades, el cumplimiento de los plazos y la identificación de riesgos potenciales. El uso de software de gestión de proyectos y herramientas de seguimiento puede simplificar este proceso.

Gestión proactiva de problemas: Cuando surjan problemas o desafíos, es crucial abordarlos de manera proactiva y encontrar soluciones rápidas. Esto puede implicar la implementación de medidas correctivas, la reasignación de tareas, la revisión de plazos o la asignación de recursos adicionales. Contar con un equipo preparado para identificar y abordar problemas de manera efectiva es esencial.

Flexibilidad y adaptabilidad: Mantener una actitud flexible y estar preparado para adaptarse a los cambios durante el avance del proyecto es importante. Esto puede incluir la revisión y actualización periódica del plan del proyecto para adaptarlo a nuevas necesidades o circunstancias.

Monitorización y control del progreso.

Flexibilidad y adaptabilidad son elementos clave en la gestión de proyectos:

Es esencial mantener una mentalidad abierta y estar preparado para ajustar el curso a medida que el proyecto evoluciona. Esto implica

revisar y actualizar regularmente el plan del proyecto para adecuarse a nuevas necesidades o circunstancias inesperadas.

Establecimiento de hitos y puntos de control:

Durante la fase de planificación, se identifican hitos y puntos de control cruciales para evaluar el avance del proyecto. Estos hitos representan logros significativos, como la finalización de un diseño o la implementación de una funcionalidad clave. Los puntos de control, por otro lado, son momentos específicos en el tiempo en los que se lleva a cabo una evaluación oficial del progreso.

Realización de pruebas y validaciones:

Después de finalizar cada fase, se llevan a cabo pruebas exhaustivas para garantizar que los entregables cumplan con los requisitos establecidos. Esto puede incluir pruebas de funcionalidad, rendimiento y usabilidad.

La metodología Six Sigma hace hincapié en la importancia de estas pruebas para detectar y corregir errores tempranamente.

Análisis de desviaciones: Durante el monitoreo del proyecto, se comparan los resultados actuales con los objetivos establecidos para identificar desviaciones significativas. Si se detectan desviaciones, se realiza un análisis para determinar las causas subyacentes y se toman medidas correctivas según sea necesario. Por ejemplo, si una fase del proyecto se retrasa, pueden asignarse más recursos o ajustarse los plazos para recuperar el tiempo perdido.

Comunicación regular: La transparencia y la comunicación abierta son fundamentales para mantener a todos los miembros del equipo y partes interesadas informados sobre el progreso del proyecto. Se deben realizar reuniones periódicas para revisar el estado del proyecto, abordar cualquier problema que surja y planificar próximos pasos. Esta comunicación constante ayuda a mantener el compromiso de todos con los objetivos del proyecto.

Ajuste del plan según sea necesario: A medida que el proyecto avanza, puede ser necesario ajustar el plan original para adaptarse a cambios en los requisitos, recursos disponibles o circunstancias externas. Siempre y cuando se mantenga el enfoque en los objetivos

del proyecto, la metodología Six Sigma permite la flexibilidad suficiente para realizar ajustes según sea necesario.

Cierre y Gestión de Riesgos:

Finalización formal, entrega de productos y lecciones aprendidas.

La finalización formal de un proyecto marca un hito significativo que indica el exitoso cumplimiento de todas las actividades y entregables planificados.

Evaluación de logros de objetivos: Antes de concluir oficialmente, se realiza una evaluación exhaustiva para determinar si se han alcanzado los objetivos y requisitos del proyecto. Cada entregable se somete a revisión y los resultados se contrastan con las expectativas del cliente y los criterios de éxito previamente establecidos.

Cierre de actividades: Se llevan a cabo todas las acciones necesarias para formalizar el cierre del proyecto. Esto puede implicar la desmovilización del equipo, la finalización de contratos con proveedores o subcontratistas externos, el cese de los recursos utilizados y la recopilación de todos los activos del proyecto.

Documentación de lecciones aprendidas: Se realiza una revisión retrospectiva del proyecto para identificar lecciones aprendidas y mejores prácticas que puedan aplicarse en proyectos futuros. Se documentan los logros, obstáculos y oportunidades de mejora con el fin de optimizar continuamente los procesos y la ejecución de proyectos en el futuro.

Mantenimiento y soporte continuo: Aunque el proyecto haya concluido,

es importante considerar que algunos entregables pueden requerir mantenimiento o soporte posterior a la implementación. Esto podría implicar actualizaciones de software, correcciones de errores, capacitación adicional para usuarios o cualquier otro tipo de asistencia técnica necesaria para garantizar la funcionalidad y satisfacción del cliente a largo plazo.

Identificación, evaluación y mitigación de riesgos.

La identificación, evaluación y mitigación de riesgos son elementos fundamentales para garantizar el éxito en la gestión de proyectos.

Identificación de riesgos: En cada etapa del proyecto, se llevan a cabo exhaustivas pruebas utilizando la metodología en forma de sigma antes de avanzar a la siguiente fase. Este proceso de evaluación permite detectar posibles problemas, deficiencias o riesgos que puedan surgir durante el desarrollo del proyecto, tales como errores de software, incompatibilidades, limitaciones de recursos o cambios en los requisitos del cliente.

Evaluación de riesgos: Una vez identificados, los riesgos se evalúan considerando tanto su probabilidad de ocurrencia como el impacto que podrían tener en el proyecto. Cada riesgo recibe una calificación en función de su gravedad y de la capacidad del proyecto para mitigarlo. Esto facilita la priorización de los riesgos, permitiendo focalizarse en los más relevantes.

Seguimiento y monitoreo continuo: Se lleva a cabo un seguimiento constante de los riesgos identificados a lo largo de todo el proyecto, así como de las medidas de mitigación implementadas. Esto posibilita detectar cualquier cambio en la naturaleza o en el impacto de los riesgos y, en consecuencia, ajustar las estrategias de mitigación. Además, conforme avanza el proyecto, pueden surgir nuevos riesgos, por lo que resulta crucial mantener una vigilancia constante.

Comunicación y gestión de riesgos: Es crucial que todas las partes involucradas en el proceso de gestión de riesgos mantengan una

comunicación efectiva. Se deben establecer canales claros de comunicación para informar sobre los riesgos identificados, las estrategias propuestas para mitigarlos y cualquier cambio en el panorama de riesgos. Esto promueve la colaboración y la transparencia en la gestión de riesgos.

Liderazgo, Comunicación y Gestión de Equipos:

Gestión de equipos y liderazgo.

A pesar de llevar a cabo el proyecto de manera individual, el liderazgo, la comunicación y la gestión de relaciones siguen siendo elementos de gran importancia.

Liderazgo personal: Aunque hayas trabajado en solitario en el proyecto, tu papel como líder fue esencial. Tuviste la responsabilidad de establecer metas claras, tomar decisiones importantes, mantener el enfoque en los objetivos del proyecto y administrar de manera eficiente los recursos disponibles.

Comunicación efectiva: A pesar de la falta de un equipo formal, mantener una comunicación efectiva con el profesor y el supervisor del proyecto fue fundamental. Mantuviste una comunicación constante con otros para discutir el avance del proyecto, resolver cualquier duda o problema, y recibir retroalimentación sobre tu trabajo. Esta comunicación permitió mantener alineados los objetivos del proyecto y garantizar el cumplimiento de los estándares de calidad esperados.

Gestión de relaciones: Es probable que hayas interactuado no solo con el profesor y el supervisor, sino también con otros recursos o partes interesadas en el proyecto, como proveedores externos o expertos en el tema. Para asegurar la colaboración y el apoyo necesarios para el éxito del proyecto, también fue crucial gestionar

estas relaciones de manera efectiva.

Autogestión y motivación: Mantener altos niveles de motivación y autodisciplina fue fundamental como líder del proyecto para cumplir con los plazos y objetivos establecidos. Esto implicó un compromiso personal con el éxito y la calidad del proyecto, así como una gestión eficaz del tiempo.

Gestión de Adquisiciones:

Proceso de adquisición eficiente de bienes y servicios.

Proceso eficaz de adquisición de bienes y servicios:

Para garantizar una adquisición óptima de bienes y servicios, se requiere una gestión eficiente y una sólida autodisciplina como líder del proyecto. Mantener altos niveles de motivación resulta crucial para cumplir con los plazos y objetivos establecidos, lo que implica un compromiso personal con el éxito y la calidad del proyecto, así como una gestión efectiva del tiempo.

Análisis económico detallado del proyecto:

Realizar un análisis económico exhaustivo del proyecto resulta esencial para determinar el presupuesto disponible y establecer límites claros en términos de calidad y tiempo. En este contexto, se estima que el proyecto tiene un valor entre ochocientos mil y un millón de colones. Este análisis proporciona una comprensión clara de los recursos financieros disponibles y su distribución durante el proyecto.

Identificación precisa de necesidades de adquisición:

Una vez establecido el presupuesto, se procede a identificar con precisión los bienes y servicios necesarios para la ejecución del proyecto.

Selección estratégica de proveedores:

Se lleva a cabo un riguroso proceso de selección de proveedores con el fin de obtener los bienes y servicios requeridos. Este proceso implica investigar y evaluar una variedad de proveedores, teniendo en cuenta criterios como calidad, precio, reputación, disponibilidad, entre otros.

Negociación detallada de contratos:

Una vez seleccionados los proveedores adecuados, se inicia una negociación exhaustiva de los términos y condiciones del contrato. Estos términos incluyen el precio, los plazos de entrega, las garantías, y otros aspectos relevantes. Es fundamental que el contrato refleje fielmente las necesidades y expectativas del proyecto.

Seguimiento y control continuo:

Durante la ejecución del proyecto, se lleva a cabo un seguimiento constante de las adquisiciones para garantizar el cumplimiento de los términos del contrato y la entrega oportuna de los bienes y servicios. Cualquier inconveniente o discrepancia que surja durante este proceso se aborda de manera inmediata y efectiva.

CAPÍTULO 5: DIAGRAMAS Y HERRAMIENTAS DE MODELADO

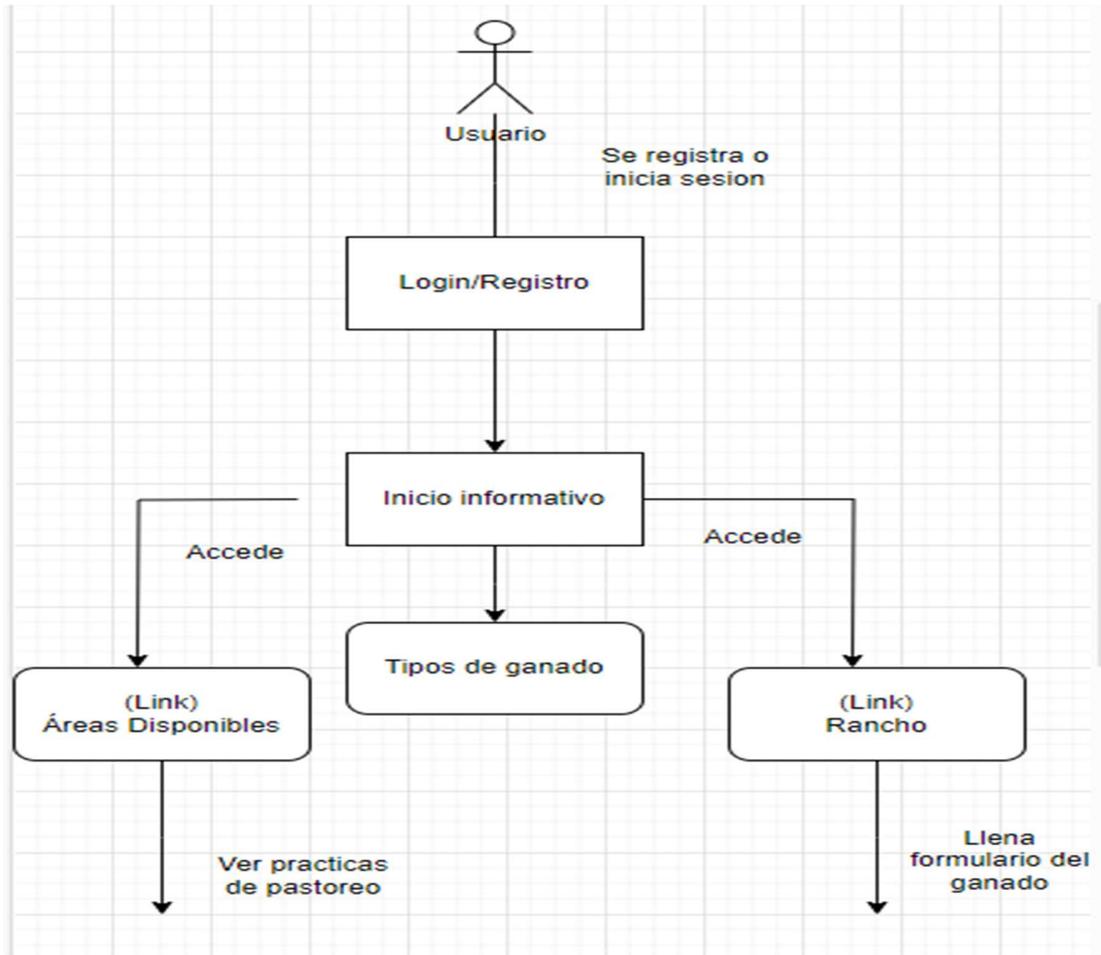
Diagramas de Procesos y Flujo:

Utilización de diagramas para representar procesos y flujos de información.

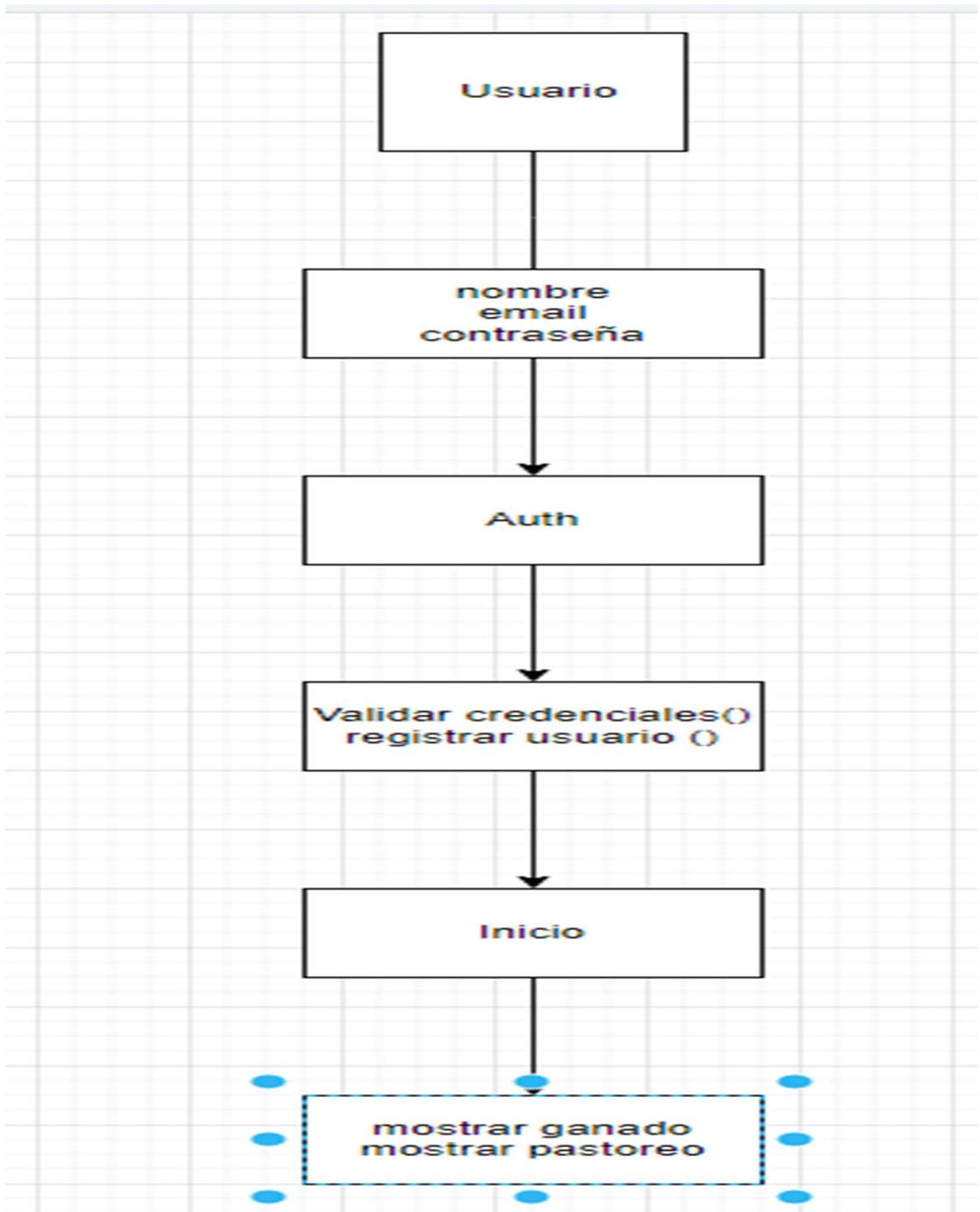
Los diagramas de procesos y flujo son cruciales porque muestran claramente cómo funcionan los procesos. Facilitan la comprensión, ayudan a identificar problemas, fomentan la colaboración y sirven como herramientas de documentación y capacitación. Su capacidad para simplificar procesos complejos, mejorar la eficiencia y garantizar la consistencia en la ejecución de tareas es lo que lo hace útil.

Diagramas UML:

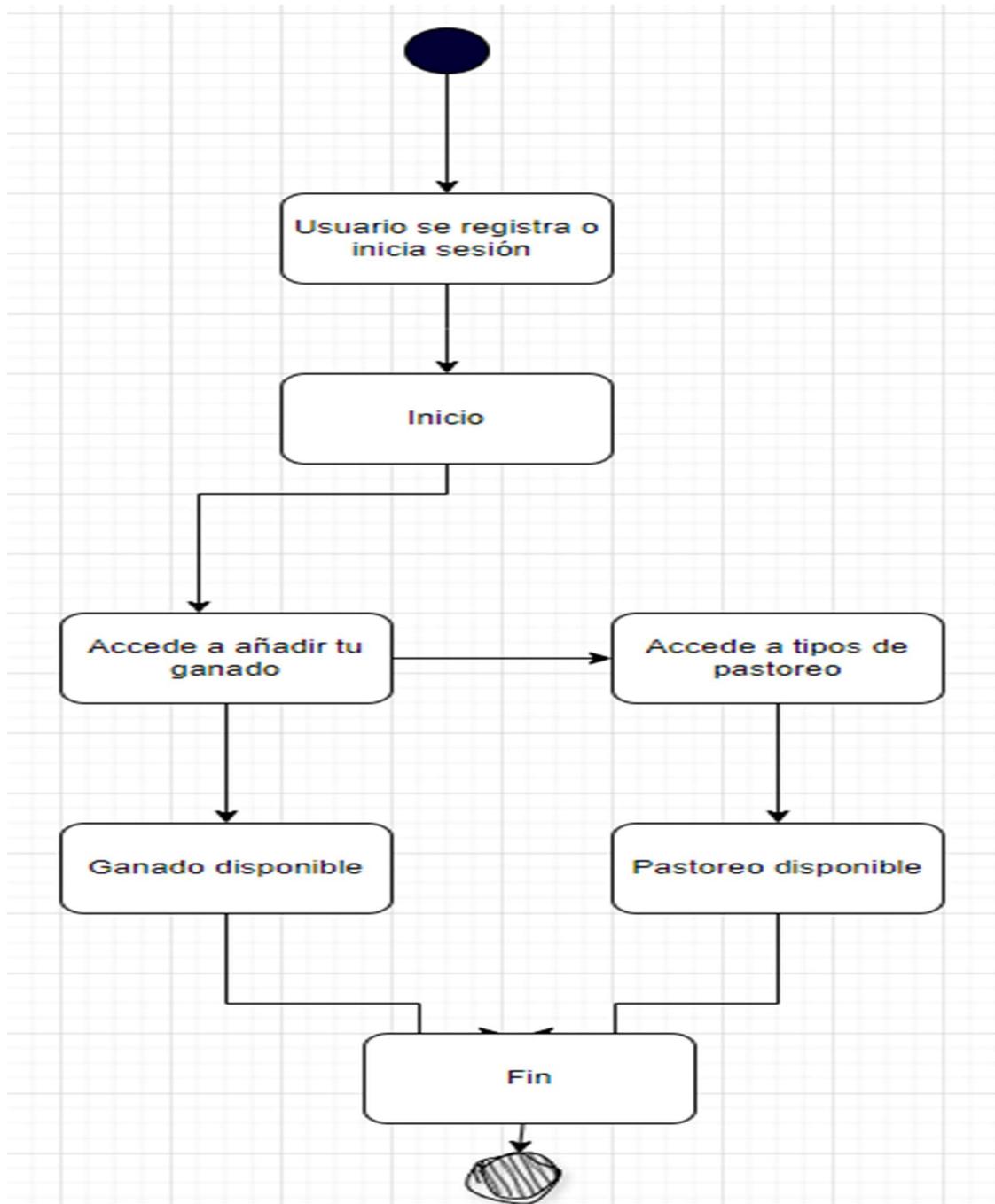
DIAGRAMA CASOS DE USO



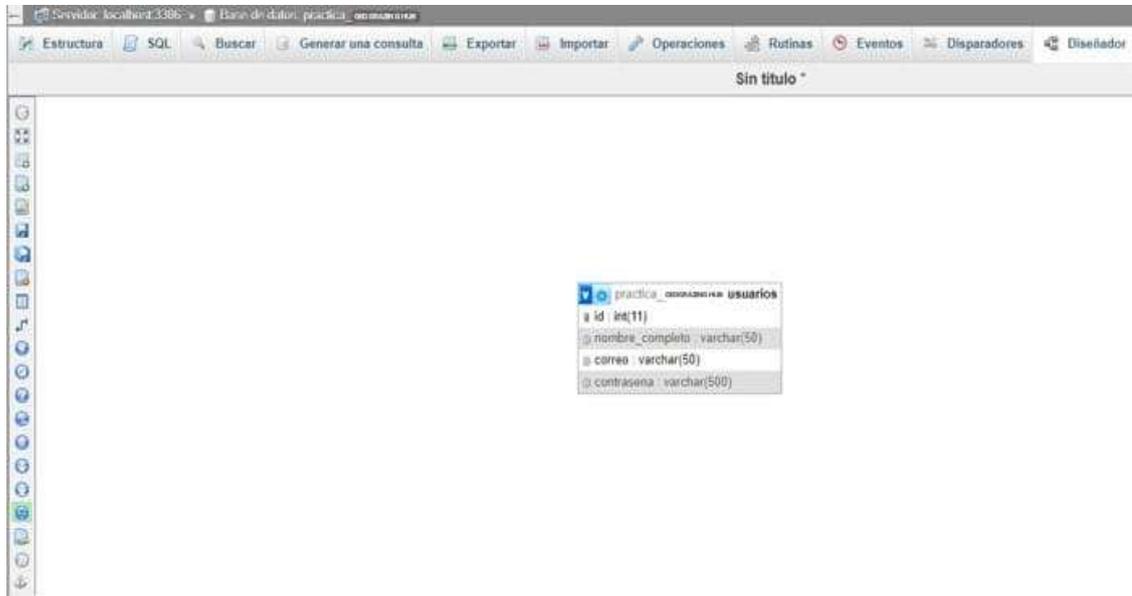
DIAGRAMAS DE CLASE



DIAGRAMAS DE ACTIVIDAD



DIAGRAMAS de Base de Datos:



Diccionarios de Datos:

6/4/24, 14:41

sv78.ifastnet.com / localhost / GEOGRAZING HUB | phpMyAdmin 5.2.1

GEOGRAZING HUB

usuarios

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (Primaria)	int(11)	No		
nombre_completo	varchar(50)	No		
correo	varchar(50)	No		
contrasena	varchar(500)	No		

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	5	A	No	

Herramientas de Modelado

Se utilizó el Visual Code como principal herramienta de trabajo para así utilizar los lenguajes de programación web necesarios para el proyecto, se utilizó XAMPP- phpMyAdmin para los lenguajes de bases de datos, se utilizó Dra.io para hacer los diagramas de la manera más eficiente posible.

Se han utilizado varias herramientas de modelado para facilitar el proceso de desarrollo en este proyecto en particular:

Visual Studio Code: Ofrece un entorno de desarrollo integrado (IDE) extremadamente flexible y poderoso como su principal herramienta de trabajo. Su capacidad de personalización y el soporte para una amplia gama de extensiones lo convierten en una opción popular para el desarrollo de aplicaciones web. Con VS Code, se pudo escribir, editar y depurar el código HTML, CSS, JavaScript.

Draw.io: Se utilizó Draw.io, una herramienta gratuita y fácil de usar en línea, para crear diagramas de flujo, diagramas de procesos, diagramas UML y otros tipos de diagramas. Diseño de diagramas es posible con Draw.io, que ofrece una amplia gama de formas y elementos predefinidos. Esta herramienta permitió la creación de diagramas detallados de la arquitectura del sistema, diagramas de casos de uso y otros diagramas necesarios para visualizar y comunicar de manera clara y efectiva los aspectos clave del proyecto.