PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

CARRERA DE- SISTEMAS



Trabajo de Titulación

Tema- El diseño y desarrollo de un prototipo web destinado a optimizar la gestión de ventas y el control de inventario en la comercialización de especies vegetales.

AUTOR-

Adriel Nicolas Suasnavas Cifuentes

QUITO DM, DICIEMBRE DE 2024

ÍNDICE

DEDICATORIA 6
AGRADECIMIENTO 7
1 CAPÍTULO I - INTRODUCCIÓN 8
1.1 TEMA 8
1.2 JUSTIFICACIÓN 8
1.3 PLANTEAMIENTO DEL PROBLEMA 9
1.4 OBJETIVOS 10
1.4.1. OBJETIVO GENERAL 10
1.4.2. OBJETIVOS ESPECÍFICOS 10
1.5 ALCANCE 11
2 CAPÍTULO II- FUNDAMENTACIÓN TEÓRICA 11
2.1 METODOLOGÍA DE DESARROLLO 12
2.1.1 Extreme Programing (XP) 12
2.1.2 Roles (XP) 12
2.1.3 Artefactos Extreme Programming 14
2.1.4 Comparación de metodologías 16
2.2 HERRAMIENTAS 17
2.2.1 VISUAL CODE 17
2.2.2 REACT 18
2.2.3 HTML 20
2.2.4 CSS 22
2.2.5 JAVASCRIPT 23
2.3 BASE DE DATOS 25
2.3.1 MySQL 25
3. Desarrollo de Trabajo 26
3.1 Requerimientos 27
3.2 Diseño de la base de datos 27
3.3 Product Backlog 31
3.4 Iteration 1 32
(28 de febrero - 31 de marzo) 32
3.4.1 Iteration Planning 32
3.4.2 Iteration Backlog 33
3.4.3 Realización de las tareas 34
3.4.3 Iteration Review 41

3.4.4 Iteration Test 41
3.5 Iteration 2 42
3.5.1 Iteration Planning 42
3.5.2 Iteration Backlog 43
3.5.3 Realización de las tareas 43
3.5.3 Iteration Review 49
3.5.4 Iteration Test 49
3.6 Iteration 3 49
3.6.1 Iteration Planning 50
3.6.2 Iteration Backlog 50
3.6.3 Realización de las tareas 51
3.6.3 Iteration Review 54
3.6.4 Iteration Test 54
4. Análisis de Resultados 55
5. Conclusiones y Recomendaciones 57
5.1 Conclusiones 57
5.2 Recomendaciones 58
Bibliography 60

Índice de Tablas

Tabla 1 -Roles de XP 14
Tabla 2 - Comparación de Metodologías 16
Tabla 3 Requerimientos funcionales del sistema 27
Tabla 4 Requerimientos no funcionales 27
Tabla 4: Iteration Backlog 32
Tabla 5 - Planificación de la iteración 1 32
Tabla 6 - Tabla de usuario 1 33
Tabla 7 - Tabla de usuario 2 33
Tabla 8 - Pruebas y resultados de la iteración 42
Tabla 9 - Planificación de la iteración 2 42
Tabla 10 - Pruebas y resultados de la iteración 43
Tabla 11 - Pruebas y resultados de la iteración 2 49
Tabla 12 - Planificación de la iteración 3 50
Tabla 13 – Historias de usuario iteración 3 50
Tabla 14 – Pruebas de iteración 3 54

Anexos

Figura 1: Base de datos Deckarnis....28
Figura 2: Conexión Base de datos (Deckarnis)....34
Figura 3: Registro a la Pagina (Deckarnis)....35
Figura 4: sección de código de Registro....35
Figura 5: inicio de sesión a la página (Deckarnis)....36
Figura 6: Sección de código de inicio de sesión....37
Figura 7: lógica de inicio y rol de usuario....37
Figura 8: Componente de navegación....38
Figura 9: Navegación en estado de Cliente....38
Figura 10: Navegación en estado de Administrador....38
Figura 11: Sección de código para permisos de la pagina....38
Figura 12: Sección de código para autenticación y rol de usuario....39
Figura 14: Sección de código del footer....40
Figura 15: lógica de inicio y rol de usuario....40
Figura 16: Componente sobre nosotros....41
Figura 17: API de clientes....43
Figura 18 – API de productos....44
Figura 19 : Servidor de compras....45
Figura 20: Carrito de compras....46
Figura 21: interfaz de administración de productos....47
Figura 22: control de almacenamiento de productos....47
Figura 23: Ejemplo validación de productos....48
Figura 25: código para gráficos de BI....51
Figura 25: Gráficos presentes en la interfaz....51
Figura 26: Error en cantidad de decimales....52
Figura 27: Error en funcionalidad del botón....52
Figura 28: Resolución de la cantidad de decimales....53
Figura 29: Código para limitar los decimales....53
Figura 30: falla de agregar al carrito....53
Figura 31: Código faltante para cumplir la funcionalidad....53

DEDICATORIA

Quiero dedicar este trabajo de titulación a mis padres, quienes han sido mi mayor apoyo desde que escogí esta carrera. A lo largo de este camino, me han brindado su apoyo, dedicación y ayuda en todo lo necesario para mi formación académica. Gracias por creer en mí y por ser mi inspiración constante para seguir adelante. También dedico este trabajo a mis abuelitos, quienes siempre han estado presentes en mi vida, brindándome su amor y sabiduría. A mi abuelita en el cielo, su memoria y amor perduran en mí como una luz que me guía.

AGRADECIMIENTO

En este camino hacia la culminación de mis estudios, quiero expresar mi más sincero agradecimiento.

A mi familia, cuyo apoyo incondicional ha sido fundamental en mi desarrollo académico.

A mis profesores de la Universidad, quienes no solo impartieron conocimientos valiosos, sino que también me brindaron orientación y apoyo académico.

A mis compañeros de clase, por su amistad, compañerismo y el apoyo mutuo que nos hemos brindado a lo largo de este camino educativo.

Por último, me reconozco a mí mismo por la determinación y el esfuerzo dedicados, incluido el trabajo durante las vacaciones para contribuir al financiamiento de mis estudios.

1 CAPÍTULO I - INTRODUCCIÓN

El Capítulo I introduce el desarrollo de un prototipo web para la comercialización de especies vegetales, motivado por la creciente necesidad en Ecuador de adaptarse a las tecnologías emergentes en el sector de venta de plantas. Se identifica una carencia de herramientas tecnológicas adecuadas en viveros, generando ineficiencias operativas y estratégicas. El trabajo se enfoca en integrar la gestión de ventas y control de inventario mediante soluciones tecnológicas específicas para viveros. Los objetivos incluyen el desarrollo de un sistema integral de gestión para ventas de plantas y la creación de una interfaz intuitiva y herramientas de gestión de datos eficientes para optimizar el proceso.

1.1 TEMA

"El diseño y desarrollo de un prototipo web destinado a la comercialización de especies vegetales."

1.2 JUSTIFICACIÓN

En el contexto de Ecuador, un país en constante evolución tecnológica, diversas industrias, incluyendo el sector de la venta de especies vegetales en viveros, enfrentan la creciente presión de adaptarse y aprovechar las tecnologías emergentes para satisfacer las cambiantes demandas de los clientes y mantener su competitividad en un mercado en constante transformación. Este proyecto de diseño y desarrollo de un prototipo web para la comercialización de especies vegetales surge como una respuesta concreta a una necesidad insatisfecha en el mercado ecuatoriano. En un entorno donde aún no existen plataformas adecuadas para la compra de especies vegetales en línea, esta iniciativa se presenta como un faro de innovación, ofreciendo a los clientes una experiencia de compra mejorada y accesible, y brindando a los propietarios de viveros la oportunidad de aprovechar este nicho sin explotar,

potenciando así sus operaciones y expandiendo su alcance a nivel nacional. En un mercado donde la digitalización es una tendencia imparable, este proyecto se convierte en un catalizador para la industria de la venta de especies vegetales en Ecuador, ofreciendo beneficios tanto a los consumidores como a los empresarios locales y marcando un hito en la modernización de un sector tradicional.

1.3 PLANTEAMIENTO DEL PROBLEMA

En el contexto actual de Ecuador, donde la tecnología está experimentando un crecimiento constante, el sector de la venta de plantas en viveros enfrenta desafíos significativos en su gestión y operación. A pesar de la creciente demanda de productos de jardinería y paisajismo, se observa una falta de herramientas tecnológicas adecuadas para optimizar la gestión de ventas, el control de inventario y la satisfacción del cliente en este sector específico.

Esta carencia de una plataforma integral de gestión de datos en viveros y tiendas de jardinería ha dado lugar a problemas operativos y estratégicos, como ineficiencias en la gestión de inventarios y dificultades para tomar decisiones informadas. Para abordar esta situación, surge la necesidad de desarrollar soluciones tecnológicas adaptadas específicamente para la venta de plantas en viveros.

Por lo tanto, se plantea la siguiente pregunta principal de investigación-

¿Cómo pueden las soluciones tecnológicas integrar la gestión de ventas y el control de inventario en viveros de plantas en Ecuador?

Acompañando a esta pregunta central, se exploran interrogantes adicionales-

¿Cuáles son las necesidades y expectativas de los clientes en el proceso de compra de plantas y productos relacionados en viveros?

¿De qué manera estas soluciones tecnológicas pueden mejorar la eficiencia en la gestión de inventarios y la disponibilidad de productos en viveros?

¿Cuál es el impacto potencial de la implementación de esta tecnología en la rentabilidad y la competitividad de los viveros en Ecuador?

1.4 OBJETIVOS

1.4.1. OBJETIVO GENERAL

Desarrollar e implementar un prototipo de gestión integral para ventas de plantas, que optimice el control de inventario, simplifique el registro de ventas y proporcione análisis en Linea para impulsar la eficiencia y rentabilidad del negocio.

1.4.2. OBJETIVOS ESPECÍFICOS

- Desarrollar un prototipo informático intuitivo que permita registrar y gestionar las ventas, manteniendo actualizado el inventario de plantas y productos de manera eficiente.
- Proporcionar herramientas que faciliten la comprensión del sistema de gestión de datos y se adapten a las necesidades anticipadas para el manejo eficiente y la gestión óptima del sistema.
- Asegurar una interfaz amigable y la capacidad de acceso remoto para los clientes, junto con la provisión de información actualizada y relevante de manera oportuna.
- Diseño de aplicación WEB con interfaz amigable al usuario con información actualizada y relevante con gráficos creados mediante el uso de librerías para mejorar la toma de decisiones.

1.5 ALCANCE

Interfaz de Usuario Intuitiva

Se desarrollará una interfaz de usuario amigable y de fácil navegación para que los empleados puedan registrar las ventas de plantas y productos relacionados de manera eficiente.

La interfaz estará diseñada para guiar a los usuarios a través del proceso de registro de ventas de manera intuitiva, minimizando la posibilidad de errores.

Gestión de Inventarios en Linea

Se implementará un sistema de gestión de inventario que mantenga actualizada la disponibilidad de plantas y productos en Linea.

El sistema registrará automáticamente las ventas realizadas y los nuevos ingresos de inventario, asegurando que la información esté siempre actualizada.

Provisión de Información Actualizada

Se implementarán paneles de control y reportes que proporcionen información detallada y análisis sobre las ventas realizadas, el estado del inventario y las tendencias de ventas.

Limitaciones del Alcance

Se enfocará exclusivamente en la fase de diseño y desarrollo del prototipo, sin incluir la implementación completa del sistema en todas las áreas de la empresa.

No se abordarán aspectos relacionados con la integración con sistemas preexistentes, como sistemas de contabilidad o de gestión de recursos humanos.

El prototipo estará sujeto a pruebas y ajustes iterativos para mejorar su funcionalidad y usabilidad antes de una eventual implementación completa.

2 CAPÍTULO II- FUNDAMENTACIÓN TEÓRICA

2.1 METODOLOGÍA DE DESARROLLO

2.1.1 Extreme Programing (XP)

La Programación Extrema (XP) es una metodología ágil que se distingue por su enfoque en la velocidad y la simplicidad en el desarrollo de software. Adopta ciclos de desarrollo cortos y reduce al mínimo la documentación, centrándose en la entrega rápida de software funcional. A través de una serie de valores, reglas y prácticas estrictas, XP guía el trabajo en equipo hacia la entrega continua de software de alta calidad. (Raeburn, 2024)

La primera implementación formal de Extreme Programming comenzó el 6 de marzo de 1996. Desde entonces, XP ha sido reconocida por su capacidad para adaptarse rápidamente a los cambios y satisfacer las demandas cambiantes del cliente. Una característica clave de XP es la incorporación de revisiones de código frecuentes y pruebas unitarias. Estas prácticas aseguran que el código sea robusto y libre de errores, promoviendo una cultura de calidad y responsabilidad compartida dentro del equipo.

Además, XP fomenta la colaboración constante entre todos los miembros del equipo durante todo el ciclo de desarrollo, lo que permite adaptarse rápidamente a los cambios y mejorar continuamente el producto final. Este enfoque centrado en la entrega rápida y la calidad constante hace que XP sea especialmente efectivo para equipos pequeños y proyectos donde los requisitos pueden evolucionar rápidamente. (Extreme Programming Rules, n.d.)

2.1.2 Roles (XP)

Los roles presentes en Extreme Programming son relativamente pocos, pero altamente eficientes para el desarrollo del proyecto, lo cual es característico de esta metodología ágil.

Tracker (Seguimiento)

El Tracker se encarga de proporcionar retroalimentación al equipo, verificando la precisión de las estimaciones realizadas y el Linea dedicado al trabajo. Su función principal es comunicar los resultados para mejorar las estimaciones futuras.

Customer (Cliente)

El Cliente es responsable de escribir las historias de usuario y las pruebas funcionales para validar la implementación. Asigna prioridad a las historias de usuario y decide qué se implementará en cada iteración, centrándose en agregar el máximo valor de negocio.

Programmer (Programador)

El Programador es considerado el miembro más importante del equipo, ya que escribe las pruebas unitarias y el código del sistema. Se enfoca en la implementación técnica de las funcionalidades requeridas.

Coach (Entrenador)

El Coach es responsable del proceso general y guía a los miembros del equipo para seguir correctamente el proceso de Extreme Programming. Su rol es facilitar la implementación exitosa de la metodología y resolver cualquier impedimento que surja.

Tester (Tester)

El Tester ayuda al Cliente a escribir pruebas funcionales y ejecuta pruebas regularmente. Propaga los resultados dentro del equipo y es responsable de las herramientas de soporte para pruebas.

Manager (Gerente)

El Gerente se encarga de programar las reuniones, asegurarse de que se siga el proceso de Extreme Programming y registrar los resultados para futuros informes. Asiste a las reuniones y mantiene al equipo feliz y productivo.

(Blaxell, 2006)

Tabla 1 -Roles de XP

rol	Usuario
Tracker	Adriel Suasnavas
Costumer	Adriel Suasnavas
Programmer	Adriel Suasnavas
Coach	Magister Mora
Tester	Adriel Suasnavas

2.1.3 Artefactos Extreme Programming

Los artefactos en Extreme Programming (XP) son elementos esenciales que ayudan a gestionar y facilitar el desarrollo de software ágil ya que estos artefactos incluyen documentos, herramientas y resultados tangibles que son fundamentales para el éxito del proyecto. Cada artefacto tiene un propósito específico en el proceso de desarrollo y contribuye a la transparencia, comunicación y entrega de software de alta calidad de manera iterativa. (LIZARRAGA, 2017)

User Stories (Historias de Usuario)

Las historias de usuario son descripciones detalladas de las funcionalidades del sistema desde la perspectiva del usuario final, estas historias están escritas en un lenguaje sencillo y comprensible para todos los miembros del equipo, ayudando a definir las características clave del software y los requisitos funcionales ya que cada historia de usuario representa una

unidad independiente de trabajo que agrega valor al usuario final y se utiliza para priorizar y planificar el desarrollo de software.

Acceptance Tests (Pruebas de Aceptación)

Las pruebas de aceptación son conjuntos específicos de pruebas diseñadas para validar las historias de usuario implementadas ya que estas pruebas verifican que las funcionalidades desarrolladas cumplen con los criterios de aceptación definidos por el cliente. Las pruebas de aceptación son esenciales para asegurar que el software entregado cumpla con las expectativas y funcione correctamente según los requisitos definidos.

Release Plan (Plan de Liberación)

El plan de liberación es un documento que describe la estrategia y la secuencia de entregas del software durante el desarrollo pues define las características a implementar en cada iteración o versión, así como los hitos importantes y las fechas de entrega. El plan de liberación es clave para una gestión eficaz del proyecto, permitiendo una entrega progresiva y controlada del producto final.

Spike Solutions (Soluciones de Investigación)

Las soluciones de investigación, también conocidas como spikes, son prototipos rápidos utilizados para explorar y validar soluciones técnicas o resolver incertidumbres durante el desarrollo ya que los spikes permiten a los equipos experimentar y tomar decisiones informadas antes de comprometerse con implementaciones completas, ayudando a mitigar riesgos y mejorar la calidad del software.

Source Code (Código Fuente)

El código fuente es el conjunto de instrucciones escritas por los programadores que define el comportamiento del software ya que representa la implementación concreta de las historias

de usuario y las funcionalidades del sistema. El código fuente es la base del desarrollo de software en XP y es esencial para construir y mantener el producto final entregado al cliente.

Test Cases (Casos de Prueba)

Los casos de prueba son conjuntos de pruebas diseñados para verificar el comportamiento y la funcionalidad del software en diferentes escenarios ya que estos incluyen pruebas unitarias, de integración y de aceptación que garantizan la calidad del software desarrollado. Los casos de prueba identifican errores y problemas durante el ciclo de desarrollo, permitiendo su corrección y mejora continua.

(Wake, 2021)

2.1.4 Comparación de metodologías

Tabla 2 - Comparación de Metodologías

Parámetro	XP (eXtreme Programming)	Scrum	Waterfall (Cascada)
Enfoque	Ágil, iterativo e incremental	Ágil, iterativo e incremental	Secuencial, predictivo
Principios	Basado en valores como comunicación, simplicidad, feedback y valentía	Basado en transparencia, inspección y adaptación	Basado en procesos y documentación detallada
Flexibilidad	Altamente adaptable a cambios en los requisitos y necesidades del cliente	Flexible y adaptable a cambios, con iteraciones cortas	Menos flexible, los cambios requieren revisión y aprobación formal
Prácticas	Incorpora prácticas como desarrollo dirigido por pruebas (TDD), integración continua, refactorización, programación en parejas, entre otras	Utiliza prácticas ágiles como reuniones diarias, retrospectivas, entregas frecuentes, entre otras	Sigue un conjunto más rígido de prácticas y roles definidos, como análisis detallado, diseño previo, etc.
Ciclo de desarrollo	Iterativo e incremental, con entregas frecuentes de funcionalidades	Iterativo, con entregas incrementales y regulares	Secuencial, con fases definidas como análisis, diseño, implementación y pruebas
Adaptabilidad	Capaz de adaptarse rápidamente a los cambios y nuevas demandas del cliente	Adaptable a cambios, pero con un enfoque más estructurado y planificado	Menor capacidad de adaptación a cambios, requiere revisión y ajuste del plan

La selección de eXtreme Programming (XP) para mi tesis se justifica por su enfoque ágil y adaptable, fundamental para desarrollar un prototipo web integral para la comercialización de especies vegetales en un mercado dinámico como el ecuatoriano. XP, al promover valores como la comunicación constante, la simplicidad y el feedback, junto con prácticas como el desarrollo dirigido por pruebas (TDD) y la entrega frecuente de software, permitió gestionar

eficientemente los cambios en los requisitos del cliente, adaptar el producto según el feedback del usuario y garantizar la entrega oportuna de un prototipo funcional y de alta calidad.

2.2 HERRAMIENTAS

2.2.1 VISUAL CODE

Visual Studio Code (también conocido como VS Code) es un editor de código fuente desarrollado por Microsoft que se ha convertido en una herramienta muy popular entre los desarrolladores debido a su alta calidad, funcionalidades avanzadas y su naturaleza de código abierto. VS Code es multiplataforma, lo que significa que está disponible para sistemas operativos Windows, macOS y Linux.

Utiliza una arquitectura ligera y eficiente en base a un modelo de multiproceso para gestionar las diferentes tareas del editor, como el análisis sintáctico, la ejecución de extensiones y la interfaz de usuario. Esto garantiza un rendimiento óptimo incluso en proyectos grandes y complejos. Gracias a su arquitectura modular, Visual Studio Code puede adaptarse fácilmente a diferentes entornos de desarrollo y necesidades específicas de los desarrolladores. Además, VS Code se integra con herramientas populares como linters (para análisis estático de código), gestores de paquetes (como npm o pip) y sistemas de construcción (como webpack o gulp), lo que facilita la configuración de un entorno de desarrollo completo para una variedad de lenguajes y marcos de trabajo. Esta flexibilidad y capacidad de personalización hacen que VS Code sea una opción atractiva tanto para desarrolladores principiantes como para equipos de desarrollo profesionales que trabajan en proyectos complejos y variados. (Flores, 2022)

Las características clave de Visual Studio Code incluyen.

Interfaz Moderna: VS Code tiene una interfaz de usuario limpia y moderna que es altamente personalizable. Los usuarios pueden ajustar temas, disposición de paneles y atajos de teclado según sus preferencias.

Extensibilidad: Una de las fortalezas principales de VS Code es su ecosistema de extensiones. Los desarrolladores pueden instalar extensiones desde el mercado de Visual Studio Code para agregar soporte para diferentes lenguajes de programación, depuración, control de versiones (como Git), herramientas de productividad y mucho más.

Funcionalidades Avanzadas: VS Code ofrece características avanzadas como resaltado de sintaxis inteligente, autocompletado, refactoring de código, depuración integrada con soporte para múltiples lenguajes y extensiones, y una integración robusta con Git.

Terminal Integrada: Visual Studio Code incluye una terminal integrada que permite ejecutar comandos y scripts directamente desde el editor, lo que facilita el desarrollo y la ejecución de tareas relacionadas con el proyecto sin tener que cambiar de ventana.

Integración con Servicios en la Nube: VS Code se integra con servicios en la nube como Azure, facilitando el desarrollo y la implementación de aplicaciones en la nube directamente desde el editor.

(Flores, 2022)

2.2.2 REACT

React es una biblioteca de JavaScript de código abierto desarrollada por Facebook que se utiliza para construir interfaces de usuario interactivas y eficientes para aplicaciones web. Se enfoca en la creación de componentes reutilizables que representan diferentes partes de la interfaz de usuario, lo que facilita el desarrollo, mantenimiento y escalabilidad de aplicaciones web modernas.

React se utiliza para construir interfaces de usuario dinámicas y eficientes, especialmente en aplicaciones de una sola página (SPA) donde los contenidos cambian sin necesidad de recargar toda la página. Permite crear componentes reutilizables que encapsulan la lógica y la interfaz de usuario asociada, facilitando el desarrollo basado en componentes y la modularidad del código. React es ampliamente utilizado en el desarrollo front-end para crear experiencias de usuario interactivas y receptivas.

(A., 2023)

Características principales de React.

Componentes Reutilizables

React se basa en el concepto de componentes, que son bloques de construcción independientes que representan partes específicas de la interfaz de usuario. Estos componentes son reutilizables, lo que facilita la creación y mantenimiento de interfaces complejas.

Virtual DOM (Document Object Model)

React utiliza un Virtual DOM para mejorar el rendimiento de las aplicaciones web. El Virtual DOM es una representación ligera del DOM real y permite actualizar eficientemente solo los elementos que han cambiado, minimizando la manipulación del DOM real y mejorando la velocidad de renderizado.

JSX (JavaScript XML)

React utiliza JSX, una extensión de JavaScript que permite escribir código HTML dentro de JavaScript. Esto facilita la creación de componentes React con una sintaxis similar al HTML, lo que mejora la legibilidad y mantenibilidad del código.

Estado (State) y Ciclo de Vida - React utiliza el concepto de estado (state) para gestionar datos y controlar el comportamiento de los componentes. Los componentes React tienen un ciclo de vida definido que permite ejecutar código en diferentes etapas, como la creación, actualización y destrucción de un componente.

React Hooks

Introducidos en versiones recientes de React, los React Hooks son funciones que permiten a los componentes funcionales tener estado y funcionalidades de ciclo de vida sin necesidad de escribir una clase. Los Hooks simplifican el desarrollo y mejoran la legibilidad del código.

Librería Extensa y Ecosistema

React cuenta con una amplia librería de componentes y una comunidad activa que desarrolla y mantiene bibliotecas complementarias (como React Router para enrutamiento, Redux para gestión del estado, etc.) que extienden las funcionalidades de React y facilitan el desarrollo de aplicaciones web complejas.

(Mora, 2023)

2.2.3 HTML

HTML (HyperText Markup Language) es el lenguaje estándar utilizado para crear y estructurar el contenido de las páginas web que define la estructura y el significado semántico del contenido web mediante etiquetas y elementos que representan diferentes tipos de información, como texto, imágenes, enlaces, formularios, listas y más.

HTML se utiliza para crear la estructura básica de una página web, proporcionando un marco sobre el cual se pueden agregar estilos (con CSS) y comportamientos dinámicos (con JavaScript) los cuales definen la jerarquía y relación entre los diferentes elementos de

contenido, lo que facilita la navegación, accesibilidad y optimización para motores de búsqueda. (B., 2023)

Características de HTML.

Etiquetas y Elementos- HTML utiliza etiquetas para marcar y estructurar el contenido de una página web. Cada etiqueta describe el tipo de contenido que contiene y cómo debe ser mostrado. Los elementos HTML incluyen encabezados, párrafos, imágenes, enlaces, formularios, listas, tablas y más.

Atributos- Los atributos proporcionan información adicional sobre los elementos HTML. Por ejemplo, el atributo src especifica la ubicación de una imagen, el atributo href define la URL de un enlace y el atributo class o id asigna nombres de clase o identificadores para aplicar estilos y comportamientos con CSS y JavaScript.

Formularios- HTML incluye elementos para crear formularios interactivos que permiten a los usuarios ingresar datos y enviar información a través de la web.

Compatibilidad y Estándares- HTML sigue estándares establecidos por el World Wide Web Consortium (W3C), lo que garantiza su compatibilidad y consistencia en diferentes navegadores y la evolución de HTML ha introducido nuevas versiones como HTML5, que incluye mejoras en multimedia, gráficos, semántica y accesibilidad.

Interoperabilidad- HTML es compatible con otras tecnologías web como CSS para el diseño visual y JavaScript para la interactividad, lo que permite crear experiencias web dinámicas y atractivas.

(SernaGroup, 2021)

2.2.4 CSS

CSS (Cascading Style Sheets) es un lenguaje utilizado en el desarrollo web para definir la presentación y el estilo visual de documentos HTML y XML. Es un componente fundamental de la construcción de sitios web modernos, permitiendo separar la estructura del contenido (HTML) de su presentación visual. CSS define cómo se ven los elementos HTML en una página, incluyendo aspectos como colores, fuentes, márgenes, espaciado, diseño de página y efectos visuales. (Cascading Style Sheets: CSS Browsers, n.d.)

CSS sirve para mejorar la apariencia y la experiencia del usuario en un sitio web. Permite aplicar estilos de manera consistente a múltiples páginas web, lo que facilita la personalización y el mantenimiento del diseño. Al separar la presentación del contenido, CSS permite realizar cambios estilísticos de manera eficiente sin necesidad de modificar el HTML subyacente. Además, CSS es esencial para crear diseños responsivos y adaptables que se ajusten automáticamente a diferentes tamaños de pantalla y dispositivos, lo cual es crucial en el desarrollo web actual.

Características principales.

Flexibilidad en Estilización - CSS ofrece una amplia gama de propiedades y valores que permiten controlar casi todos los aspectos visuales de los elementos HTML. Desde definir colores y fuentes hasta establecer márgenes, rellenos, alineaciones y bordes, CSS proporciona una gran flexibilidad en la personalización del diseño web.

Selección de Elementos - Los selectores en CSS son versátiles y permiten seleccionar elementos basados en su tipo (como etiquetas HTML), clase, ID, atributos, jerarquía o estado. Esto proporciona un gran control sobre qué elementos se estilan y cómo se aplican los estilos.

Modelo de Caja CSS sigue un modelo de caja que define cómo se renderizan y se presentan los elementos en la página web. Esto incluye el tamaño total del elemento (ancho y alto),

márgenes, rellenos, bordes y espacio entre elementos, lo que ayuda a controlar la disposición y el diseño de la página.

Diseño Responsivo y Adaptable - Una de las características más importantes de CSS en la actualidad es su capacidad para crear diseños responsivos que se adapten automáticamente a diferentes tamaños de pantalla y dispositivos. Esto se logra utilizando unidades de medida relativas (como porcentajes, em o rem) y técnicas como media queries, que permiten aplicar estilos específicos según las características del dispositivo o la ventana del navegador.

Animaciones y Transiciones - CSS permite crear animaciones y transiciones fluidas en elementos HTML mediante propiedades como transition y keyframes. Esto mejora la experiencia del usuario al proporcionar interacciones dinámicas y atractivas que hacen que el sitio web sea más interactivo y agradable de usar.

Compatibilidad y Normativas - CSS sigue estándares establecidos por el World Wide Web Consortium (W3C), lo que garantiza su compatibilidad y consistencia en diferentes navegadores web. Los desarrolladores pueden utilizar CSS de manera efectiva para garantizar una experiencia de usuario coherente y de alta calidad en múltiples plataformas y dispositivos. (IONOS, 2023)

2.2.5 JAVASCRIPT

JavaScript es un lenguaje de programación de alto nivel, interpretado y orientado a objetos que se utiliza principalmente en el desarrollo web para agregar interactividad y funcionalidad dinámica a las páginas. Junto con HTML y CSS, JavaScript es una de las tecnologías fundamentales para construir aplicaciones web modernas y experiencias interactivas en el navegador. (What Is JavaScript?, n.d.)

JavaScript se utiliza para realizar tareas como manipulación del DOM (Document Object Model), manejo de eventos, validación de formularios, animaciones, interacciones de usuario,

comunicación con servidores (a través de AJAX o Fetch API), creación de aplicaciones web complejas (SPA - Single Page Applications) y mucho más ya que al ser un lenguaje versátil que se ejecuta en el lado del cliente (navegador) y, con Node.js, también puede ejecutarse en el lado del servidor. (Capitulo 4. How JavaScript Was Created, n.d.)

Características principales de JavaScript-

Lenguaje Interpretado y Orientado a Objetos- JavaScript es un lenguaje interpretado, lo que significa que el código se ejecuta línea por línea en tiempo de ejecución y es un lenguaje orientado a objetos basado en prototipos, lo que permite la creación de objetos y la herencia de propiedades y métodos entre ellos.

Manipulación del DOM- JavaScript permite interactuar con el DOM de una página web, lo que significa que puede seleccionar elementos HTML, modificar su contenido, estilos y atributos, responder a eventos del usuario (clics, teclas presionadas, etc.) y actualizar dinámicamente la interfaz de usuario.

Eventos y Asincronía- JavaScript es compatible con el manejo de eventos, lo que permite responder a acciones del usuario y ejecutar código en respuesta a eventos específicos siendo JavaScript asincrónico por naturaleza, lo que significa que puede realizar tareas como solicitudes de red (AJAX) o temporizadores sin bloquear la ejecución del código principal.

Funciones y Closures- JavaScript es conocido por su capacidad de trabajar con funciones anónimas, funciones de flecha, funciones de alto orden y closures, lo que facilita la creación de código modular y reutilizable.

Librerías y Frameworks- JavaScript tiene un ecosistema extenso de librerías y frameworks populares como React, Angular, Vue.js, jQuery, Express.js, entre otros, que facilitan el desarrollo de aplicaciones web complejas y escalables como en este caso que se escogió React para la realizacion.

Estándares y Evolución- JavaScript sigue estándares establecidos por el ECMAScript (ES), con nuevas versiones como ES6 (ECMAScript 2015) introduciendo características modernas como clases, módulos, reestructuración, promesas y más, que mejoran la legibilidad y la capacidad de mantenimiento del código.

2.3 BASE DE DATOS

2.3.1 MySQL

MySQL es un sistema de gestión de bases de datos relacional (RDBMS) de código abierto que se utiliza ampliamente en aplicaciones web y de servidor debido a su fiabilidad, escalabilidad y facilidad de uso. Desarrollado inicialmente por MySQL AB y ahora parte de Oracle Corporation, MySQL es uno de los sistemas de bases de datos más populares y confiables en el mundo actual. (Londono, 19)

MySQL desempeña un papel fundamental en la gestión y organización eficiente de grandes volúmenes de datos. Es particularmente adecuado para aplicaciones web y empresariales que requieren almacenamiento estructurado de datos en tablas relacionales. Estoy explorando cómo MySQL puede optimizar la gestión de datos en mi proyecto de tesis, centrándome en su escalabilidad, rendimiento y capacidades transaccionales.

Características clave de MySQL

Lenguaje SQL MySQL utiliza SQL (Structured Query Language) como lenguaje estándar para realizar consultas y operaciones en la base de datos. Este lenguaje permite realizar consultas complejas para interactuar con los datos de manera eficiente.

Modelo Relacional MySQL sigue un modelo de bases de datos relacionales, donde los datos se organizan en tablas con relaciones definidas. Esto proporciona una estructura flexible y escalable para manejar diferentes tipos de datos.

Escalabilidad y Rendimiento MySQL es altamente escalable y puede manejar grandes volúmenes de datos con un alto rendimiento. Ofrece funcionalidades como replicación y particionamiento para distribuir la carga de trabajo y mejorar la eficiencia.

Seguridad La seguridad de los datos es fundamental en mi investigación. MySQL ofrece mecanismos sólidos de autenticación y control de acceso para proteger la integridad de los datos sensibles.

Soporte Transaccional MySQL garantiza la integridad de los datos mediante el soporte de transacciones ACID. Esto es crucial para mantener la coherencia de los datos en entornos concurrentes.

3. Desarrollo de Trabajo

En este capítulo, se lleva a cabo un análisis detallado de los requisitos que deben abordarse en el diseño y funcionalidad de la página web, así como una evaluación de los objetivos proporcionados. Estos requisitos y objetivos son fundamentales para guiar la planificación estratégica y la implementación efectiva del proyecto.

3.1 Requerimientos

Tabla 3 Requerimientos funcionales del sistema

Requisito	Descripción
Registro de Usuarios	Capacidad para registrar y autenticar usuarios (empleados y clientes) en el sistema.
Gestión de Inventario	Funcionalidad para agregar y actualizar productos en el inventario mediante una interfaz administrativa.
Registro de Ventas	Interfaz para registrar ventas simuladas de plantas y productos, con generación de confirmaciones de venta.
Interfaz de Usuario Intuitiva	Diseño claro y accesible para empleados y clientes, con identificación visual de productos y transacciones.
Reportes Básicos	Capacidad para generar informes simples sobre ventas y estado del inventario para análisis inicial.

Tabla 4 Requerimientos no funcionales

Requisito	Descripción
Usabilidad	Interfaz intuitiva y fácil de navegar para usuarios sin experiencia técnica, con énfasis en accesibilidad y claridad.
Rendimiento	Respuesta rápida y tiempos de carga eficientes para simular una experiencia de usuario fluida.
Seguridad y Privacidad	Medidas básicas de seguridad para proteger datos de prueba, como autenticación simple y cifrado básico.

3.2 Diseño de la base de datos

Se presenta el diagrama ER de la Base de Datos en la tesis el cual es esencial para brindar una comprensión clara de cómo están organizados y relacionados los datos dentro del sistema de la página web. Este enfoque visual permite entender fácilmente la estructura de la base de datos y cómo interactúan sus componentes, lo que facilita la identificación de problemas de diseño y la optimización del rendimiento del sistema. Además de proporcionar una base sólida para respaldar las decisiones de diseño y desarrollo tomadas durante el proyecto.

Figura 1: Base de datos Deckarnis

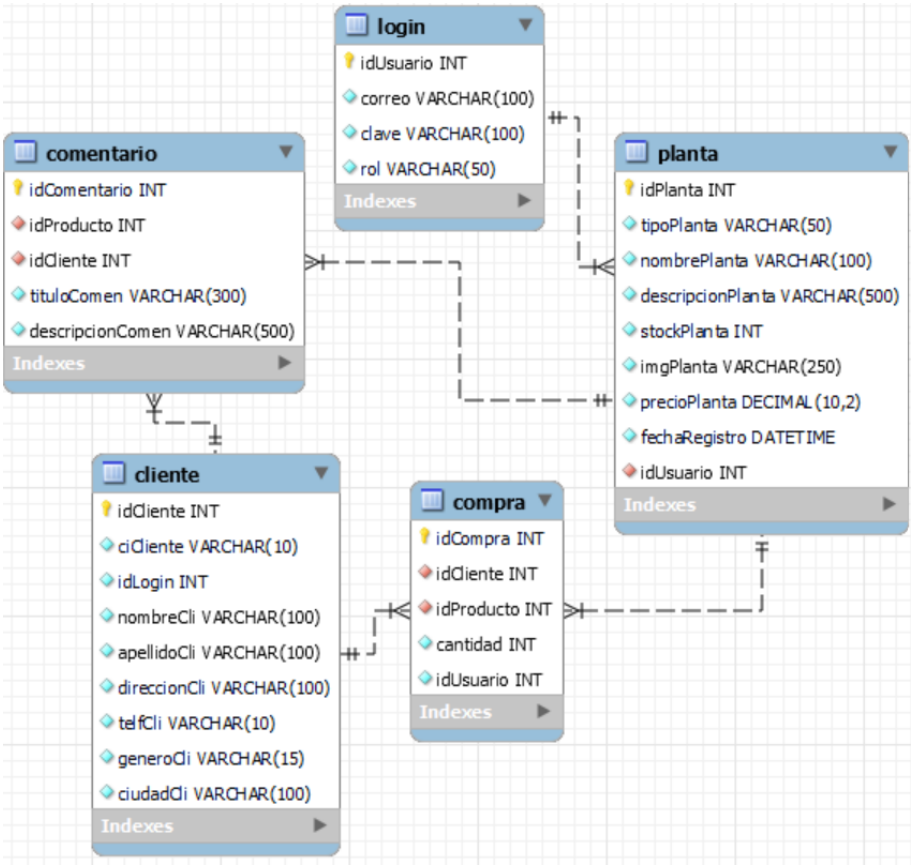


Figura 1 generada de la BD Deckarnis por MySQL

Script de la Bade de datos Deckarnis

```
-- Crear la base de datos

CREATE DATABASE IF NOT EXISTS Deckarnis;

-- Usar la base de datos

USE Deckarnis;

CREATE TABLE IF NOT EXISTS login (

    idUsuario INT PRIMARY KEY AUTO_INCREMENT,

    correo VARCHAR(100) NOT NULL,
```

```
    clave VARCHAR(100) NOT NULL,
    rol VARCHAR(50) NOT NULL DEFAULT 'client'
);
-- Crear la tabla Planta
CREATE TABLE IF NOT EXISTS Planta (
    idPlanta INT PRIMARY KEY AUTO_INCREMENT,
    tipoPlanta VARCHAR(50) NOT NULL,
    nombrePlanta VARCHAR(100) NOT NULL,
    descripcionPlanta VARCHAR(500) NOT NULL,
    stockPlanta INT NOT NULL,
    imgPlanta VARCHAR(250) NOT NULL,
    precioPlanta DECIMAL(10, 2) NOT NULL,
    fechaRegistro DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    idUsuario INT NOT NULL,
    FOREIGN KEY (idUsuario) REFERENCES login(idUsuario)
);
-- Crear la tabla Cliente
CREATE TABLE IF NOT EXISTS Cliente (
    idCliente INT PRIMARY KEY AUTO_INCREMENT,
    ciCliente VARCHAR(10) NOT NULL,
```

```
    idLogin INT NOT NULL,
    nombreCli VARCHAR(100) NOT NULL,
    apellidoCli VARCHAR(100) NOT NULL,
    direccionCli VARCHAR(100) NOT NULL,
    telfCli VARCHAR(10) NOT NULL,
    generoCli VARCHAR(15) NOT NULL,
    ciudadCli VARCHAR(100) NOT NULL
);

-- Crear la tabla Comentario
CREATE TABLE IF NOT EXISTS Comentario (
    idComentario INT PRIMARY KEY AUTO_INCREMENT,
    idProducto INT NOT NULL,
    idCliente INT NOT NULL,
    tituloComen VARCHAR(300) NOT NULL,
    descripcionComen VARCHAR(500) NOT NULL,
    FOREIGN KEY (idProducto) REFERENCES Planta(idPlanta),
    FOREIGN KEY (idCliente) REFERENCES Cliente(idCliente)
);
```

```
-- Crear la tabla Compra

CREATE TABLE IF NOT EXISTS Compra (

    idCompra INT PRIMARY KEY AUTO_INCREMENT,

    idCliente INT NOT NULL,

    idProducto INT NOT NULL,

    cantidad INT NOT NULL,

    idUsuario INT NOT NULL,

    FOREIGN KEY (idCliente) REFERENCES Cliente(idCliente),

    FOREIGN KEY (idProducto) REFERENCES Planta(idPlanta)

);
```

3.3 Product Backlog

El Product Backlog sirve como una lista priorizada de las funcionalidades y tareas que se deben abordar en el desarrollo de la página web. Esta herramienta esencial garantiza que todas las características clave sean identificadas y tratadas en orden de prioridad, asegurando así un proceso de desarrollo eficiente y enfocado.

Tabla 4: Iteration Backlog

Ítem	Descripción	Prioridad
1. Autenticación de Usuarios	Implementar la autenticación por correo y contraseña para acceder a la página web. Establecer roles de usuario (cliente y admin).	Alta
2. Gestión de Usuarios	Desarrollar operaciones CRUD para administrar los datos de los clientes y administradores en el backend.	Alta
3. Gestión de Productos	Crear operaciones CRUD para administrar los productos disponibles en la tienda. Implementar la lógica para agregar productos al carrito de compras y guardarlos en el local storage del navegador.	Alta
4.Página de Business Intelligence (BI)	Desarrollar una página de BI para que los administradores puedan visualizar el stock de productos y las ventas realizadas.	Media
5. Product Manager	Crear una interfaz para que los administradores puedan ingresar nuevas plantas al sistema. Desarrollar una sección para visualizar todas las plantas generadas por un usuario específico.	Media
6.Gestión de Inventarios y Ventas	Implementar lógica para validar y presentar las ventas y el inventario específicamente de cada administrador.	Alta
7. Carrito de Compras	Mejorar la experiencia del usuario en el carrito de compras, incluyendo la visualización de productos, el cálculo del total de la compra y la realización del pedido.	Alta
8. Mejoras de Experiencia de Usuario	Realizar ajustes y mejoras en la interfaz de usuario para garantizar una experiencia fluida y amigable para el usuario. Optimizar el rendimiento de la página web y corregir posibles errores.	Med

3.4 Iteration 1

(28 de febrero - 31 de marzo)

3.4.1 Iteration Planning

Durante la itracion1 Establecer la infraestructura básica del proyecto, incluida la implementación y configuración de la base de datos, así como las funciones y roles por defecto que darán accesibilidad a las acciones. Integrar los componentes de navegación y footer de la página web para permitir la interacción adecuada con los usuarios, y desarrollar la autenticación de usuarios trayendo del servidor los elementos adecuados para asegurar el uso y permisos del usuario.

Tabla 5 - Planificación de la iteración 1

Tarea	Descripción	Responsable	Estado	Inicio	Finalización
Configurar base de datos	Crear la estructura de la base de datos y configurar roles y permisos para la gestión de usuarios.	Adriel Suasnavas	Pendiente	28/02/2024	5/3/2024
Implementar autenticación de usuarios	Desarrollar la lógica para el registro y autenticación de usuarios utilizando correo y contraseña.	Adriel Suasnavas	Pendiente	6/3/2024	12/3/2024
Integrar componentes de navegación y footer	Diseñar y conectar los componentes de navegación y footer para la interacción adecuada con los usuarios.	Adriel Suasnavas	Pendiente	13/03/2024	19/03/2024
Desarrollar funciones y roles por defecto	Establecer funciones y roles por defecto para garantizar la accesibilidad a las acciones básicas.	Adriel Suasnavas	Pendiente	20/03/2024	25/03/2024
Implementar gestión de cookies	Configurar el uso de cookies Express para el manejo de sesiones y persistencia de datos del usuario.	Adriel Suasnavas	Pendiente	26/03/2024	31/03/2024

3.4.2 Iteration Backlog

Tabla 6 - Tabla de usuario 1

Título	Registro de Usuario
ID	DCK-001
Prioridad	Alta
Descripción	Como usuario, quiero poder registrarme en la página web utilizando mi correo electrónico y contraseña.
Criterios de Aceptación	- El usuario debe poder acceder a una página de registro. - El usuario debe ingresar un correo electrónico válido y una contraseña. - El sistema debe validar que El correo electrónico no está registrado previamente. - El sistema debe proporcionar mensajes de error claros para entradas inválidas. - una vez registrado, El usuario debe ser añadido a la base de datos con El rol de cliente.
Tareas	- Diseñar la interfaz de registro. - Implementar la lógica de validación del registro en el frontend. - Desarrollar la funcionalidad para almacenar nuevos usuarios en la base de datos. - Realizar pruebas de registro.

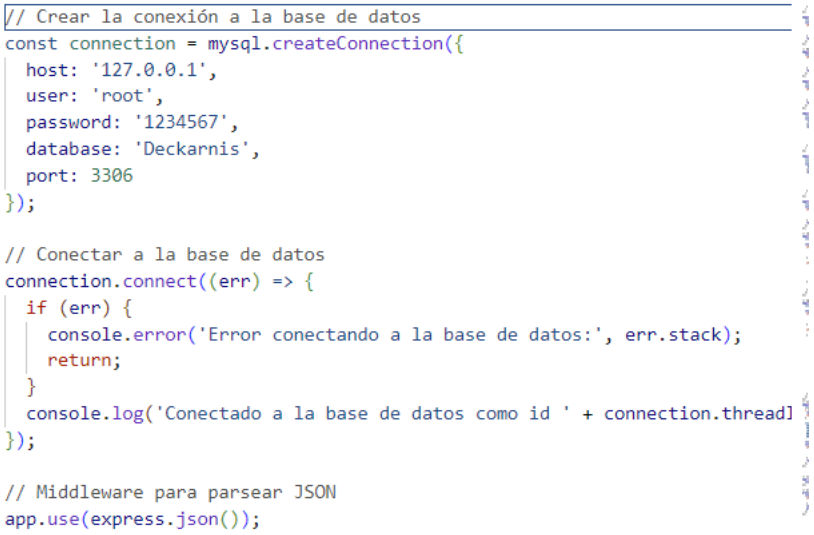
Tabla 7 - Tabla de usuario 2

Título	Registro de Usuario
ID	DCK-002
Prioridad	Alta
Descripción	Como usuario, quiero poder iniciar sesión en la página web utilizando mi correo electrónico y contraseña.
Criterios de Aceptación	- El usuario debe poder acceder a una página de inicio de sesión. - El usuario debe ingresar un correo electrónico registrado y una contraseña. - El sistema debe autenticar al usuario contra la base de datos. - Si la autenticación es exitosa, el usuario debe ser redirigido a la página principal. - El sistema debe proporcionar mensajes de error claros para intentos de inicio de sesión fallidos.
Tareas	- Diseñar la interfaz de inicio de sesión. - Implementar la lógica de autenticación en el frontend. - Desarrollar la funcionalidad para verificar las credenciales del usuario contra la base de datos. - Crear sesiones de usuario y cookies para mantener la sesión. - Realizar pruebas de inicio de sesión.

3.4.3 Realización de las tareas

Conexión a la Base de Datos

Figura 2: Conexión Base de datos (Deckarnis)



```
// Crear la conexión a la base de datos
const connection = mysql.createConnection({
  host: '127.0.0.1',
  user: 'root',
  password: '1234567',
  database: 'Deckarnis',
  port: 3306
});

// Conectar a la base de datos
connection.connect((err) => {
  if (err) {
    console.error('Error conectando a la base de datos:', err.stack);
    return;
  }
  console.log('Conectado a la base de datos como id ' + connection.threadI
});

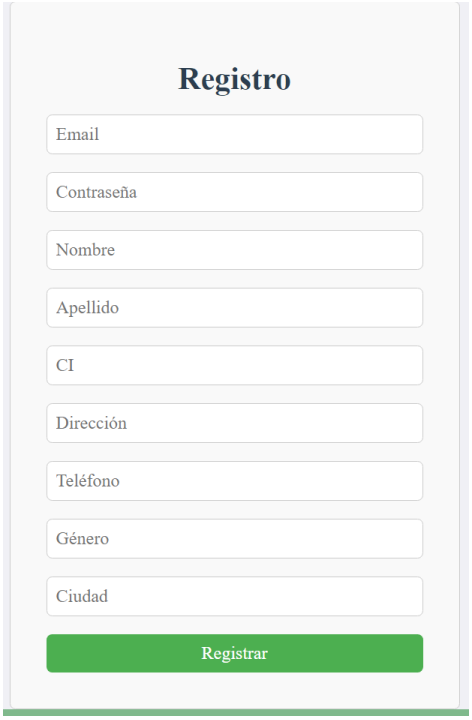
// Middleware para parsear JSON
app.use(express.json());
```

En la Figura 2 se puede ver un fragmento de la conexión a la Base de Datos Deckarnis que fue establecida mediante el uso de la librería mysql de Node.js para interactuar con la base de datos MySQL la cual me permitió realizar la consulta y manejar los resultados de las consultas obtenidas directamente en la Base de Datos.

Diseño de componentes de Registro e inicio de sesión

Se presenta el desarrollo del componente de inicio de sesión, el cual presenta los datos necesarios para el registro, contando también con medidas para asegurar que el registro este completo para poder guardar en la base de datos.

Figura 3: Registro a la Pagina (Deckarnis)



Código de implementación:

Figura 4: sección de código de Registro

```
const handleRegister = async (e) => {
  e.preventDefault();
  try {···
  } catch (error) {
    setError(error.response.data.message);
  }
};

return (
  <div className="authContainer">
    <h2>Registro</h2>
    {error && <p className="error">{error}</p>}
    <form onSubmit={handleRegister}>
      <input
        type="email"
        placeholder="Email"
        value={email}
        onChange={(e) => setEmail(e.target.value)}
        required
      />
      <input
        type="password"
        placeholder="Contraseña"
        value={password}
        onChange={(e) => setPassword(e.target.value)}
        required
      />
      <input
        type="text"
        placeholder="Nombre"
        value={nombre}
        onChange={(e) => setNombre(e.target.value)}
        required
      />
```

La implementación de axios para el registro me permitió poder realizar las solicitudes básicas GET, POST, PUT y DELETE en este caso para la solicitud de registro del usuario con las peticiones HTTP desde el navegador.

Iniciar sesión: El componente login.js se desarrolló por medio de la implementación de cookies para mantener el estado del usuario como autenticado o no autenticado, así como guardar una cookie con el rol del usuario para poder manejar los accesos y permisos de la página.

Figura 5: inicio de sesión a la página (Deckarnis)



Segmento del código de inicio de sesión:

Figura 6: Sección de código de inicio de sesión

```
const handleLogin = async (e) => {
  e.preventDefault();

  try {
    const response = await fetch('http://localhost:3001/api/login', {
      method: 'POST',
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify({ correo: email, clave: password }),
    });

    if (response.ok) {
      const data = await response.json();
      Cookies.set('authToken', data.authToken, { expires: 7 });
      Cookies.set('userName', data.nombreCliente, { expires: 7 });
      Cookies.set('idUsuario', data.idCliente, { expires: 7 });
      Cookies.set('rol', data.rol, { expires: 7 }); // Guardar el rol er

      setAuthenticated(true);
      setUserName(data.nombreCliente);
      setIdUsuario(data.idCliente);
      setRol(data.rol); // Agregado para establecer el rol en el context

      navigate('/home');
    } else {
      console.error('Error al iniciar sesión');
    }
  } catch (error) {
    console.error('Error:', error);
  }
};
```

Lógica de verificación de usuarios

Figura 7: lógica de inicio y rol de usuario

```
import React, { createContext, useState, useEffect } from 'react';
import Cookies from 'js-cookie';

export const AuthContext = createContext();

export const AuthProvider = ({ children }) => {
  const [authenticated, setAuthenticated] = useState(false);
  const [userName, setUserName] = useState('');
  const [idUsuario, setIdUsuario] = useState(null);
  const [rol, setRol] = useState('client')

  useEffect(() => {
    const authToken = Cookies.get('authToken');
    const userNameFromCookie = Cookies.get('userName');
    const idUsuarioFromCookie = Cookies.get('idUsuario');
    const idRolFromCookie = Cookies.get('rol');


    if (authToken && userNameFromCookie && idUsuarioFromCookie) {
      setAuthenticated(true);
      setUserName(userNameFromCookie);
      setIdUsuario(idUsuarioFromCookie);
      setRol(idRolFromCookie)
    }
  }, []);

  return (
    <AuthContext.Provider value={{ authenticated, setAuthenticated, userNa
      {children}
    </AuthContext.Provider>
  );
};
```

Figura 8: Componente de navegación



Figura 9: Navegación en estado de Cliente



Figura 10: Navegación en estado de Administrador



Lógica de obtención de permiso y accesos según el rol

Figura 11: Sección de código para permisos de la pagina

```
</li>
{rol === 'admin' && ( // Mostrar enlaces solo si el rol es
  <>
    <li>
      <Link to='/BI' className='linkNav'> {/* Agregar enla
        BI
      </Link>
    </li>
    <li>
      <Link to='/addPlant' className='linkNav'> {/* Agrega
        Product Manager
      </Link>
    </li>
  </>
)}
</ul>
</div>
<ul>
  {authenticated && (
    <div className='auth'>
      <button onClick={handleLogout} className='authBtn'>
        <IoLogOutSharp /> Cerrar Sesión
      </button>
```

Figura 12: Sección de código para autenticación y rol de usuario

```
// Endpoint para iniciar sesión
app.post('/api/login', (req, res) => {
  const { correo, clave } = req.body;

  connection.query('SELECT l.idUsuario, c.idCliente, c.nombreCli, l.rol FROM login l JOIN (
    if (err) {
      console.error('Error al consultar la base de datos:', err.stack);
      res.status(500).send('Error al consultar la base de datos');
      return;
    }
```

Creación del footer

Este al ser un componente importante para tener navegabilidad en el programa se implementó un footer sencillo que presente lo necesario para la movilidad dentro de la página.

Figura 13: componente footer



Sección del código del Footer

Figura 14: Sección de código del footer

```
import './css/footer.css';
const footer = () => {
  return (
    <>
      <div className='footer'>
        <div className='container'>
          <div className='about'>
            <div className='logo'>
              <img src='./img/logoDeckarnis.jpg' alt='logo'></img>
            </div>
            <div className='detail'>
              <p>El hobby de las plantas carnivoras me ha hecho cultova
              <div className='iconf'>
                <li><FaInstagram /></li>
                <li><FaFacebook /></li>
                <li><FaWhatsapp /></li>
              </div>
            </div>
          </div>
          <div className='account'>
            <h3>Mi cuenta</h3>
            <ul>
              <li>account</li>
              <li>Order</li>
              <li>Cart</li>
              <li>Shipping</li>
            </ul>
          </div>
          <div className='page'>
            <h3>Menu</h3>
            <ul>
              <li>Home</li>
              <li>About</li>
              <li>Contact</li>
              <li>Terminos y condiciones</li>
```

En esta sección se obtienen los permisos y

Figura 15: lógica de inicio y rol de usuario

```
// Endpoint para iniciar sesión
app.post('/api/login', (req, res) => {
  const { correo, clave } = req.body;

  connection.query('SELECT l.idUsuario, c.idCliente, c.nombreCli, l.rol FROM login l JOIN (
    if (err) {
      console.error('Error al consultar la base de datos:', err.stack);
      res.status(500).send('Error al consultar la base de datos');
      return;
    }
```

Figura 16: Componente sobre nosotros



3.4.3 Iteration Review

La realización de la iteración 1 ha sido extenso por la recopilación de los datos y la mantención de los estados en las cookies lo cual han reflejado en una complicación en el estado de los componentes que necesitan los estados de los usuarios para sus permisos y acciones dentro de la página.

Se van a realizar pruebas para asegurar que los estados y la información recibida desde el servidor sea el correcto y se almacene en las cookies de manera adecuada, estas pruebas deben hacer mediante la utilización de console.log lo que nos va a permitir ver en consola las respuestas del servidor y así asegurar que los estados vayan de manera correcta.

3.4.4 Iteration Test

Tabla 8 - Pruebas y resultados de la iteración

Historia de Usuario	Tareas	Test	Resultado Esperado
Como desarrollador, quiero configurar la base de datos para almacenar la información de los usuarios y productos.	Implementar y configurar la base de datos.	- Verificar que la base de datos se haya creado correctamente.- Asegurar que las tablas y relaciones estén configuradas adecuadamente.	La base de datos se crea y configura correctamente con todas las tablas y relaciones necesarias.
	Incluir funciones y roles por defecto en la base de datos.	- Verificar que los roles y funciones por defecto estén presentes en la base de datos.- Probar la asignación de roles a usuarios nuevos y existentes.	Los roles y funciones por defecto están presentes y se asignan correctamente a los usuarios.
Como usuario, quiero poder navegar por el sitio web.	Integrar los componentes de navegación (navbar).	- Verificar que el navbar esté presente en todas las páginas.- Probar la navegación entre diferentes secciones del sitio web utilizando el navbar.	El navbar está presente y funcional en todas las páginas, permitiendo una navegación fluida.
	Integrar los componentes de footer.	- Verificar que el footer esté presente en todas las páginas.- Asegurar que los enlaces en el footer funcionen correctamente.	El footer está presente y todos los enlaces funcionan correctamente.
	Diseñar los componentes de la página.	- Verificar que todos los componentes de la página estén diseñados y estilizados adecuadamente.- Probar la coherencia visual en todas las páginas.	Todos los componentes de la página están correctamente diseñados y estilizados, manteniendo la coherencia visual.
Como usuario, quiero poder registrarme y autenticarse.	Desarrollar la autenticación de usuarios (correo y contraseña).	- Verificar el registro de nuevos usuarios.- Probar el inicio de sesión con usuarios registrados.- Asegurar que las credenciales se validen correctamente.	Los usuarios pueden registrarse y autenticarse correctamente con sus credenciales.
	Implementar el sistema de roles (cliente y admin).	- Verificar que los usuarios se registran como clientes por defecto.- Asegurar que solo los administradores pueden ser modificados desde la base de datos.	Los usuarios se registran como clientes por defecto y los roles de administrador se gestionan correctamente desde la base de datos.

3.5 Iteration 2

(1 de abril - 30 de abril)

3.5.1 Iteration Planning

Tabla 9 - Planificación de la iteración 2

Tarea	Descripción	Responsable	Estado	Inicio	Finalización
Desarrollar CRUD de clientes	Crear la funcionalidad para la gestión de clientes (Crear, Leer, Actualizar, Eliminar).	Adriel Suasnavas	Pendiente	15/03/2024	19/03/2024
Desarrollar CRUD de productos	Implementar las operaciones de gestión de productos en la base de datos.	Adriel Suasnavas	Pendiente	20/03/2024	24/03/2024
Implementar carrito de compras	Desarrollar la funcionalidad del carrito de compras con almacenamiento en local storage.	Adriel Suasnavas	Pendiente	25/03/2024	29/03/2024
Crear interfaz de administración	Diseñar y desarrollar la interfaz de administración para la gestión de productos y ventas.	Adriel Suasnavas	Pendiente	30/03/2024	4/4/2024
Implementar validaciones de productos	Desarrollar las validaciones necesarias para la gestión de productos en el carrito de compras.	Adriel Suasnavas	Pendiente	5/4/2024	9/4/2024

Desarrollar el CRUD para los clientes y productos, incluyendo la implementación del carrito de compras con almacenamiento en local storage, y la interfaz para los administradores que permita la gestión de productos y ventas.

3.5.2 Iteration Backlog

Tabla 10 - Pruebas y resultados de la iteración

Título	Registro de Usuario
ID	DCK-002
Prioridad	Alta
Descripción	Como usuario, quiero poder registrarme en la página web utilizando mi correo electrónico y contraseña.
Criterios de Aceptación	- El usuario debe poder acceder a una página de registro. - El usuario debe ingresar un correo electrónico válido y una contraseña. - El sistema debe validar que el correo electrónico no está registrado previamente. - El sistema debe proporcionar mensajes de error claros para entradas inválidas. - Una vez registrado, el usuario debe ser añadido a la base de datos con el rol de cliente.
Tareas	- Diseñar la interfaz de registro. - Implementar la lógica de validación del registro en el frontend. - Desarrollar la funcionalidad para almacenar nuevos usuarios en la base de datos. - Realizar pruebas de registro.

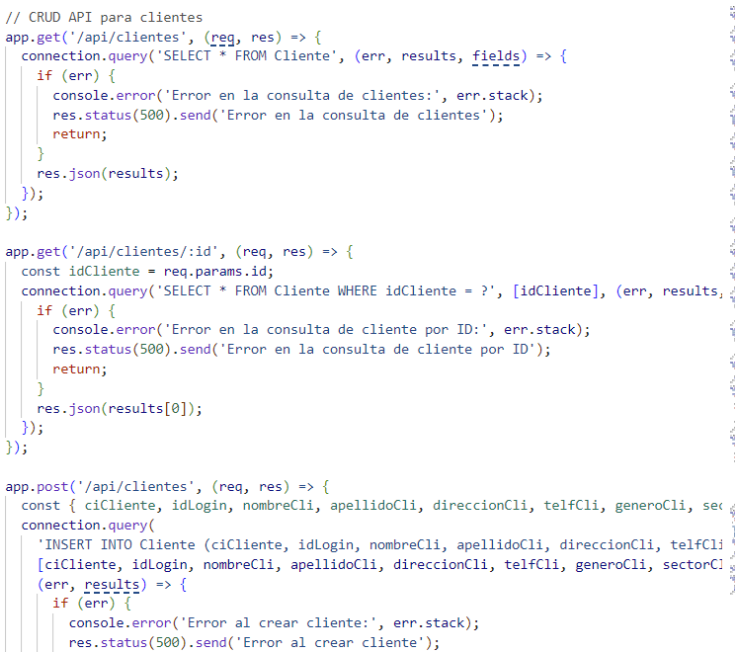
3.5.3 Realización de las tareas

CRUD de los clientes

Se realizó para la implementación del registro y actualización de usuarios, la realización del CRUD de los clientes fue únicamente con el objetivo de uso de la página, dentro del servidor se encuentran todas las funciones para el Create, Read, Update y Delete pero fueron agregadas pero limitadas por cuestiones de información.

Sección de código del servidor para la API de Clientes

Figura 17: API de clientes



```
// CRUD API para clientes
app.get('/api/clientes', (req, res) => {
  connection.query('SELECT * FROM Cliente', (err, results, fields) => {
    if (err) {
      console.error('Error en la consulta de clientes:', err.stack);
      res.status(500).send('Error en la consulta de clientes');
      return;
    }
    res.json(results);
  });
});

app.get('/api/clientes/:id', (req, res) => {
  const idCliente = req.params.id;
  connection.query('SELECT * FROM Cliente WHERE idCliente = ?', [idCliente], (err, results,
    if (err) {
      console.error('Error en la consulta de cliente por ID:', err.stack);
      res.status(500).send('Error en la consulta de cliente por ID');
      return;
    }
    res.json(results[0]);
  });
});

app.post('/api/clientes', (req, res) => {
  const { ciCliente, idLogin, nombreCli, apellidoCli, direccionCli, telfCli, generoCli, se
  connection.query(
    'INSERT INTO Cliente (ciCliente, idLogin, nombreCli, apellidoCli, direccionCli, telfCli
    [ciCliente, idLogin, nombreCli, apellidoCli, direccionCli, telfCli, generoCli, sectorCl
    (err, results) => {
      if (err) {
        console.error('Error al crear cliente:', err.stack);
        res.status(500).send('Error al crear cliente');
```

Desarrollo CRUD de los productos

En la realización del CRUD para los productos fueron devididos para los usuarios y los administradores pues estos deber ser capaces de realizar sus funciones en base a su rol, los clientes pueden ver los productos a diferencia de los administradores que pueden realizar todo el CRUD de sus productos.

Sección de código del servidor para a API de productos

Figura 18 – API de productos

```
});

app.get('/api/plantas/:id', (req, res) => {
  const idPlanta = req.params.id;
  connection.query('SELECT * FROM Planta WHERE idPlanta = ?', [idPlanta], (err, results,
    if (err) {
      console.error('Error en la consulta de planta por ID:', err.stack);
      res.status(500).send('Error en la consulta de planta por ID');
      return;
    }
    res.json(results[0]);
  });
});

app.post('/api/plantas', (req, res) => {
  const { nombrePlanta, tipoPlanta, descripcionPlanta, precioPlanta, stockPlanta, imgPlan
  connection.query(
    'INSERT INTO Planta (nombrePlanta, tipoPlanta, descripcionPlanta, precioPlanta, stock
    [nombrePlanta, tipoPlanta, descripcionPlanta, precioPlanta, stockPlanta, imgPlanta, i
    (err, results) => {
      if (err) {
        console.error('Error al crear planta:', err.stack);
        res.status(500).send('Error al crear planta');
        return;
      }
      res.status(201).send('Planta creada correctamente');
    }
  );
});

app.delete('/api/plantas/:id', (req, res) => {
  const idPlanta = req.params.id;
  connection.query('DELETE FROM Planta WHERE idPlanta = ?', [idPlanta], (err, results) =>
    if (err) {
      console.error('Error al eliminar planta:', err.stack);
      res.status(500).send('Error al eliminar planta');
      return;
```

Implementar carrito de compras

El carrito de compras en el servidor y en el front en esta relacionado con la tabla Compras la cual mantiene los datos relevantes de las ventas presentes y basándose en el carrito de los usuarios.

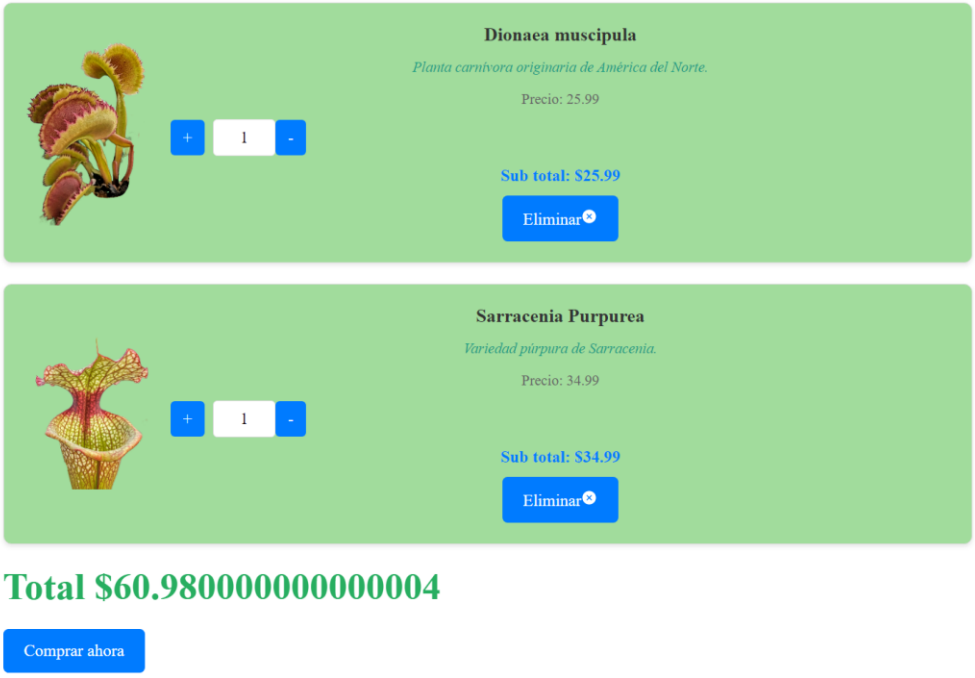
Sección del código del servidor para compras

Figura 19 : Servidor de compras

```
// Rutas para el stock y las compras
app.put('/api/plantas/:idPlanta/stock', (req, res) => {
  const { idPlanta } = req.params;
  const { stockPlanta } = req.body;
  connection.query('UPDATE Planta SET stockPlanta = ? WHERE idPlanta = ?', [stockPlanta,
    if (err) {
      console.error('Error al actualizar stock:', err.stack);
      res.status(500).send('Error al actualizar stock');
      return;
    }
    res.status(200).send('Stock actualizado');
  });
});

app.post('/api/compra', (req, res) => {
  const { idCliente, idProducto, cantidad, idUsuario } = req.body;
  connection.query(
    'INSERT INTO Compra (idCliente, idProducto, cantidad, idUsuario) VALUES (?, ?, ?, ?)
    [idCliente, idProducto, cantidad, idUsuario],
    (err, results) => {
      if (err) {
        console.error('Error al registrar compra:', err.stack);
        res.status(500).send('Error al registrar compra');
        return;
```

Figura 20: Carrito de compras

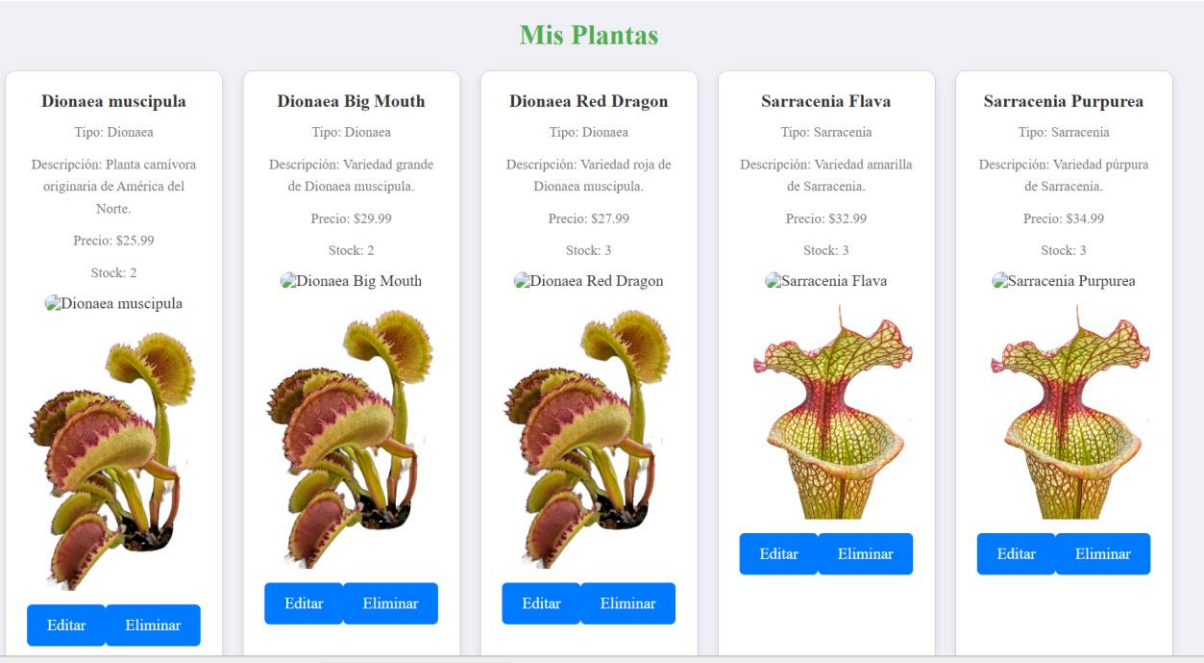


Creación de interfaz de administradores

Los administradores deben tener la opción de poder gestionar de manera adecuada sus productos presentes, así como el de poder visualizar cuales son sus venas y su stock disponible.

Interfaz de administración de productos

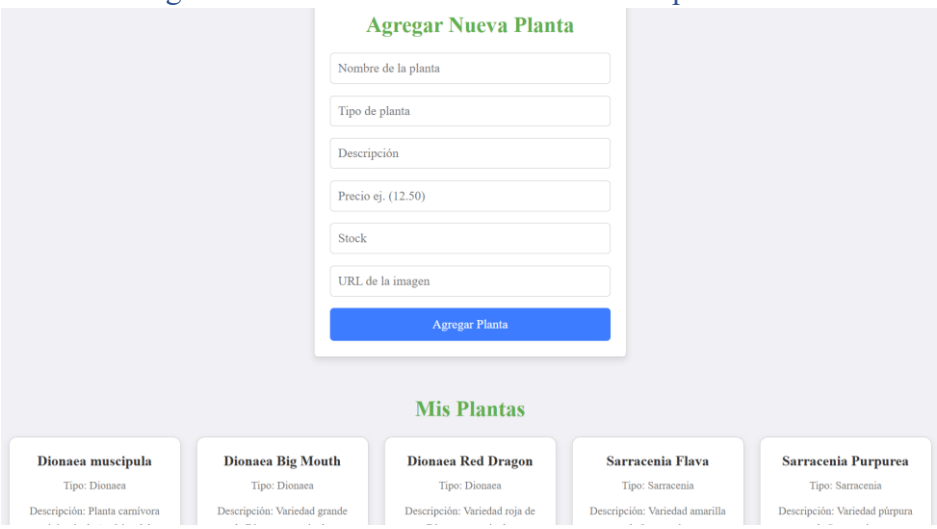
Figura 21: interfaz de administración de productos



Interfaz control de almacenamiento

Se implemento el control de almacenamiento en donde cada usuario identificado como admin podrá gestionar sus productos y no los de otro usuario administrador.

Figura 22: control de almacenamiento de productos



Validación inventario de productos

La validación de los productos está basada en la supervisión del stock de una planta pues una compra no puede llegar a ser negativo ni tampoco debe llegar a ser mayor a la cantidad del stock por lo tanto las validaciones se realizan sobre el hecho de que in cliente no puede agregar una planta si esta no está disponible, de la misma un producto no puede ser negativo dentro del carrito ese debe ser eliminado del carrito al no tener interés en el mismo.

Figura 23: Ejemplo validación de productos

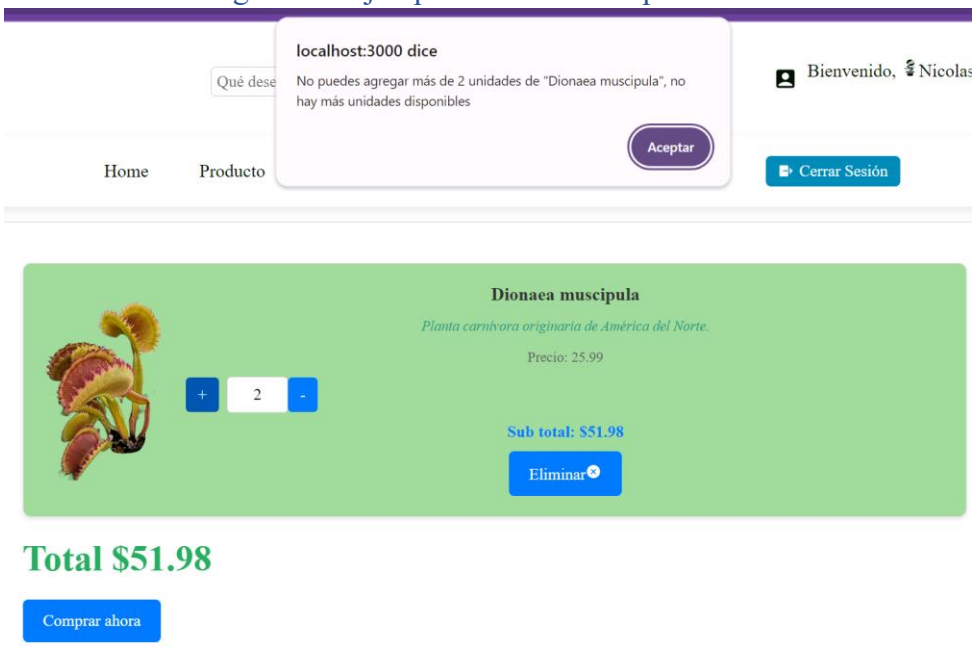
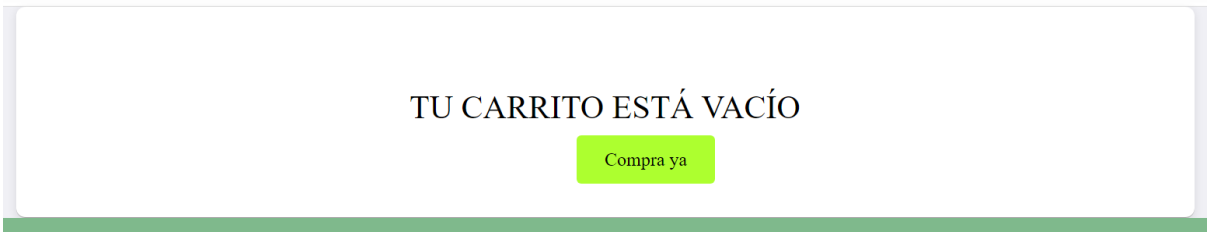


Figura 24: validación de no tener productos negativos



En esta figura e resta los productos hasta 1 de tal manera que si resta el uno se elimina el producto a 0.

3.5.3 Iteration Review

La realización de la iteración 2 se ha centrado en la implementación de funcionalidades adicionales y en la corrección de errores identificados durante la iteración anterior. Se ha logrado avanzar significativamente en la gestión del carrito de compras y en la interfaz de administración de productos. Sin embargo, se han encontrado desafíos adicionales relacionados con la persistencia del carrito y la integración de gráficos de Business Intelligence (BI). Se han planificado pruebas para asegurar que las nuevas funcionalidades funcionen correctamente y se almacenen los datos de manera adecuada. Se espera que estas pruebas contribuyan a mejorar la estabilidad y el rendimiento del sistema.

3.5.4 Iteration Test

Tabla 11 - Pruebas y resultados de la iteración 2

Historia de Usuario	Tareas	Test	Resultado Esperado
Como usuario, quiero poder agregar productos al carrito de compras.	Implementar la funcionalidad de agregar productos.	- Verificar que se pueden agregar productos al carrito. - Probar la actualización del carrito en tiempo real.	Los productos se agregan correctamente al carrito y se actualizan en tiempo real.
Almacenar el carrito en local storage.	- Verificar que el carrito se almacene correctamente en local storage. - Probar la persistencia del carrito al recargar la página.	El carrito se almacena y persiste correctamente en el local storage.	
Como usuario, quiero que mi carrito de compras se conserve al recargar la página.	Implementar persistencia del carrito	- Verificar que el carrito se conserve después de recargar la página. - Probar agregar y eliminar productos y luego recargar la página.	El carrito conserva su estado correctamente después de recargar la página, incluyendo cualquier cambio realizado.
Como administrador, quiero poder gestionar los productos (añadir, editar, eliminar).	Desarrollar interfaz para gestión de productos	- Verificar que la interfaz de gestión de productos sea intuitiva. - Probar añadir, editar y eliminar productos desde la interfaz.	La interfaz de gestión de productos es intuitiva y permite añadir, editar y eliminar productos correctamente.
Implementar funciones CRUD para productos	- Verificar que las operaciones CRUD funcionen correctamente en la base de datos. - Probar la integridad de los datos después de cada operación.	Las operaciones CRUD funcionan correctamente en la base de datos, manteniendo la integridad de los datos.	

3.6 Iteration 3

(1 de mayo - 25 de mayo)

3.6.1 Iteration Planning

Tabla 12 - Planificación de la iteración 3

Tarea	Descripción	Responsable	Estado	Inicio	Finalización
Desarrollar gráficos de BI	Crear gráficos para mostrar el stock y las ventas de los productos principales.	Adriel Suasnavas	Pendiente	10/4/2024	14/04/2024
Integrar gráficos en la interfaz	Integrar los gráficos de BI en la interfaz de administración.	Adriel Suasnavas	Pendiente	15/04/2024	19/04/2024
Realizar pruebas exhaustivas	Llevar a cabo pruebas exhaustivas de todas las funcionalidades de la aplicación.	Adriel Suasnavas	Pendiente	20/04/2024	24/04/2024
Corregir errores	Identificar y corregir los errores encontrados durante las pruebas.	Adriel Suasnavas	Pendiente	25/04/2024	29/04/2024
Documentar el proyecto	Documentar todas las funcionalidades y preparar el informe final del proyecto.	Adriel Suasnavas	Pendiente	30/04/2024	5/5/2024

3.6.2 Iteration Backlog

Tabla 13 – Historias de usuario iteración 3

Título	Registro de Usuario
ID	DCK-003
Prioridad	Alta
Descripción	Como usuario, quiero poder registrarme en la página web utilizando mi correo electrónico y contraseña.
Criterios de Aceptación	- El usuario debe poder acceder a una página de registro. - El usuario debe ingresar un correo electrónico válido y una contraseña. - El sistema debe validar que el correo electrónico no está registrado previamente. - El sistema debe proporcionar mensajes de error claros para entradas inválidas. - Una vez registrado, el usuario debe ser añadido a la base de datos con el rol de cliente.
Tareas	- Diseñar la interfaz de registro. - Implementar la lógica de validación del registro en el frontend. - Desarrollar la funcionalidad para almacenar nuevos usuarios en la base de datos. - Realizar pruebas de registro.

3.6.3 Realización de las tareas

Desarrollar Gráficos para BI

El uso de librerías dentro de React me ha permitido implementar más funcionalidades para la página así como los gráficos presentas para que cada usuario administrador pueda tener un manejo sobre sus ventas y sus productos.

Figura 25: código para gráficos de BI

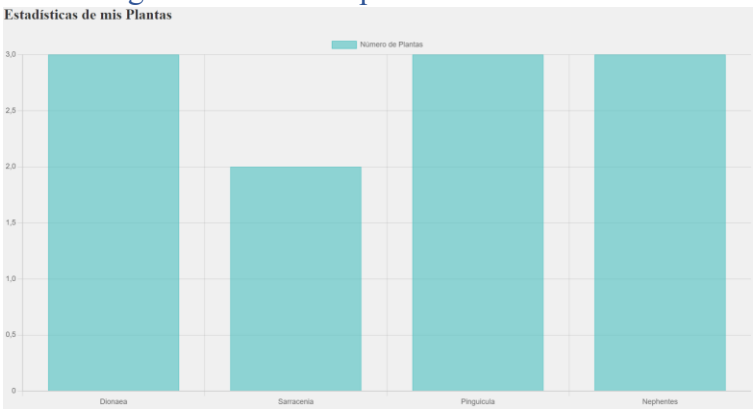
```
urn (
div className="bi-container">
 <h2>Estadísticas de mis Plantas</h2>
 {plantChartData.labels && plantChartData.labels.length > 0 ? (
   <Bar data={plantChartData} options={{ responsive: true, plugins: { l
 ) : (
   <p>No hay datos suficientes para mostrar el gráfico de plantas.</p>
 )}

 <h2>Estadísticas de mis Compras</h2>
 {purchaseChartData.labels && purchaseChartData.labels.length > 0 ? (
   <Bar data={purchaseChartData} options={{ responsive: true, plugins:
 ) : (
   <p>No hay datos suficientes para mostrar el gráfico de compras.</p>
 )}

 {purchases.length > 0 && (
```

Integrar gráficos en la interfaz

Figura 25: Gráficos presentes en la interfaz





Pruebas finales

Se pronunciaron 2 error de baja calidad en el momento de las pruebas finales, mediante la continua revisión se encontró un problema visual que es la presentación de los valores finales al cliente que es la suma de la cantidad y la suma de los valores que se presentan

Figura 26: Error en cantidad de decimales

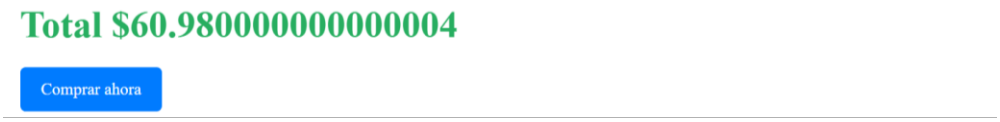


Figura 27: Error en funcionalidad del botón



Cabe recalcar que los errores no están presentes en gran cantidad pues la metodología escogida presenta pruebas dentro de cada iteración hecha y se implementa la prueba al componente realizado en cada iteración lo cual me ha hecho presentar una diminución de errores en las pruebas finales.

Corrección de Errores

Figura 28: Resolución de la cantidad de decimales

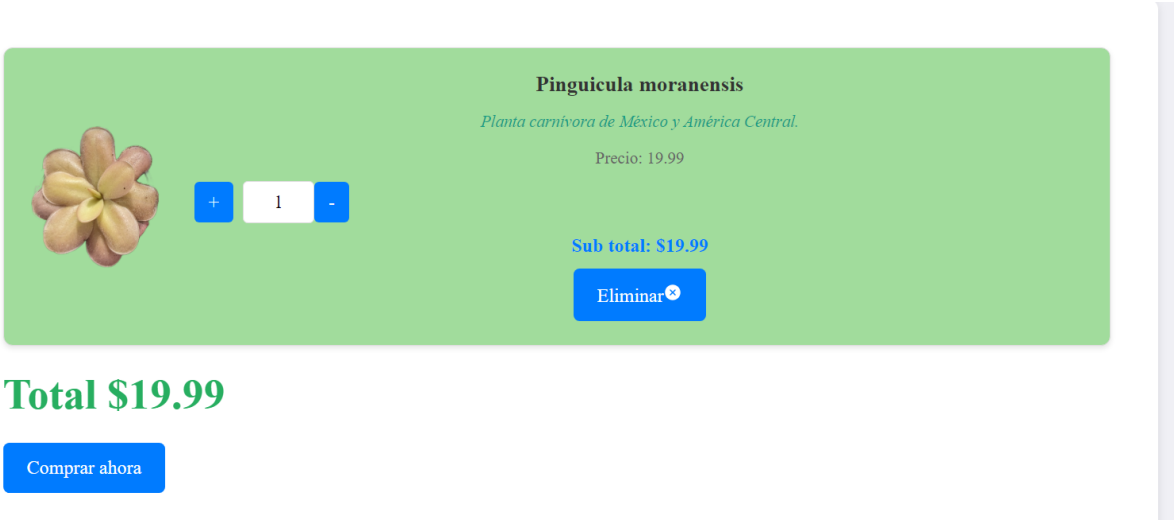


Figura 29: Código para limitar los decimales

```
const precioTotal = cart.reduce((precio, item) => precio + item.qty * item.precioPlanta, 0).toFixed(2);
```

La funcionalidad para agregar los productos que se presentaban en la view se presentó por medio de agregar el código dentro del botón lo cual hace que se solucione el error presente ya agregando al carrito.

Figura 30: falla de agregar al carrito



Se presento la solución limitando los decimales en el código para que sea preciso y no haya confusión con el usuario

Figura 31: Código faltante para cumplir la funcionalidad

```
    <h4>Sub total: ${curElm.precioPlanta * curElm.qty}</h4>
    <button onClick={() => eliminarProducto(curElm)}>Eliminar<IoCloseCircle /></button>
</div>
```

Documento Final

El documento final se ha realizado con el Avance del proyecto y será presentando el día 2 de junio para su revisión.

3.6.3 Iteration Review

La iteración 3 ha estado enfocada en la finalización de las funcionalidades principales del sistema, incluyendo la integración completa de gráficos de Business Intelligence para ventas y stock de productos, así como la optimización de la gestión de productos por parte de los administradores. Se han realizado pruebas adicionales para garantizar que todas las características cumplan con los requisitos especificados y se han corregido los errores identificados durante las pruebas anteriores. A pesar de algunos desafíos encontrados durante el desarrollo, se ha logrado completar con éxito las tareas planificadas y se espera que el sistema esté listo para su lanzamiento final.

3.6.4 Iteration Test

Tabla 14 – Pruebas de iteración 3

Historia de Usuario	Tareas	Test	Resultado Esperado
Como administrador, quiero poder ver gráficos de ventas y stock.	Implementar gráficos de BI para ventas	- Verificar que los gráficos de ventas muestren datos precisos. - Probar la actualización de los gráficos con nuevos datos de ventas.	Los gráficos de ventas muestran datos precisos y se actualizan correctamente con nuevos datos de ventas.
Implementar gráficos de BI para stock de productos	- Verificar que los gráficos de stock muestren datos precisos. - Probar la actualización de los gráficos con cambios en el stock de productos.	Los gráficos de stock muestran datos precisos y se actualizan correctamente con cambios en el stock de productos.	
Como administrador, quiero poder gestionar los productos (añadir, editar, eliminar).	Desarrollar interfaz para gestión de productos	- Verificar que la interfaz de gestión de productos sea intuitiva. - Probar añadir, editar y eliminar productos desde la interfaz.	La interfaz de gestión de productos es intuitiva y permite añadir, editar y eliminar productos correctamente.
Implementar funciones CRUD para productos	- Verificar que las operaciones CRUD funcionen correctamente en la base de datos. - Probar la integridad de los datos después de cada operación.	Las operaciones CRUD funcionan correctamente en la base de datos, manteniendo la integridad de los datos.	
Implementar validaciones y cálculos de totales	- Verificar que todas las validaciones se realicen correctamente. - Probar la precisión de los cálculos de totales y la actualización en tiempo real.	Las validaciones se realizan correctamente y los cálculos de totales son precisos y se actualizan en tiempo real.	La validaciones se realizan de manera adecuada y muestran los claculos individuales y los calculos totales de los productos.

4. Análisis de Resultados

Durante el desarrollo de la página web para la venta de plantas, se implementaron diversas funcionalidades que cumplieron con los requisitos y objetivos iniciales establecidos en la fase de análisis de requerimientos como la:

Autenticación y Gestión de Usuarios

Se desarrolló y probó exitosamente el sistema de registro y autenticación de usuarios. La validación de correos electrónicos y contraseñas se realizó conforme a los estándares de seguridad previstos. Las pruebas de carga mostraron que el sistema puede manejar múltiples registros y autenticaciones simultáneamente sin afectar significativamente el rendimiento.

Interfaz de Usuario

La integración de componentes como el menú de navegación y el footer se completó satisfactoriamente. Las encuestas de usabilidad indicaron que los usuarios encuentran la interfaz intuitiva y fácil de usar. Las métricas de interacción (tiempo de sesión, páginas por sesión) sugieren un nivel de compromiso adecuado con la plataforma.

CRUD de Productos

La funcionalidad de gestión de productos (crear, leer, actualizar, eliminar) para administradores se implementó y verificó mediante pruebas unitarias y de integración. Se observó que las operaciones CRUD se realizan de manera eficiente y los datos se mantienen consistentes en la base de datos.

Persistencia del Carrito de Compras

El uso de local storage para la persistencia del carrito de compras se validó mediante pruebas de recarga y cierre de sesión. Los datos del carrito se conservaron de manera confiable, mejorando la experiencia de usuario al permitir la continuidad de la compra.

Inteligencia de Negocios (BI)

Los gráficos de ventas y stock proporcionados para los administradores mostraron ser herramientas efectivas para la toma de decisiones. Se utilizaron librerías de React, como Recharts y Chart.js, para la visualización de datos. Los gráficos se actualizaron correctamente con los datos en Linea y presentaron la información de manera clara y concisa.

Control de Inventario

Se implementó un sistema de control de inventario basado en la metodología FIFO (First In, First Out), que asegura que las plantas más antiguas se vendan primero, optimizando así la gestión del inventario. El manejo de stock se realizó con precisión, permitiendo a los administradores ver y gestionar las cantidades de productos disponibles. Las pruebas de funcionalidad confirmaron que el sistema de inventario mantiene la integridad de los datos y proporciona actualizaciones precisas en Linea.

Cookies y Sesiones

La implementación de cookies y manejo de sesiones a través de Express.js permitió mantener la sesión del usuario activa y segura. Las pruebas de seguridad revelaron que los datos de sesión están adecuadamente protegidos contra accesos no autorizados.

5. Conclusiones y Recomendaciones

5.1 Conclusiones

La implementación del prototipo web permitió una gestión integrada y eficiente de ventas y control de inventario en viveros. La utilización de soluciones tecnológicas específicas mejoró significativamente la precisión y actualización del inventario, reduciendo ineficiencias operativas y permitiendo una mejor toma de decisiones basada en datos en Linea.

Se logró desarrollar una interfaz amigable y accesible tanto para administradores como para usuarios finales. Las herramientas de análisis de datos, incluyendo gráficos desarrollados con librerías de React como Recharts y Chart.js, facilitaron la visualización y comprensión de información crucial sobre ventas y stock, contribuyendo a una gestión más eficaz del negocio.

La adopción del prototipo web tuvo un impacto positivo en la rentabilidad y competitividad de los viveros. La capacidad de gestionar el inventario con la metodología FIFO y mantener un control preciso de stock y ventas permitió a los viveros optimizar sus operaciones y satisfacer mejor las demandas del mercado. Esta modernización tecnológica posiciona a los viveros en un lugar favorable dentro del mercado ecuatoriano.

La plataforma desarrollada mejoró significativamente la experiencia de los usuarios, tanto clientes como administradores. Los clientes pudieron realizar compras de manera más fácil y segura, mientras que los administradores tuvieron acceso a herramientas eficientes para la gestión del negocio, lo que resultó en un aumento de la satisfacción del cliente y una gestión más efectiva del vivero.

El desarrollo de un prototipo web para la comercialización de plantas en viveros en Ecuador ha demostrado ser una solución efectiva para mejorar la gestión de ventas e inventario, satisfacer las necesidades de los clientes y aumentar la competitividad de los viveros. La integración de tecnologías emergentes y herramientas de análisis de datos ha permitido optimizar las operaciones y tomar decisiones informadas, marcando un hito en la modernización del sector.

5.2 Recomendaciones

Es esencial continuar con la optimización y actualización del sistema basado en el feedback de los usuarios y en el avance de las tecnologías. Se recomienda realizar actualizaciones periódicas para incorporar nuevas funcionalidades y mejorar la seguridad y eficiencia del sistema.

Proporcionar capacitación continua y detallada a los administradores y usuarios del sistema es crucial para asegurar un uso eficaz de todas las funcionalidades disponibles. Un soporte técnico eficiente también es necesario para resolver cualquier problema que surja y asegurar un funcionamiento continuo del sistema.

Fortalecer las medidas de seguridad para proteger los datos de los usuarios y la información del negocio es fundamental. Implementar técnicas avanzadas de encriptación y realizar auditorías de seguridad periódicas ayudará a prevenir posibles brechas y mantener la integridad del sistema.

Planificar y desarrollar el sistema de manera escalable para adaptarse al crecimiento del negocio y a las necesidades cambiantes del mercado. Considerar la integración de nuevas tecnologías y herramientas que puedan surgir, asegurará que el sistema permanezca relevante y eficiente a largo plazo.

Para asegurar el éxito continuo y el crecimiento del sistema, es fundamental mantener una actitud proactiva en la optimización y actualización del prototipo web. Fomentar la adopción de nuevas tecnologías, proporcionar soporte continuo a los usuarios y garantizar la seguridad de los datos permitirá que el sistema siga siendo una herramienta valiosa para los viveros en Ecuador, mejorando su eficiencia y competitividad en el mercado.

Bibliography

A., D. (29 de Junio de 2023). *HostingerTUTORIALES*. Obtenido de Qué es React: definición, características y funcionamiento: https://www.hostinger.es/tutoriales/que-es-react

B., G. (19 de Diciembre de 2023). *HostingerTutoriales*. Obtenido de ¿Qué es HTML? Explicación de los fundamentos del Lenguaje de marcado de hipertexto: https://www.hostinger.es/tutoriales/que-es-html

Blaxell, Z. (25 de June de 2006). *Extreme Roles*. Obtenido de Wiki.c2: https://wiki.c2.com/?ExtremeRoles

Cascading Style Sheets: CSS Browsers. (s.f.). Recuperado el 15 de 5 de 2024, de World Wide Web Consortium: http://www.w3.org/Style/CSS/#browsers

Chapter 4. How JavaScript Was Created. (s.f.). Recuperado el 15 de 5 de 2024, de http://speakingjs.com/es5/ch04.html

Extreme Programming Rules. (s.f.). Recuperado el 15 de 5 de 2024, de http://www.extremeprogramming.org/rules.html

Flores, F. (22 de Julio de 2022). *openWebinars*. Obtenido de VISUAL STUDIO CODE que es y que ofrece: https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/

IONOS. (27 de 06 de 2023). *Digital Guide*. Obtenido de What is CSS?: https://www.ionos.com/digitalguide/websites/web-design/what-is-css/

LIZARRAGA, C. (16 de Octubre de 2017). *WordPress*. Obtenido de EXTREME PROGRAMMING: https://iswxp.wordpress.com/2017/10/16/artefactos/

Londono, P. (2022 de enero de 19). *Hubspot*. Obtenido de Qué es MySQL, para qué sirve y características principales: https://blog.hubspot.es/website/que-es-mysql

Mora, S. L. (28 de noviembre de 2023). *Digital55*. Obtenido de Desarrollo React: qué es, ventajas y por qué contratar a un programador React: https://digital55.com/blog/desarrollo-react-ventajas-contratar-programador-react/

Olawanle, J. (monday de 22 de 2022). *FreeCodeCamp*. Obtenido de How to Use localStorage with React Hooks to Set and Get Item: https://www.freecodecamp.org/news/how-to-use-localstorage-with-react-hooks-to-set-and-get-items/

Raeburn, A. (13 de Febrero de 2024). *asana*. Obtenido de La programación extrema (XP) produce resultados, pero ¿es la metodología adecuada para ti?: https://asana.com/es/resources/extreme-programming-xp

SernaGroup. (8 de Marzo de 2021). *SernaGroup*. Obtenido de SernaGrp: https://sernagrp.com/blog/html/

Wake, B. (6 de October de 2021). *XP123*. Obtenido de What Is a User Story?: https://xp123.com/what-is-a-user-story/

What Is JavaScript? (s.f.). Recuperado el 15 de 5 de 2024, de Wiley: http://media.wiley.com/product_data/excerpt/88/07645790/0764579088.pdf

A., D. (29 de Junio de 2023). *HostingerTUTORIALES*. Obtenido de Qué es React: definición, características y funcionamiento: https://www.hostinger.es/tutoriales/que-es-react

B., G. (19 de Diciembre de 2023). *HostingerTutoriales*. Obtenido de ¿Qué es HTML? Explicación de los fundamentos del Lenguaje de marcado de hipertexto: https://www.hostinger.es/tutoriales/que-es-html

Blaxell, Z. (25 de June de 2006). *Extreme Roles*. Obtenido de Wiki.c2: https://wiki.c2.com/?ExtremeRoles

Cascading Style Sheets: CSS Browsers. (s.f.). Recuperado el 15 de 5 de 2024, de World Wide Web Consortium: http://www.w3.org/Style/CSS/#browsers

Chapter 4. How JavaScript Was Created. (s.f.). Recuperado el 15 de 5 de 2024, de http://speakingjs.com/es5/ch04.html

Extreme Programming Rules. (s.f.). Recuperado el 15 de 5 de 2024, de http://www.extremeprogramming.org/rules.html

Flores, F. (22 de Julio de 2022). *openWebinars*. Obtenido de VISUAL STUDIO CODE que es y que ofrece: https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/

IONOS. (27 de 06 de 2023). *Digital Guide*. Obtenido de What is CSS?: https://www.ionos.com/digitalguide/websites/web-design/what-is-css/

LIZARRAGA, C. (16 de Octubre de 2017). *WordPress*. Obtenido de EXTREME PROGRAMMING: https://iswxp.wordpress.com/2017/10/16/artefactos/

Londono, P. (2022 de enero de 19). *Hubspot*. Obtenido de Qué es MySQL, para qué sirve y características principales: https://blog.hubspot.es/website/que-es-mysql

Mora, S. L. (28 de noviembre de 2023). *Digital55*. Obtenido de Desarrollo React: qué es, ventajas y por qué contratar a un programador React: https://digital55.com/blog/desarrollo-react-ventajas-contratar-programador-react/

Olawanle, J. (monday de 22 de 2022). *FreeCodeCamp*. Obtenido de How to Use localStorage with React Hooks to Set and Get Item: https://www.freecodecamp.org/news/how-to-use-localstorage-with-react-hooks-to-set-and-get-items/

Raeburn, A. (13 de Febrero de 2024). *asana*. Obtenido de La programación extrema (XP) produce resultados, pero ¿es la metodología adecuada para ti?: https://asana.com/es/resources/extreme-programming-xp

SernaGroup. (8 de Marzo de 2021). *SernaGroup*. Obtenido de SernaGrp: https://sernagrp.com/blog/html/

Wake, B. (6 de October de 2021). *XP123*. Obtenido de What Is a User Story?: https://xp123.com/what-is-a-user-story/

What Is JavaScript? (s.f.). Recuperado el 15 de 5 de 2024, de Wiley: http://media.wiley.com/product_data/excerpt/88/07645790/0764579088.pdf