



Centro Industrial y Desarrollo
Empresarial Soacha
Regional Cundinamarca



Manual Técnico

VineFresh



Aprendices SENA

JOAN SEBASTIAN OLAVE MARIÑO

LING GELEN ROZO RODRIGUEZ

INGRID DANIELA SANCHEZ LARA

CENTRO INDUSTRIAL Y DESARROLLO EMPRESARIAL – CIDE

ANÁLISIS Y DESARROLLO DE SOFTWARE - ADSO

SOACHA, CUNDINAMARCA

2025



1. Introducción
2. Objetivos del Proyecto
 - 2.1 Objetivo general
 - 2.2 Objetivos específicos
3. Alcance del proyecto
4. Personal Involucrado
 - 4.1 Diagramas de Casos de Uso
5. Características de los usuarios
6. Requerimientos
 - 6.1 Requerimientos funcionales
 - 6.2 Requerimientos no funcionales
7. Arquitectura del proyecto
 - 7.1 Definición metodología
 - 7.2 Ciclo de vida
 - 7.2 Diagrama de despliegue
8. Fundamentos y herramientas utilizadas
9. Requisitos del Sistema
10. Procesos del Software
11. Instalación de Aplicaciones
12. Diagrama de Clases
13. Modelo entidad Relación
14. Diagrama entidad relación
15. Diccionario de Datos
16. Glosario
17. Referencias



Introducción

El presente Manual Técnico del sistema VINEFRESH tiene como objetivo documentar de manera detallada los aspectos técnicos, funcionales y estructurales del software desarrollado como parte del programa Análisis y Desarrollo de Software (ADSO) del SENA.

VINEFRESH es una aplicación orientada a la gestión y control de productos vinícolas, diseñada para optimizar los procesos de registro, inventario, ventas y administración de bodegas de vino. Su propósito es ofrecer una herramienta moderna, intuitiva y eficiente que facilite la trazabilidad y el manejo de información dentro del negocio.

Este manual contiene la descripción técnica del sistema, incluyendo la arquitectura, diagramas, base de datos, procedimientos almacenados, vistas, disparadores, y demás componentes que sustentan su funcionamiento. Además, se detallan los requerimientos técnicos, herramientas utilizadas y la metodología empleada durante las etapas de análisis, diseño, desarrollo y pruebas.

El documento está dirigido a desarrolladores, analistas y personal técnico que requiera comprender, mantener o mejorar el sistema, garantizando así su correcta implementación y evolución dentro del entorno de producción.



Objetivos del Proyecto

Objetivo general

Desarrollar un aplicativo web para una vinícola que ofrezca una experiencia de usuario atractiva y funcional, con una interfaz intuitiva, un catálogo organizado de productos y combos de vinos de alta calidad, optimizada para facilitar la navegación, selección y compra de forma eficiente.

Objetivos específicos

- Analizar la usabilidad y la fluidez del recorrido del usuario, desde la exploración del catálogo hasta la decisión de compra, eliminando fricciones y simplificando los pasos.
- Diseñar una interfaz clara y responsiva que permita a los usuarios navegar fácilmente entre productos y combos, estructurados por categorías y tipos.
- Implementar herramientas que permitan filtrar productos por precio, tipo, disponibilidad u otras características relevantes, optimizando la navegación sin recargar la página.



Alcance del proyecto

El proyecto de un aplicativo web de ventas online tiene como alcance la creación de una plataforma digital que permita a los clientes de todo el mundo adquirir productos de alta calidad de manera segura y eficiente. La página web estará diseñada para ser accesible desde cualquier dispositivo con conexión a internet, permitiendo a los clientes navegar y realizar compras en cualquier momento y lugar.

Personal Involucrado

En el desarrollo del sistema VINEFRESH se contó con la participación activa de aprendices del programa análisis y desarrollo (ADSO), quienes asumieron diferentes roles a lo largo de las fases de análisis, diseño, desarrollo pruebas e implementación. cada integrante aportó sus conocimientos técnicos y compromisos para lograr un producto funcional y alineado con los requerimientos establecidos.

A continuación se detalla el personal involucrado en el proyecto:

NOMBRE	ROL	ENTIDAD/PROGRAMA
Ling Gelen Rozo Rodriguez	Analista y desarrollador de software.	SENA - Programa ADSO.
Joan Sebastian Olave Mariño	Analista y desarrollador de software.	SENA - Programa ADSO.
Ingrid Daniela Sanchez Lara	Analista y desarrollador de software.	SENA - Programa ADSO.
Paola Andrea Gutiérrez Mendieta	Instructor/tutor del proyecto.	SENA - Centro industrial y de desarrollo empresarial de Soacha (CIDE).

Diagramas de Casos de Uso

CUADRO Y FOTO DE LOS CASOS DE USO

CASO DE USO 1 :

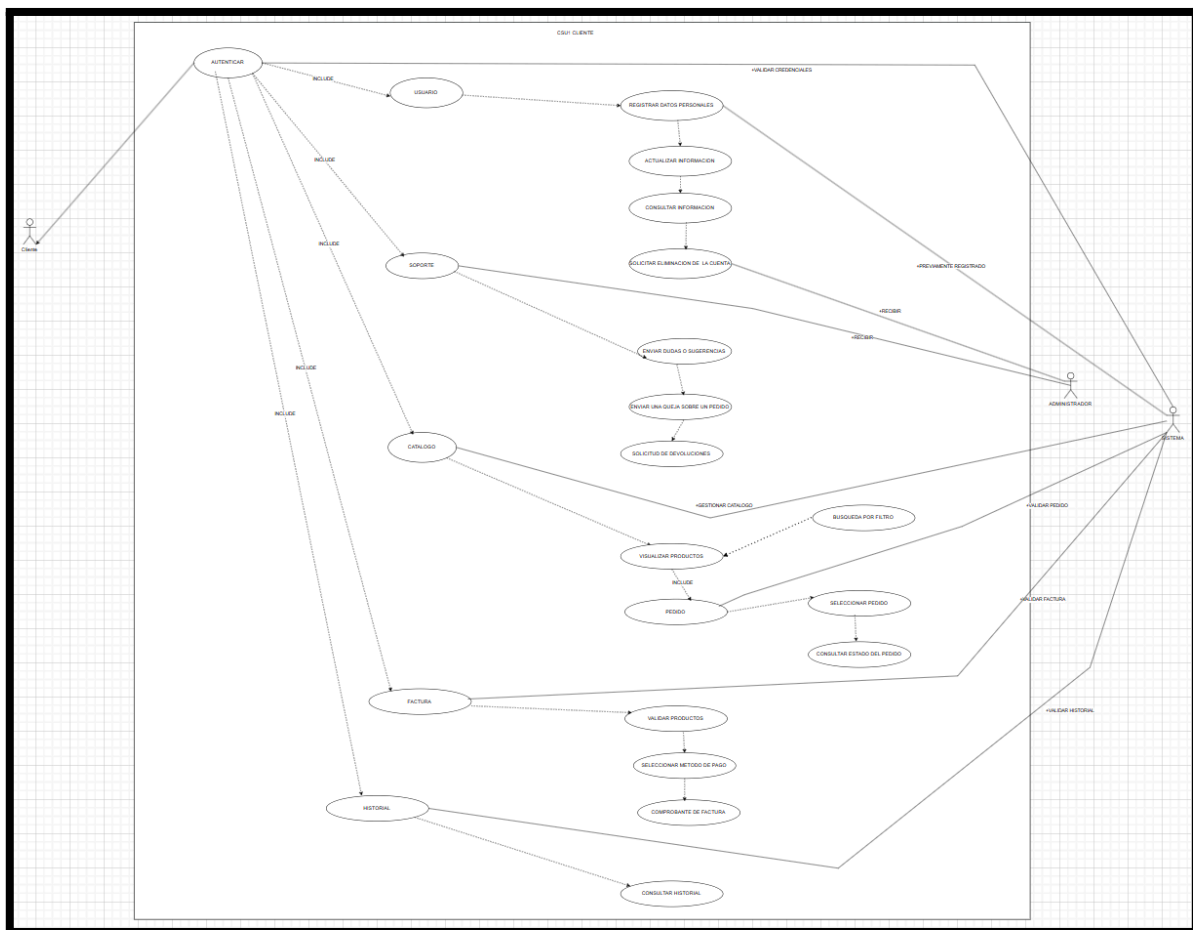


TABLA DE CASO DE USO 1 :

1. 0111IDENTIFICACIÓN DE CASO DE USO			
1.1 Id Caso	CU001	1.2 Nombre	Actividades del cliente en el sitio web vinefresh



1. HISTÓRICO DE CASO DE USO			
2.1 Autor	Ingrid Daniela Sánchez Lara		
2.2 Fecha Creación	12/03/2025	2.3 Última Actualización	19/10/2025
2.4 Actualizado por	Ingrid Daniela Sánchez Lara	2.5 Versión	1.1
1. DEFINICIÓN DE UN CASO DE USO			
3.1 DESCRIPCIÓN			
<p>El presente caso de uso permite evidenciar todas las actividades que el cliente puede realizar dentro del sistema VineFresh, desde que se autentica hasta que accede a los diferentes módulos del sitio web. Incluye acciones como registrar o actualizar información, gestionar pedidos, consultar historial, validar facturas y comunicarse con el soporte o administración.</p>			
3.2 ACTORES			
<p>Cliente: Usuario final que interactúa con el sistema web.</p> <p>Administrador: Encargado de recibir solicitudes, quejas o devoluciones y gestionar los procesos internos del sistema.</p> <p>Sistema: Plataforma tecnológica que valida, registra y gestiona las operaciones realizadas por el cliente.</p>			
3.3 PRECONDICIONES			



Manual Técnico

1. Si el cliente no está registrado, podrá crear una cuenta antes de iniciar sesión.
2. Si el cliente ya está registrado, debe autenticarse correctamente para acceder a las funcionalidades.
3. El cliente debe haber iniciado sesión para poder:
 - Registrar, actualizar o consultar su información personal.
 - Enviar dudas, sugerencias, quejas o solicitudes de devolución (gestionadas por el administrador).
 - Visualizar el catálogo y los productos disponibles.
 - Realizar pedidos y consultar su estado.
 - Validar productos, seleccionar método de pago y obtener su comprobante.
 - Consultar el historial de pedidos o facturas.

3.4 FLUJO NORMAL



Manual Técnico

Contando con las precondiciones el flujo normal será el siguiente:

Paso	Cliente	Sistema
1	Intenta iniciar sesión en el sistema. Si no tiene cuenta, selecciona la opción de registrarse y completa el formulario	Si el usuario está registrado, valida las credenciales y permite el acceso
2	Registra, actualiza o consulta su información personal; también puede solicitar la eliminación de su cuenta.	Verifica la información y comunica al administrador las solicitudes de eliminación.
3	El cliente realiza algunas de las siguientes operaciones: enviar dudas, quejas o sugerencias y solicitar devoluciones	Se comprueba la validez de las solicitudes y el sistema envía una respuesta para actualizar el proceso de devolución si es el caso, a su vez esto es recibido por el administrador
4 4.1	El cliente realiza algunas de las siguientes operaciones: visualizar el catálogo, visualizar los productos y realizar una búsqueda por filtros El cliente podrá solicitar pedidos y podrá consultar su estado	Se hará la verificación de todos los productos y que estos se encuentren en el catálogo y sea posible visualizarlos y así poder permitirle al cliente consultar el estado de los pedidos
5	El cliente realiza algunas de las siguientes operaciones: validar productos, seleccionar el método de pago y recibir el comprobante de factura	Se realizará la verificación de los productos y del método de pago, lo cual permitirá al cliente generar una próxima factura, al finalizar la anterior al momento de generar el pago se generará un comprobante de factura automático
6	El cliente realiza algunas de las siguientes operaciones: consultar historial	El cliente podrá consultar el historial de sus pedidos o



Manual Técnico

		consultas anteriores mas no podrá modificar
3.5 FLUJO ALTERNATIVO		
Si existe otra forma de acceder al caso de uso, describir los pasos.		
Paso	Cliente	Sistema
1	El cliente se autentica sin estar registrado	Se mostrará un error que hará al cliente volver al módulo donde deberá registrarse
2	El cliente se registra 2 veces	Se mostrará un error que indicará que ese usuario ya existe y que no más debe iniciar sesión
3	El cliente no puede solicitar devoluciones, ni enviar pqrs	Se enviará una alerta al administrador en el cual se indique la falla del sistema, generando la posibilidad de que el cliente vuelva a solicitar su petición
4	Selecciona un método de pago incorrecto.	Muestra mensaje de error indicando datos no válidos y solicita reintentar.
5	El cliente selecciona el método de pago incorrecto	Se mostrará en la pantalla que los datos ingresados no coinciden y deberá volver a realizar el pago
6	El cliente no puede consultar su historial	Se le mostrará un error y se le pedirá reintentar en cierta cantidad de minutos, se le avisará al administrador para generar una solución



3.5 FLUJO EXCEPCIONAL

Paso	Cliente	Sistema
1	Ya está registrado, pero el sistema no le permite autenticarse	Muestra mensaje con información de contacto al servicio al cliente.
2	Actualiza su información con datos incorrectos.	Detecta datos inválidos, bloquea temporalmente la cuenta y solicita verificación.
3	Genera spam o abuso en el módulo de soporte.	Envía advertencia y bloquea temporalmente la cuenta, indicando contacto con servicio al cliente.
4	Intenta realizar un pago y se interrumpe la conexión o el sistema no recibe confirmación.	Anula la transacción automáticamente, registra el intento fallido y notifica al cliente del error.
5	Cierra la sesión sin guardar los cambios realizados en su perfil o pedido.	Descarta los cambios no confirmados y guarda únicamente los registros anteriores.

3.7 POS CONDICIONES

FL normal:

- 1.El cliente obtiene acceso completo a las funcionalidades del sitio.
- 2.El sistema registra y almacena todas las operaciones realizadas.
- 3.El administrador recibe las notificaciones correspondientes a solicitudes o reclamos.



Manual Técnico

4. Se generan comprobantes, actualizaciones y registros en el historial del cliente.

3.8 FRECUENCIA

Alta: Este caso de uso se ejecuta de forma continua, ya que los clientes acceden constantemente al sistema para interactuar con la plataforma

CASO DE USO 2:

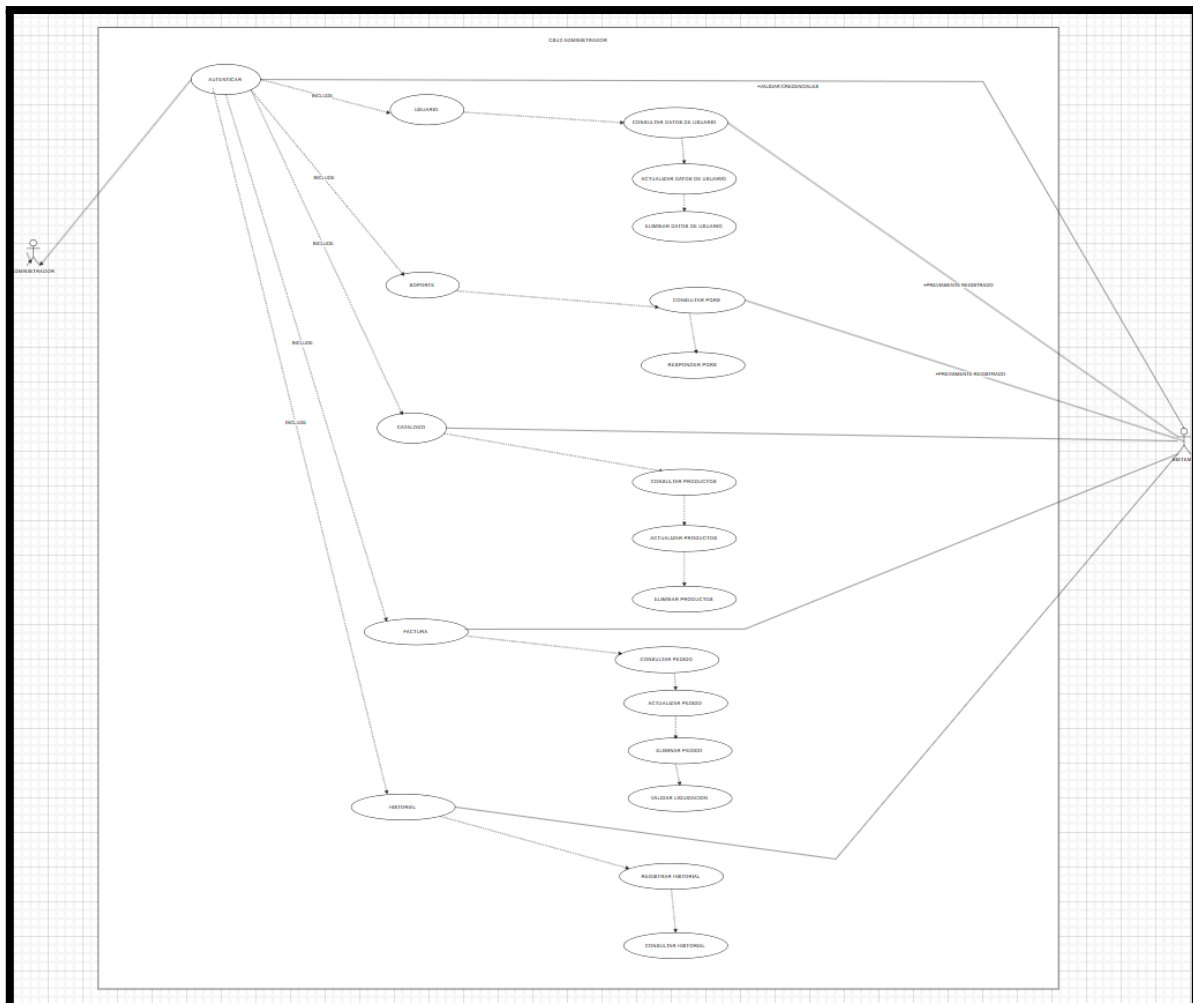


TABLA DE CASO DE USO 2 :



1. 0111IDENTIFICACIÓN DE CASO DE USO			
1.1 Id Caso	CU002	1.2 Nombre	Actividades del administrador en el sitio web vinefresh
1. HISTÓRICO DE CASO DE USO			
2.1 Autor	Ingrid Daniela Sánchez Lara		
2.2 Fecha Creación	12/03/2025	2.3 Última Actualización	19/10/2025
2.4 Actualizado por	Ingrid Daniela Sánchez Lara	2.5 Versión	1.1
1. DEFINICIÓN DE UN CASO DE USO			
3.1 DESCRIPCIÓN			
<p>Este caso de uso describe las actividades que el administrador puede realizar dentro del sistema VineFresh. Incluye las operaciones de autenticación, gestión de usuarios, productos, pedidos, soporte, facturas e historial. El administrador tiene control total sobre el mantenimiento de la información y la supervisión del funcionamiento general del sistema.</p> <p>Aunque el sistema VineFresh automatiza varios procesos de gestión, requiere la supervisión y mantenimiento del administrador, quien se encarga de validar y controlar la información registrada para asegurar la operatividad del sitio</p>			
3.2 ACTORES			
<p>Administrador: Actor principal encargado de gestionar los módulos del sistema y responder las solicitudes de los clientes.</p> <p>Sistema: Actor secundario que valida, ejecuta y actualiza las operaciones realizadas por el administrador.</p>			
3.3 PRECONDICIONES			



1. El administrador debe estar registrado en el sistema.
2. El administrador debe autenticarse correctamente antes de acceder a cualquier módulo.
3. El administrador debe tener permisos válidos para consultar, actualizar o eliminar información.
4. El sistema debe estar en funcionamiento para validar credenciales y procesar los cambios.

3.4 FLUJO NORMAL

Contando con las precondiciones el flujo normal será el siguiente:

Paso	Administrador	Sistema
1	Se autentica en el sistema ingresando sus credenciales.	Valida las credenciales; si son correctas, permite el acceso a los módulos.
2	Consulta, actualiza o elimina datos de usuarios registrados.	Muestra los datos del usuario y aplica los cambios solicitados.
3	Consulta o responde las PQRS (Petitionen, Quejas, Reclamos o Sugerencias) enviadas por los clientes.	Visualiza las solicitudes pendientes y actualiza el estado según la respuesta enviada.
4	Gestiona el catálogo de productos: puede consultar, actualizar o eliminar productos.	Aplica los cambios realizados y actualiza el catálogo en tiempo real.
5	Gestiona los pedidos: consulta, actualiza o elimina pedidos y valida la liquidación correspondiente.	Procesa la información del pedido y genera la factura correspondiente.



Manual Técnico

6	Registra o consulta el historial de transacciones o pedidos.	Guarda los cambios en la base de datos y permite la visualización del historial.
---	--	--

3.5 FLUJO ALTERNATIVO

Si existe otra forma de acceder al caso de uso, describir los pasos.

Paso	Administrador	Sistema
1	Intenta autenticarse con credenciales incorrectas.	Muestra mensaje de error e impide el acceso al sistema.
2	Intenta eliminar un usuario inexistente.	Envía alerta y no permite realizar la acción.
3	Intenta eliminar o actualizar un producto no disponible en el catálogo.	Envía alerta y no permite realizar la acción.
4	Intenta acceder a un módulo sin permisos suficientes.	Muestra mensaje de acceso restringido.
5	Intenta generar una factura con información incompleta del pedido.	Muestra alerta solicitando completar los datos antes de continuar.

3.5 FLUJO EXCEPCIONAL



Manual Técnico

Paso	Administrador	Sistema
1	El sistema no responde o se presenta un fallo en la autenticación.	Muestra mensaje de error de servidor y solicita intentar nuevamente más tarde.
2	Se detecta inconsistencia en los datos de usuarios o pedidos.	Bloquea temporalmente la acción y genera un reporte al área técnica.
3	Intenta cargar productos con información errónea (por ejemplo, precios o stock negativos).	Emite advertencia y no permite registrar la información inválida.
4	El sistema pierde conexión con la base de datos.	Detiene las operaciones y guarda los cambios previos de forma segura.
3.7 POS CONDICIONES		
<p><u>FL normal:</u></p> <p><u>1.Los datos de usuarios, productos y pedidos quedan actualizados en la base de datos.</u></p> <p><u>2.Se generan o actualizan facturas e historiales correspondientes.</u></p> <p><u>3.Las PQRS son gestionadas y marcadas como atendidas.</u></p> <p><u>4.El sistema mantiene un registro de auditoría de todas las acciones realizadas por el administrador.</u></p>		
3.8 FRECUENCIA		



Alta: Este caso de uso se ejecuta de manera frecuente, ya que el administrador interactúa constantemente para mantener la operatividad del sistema.

Características de los usuarios

El sistema VINEFRESH está orientado a dos tipos de usuarios principales, Administrador y Cliente. Cada uno cuenta con un nivel de acceso y funciones específicas dentro del sistema.

Administrador

- Posee conocimientos intermedios en herramientas informáticas y gestión de inventarios.
- Se encarga de registrar, modificar y eliminar productos vinícolas.
- Gestiona usuarios, proveedores y reportes de ventas.
- Supervisa el correcto funcionamiento del sistema y mantiene actualizada la información del catálogo.

Cliente

- Posee conocimientos básicos en el uso de computadores o dispositivos móviles.
- Puede visualizar el catálogo de vinos, realizar pedidos y consultar información básica del producto.
- No tiene acceso a configuraciones administrativas ni a información interna del negocio.
- Interactúa con una interfaz sencilla, intuitiva y visualmente atractiva.



Requerimientos

Requerimientos funcionales

1. El sistema debe permitir registrar, editar y eliminar productos vinícolas.
2. El sistema debe permitir registrar usuarios administradores y clientes.
3. El sistema debe gestionar inventarios, controlando existencias, entradas y salidas.
4. El sistema debe permitir realizar pedidos o compras de productos por parte del cliente.
5. El sistema debe validar credenciales de acceso (login/logout).
6. El sistema debe mostrar el catálogo de productos disponibles.
7. El sistema debe filtrar o buscar vinos por nombre, tipo, marca o año.
8. El sistema debe permitir generar reportes de ventas e inventario para el administrador.
9. El sistema debe notificar errores o confirmaciones al ejecutar operaciones.
10. El sistema debe mantener la integridad de los datos en todo momento.

Requerimientos no funcionales

1. El sistema debe ofrecer una interfaz amigable, accesible y responsive.
2. El sistema debe ser compatible con los principales navegadores web (Google Chrome, Edge, Firefox).
3. El sistema debe procesar las solicitudes en menos de 3 segundos.
4. El sistema debe estar desarrollado bajo el patrón MVC (Modelo Vista Controlador).
5. El sistema debe utilizar MySQL como motor de base de datos.
6. Las contraseñas deben ser cifradas para garantizar la seguridad de los usuarios.



7. El sistema debe permitir fácil mantenimiento y actualización del código fuente.
8. La aplicación debe tener un diseño coherente con la identidad visual de VINEFRESH.
9. El sistema debe controlar el acceso por roles (cliente y administrador).
10. El sistema debe guardar registros de actividad o bitácoras básicas (opcional).

Arquitectura del proyecto

El sistema VINEFRESH se desarrolla bajo una arquitectura en tres capas, la cual permite una organización modular, escalable y fácil de mantener. Esta estructura separa el sistema en tres niveles principales:

Capa de Presentación: Corresponde a la interfaz gráfica con la que interactúa el usuario. En esta capa se desarrollan las vistas utilizando tecnologías como HTML, CSS, JavaScript y Bootstrap, brindando una experiencia visual clara e intuitiva para la gestión de productos vinícolas, clientes y ventas.

Capa de Lógica o Negocio: Contiene las reglas, procesos y validaciones que definen el comportamiento del sistema. Aquí se controlan operaciones como el registro, actualización y control de inventarios, ventas y usuarios, utilizando el lenguaje PHP como principal motor lógico.

Capa de Datos: Es la encargada del almacenamiento y manejo de la información dentro de una base de datos PostgreSQL, donde se encuentran estructuradas las tablas, vistas, procedimientos almacenados y disparadores que garantizan la integridad y consistencia de los datos.

Esta arquitectura se complementa con el patrón Modelo Vista Controlador (MVC), que separa la presentación, la lógica y el control de las acciones del usuario, facilitando el mantenimiento y la escalabilidad del sistema VINEFRESH

Definición metodología



Manual Técnico

Para el desarrollo del sistema VINEFRESH, se empleó la metodología Ágil SCRUM, que permite trabajar de forma iterativa e incremental. Esta metodología fomenta la colaboración constante entre los miembros del equipo y el instructor, asegurando entregas parciales funcionales en cada iteración o “sprint”.

Durante cada sprint, se llevaron a cabo reuniones de seguimiento, revisión de avances y planificación de nuevas tareas, permitiendo ajustar los requerimientos según las necesidades del cliente o los hallazgos del proceso de desarrollo.

Gracias a SCRUM, el equipo logró una mayor adaptabilidad, retroalimentación continua y una entrega final más alineada con los objetivos del proyecto.

Ciclo de vida

El ciclo de vida del sistema VINEFRESH se desarrolló siguiendo las etapas definidas en el modelo en cascada, aplicadas de manera estructurada dentro de los sprints ágiles. Cada fase permitió avanzar de forma ordenada en la construcción del proyecto:

Análisis de Requerimientos: se identificaron las necesidades del cliente, los objetivos del sistema y los requerimientos funcionales y no funcionales.

Diseño: se elaboraron los diagramas UML, el diseño de base de datos (DER), estructura de la aplicación y la interfaz de usuario.

Desarrollo o Codificación: se implementa la lógica del sistema utilizando PHP, HTML, CSS, JavaScript y SQL.8

Pruebas: se validó el correcto funcionamiento del sistema, identificando y corrigiendo errores antes de su implementación.

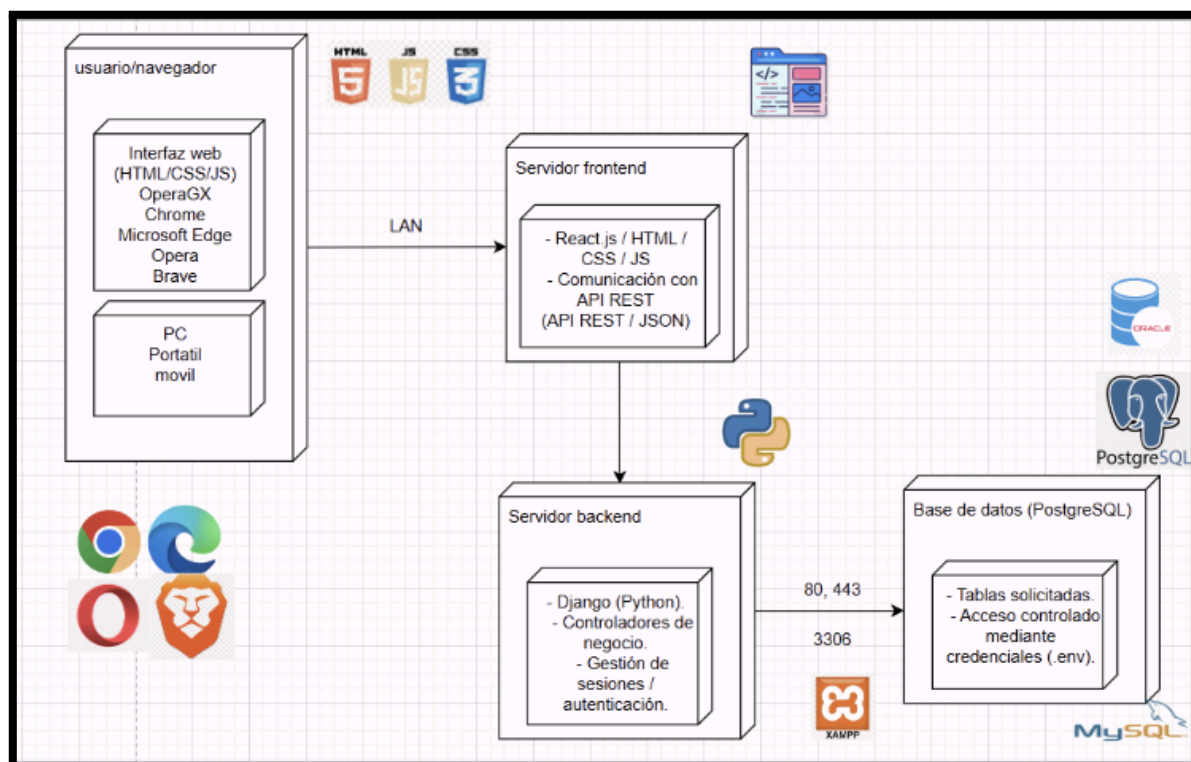
Implementación: se desplegó el sistema para su uso, garantizando que los módulos cumplieran los objetivos establecidos.

Mantenimiento: se definieron las actividades futuras para la corrección de errores, actualización de versiones y mejora de funcionalidades.

Este ciclo de vida permitió mantener una secuencia lógica de trabajo, asegurando la calidad, funcionalidad y estabilidad del sistema VINEFRESH.



Diagrama de despliegue



Fundamentos y herramientas utilizadas

- Jira -**Herramienta principal de planificación y seguimiento** del proyecto *VINEFRESH*. Permitirá organizar las tareas de desarrollo, registrar los avances, asignar responsabilidades y mantener el control de los tiempos y entregables de cada fase.



- La metodología orientada a objetos (O.O.) se utilizará para estructurar el proyecto *VINEFRESH* de forma modular, reutilizable y fácil de mantener. Esta metodología permite representar los elementos del sistema (usuarios, productos, pedidos, reseñas, etc.) como objetos, cada uno con sus propios atributos (datos) y métodos (funciones o comportamientos).



- Google Docs se utilizará como la herramienta principal para elaborar, compartir y mantener la documentación técnica del sistema. Su uso permitirá trabajar de manera colaborativa, garantizando que todos los integrantes actualicen la información en tiempo real.

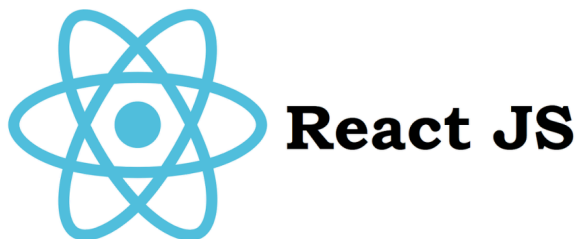


Desarrollo (Frontend + Backend)

Frontend (lo que ve el usuario)

Lenguajes y frameworks:

- [React.js](https://reactjs.org/)- Es el principal framework para el desarrollo del frontend. React permitirá construir una interfaz web moderna, dinámica y responsiva que mejore la experiencia del usuario durante la navegación y compra de productos.



- El proyecto VINEFRESH contempla la posibilidad de ampliar su alcance a dispositivos móviles mediante el uso de React Native, un framework basado en JavaScript que permite desarrollar aplicaciones móviles multiplataforma (Android y iOS) utilizando la misma lógica y componentes de React.js.



El desarrollo del frontend del aplicativo web VINEFRESH se apoyará en los lenguajes base del entorno web —HTML5, CSS3 y JavaScript— junto con el framework Bootstrap, con el fin de crear una interfaz moderna, funcional y adaptable a diferentes dispositivos



1. HTML5

Se utilizará para definir la estructura semántica del sitio web. Organizando los elementos principales: encabezados, secciones, formularios, tablas y pies de página. Permitiendo integrar componentes dinámicos de React.js, como el catálogo de vinos y los formularios de autenticación.

2. CSS3



Se emplea para definir los estilos visuales del aplicativo: colores, tipografías, márgenes, fondos e interacciones. A través de hojas de estilo modulares, garantizando la coherencia estética de todas las páginas. Se implementarán reglas de diseño responsivo para que el sistema se adapte correctamente a pantallas de escritorio, tablets y móviles.

3. JavaScript

Será el lenguaje encargado de dar interactividad y dinamismo a la interfaz. Permitirá controlar eventos del usuario (clics, validaciones, búsquedas) y realizar peticiones asíncronas (AJAX/Fetch) al backend de Django. También servirá como base para React.js, donde se maneja la lógica de los componentes.

4. Bootstrap

Se integrará para acelerar el diseño visual del sistema. Aportando componentes predefinidos como botones, modales, menús y formularios adaptativos. Garantizará una apariencia profesional, limpia y totalmente responsive sin necesidad de crear estilos desde cero.

Backend (lógica del servidor)

El backend del proyecto VINEFRESH estará enfocado en la gestión de la lógica del sistema, la comunicación con la base de datos y la administración de las operaciones del servidor. Para su desarrollo se emplearán los lenguajes Python y Java, junto con el framework Django, con el fin de garantizar un sistema seguro, escalable y eficiente.



python & django

1. Python

Será el lenguaje principal para el desarrollo del backend. Encargándose de implementar la lógica del servidor, procesar peticiones y gestionar la conexión con la base de datos PostgreSQL. Gracias a su sintaxis clara y su integración con Django, permitirá desarrollar módulos como autenticación, registro de usuarios, manejo de productos, pedidos y reseñas. Además, se utilizarán librerías como Django REST Framework para crear las APIs REST que conectarán el backend con el frontend en [React.js](https://reactjs.org/).

1.1 Django:

Django servirá como el núcleo del backend, proporcionando una estructura modular basada en el patrón MVC (Modelo–Vista–Controlador). Gestionará la creación de modelos (clases que representan las tablas de la base de datos), vistas (rutas y controladores) y plantillas (si se usa renderizado del lado del servidor).

2. Java



Se utilizará como lenguaje complementario, especialmente para el desarrollo de posibles módulos adicionales o integración con aplicaciones externas. Viéndolo integrado en: Automatización de reportes o análisis de datos, validaciones adicionales en el flujo de negocio y compatibilidad futura con aplicaciones móviles nativas o microservicios. Garantizando que el proyecto mantiene compatibilidad con entornos empresariales y facilita la escalabilidad a largo plazo.

Base de datos (B.D.)





Manual Técnico

En el proyecto VINEFRESH, se utiliza PostgreSQL como el sistema gestor de base de datos principal. Su función será almacenar, organizar y administrar toda la información del sistema, garantizando integridad, seguridad y alto rendimiento en las operaciones de lectura y escritura.

Control de versiones:



Para el sistema de control de versiones estaremos utilizando Git y GitHub, con el fin de mantener un registro organizado y seguro del código fuente durante todas las fases de desarrollo. GitHub es una plataforma en línea que utiliza Git para almacenar los repositorios en la nube, facilitando el trabajo colaborativo y la sincronización entre los miembros del equipo. Dentro del proyecto, cada integrante contará con su propia rama de desarrollo y realizará push de los cambios hacia el repositorio remoto en GitHub.

Visual Studio Code

Para el desarrollo del aplicativo web VINEFRESH, se utilizará Visual Studio Code (VS Code) como el entorno de desarrollo integrado (IDE) principal. VS Code ofrece un entorno liviano, flexible y multiplataforma que facilita la programación tanto del frontend (React.js, HTML, CSS, JavaScript) como del backend (Python con Django).



Plataformas:

Vercel - Plataforma para el despliegue del frontend desarrollado en [React.js](https://reactjs.org/). Vercel permite alojar aplicaciones web modernas de manera sencilla, rápida y gratuita, brindando un entorno de producción accesible para los usuarios finales.



Render - Para el despliegue del backend del aplicativo web VINEFRESH, se utilizará la plataforma Render, una herramienta en la nube que permite alojar aplicaciones desarrolladas en Python con Django de forma segura, automatizada y eficiente. Tiene compatibilidad directa con entornos Django, bases de datos PostgreSQL y despliegues continuos conectados a repositorios GitHub.



Pruebas y calidad

Postman - Herramienta utilizada para realizar las pruebas de las API REST desarrolladas en el backend de VINEFRESH. Permitirá verificar que los distintos endpoints del sistema funcionen correctamente y que la comunicación entre el frontend y el backend sea estable y segura.



Jest - Validar el comportamiento de los componentes del frontend, asegurando que la interfaz responda correctamente a las acciones del usuario y que los datos renderizados desde el backend sean los esperados.



PyTest- Comprobar la correcta ejecución de las funciones del servidor, los endpoints de la API REST y las operaciones con la base de datos PostgreSQL.



Centro Industrial y Desarrollo
Empresarial Soacha
Regional Cundinamarca



Manual Técnico



SonarQube– Para asegurar la calidad, mantenibilidad y seguridad del código fuente de VINEFRESH, se emplea la herramienta SonarQube. Esta herramienta analiza de forma estática el código del proyecto, detectando errores, vulnerabilidades y malas prácticas de programación.



Presentación y documentación final

Canva - Diseños visuales o prototipos y presentación del proyecto.



Proyecto académico - Cumplir con los requisitos de presentación (documento técnico, video, matriz QA.).

Requisitos del Sistema

Hardware:



Manual Técnico

- Procesador: Intel Core i5 o i7 (o equivalente)
- Memoria RAM: 8 GB o más - Almacenamiento: 256 GB o más (SSD recomendado)
- Tarjeta gráfica: Integrada o dedicada (dependiendo de las necesidades)
- Pantalla: 14" o más (Full HD o 4K)
- Conectividad: Wi-Fi, Bluetooth, puertos USB, HDMI, etc.
- Sistema de enfriamiento: Adecuado para el procesador y otros componentes.

Software

- Sistema operativo: Windows 10 o Linux (dependiendo de las necesidades)
- Navegador web: Actualizado (Google Chrome, Mozilla Firefox, etc.)
- Suite de oficina: Microsoft Office o LibreOffice - Antivirus: Actualizado y
- confiable
- Software de gestión de sistemas: Dependiendo de las necesidades (ej. Nagios, Zabbix, etc.)

Procesos del Software

El sistema VINEFRESH sigue una secuencia de procesos que garantizan la interacción correcta entre el usuario y el sistema. Los principales procesos son:



Manual Técnico

- **Inicio de sesión:** el usuario ingresa sus credenciales y accede al sistema según su rol.
- **Gestión de productos:** el administrador puede agregar, modificar, eliminar y consultar información de vinos.
- **Gestión de inventario:** se controlan las existencias y actualizaciones automáticas tras las ventas.
- **Visualización de catálogo:** el cliente puede explorar los vinos disponibles con filtros o búsquedas.
- **Proceso de compra o pedido:** el cliente selecciona productos y genera un pedido.
- **Generación de reportes:** el administrador consulta reportes de ventas, clientes o inventarios.
- **Cierre de sesión y respaldo:** el sistema guarda cambios y puede generar copias de seguridad automáticas.

Instalación de Aplicaciones

Para la correcta instalación y funcionamiento del sistema VINEFRESH se deben cumplir los siguientes pasos y requerimientos técnicos:

Requerimientos del entorno

- Servidor local: XAMPP o WAMP con Apache y MySQL instalados.
- Navegador: Google Chrome, Mozilla Firefox o Microsoft Edge.
- Editor de código: Visual Studio Code o Sublime Text.
- Sistema operativo: Windows 10 o superior.

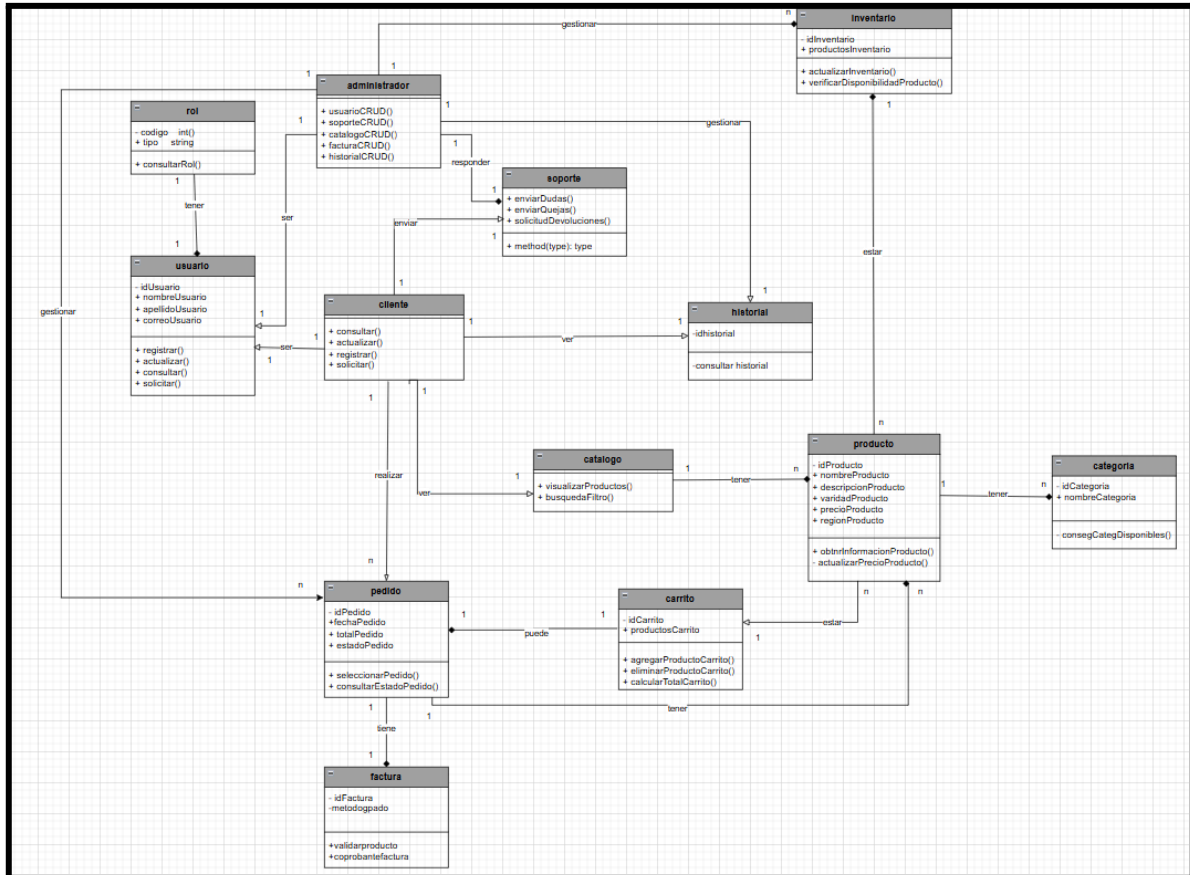
Pasos de instalación

1. Instalar XAMPP y activar los módulos Apache y MySQL.
2. Copiar la carpeta del proyecto VINEFRESH dentro del directorio htdocs.
3. Abrir phpMyAdmin e importar el archivo de la base de datos (VINEFRESH.sql).
4. Configurar las credenciales de conexión en el archivo config.php o equivalente.
5. Abrir el navegador y acceder a la URL: <http://localhost/VINEFRESH/>
6. Iniciar sesión con las credenciales de administrador predefinidas.

Diagrama de Clases



Manual Técnico



Modelo entidad Relación



Manual Técnico

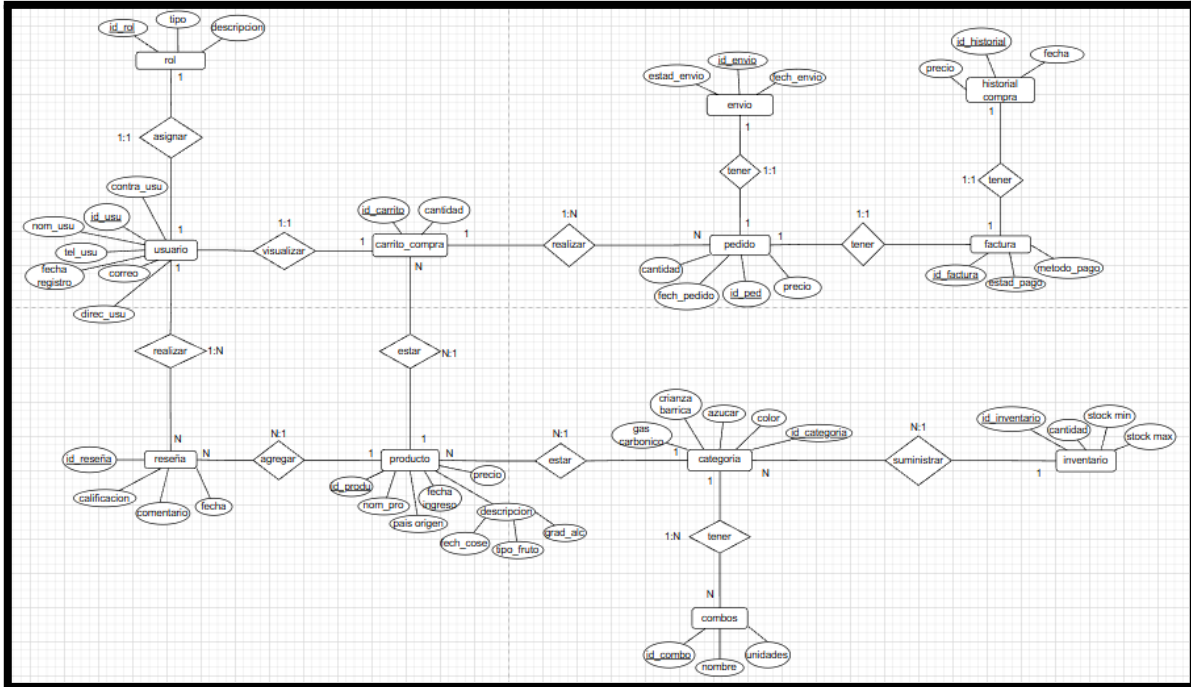
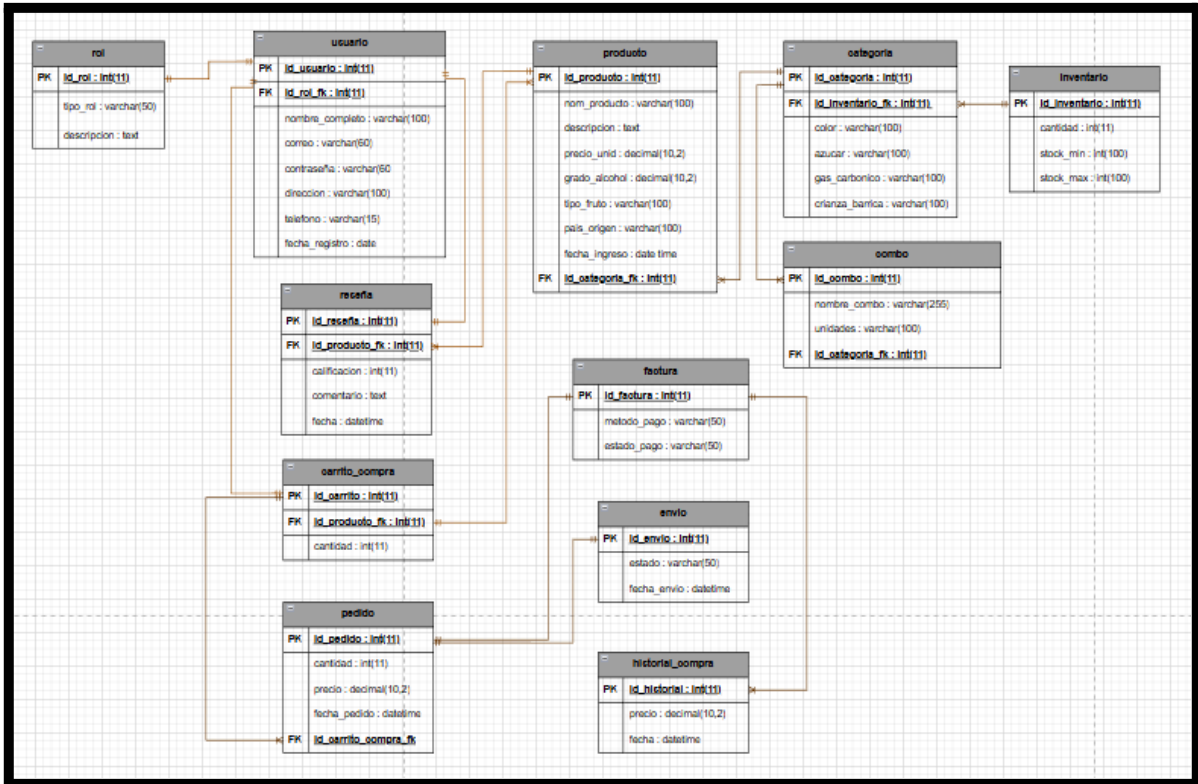


Diagrama entidad relación



Diccionario de Datos

DICCIONARIO DE DATOS . TABLA No.1 "CARRITO_COMPRA"

carrito_compra

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio
id_carrito (Primaria)	int(11)	No		usuario -> id_usuario	identificador unico del carrito de compras	
id_producto_fk	int(11)	No		producto -> id_producto	identificador del producto agreagado al carrito de compras	
cantidad	int(11)	No			cantidad de productos	

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id_carrito	501	A	No	
id_producto_fk	BTREE	No	No	id_producto_fk	501	A	No	



DICCIONARIO DE DATOS . TABLA No.2 “CATEGORIA”

categoria

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio
id_categoria (Primaria)	int(11)	No			identificador unico de las categorias	
id_inventario_fk	int(11)	No		inventario -> id_inventario	identificador del inventario clasificado por las categorias	
color	varchar(100)	No			color del vino	
azucar	varchar(100)	No			cantidad de azucar del vino	
gas_carbonico	varchar(100)	No			cantidad de gas carbonico del vino	
crianza_barrica	varchar(100)	No			tipo de crianza barrica	

DICCIONARIO DE DATOS . TABLA No.3 “COMBO”

combo

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio
id_combo (Primaria)	int(11)	No			identificador unico del combo	
nombre_combo	varchar(255)	No			nombre asignado al combo	
unidades	varchar(100)	No	current_timestamp()		unidades de productos	
id_categoria_fk	int(10)	No		categoria -> id_categoria	identificador de la categoria aignada al combo	

DICCIONARIO DE DATOS . TABLA No.4 “CUPON”



Manual Técnico

cupon

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio
id_cupon (Primaria)	int(11)	No			identificador unico del cupon	
codigo	varchar(50)	No			codigo del cupon	
descuento	decimal(5,2)	No			descuento que ejerse el cupon	
fecha_expiracion	date	No			fecha de expiracion del cupon	
activo	varchar(10)	Sí	1		estado del cupon	
id_factura_fk	int(11)	No		factura -> id_factura	identificador de la factura a la que se le aplica el cupon	

DICCIONARIO DE DATOS . TABLA No.5 “ENVIA”

envio

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio
id_envio (Primaria)	int(11)	No		pedido -> id_pedido	identificador unico del envio	
estado	varchar(50)	No			estado de rastreo del pedido	
fecha_envio	datetime	Sí	NULL		fecha de llegada del pedido	

DICCIONARIO DE DATOS . TABLA No.6 “FACTURA”

factura

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio
id_factura (Primaria)	int(11)	No		pedido -> id_pedido	identificador unico de la factura	
metodo_pago	varchar(50)	No			metodo de pago seleccionado	
estado_pago	varchar(50)	No			estado del pago, aprobado, desaprobado	

DICCIONARIO DE DATOS . TABLA No.7 “HISTORIAL_COMPRA”



historial_compra

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio
id_historial (<i>Primaria</i>)	int(11)	No		factura -> id_factura	identificaodr unico del historial	
precio	decimal(10,2)	No			precios registrados de las compras	
fecha	datetime	No			fecha registrada de las compras	

DICCIONARIO DE DATOS . TABLA No.8 “INVENTARIO”

inventario

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio
id_inventario (<i>Primaria</i>)	int(11)	No			identificador unico del inventario	
cantidad	int(11)	No			cantidad de productos existentes	
stock_min	int(100)	No			la cantidad de producto que debe haber en el inventario	
stock_max	int(100)	No			cantidad maxima del producto que debe haber en el inventario	

DICCIONARIO DE DATOS . TABLA No.9 “PEDIDO”



Manual Técnico

pedido

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio
id_pedido (Primaria)	int(11)	No			identificador unico del pedido	
cantidad	int(11)	No			cantidad de productos	
precio	decimal(10,2)	No			precio del pedido total o suma de los productos	
fecha_pedido	datetime	Sí	current_timestamp()		fecha de de cuando se realizo el pedido	
id_carrito_compra_fk	int(11)	No		carrito_compra -> id_carrito	identificador del carrito de compras donde se extregieron los productos para el pedido	

DICCIONARIO DE DATOS . TABLA No.10 “PRODUCTO”

producto

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio
id_producto (Primaria)	int(11)	No		reseña -> id_producto_fk	identificador unico del producto	
nom_producto	varchar(100)	No			nombre del producto	
descripcion	varchar(500)	No			detalles del producto	
precio_unid	decimal(10,2)	No			precio unico del producto	
grado_alcohol	decimal(10,2)	No			cantidad del alcohol del productos	
tipo_fruto	varchar(100)	No			tipo de fruto del que esta hecho el producto	
pais_origen	varchar(100)	No			pais de origen donde se exporto el fruto o el producto	
fecha_ingreso	datetime	No			fecha de ingreso del producto	
id_categoria_fk	int(11)	No		categoria -> id_categoria	identificador de la categoria a la que pertenece el grupo	

DICCIONARIO DE DATOS . TABLA No.11 “RESEÑA”



reseña

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio
id_reseña (Primaria)	int(11)	No		usuario -> id_usuario	identificador unico de la reseña	
id_producto_fk	int(11)	No			identificador del producto el que se da la reseña	
calificacion	int(11)	Sí	NULL		calificacion de la calidad del producto	
comentario	text	Sí	NULL		comentarios acerca del producto	
fecha	datetime	No			fecha de publicacion del comentario	

ARIO DE DATOS . TABLA No.12 “ROL”

rol

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio
id_rol (Primaria)	int(11)	No			identificador unico del rol	
tipo_rol	varchar(50)	No			tipo de rol asignado a la cuenta, administrador o cliente	
descripcion	text	No			descripcion de actividades o tareas que puede realizar el usuario	

DICCIONARIO DE DATOS . TABLA No.13 “SOPORTE”

soporte

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio
id_soporte (Primaria)	int(11)	No			identificador del soporte	
id_usuario_fk	int(11)	No		usuario -> id_usuario	identificador del usuario que solicito el soporte	
mensaje	text	No			mensaje del soporte	
PQRS	varchar(100)	No			peticion, queja, reclamo y solicitud	
estado	enum('Abierto', 'En Proceso', 'Cerrado')	Sí	Abierto		estado del soporte, en proceso o solucionado	
fecha	datetime	No			fecha de envio del soporte	

DICCIONARIO DE DATOS . TABLA No.14 “USUARIO”



usuario

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio
id_usuario (Primaria)	int(11)	No			identificador único del usuario	
id_rol_fk	int(11)	No		rol -> id_rol	identificador del rol asignado al usuario	
nombre_completo	varchar(100)	No			nombre y apellido del usuario	

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio
correo	varchar(60)	No			direccion de correo electronico del usuario	
contraseña	varchar(60)	No			contraseña de la cuenta del usuario	
direccion	varchar(100)	No			direccion de vivienda del usuario	
telefono	varchar(15)	No			telefono de contacto del usuario	
fecha_registro	date	No			fecha donde se registro el usuario al sistema	

DICCIONARIO DE DATOS . TABLA No.15 "VISTA_HISTORIAL_COMPRAS_CLIENTES"

vista_historial_compras_clientes

Comentarios de la tabla: VIEW

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio
nombre_completo	varchar(100)	No			nombre y apellido del usuario	
precio	decimal(10,2)	No			precios registrados de las compras	
fecha	datetime	No			fecha registrada de las compras	



DICCIONARIO DE DATOS . TABLA No.16 “VISTA_MIS_PEDIDOS”

vista_mis_pedidos

Comentarios de la tabla: *VIEW*

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio
id_usuario	int(11)	No	0		identificador único del usuario	
nombre_completo	varchar(100)	No			nombre y apellido del usuario	
id_pedido	int(11)	No			identificador unico del pedido	
cantidad	int(11)	No			cantidad de productos	
precio	decimal(10,2)	No			precio del pedido total o suma de los productos	
fecha_pedido	datetime	Sí	current_timestamp()		fecha de de cuando se realizo el pedido	
estado_envio	varchar(50)	Sí	NULL		estado de rastreo del pedido	

DICCIONARIO DE DATOS . TABLA No.17 “VISTA_PRODUCTOS_DISPONIBLES”

vista_productos_disponibles

Comentarios de la tabla: *VIEW*

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio
id_producto	int(11)	No			identificador unico del producto	
nom_producto	varchar(100)	No			nombre del producto	
descripcion	varchar(500)	No			detalles del producto	
precio_unid	decimal(10,2)	No			precio unico del producto	
stock_disponible	int(11)	No			cantidad de productos existentes	
categoria_color	varchar(100)	No			color del vino	

DICCIONARIO DE DATOS . TABLA No.18 “VISTA_SOPORTE_ACTIVADO”

vista_soporte_activo

Comentarios de la tabla: *VIEW*

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio
id_soporte	int(11)	No			identificador del soporte	
nombre_completo	varchar(100)	No			nombre y apellido del usuario	
PQRS	varchar(100)	No			peticion,queja,reclamo y solicitud	
estado	enum("Abierto", "En Proceso", "Cerrado")	Sí	Abierto		estado del soporte,en proseso o solucionado	
fecha	datetime	No			fecha de envio del soporte	



Glosario

- **API (Application Programming Interface):** Conjunto de reglas y herramientas que permiten que diferentes aplicaciones se comuniquen entre sí.
- **Agile:** Metodología de desarrollo de software basada en la colaboración, la flexibilidad y la entrega continua de valor.
- **Algoritmo:** Conjunto de instrucciones lógicas y finitas para resolver un problema o realizar una tarea específica.
- **Análisis de Requisitos:** Proceso de recopilación y evaluación de las necesidades del usuario para definir las características del software.
- **Automatización de Procesos:** Uso de software para ejecutar tareas repetitivas sin intervención humana.
- **Arquitectura de Software:** Diseño estructural fundamental que organiza componentes, módulos y sus interacciones en un sistema.
- **AOP (Programación Orientada a Aspectos):** Paradigma que permite separar preocupaciones transversales del código principal
- **Back-end:** Parte del software que se ejecuta en el servidor, gestionando la lógica de negocio y la base de datos.
- **Base de Datos:** Sistema que almacena y organiza información de forma estructurada para su fácil acceso y manipulación.
- **Big Data:** Conjunto de datos de gran volumen y complejidad que requieren herramientas especializadas para su procesamiento y análisis.
- **Bug:** Error o falla en el código de un programa que impide su correcto funcionamiento.



Manual Técnico

- **Bootstrap:** Framework front-end que facilita el diseño de interfaces web responsivas y modernas.
- **Caché:** Memoria de acceso rápido que almacena datos temporales para acelerar la ejecución de aplicaciones.
- **Código Abierto (Open Source):** Software cuyo código fuente está disponible para ser utilizado, modificado y distribuido públicamente.
- **Compilador:** Programa que traduce código fuente a un lenguaje de máquina que puede ejecutar el procesador.
- **CRUD (Create, Read, Update, Delete):** Conjunto básico de operaciones para gestionar información en bases de datos.
- **CSS (Cascading Style Sheets):** Lenguaje que define el estilo y diseño de las páginas web.
- **DevOps:** Cultura y conjunto de prácticas que integran el desarrollo y las operaciones para optimizar la entrega de software.
- **Documentación:** Conjunto de información escrita que describe el funcionamiento, diseño y uso de un sistema.
- **Django:** Framework de desarrollo web en Python que sigue el patrón MVC y agiliza la creación de aplicaciones robustas.
- **Despliegue:** Proceso de poner en producción una aplicación o sistema.
- **Design Patterns (Patrones de Diseño):** Soluciones reutilizables y probadas para problemas comunes en la ingeniería de software.
- **Escalabilidad:** Capacidad de un sistema para adaptarse y manejar un aumento en la carga de trabajo sin perder rendimiento.
- **E-Commerce:** Comercio electrónico que se lleva a cabo a través de plataformas digitales.



Manual Técnico

- **Framework:** Conjunto de herramientas y bibliotecas que facilitan el desarrollo estructurado de software.
- **Front-end:** Parte de la aplicación que interactúa directamente con el usuario, enfocada en la interfaz y experiencia.
- **Función:** Bloque de código reutilizable que realiza una tarea específica dentro de una aplicación.
- **Gestión de versiones:** Control del historial de cambios en el código mediante herramientas especializadas.
- **Git:** Sistema de control de versiones distribuido ampliamente utilizado para gestionar proyectos de desarrollo.
- **Hosting:** Servicio que permite almacenar y entregar aplicaciones o sitios web a través de internet.
- **HTML (HyperText Markup Language):** Lenguaje de marcado estándar para estructurar páginas web.
- **IDE (Integrated Development Environment):** Entorno que integra herramientas de programación para facilitar el desarrollo.
- **IA (Inteligencia Artificial):** Tecnología que simula procesos cognitivos humanos, como el aprendizaje y la toma de decisiones.
- **Índice de Base de Datos:** Estructura que mejora la velocidad de las consultas en una base de datos.
- **JSON (JavaScript Object Notation):** Formato ligero y legible para el intercambio de datos entre sistemas.
- **Java:** Lenguaje de programación orientado a objetos muy utilizado en aplicaciones empresariales.



Manual Técnico

- **JUnit:** Framework de pruebas unitarias para aplicaciones Java, facilitando la validación del código.
- **Jest:** Framework de pruebas para JavaScript, ampliamente usado para validar el comportamiento de aplicaciones y librerías.
- **Kotlin:** Lenguaje de programación moderno que se ejecuta sobre la JVM y ofrece sintaxis concisa y seguridad en tipos.
- **Linux:** Sistema operativo de código abierto ampliamente utilizado en servidores y entornos de desarrollo.
- **MongoDB:** Base de datos NoSQL orientada a documentos, reconocida por su alta escalabilidad y flexibilidad.
- **MVC (Model-View-Controller):** Patrón de diseño que separa la aplicación en tres componentes para facilitar el mantenimiento.
- **Node.js:** Entorno de ejecución para JavaScript en el servidor, ideal para aplicaciones escalables y en tiempo real.
- **Programación Orientada a Objetos (POO):** Paradigma basado en la creación de “objetos” que agrupan datos y comportamientos en una sola entidad.
- **Python:** Lenguaje de programación interpretado y versátil, utilizado en desarrollo web, ciencia de datos, automatización y más.
- **Protocolo:** Conjunto de reglas que rigen la transmisión y recepción de datos en redes de comunicaciones.
- **PHP:** Lenguaje de programación del lado del servidor, ampliamente usado para el desarrollo web.
- **PostgreSQL:** Sistema de gestión de bases de datos relacionales, reconocido por su robustez y conformidad con estándares SQL.



Manual Técnico

- **Query:** Consulta o solicitud de datos a una base de datos, generalmente escrita en SQL.
- **Redundancia:** Duplicación de componentes o funciones en un sistema para aumentar la fiabilidad y disponibilidad.
- **React:** Biblioteca de JavaScript desarrollada por Facebook para la construcción de interfaces de usuario interactivas.
- **Scrum:** Marco de trabajo ágil para el desarrollo incremental de software, basado en iteraciones (sprints) y reuniones periódicas.
- **SQL (Structured Query Language):** Lenguaje estandarizado para gestionar y manipular datos en bases de datos relacionales.
- **UML (Unified Modeling Language):** Lenguaje de modelado gráfico utilizado para visualizar, diseñar y documentar sistemas de software.
- **User Interface (UI):** Diseño y disposición de los elementos interactivos de una aplicación, enfocados en la facilidad de uso.
- **Upgrade:** Proceso de actualización o mejora de hardware o software para incorporar nuevas funciones o mejoras de rendimiento.
- **Use Case (Caso de Uso):** Descripción detallada de cómo un usuario interactúa con un sistema para lograr un objetivo específico.
- **Visual Studio:** Entorno de desarrollo integrado (IDE) de Microsoft para la creación de aplicaciones en múltiples lenguajes y plataformas.
- **Version Control (Control de Versiones):** Sistema que permite rastrear, gestionar y revertir cambios en el código fuente.
- **Variable:** Elemento que almacena datos en programación y que puede modificarse durante la ejecución del código.



Centro Industrial y Desarrollo
Empresarial Soacha
Regional Cundinamarca



Manual Técnico

- **Windows:** Sistema operativo de Microsoft utilizado en entornos personales y empresariales.
- **Wireframe:** Esquema o boceto preliminar del diseño de una interfaz que ayuda a planificar la disposición de elementos.



Referencias

Quillin, B. (2023, May 11). *The Benefits of a Three-Layered Application*

Architecture. VFunction.

<https://vfunction.com/blog/the-benefits-of-a-three-layered-application-architecture/>

Özkaya, M. (2021, October 2). *Layered (N-Layer) Architecture*. Design

Microservices Architecture with Patterns & Principles.

<https://medium.com/design-microservices-architecture-with-patterns/layered-n-layer-architecture-e15ffdb7fa42>

MVC (Model, View, Controller) explicado. (n.d.). CódigoFacilito.

<https://codigofacilito.com/articulos/mvc-model-view-controller-explicado>

AlexHost. (2024, December 6). *¿Qué es MVC? Ventajas y desventajas de MVC*.

AlexHost SRL; AlexHost.

<https://alexhost.com/es/faq/que-es-mvc-ventajas-y-desventajas-de-mvc/>

Arquitectura Cliente-Servidor. (n.d.). Reactiveprogramming.io.

<https://reactiveprogramming.io/blog/es/estilos-arquitectonicos/cliente-servidor>

Scrumstudy. (2017). *Phases and processes in Scrum project* | SCRUMstudy.

Scrumstudy.com.

<https://www.scrumstudy.com/whyscrum/scrum-phases-and-processes>



▷ *Fases de la Metodología Scrum [5 Etapas y 16 Procesos]*. (2020, July 10).

ComparaSoftware Blog.

<https://blog.comparasoftware.com/fases-metodologia-scrum/>

Asana. (n.d.). *Qué es Scrum y cómo aplicarlo en gestión de proyectos* • Asana.

Asana. <https://asana.com/es/resources/what-is-scrum>

de. (2019, March 11). *El modelo en cascada: desarrollo secuencial de software*.

IONOS Digital Guide; IONOS.

<https://www.ionos.com/es-us/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/>

varkjtfvknjz. (2014, May 21). *Ciclo de vida de un software, según Pressman*.

Prezi.com.

<https://prezi.com/4a7bhv4ztme9/ciclo-de-vida-de-un-software-segun-pressman/>

Ciclo de vida de desarrollo de software | Microsoft Power Automate. (2025).

Microsoft.com.

<https://www.microsoft.com/es-co/power-platform/topics/phases-of-the-software-development-lifecycle>

Diferente modelos de ciclo de vida de un software - inlogiq. (2021, November 10). <https://inlogiq.com/diferentes-modelos-de-ciclos-de-vida-del-software/>

ISO/IEC 12207. Ciclo de vida del software | Bureau Veritas. (n.d.).

Www.bureauveritas.es.



Manual Técnico

<https://www.bureauveritas.es/certificacion/certificacion-de-equipos-e-instalaciones/isoiec-12207-ciclo-de-vida-del-software>

Prieto, E. (2022, December 30). *¿Cuáles son las etapas del Desarrollo de Software?* - Tiffin University. Tiffin University; Tiffin University.

<https://global.tiffin.edu/blog/cuales-son-las-etapas-del-desarrollo-de-software>

Solutions, T. (2020, May 23). *Web App Development Process - 7 Critical Stages to Build Amazing Apps*. TechoSquare Solutions - Your Technology Partner.

<https://www.techosquare.com/blog/web-app-development-process-model>

Asana. (2022, December 14). *Workflow diagrams help visualize business processes*. Asana. <https://asana.com/es/resources/workflow-diagram>

XAMPP FAQs for Windows. (2024). Apachefriends.org.

https://www.apachefriends.org/es/faq_windows.html

(2025). Scriptcase.net.

<https://help.scriptcase.net/portal/es/kb/articles/c%C3%B3mo-importar-una-base-de-datos-en-phpmyadmin>

Tutorial de XAMPP: Como instalar y configurar XAMPP servidor local. (2023, March). IONOS Digital Guide.

<https://www.ionos.com/es-us/digitalguide/servidores/herramientas/instala-tu-servidor-local-xampp-en-unos-pocos-pasos/>



Manual Técnico

Los requisitos ideales de PHP y del servidor de WordPress. (2022, May 16).

Kinsta®. <https://kinsta.com/es/blog/requisitos-servidor-wordpress/>

Asana. (n.d.). *¿Qué es un diagrama de flujo? Los símbolos, los tipos de diagrama y cómo leerlos* • Asana. Asana.

<https://asana.com/es/resources/what-is-a-flowchart>

¿Qué es el SDLC? - Explicación del ciclo de vida del desarrollo de software - AWS.

(n.d.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/sdlc/>

oosorioj. (2025). *Seguridad en el control de aplicaciones.* Slideshare.

[https://es.slideshare.net/slideshow/seguridad-en-el-control-de-aplicaciones/896](https://es.slideshare.net/slideshow/seguridad-en-el-control-de-aplicaciones/8964658)

[4658](https://es.slideshare.net/slideshow/seguridad-en-el-control-de-aplicaciones/8964658)