Universidad Internacional San Isidro Labrador

PROYECTO FINAL

Nombre del Proyecto: ViveGreen

Estudiante: David Zumbado Rodríguez

Curso: Programación Avanzada

Profesora: Estefania Boza Villalobos

Fecha: 2025

Tabla de Contenido

Contenido

Introducción

El proyecto **ViveGreen** consiste en el diseño, desarrollo e implementación de un sitio web integral para un vivero localizado en Costa Rica, cuyo objetivo principal es modernizar la gestión de información y la interacción con los clientes mediante una plataforma tecnológica robusta, intuitiva y escalable.
El sistema responde a la creciente necesidad de los viveros de contar con presencia digital, mejorar la experiencia del usuario y automatizar procesos internos como la gestión del catálogo de plantas, la administración de solicitudes de cotización y la publicación de contenido educativo.

La plataforma está concebida bajo principios de **usabilidad, accesibilidad, seguridad, rendimiento y escalabilidad**, permitiendo que el vivero expanda su operación en futuros ciclos de desarrollo.
Además, el sitio web combina funcionalidades informativas, administrativas y transaccionales, integrando desde vistas públicas dirigidas a los clientes hasta herramientas especializadas para el personal encargado de la gestión del vivero.

Objetivo General

Desarrollar un sitio web informativo, administrativo y transaccional que centralice el catálogo de plantas y productos del vivero, permita gestionar solicitudes de cotización de manera eficiente, brinde contenido educativo para los clientes y fortalezca la presencia digital del negocio mediante una plataforma moderna, segura y escalable.

Objetivos Específicos

- **Diseñar una arquitectura de software modular** que permita la correcta interacción entre el frontend, backend y la base de datos, garantizando mantenibilidad y facilidad de expansión futura.
- **Implementar un catálogo digital dinámico** que incluya categorías, filtros avanzados, fichas técnicas detalladas y un sistema de búsqueda optimizada para mejorar la experiencia del cliente.
- **Desarrollar un flujo completo de cotización** que permita al usuario seleccionar productos, revisar detalles y enviar una solicitud formal que llegue al correo o WhatsApp del vivero.
- **Incorporar una sección de blog o guías educativas**, permitiendo al vivero compartir contenido relevante sobre cuidados de plantas, recomendaciones, tendencias y buenas prácticas de jardinería.
- **Construir un panel administrativo con autenticación por roles**, desde el cual se puedan gestionar productos, categorías, imágenes, entradas del blog y solicitudes de cotización.
- **Optimizar la plataforma para buscadores web (SEO)**, asegurando que el vivero mejore su visibilidad en Google y aumente el alcance de clientes potenciales.
- **Garantizar la accesibilidad universal conforme a estándares WCAG 2.1**, permitiendo que personas con discapacidades puedan navegar el sitio sin limitaciones.
- **Integrar mecanismos avanzados de seguridad**, incluyendo cifrado de contraseñas, validación de entradas, sanitización de formularios y políticas de control de acceso.
- **Asegurar un rendimiento óptimo**, priorizando tiempos cortos de carga, compresión de imágenes, uso de caché y buenas prácticas de optimización web para ofrecer una experiencia fluida.

Justificación

La digitalización del comercio ha transformado la manera en que los clientes interactúan con los negocios; el sector de viveros no es la excepción. En la actualidad, los usuarios buscan información rápida, visualmente atractiva y confiable antes de realizar una compra o solicitar asesoría. Un sitio web profesional brinda al vivero una ventaja competitiva significativa, ya que permite:

- **Exponer su catálogo de forma atractiva y fácil de navegar**, aumentando la probabilidad de que el cliente encuentre la planta o producto que busca.
- **Reducir la carga administrativa**, mediante la automatización de cotizaciones y la centralización de la información.
- **Generar confianza**, ya que un sitio moderno y funcional transmite profesionalismo y seriedad.
- **Crear una comunidad**, gracias a la publicación de contenido educativo que fortalece la relación entre el vivero y sus clientes.
- **Aumentar la visibilidad**, mediante estrategias básicas de SEO y presencia digital.
- **Preparar el negocio para etapas futuras**, como integrar pagos en línea, envíos, inventarios avanzados o reservas de talleres.

Alcances Esperados

El sistema desarrollado para ViveGreen ofrecerá, como mínimo, las siguientes capacidades:

1. Sección pública

- Catálogo de productos con fichas técnicas profesionales (nombre común/científico, cuidados, riego, luz, tamaño y precio referencial).
- Sistema de búsqueda inteligente y filtros por categorías, tipo de planta, tamaño, condiciones ambientales y disponibilidad.

- Blog/Guías con artículos optimizados para SEO y enriquecidos con imágenes.
- Formulario de cotización con resumen de productos seleccionados.
- Página de contacto con mapa, WhatsApp, correo y horarios.

2. Sección administrativa

- Panel admin con roles (admin/editor).
- CRUD completo de productos, categorías e imágenes.
- Gestión del contenido del blog.
- Sección para consultar solicitudes de cotización recibidas.
- Opciones de actualización y mantenimiento.

3. Requerimientos transversales

- Diseño responsive adaptable a móviles, tablets y computadoras.
- Arquitectura escalable para futuras integraciones como pagos o envíos.
- Integración con plataformas externas (WhatsApp, Google Maps, correo SMTP).
- Seguridad reforzada en cada formulario y endpoint.

Requerimientos del Proyecto

Requerimientos Funcionales

- Gestión de productos, categorías, descripción detallada, características y galería de imágenes.
- Sistema de cotización que permita agregar productos, ingresar datos del cliente y generar una solicitud formal.
- Módulo de contacto con validación de campos y envío de mensajes.
- Blog informativo con capacidad para crear, editar y eliminar artículos.

- Panel administrativo con login seguro, roles, permisos y control de acceso.
- Integración con WhatsApp y correo para notificaciones rápidas.

Requerimientos No Funcionales

Seguridad:

- Cifrado de contraseñas con bcrypt
- Validación/sanitización de datos
- Protección CSRF y XSS
- HTTPS obligatorio

Usabilidad: interfaz intuitiva, navegación simple y accesible según WCAG AA.

Rendimiento: optimización de imágenes, carga diferida, minificación y caché.

SEO: metaetiquetas, sitemap, robots.txt y datos estructurados.

Escalabilidad: arquitectura por capas y uso de API REST para crecimiento futuro.

Disponibilidad: uptime objetivo ≥ 99.5% según hosting.

Compatibilidad: soporte para navegadores modernos (Chrome, Edge, Firefox).

Parte II: Pre-Diseño Visual

Paleta de Colores

- Turquesa (#3BB9A3) – elemento principal.
- Gris claro (#F4F4F4) – fondos.
- Verde hoja (#6DBE45) – énfasis natural.
- Blanco (#FFFFFF) – áreas limpias.
- Gris oscuro (#333333) – texto.

Tipografía

- Segoe UI (principal)
- Roboto (secundaria)

Íconos

Librería recomendada: Lucide React o Material Icons. Estilo minimalista y coherente con la línea visual del sistema.

Descripción Visual General

El diseño del sistema busca transmitir frescura, modernidad y simplicidad. La distribución de pantallas incluye login, inicio, catálogo, detalle de planta, cotización, blog, contacto y panel administrativo.

Parte III: Base de Datos

Definición

La base de datos es no relacional y se diseñó siguiendo principios de normalización hasta 3FN. Cada tabla cuenta con llaves primarias y foráneas para mantener la integridad referencial.

Elementos Principales

- Tablas: Productos, Categorías, Clientes, Cotizaciones, BlogPosts.
- Tipos de datos: INT, VARCHAR, DECIMAL, DATE/DATETIME, BOOLEAN.
- Llaves foráneas: relaciones entre productos y categorías, blog y usuario admin.
- Índices para búsquedas rápidas.

Parte IV: Desarrollo del Backend

Arquitectura Utilizada

El backend se desarrolló bajo una arquitectura por capas (Presentación, Negocio, Datos), tal como recomiendan las buenas prácticas para mantenimiento y escalabilidad.

Componentes del Backend

- Modelos: representan entidades del sistema.
- Controladores: gestionan peticiones HTTP.
- Servicios: lógica de procesamiento.
- Repositorio: acceso a base de datos.
- DTOs: transferencia limpia de datos.

Seguridad

- Incluye autenticación por roles, cifrado de contraseñas mediante bcrypt, validaciones, sanitización de entradas y uso obligatorio de HTTPS.

Gestión de Errores y Logs

- El sistema implementa manejo de excepciones centralizado y generación de logs para monitoreo y depuración.

Conexión con Base de Datos y Frontend

- La API REST expone endpoints para productos, categorías, cotización y blog. El frontend consume estos servicios mediante peticiones AJAX/Fetch.

Fragmentos de Código

- ```
  // Ejemplo ilustrativo
  app.get('/api/productos', async (req, res) => {
    const productos = await Producto.find();
    res.json(productos);
  });
  ```
  ```

El pre-diseño visual constituye la primera representación conceptual del sistema ViveGreen. Esta etapa define la identidad visual, estructura de navegación y experiencia de usuario que se implementará posteriormente en el frontend. Los elementos seleccionados buscan transmitir modernidad, naturaleza y simplicidad, ajustándose al propósito del proyecto: ofrecer una plataforma digital atractiva y funcional para un vivero costarricense.

1. Paleta de Colores

La paleta fue elegida para reflejar los valores del vivero: vida, frescura, naturaleza y profesionalismo.

Colores principales

- **Turquesa (#3BB9A3)**
  Color predominante en botones, encabezados y elementos interactivos. Evoca frescura, naturaleza y modernidad; genera una interfaz agradable y energizante.
- **Verde hoja (#6DBE45)**
  Se emplea como color de acento para destacar información importante, estados positivos o indicadores visuales relacionados con plantas y vida natural.

Colores secundarios

- **Gris claro (#F4F4F4)**
  Utilizado en fondos para secciones amplias, tarjetas y áreas de descanso visual. Facilita la lectura y aporta equilibrio.
- **Blanco (#FFFFFF)**
  Base de la mayoría de pantallas, garantiza contraste, limpieza y armonía con las plantas e imágenes.

- **Gris oscuro (#333333)**
  Color para textos principales, ya que ofrece alta legibilidad sin ser tan agresivo como el negro.

Justificación visual

La combinación refuerza una identidad moderna, verde y profesional. Los tonos turquesa y verde comunican naturaleza, mientras que los grises aportan neutralidad. Esto permite que las fotos de las plantas sean las protagonistas visuales.

2. Tipografías

Se seleccionaron tipografías modernas, legibles y estéticamente coherentes con el diseño minimalista del sistema:

- **Segoe UI – Fuente principal**
  Elegida por su excelente legibilidad en pantallas y su estilo limpio, ampliamente utilizada en interfaces modernas.
- **Roboto – Fuente secundaria**
  Se utiliza en elementos secundarios como formularios, descripciones y textos de apoyo. Roboto mantiene coherencia estética y presenta excelente rendimiento en web.

Ventajas:

- Alta legibilidad en móviles.
- Apariencia moderna y profesional.
- Fácil integración en entornos HTML/CSS.
- Variedad de grosores que permiten jerarquizar la información.

3. Íconos

Se propone utilizar una librería de íconos minimalistas y altamente compatibles con frameworks modernos:

Librerías sugeridas

- **Lucide React**
- **Material Icons**

Características del estilo

- Trazos finos y limpios.
- Estética minimalista que coincide con la paleta verde/turquesa.
- Disponibles en formato SVG, optimizando tiempos de carga.

Ejemplos de iconografía:

- 🏠 Inicio
- 🌱 Catálogo
- 💮 Detalle de planta
- 💬 Cotización
- ✉️ Contacto
- ⚙️ Administración

4. Estructura Visual del Sistema

Se definió la organización general de pantallas para asegurar una navegación fluida e intuitiva:

Pantallas principales

1. **Login:** acceso seguro para el panel administrativo.
2. **Inicio:** hero banner, eslogan del vivero, accesos rápidos.
3. **Catálogo:** tarjetas con foto, nombre, precio y disponibilidad.
4. **Detalle de Planta:** ficha técnica completa.
5. **Cotización:** listado de productos seleccionados + formulario del cliente.
6. **Contacto:** formulario validado + mapa + enlaces.
7. **Blog/Guías:** artículos educativos con diseño responsivo.

8. **Panel Administrativo:** CRUD de productos, categorías, blog e imágenes

5. Experiencia de Usuario (UX)

- Navegación clara basada en una barra superior fija.
- Animaciones livianas para interacción (hover, botones, transiciones suaves).
- Diseño totalmente responsive para asegurar usabilidad en móviles.
- Enfoque “mobile-first” priorizando carga rápida y estructura vertical.

Justificación

Una base relacional permite escalabilidad e integración futura con inventario, pagos y analítica. Además, facilita reporting y administración desde el panel principal.

PARTE IV: Desarrollo del Backend

1. Arquitectura Utilizada

El backend se desarrolló bajo una **arquitectura por capas**, considerada una de las más adecuadas para sistemas escalables y mantenibles. Esta separación permite:

- **Aislamiento de responsabilidades**.
- **Escalabilidad futura** (agregar nuevas funcionalidades sin romper el sistema).
- **Mantenibilidad** gracias al orden lógico del código.
- **Pruebas más simples** al separar lógica, datos y controladores.

Capas implementadas

1. **Capa de Presentación (Controllers):**
   Gestiona peticiones HTTP y coordina las respuestas JSON.
2. **Capa de Negocio (Services):**
   Contiene la lógica principal del proyecto: validaciones, cálculos, flujo de procesos.
3. **Capa de Datos (Repositories):**
   Responsable de acceder a la base de datos utilizando consultas u ORM.
4. **DTOs y Mappers:**
   Aseguran transferencia de datos limpia y estandarizada.

2. Componentes del Backend

Modelos

Representan las entidades centrales del sistema:

- Producto

- Categoría
- Cliente
- Cotización
- BlogPost

  Estos modelos definen campos, tipos de datos y reglas básicas.

Controladores

Gestionan las rutas expuestas por la API REST:

- /api/productos
- /api/categorias
- /api/cotizacion
- /api/blog
- /api/auth

Servicios

- Procesan lógica de negocio.
- Validan datos.
- Aplican reglas (ej.: cotizaciones con mínimo de productos).

Repositorio

- Ejecuta consultas a la base.
- Maneja inserciones, actualizaciones y eliminaciones.

3. Seguridad

El backend implementa múltiples capas de seguridad:

Autenticación y Autorización

- Login seguro con usuario y contraseña.
- Roles: administrador y editor.
- Tokens de sesión con expiración.

Cifrado

- Contraseñas cifradas con **bcrypt**.

Validación / Sanitización

- Prevención de ataques XSS en formularios.
- Filtrado de entradas sospechosas.

Uso de HTTPS

- Encriptación de datos en tránsito.

Protección de endpoints

- Rutas críticas solo accesibles para roles autorizados.

4. Gestión de Errores y Logs

El backend cuenta con:

- Middleware para manejo global de errores.
- Respuestas controladas en formato JSON.
- Registro de eventos importantes (logs) para auditoría:
  – Intentos de acceso fallidos
  – Errores del servidor
  – Acciones administrativas

5. Conexión con Base de Datos y Frontend

La comunicación se realiza mediante una **API REST** bien estructurada.

Flujo general

Frontend → solicita datos → Backend → consulta BD → responde en JSON.

Tecnologías utilizadas

- Node.js + Express
- Mongoose / consultas SQL
- Fetch API en el frontend

6. Ejemplo de Endpoint Real

```
app.get('/api/productos', async (req, res) => {
  try {
    const productos = await Producto.find();
    res.status(200).json(productos);
  } catch (error) {
    res.status(500).json({ mensaje: "Error obteniendo productos", error });
  }
});
```

Incluye manejo de errores, respuesta JSON y acceso al modelo.

Conclusión

El proyecto **ViveGreen** representa la consolidación de una solución tecnológica integral orientada a satisfacer las necesidades actuales de un vivero moderno mediante la implementación de un sitio web robusto, escalable y altamente funcional. A lo largo del desarrollo del sistema se definieron y aplicaron principios fundamentales de ingeniería de software, diseño de interfaces, arquitectura de sistemas y gestión de datos, logrando un producto coherente, eficiente y alineado con los objetivos iniciales del proyecto.

En primera instancia, se diseñó una identidad visual clara y consistente que combina una paleta de colores fresca, tipografías modernas y elementos gráficos minimalistas. Este enfoque de diseño permite transmitir la esencia natural del vivero, al tiempo que garantiza una navegación agradable y accesible para todo tipo de usuarios. El uso de una estructura visual intuitiva y un diseño responsivo facilita que el sistema sea funcional desde distintos dispositivos, requisito indispensable en el entorno digital actual.

Desde el punto de vista técnico, se implementó una **base de datos relacional normalizada**, lo cual asegura integridad, coherencia y un rendimiento óptimo en las operaciones de consulta, registro y actualización de información. El diseño de las tablas y relaciones fue pensado para permitir que el proyecto pueda crecer e incorporar nuevas funcionalidades en etapas futuras, tales como gestión de inventarios, pedidos en línea o métricas de venta, sin comprometer la estabilidad del sistema.

En cuanto al backend, se desarrolló una **arquitectura modular por capas**, lo cual permite separar adecuadamente la lógica de negocio, el acceso a datos y la capa de presentación. Esta arquitectura facilita la mantenibilidad del sistema, disminuye la complejidad del código y mejora la escalabilidad del proyecto, permitiendo la integración futura de nuevas características sin necesidad de realizar cambios estructurales profundos. Asimismo, se incorporaron prácticas de seguridad

esenciales, tales como cifrado de contraseñas, validación de datos, sanitización de entradas e implementación de rutas protegidas para los procesos administrativos.

El sistema ViveGreen también integra un conjunto de funcionalidades relevantes para el negocio, entre las cuales destacan el catálogo interactivo de plantas, el módulo de cotizaciones, la sección de blog educativo y un panel administrativo completo. Cada una de estas funcionalidades fue diseñada para optimizar los procesos internos del vivero, mejorar la experiencia del cliente, reducir tiempos de atención y fortalecer la presencia digital del negocio. En especial, el sistema de cotizaciones constituye una mejora considerable, ya que automatiza una tarea que tradicionalmente se realiza de forma manual, incrementando la precisión y rapidez de respuesta.

Finalmente, el proyecto demuestra la aplicabilidad de los conceptos teóricos aprendidos en el curso y evidencia la capacidad de diseñar, estructurar y desarrollar un sistema completo desde cero. ViveGreen no solo cumple con los objetivos planteados, sino que establece una base sólida para su evolución futura. El sistema queda preparado para incorporar nuevas funcionalidades, integrarse con servicios externos e incluso transformarse en una plataforma de comercio electrónico en fases posteriores.

En conclusión, **ViveGreen es una solución tecnológica moderna, funcional y escalable**, que aprovecha buenas prácticas de diseño, programación y arquitectura para ofrecer un producto con alto valor agregado tanto para el vivero como para sus clientes. Este proyecto refleja un trabajo integral que combina creatividad, conocimiento técnico y enfoque profesional, convirtiéndose en una herramienta real de crecimiento digital para el negocio.