**Informe Técnico del Proyecto: VcBikes**

**1. Introducción**

Este documento tiene como objetivo detallar el proceso de desarrollo, implementación y validación del aplicativo web VcBikes, diseñado para la gestión integral de operaciones en el ámbito de bicicletas (ventas, inventario, clientes, empleados, etc.). Se explica la concepción de la idea, la selección de tecnologías, la estructura del proyecto, las funcionalidades implementadas y las pruebas realizadas para garantizar un producto robusto y seguro.

**2. Título del Proyecto**

**VcBikes – Aplicativo Web de Gestión Integral para Negocios de Bicicletas**

**3. Resumen**

El sistema VcBikes se ha desarrollado para resolver los problemas de organización y gestión que enfrentan los negocios del sector de bicicletas, facilitando el manejo de clientes, empleados, productos, ventas y proveedores. Entre sus beneficios se encuentran:

* Centralización de la información y procesos.
* Gestión de inventario y stock en tiempo real.
* Control de ventas con generación automática de facturas.
* Sistema de autenticación con roles diferenciados para administradores y empleados.
* Interfaz responsiva basada en Bootstrap 5.3 que garantiza la accesibilidad en diversos dispositivos.

**4. Objetivos**

**4.1 Objetivo General**

Desarrollar un aplicativo web robusto y escalable que permita la administración integral de las operaciones de un negocio de bicicletas, optimizando procesos y facilitando la toma de decisiones.

**4.2 Objetivos Específicos**

* **Diseño de Interfaz:** Crear una experiencia de usuario intuitiva y responsiva mediante el uso de HTML, CSS, JavaScript y Bootstrap 5.3.
* **Gestión de Entidades:** Implementar módulos CRUD para la gestión de clientes, empleados, productos, ventas, categorías, subcategorías y proveedores.
* **Seguridad y Autenticación:** Integrar mecanismos de autenticación y autorización, utilizando sesiones, roles de usuario y cifrado de contraseñas.
* **Reportes y Estadísticas:** Desarrollar dashboards y reportes que proporcionen estadísticas clave (ventas mensuales, stock bajo, productos más vendidos, etc.).
* **Mantenimiento y Escalabilidad:** Estructurar el proyecto de forma modular, utilizando Flask-Migrate y modelos ORM para facilitar futuras ampliaciones o modificaciones.

**5. Alcance del Proyecto**

**Incluye:**

* **Gestión de Usuarios:** Registro, inicio de sesión, actualización de perfil y cambio de contraseña.
* **Administración de Datos:** Módulos para la gestión de productos, clientes, empleados, proveedores, categorías y subcategorías.
* **Control de Ventas:** Registro de ventas, generación de facturas y ajuste de stock en tiempo real.
* **Interfaz de Usuario:** Páginas responsivas y accesibles que faciliten la interacción, diseñadas con Bootstrap 5.3.
* **Manejo de Archivos:** Subida y gestión de imágenes para productos.

**No Incluye:**

* Integración con pasarelas de pago u otros sistemas de procesamiento financiero.
* Desarrollo de aplicaciones móviles nativas.

**6. Análisis del Problema**

**6.1 Planteamiento del Problema**

Los negocios de bicicletas suelen enfrentar dificultades en la administración de su inventario, seguimiento de ventas y gestión de clientes, lo que puede traducirse en procesos ineficientes y errores en el manejo de la información. La falta de integración entre los diferentes módulos administrativos puede retrasar la toma de decisiones y afectar la experiencia del cliente.

**6.2 Justificación**

El desarrollo de VcBikes surge como respuesta a la necesidad de optimizar los procesos administrativos, garantizando mayor eficiencia, control y visibilidad sobre las operaciones del negocio. La solución propuesta centraliza la información y facilita la integración de distintos módulos, lo que permite una gestión ágil y segura.

**6.3 Usuarios Involucrados**

* **Administradores:** Responsables de la configuración del sistema, gestión de usuarios y supervisión general de las operaciones.
* **Empleados:** Encargados de realizar ventas, gestionar clientes y actualizar la información de productos.
* **Clientes (Indirectamente):** Aunque el sistema no es de cara al público final, la correcta administración repercute en una mejor atención y servicio.

**7. Diseño del Aplicativo Web**

**7.1 Arquitectura del Sistema**

El aplicativo se ha desarrollado siguiendo el patrón Modelo-Vista-Controlador (MVC):

* **Modelo:** Gestiona la lógica de negocio y la interacción con la base de datos MySQL a través de SQLAlchemy y Flask-Migrate.
* **Vista:** Se implementa mediante plantillas HTML, con estilos y componentes interactivos utilizando CSS, JavaScript y Bootstrap 5.3.
* **Controlador:** Maneja las solicitudes HTTP, procesando formularios, autenticación, validación de datos y generando respuestas dinámicas.

**7.2 Diseño de Interfaces**

Las principales interfaces desarrolladas incluyen:

* **Página de Inicio y Contacto:** Acceso libre a información general.
* **Autenticación:** Formularios de registro, inicio de sesión y recuperación de contraseña.
* **Dashboard:** Panel principal con estadísticas (ventas, stock, productos, etc.).
* **Módulos de Gestión:** Interfaces específicas para la administración de clientes, empleados, productos, proveedores, categorías y ventas.
* **Facturación:** Vista para la generación y visualización de facturas.

**7.3 Flujo de Navegación**

El usuario inicia en la página de inicio y puede acceder a la sección de contacto o iniciar sesión. Una vez autenticado, el usuario es redirigido al dashboard, desde donde puede navegar a las distintas secciones del sistema (clientes, productos, ventas, etc.) y regresar a la pantalla anterior según sea necesario.

**8. Desarrollo e Implementación**

**8.1 Tecnologías Utilizadas**

* **Backend:** Python con el framework Flask.
* **Frontend:** HTML, CSS, JavaScript y Bootstrap 5.3.
* **Base de Datos:** MySQL gestionado mediante SQLAlchemy.
* **Control de Versiones:** Git/GitHub.
* **Otros:** Flask-Migrate para la gestión de esquemas de base de datos y manejo de sesiones, y la librería dotenv para la gestión de variables de entorno.

**8.2 Estructura del Proyecto**

El proyecto se organiza de la siguiente manera:

/VcBikes

/static --> Archivos estáticos (CSS, JS, imágenes)

/templates --> Plantillas HTML

app.py --> Archivo principal que contiene la configuración y rutas de la aplicación

/models --> Definición de los modelos de la base de datos

/forms --> Formularios para la interacción del usuario

.env --> Variables de entorno (credenciales, configuración, etc.)

app.log --> Archivo de log para el registro de eventos y errores

**8.3 Seguridad y Autenticación**

Se han implementado diversas medidas para garantizar la seguridad del sistema:

* **Autenticación y Roles:** Uso de sesiones para identificar a los usuarios y control de acceso basado en roles (administrador y empleado).
* **Cifrado de Contraseñas:** Las contraseñas se almacenan de forma segura utilizando hashing (werkzeug.security).
* **Validación y Manejo de Errores:** Validación de formularios y manejo de excepciones con rollback de transacciones en caso de error.
* **Protección de Rutas:** Decoradores para restringir el acceso a rutas sensibles, exigiendo autenticación y roles específicos.

**8.4 Implementación de Funcionalidades**

Entre las funcionalidades clave implementadas se encuentran:

* **Registro e Inicio de Sesión:** Permite el acceso seguro al sistema y asignación de roles.
* **Gestión de Inventario y Ventas:** Registro de productos, control de stock, generación de facturas y registro de ventas con cálculo automático de impuestos y descuentos.
* **Mantenimiento de Información:** Funciones CRUD para la administración de clientes, empleados, proveedores, categorías y subcategorías.
* **Generación de Reportes:** Dashboard con estadísticas en tiempo real sobre ventas, ingresos y productos con stock bajo.
* **Manejo de Archivos:** Gestión de imágenes de productos con validación de formatos y almacenamiento en servidor.

**9. Pruebas y Validación**

**9.1 Pruebas de Funcionalidad**

Se realizaron pruebas unitarias y de integración en cada módulo para confirmar que:

* El proceso de registro y autenticación funcione correctamente.
* Los formularios de entrada validen la información requerida.
* Las operaciones CRUD se ejecuten sin errores y actualicen la base de datos de manera consistente.

**9.2 Pruebas de Interfaz**

Se evaluó la usabilidad y responsividad del sistema en diferentes dispositivos y navegadores, asegurando que:

* La interfaz sea intuitiva y de fácil navegación.
* El diseño se ajuste adecuadamente en dispositivos móviles y de escritorio.

**9.3 Pruebas de Seguridad**

Se llevaron a cabo pruebas para detectar vulnerabilidades como inyecciones SQL y ataques CSRF, además de verificar:

* El correcto cifrado de las contraseñas.
* El control de acceso a rutas protegidas mediante roles.

**10. Despliegue del Aplicativo**

**10.1 Configuración del Servidor**

El aplicativo se ha configurado para ejecutarse en un entorno de producción mediante:

* Variables de entorno para la configuración segura de la aplicación (credenciales de base de datos, claves secretas, etc.).
* Un servidor web (por ejemplo, mediante un VPS o plataformas en la nube como AWS o Heroku) que gestione las peticiones y garantice la escalabilidad.

**10.2 Instalación y Ejecución**

Para instalar y ejecutar VcBikes se siguieron los siguientes pasos:

* **Instalación de Dependencias:** Uso de un entorno virtual de Python y ejecución de pip install -r requirements.txt.
* **Migraciones de Base de Datos:** Uso de Flask-Migrate para crear y actualizar las tablas de la base de datos.
* **Ejecución del Servidor:** Iniciar la aplicación con el comando python app.py, la cual se ejecuta en el puerto configurado (por defecto, 5000).

**11. Conclusiones y Recomendaciones**

**11.1 Resultados Obtenidos**

El desarrollo de VcBikes permitió consolidar un sistema completo de gestión que integra múltiples módulos administrativos, mejora la eficiencia operativa y ofrece herramientas de análisis en tiempo real, cumpliendo con los objetivos planteados.

**11.2 Limitaciones**

* **Integración de Pagos:** El sistema actual no contempla la integración con pasarelas de pago.
* **Interfaz de Usuario Avanzada:** Aunque se garantiza responsividad, podrían mejorarse aspectos visuales y de experiencia de usuario.

**11.3 Mejoras Futuras**

* **Integración de Pagos en Línea:** Incorporar módulos que permitan la conexión con sistemas de pago para facilitar transacciones.
* **Aplicación Móvil:** Desarrollar una versión móvil nativa para ampliar el alcance del sistema.
* **Optimización de Reportes:** Ampliar la funcionalidad de análisis de datos y reportes, integrando herramientas de Business Intelligence.

**12. Bibliografía y Referencias**

* **Documentación Oficial de Flask:** [Flask Documentation](https://flask.palletsprojects.com/)
* **SQLAlchemy:** [SQLAlchemy Documentation](https://www.sqlalchemy.org/)
* **Bootstrap 5:** [Bootstrap Documentation](https://getbootstrap.com/)
* **Python Dotenv:** [python-dotenv](https://github.com/theskumar/python-dotenv)
* **Guías de Seguridad en Web:** Artículos y publicaciones especializadas sobre prácticas de desarrollo seguro en aplicaciones web.