



**Escuela Politécnica Nacional
Facultad de Ingeniería en Sistemas**

**Construcción y evolución de software
2025-B**

**Documento de Construcción y Evolución de
Software**

Docente:

Vicente Adrián Eguez Sarzosa

Estudiantes:

Julián Alexsei Narvaéz Vinueza

Juan Mateo Quisilema Flores

Sebastián Patricio Ramos Carrillo



1. Introducción

El presente documento describe la construcción y evolución del software correspondiente al proyecto Sistema Web de Gestión de Ventas, una aplicación web orientada a apoyar la administración de procesos comerciales como el registro de productos, clientes y ventas, permitiendo un control centralizado y automatizado de la información.

Este sistema surge como respuesta a la necesidad de reducir errores derivados del manejo manual de datos, mejorar la trazabilidad de las transacciones y facilitar la toma de decisiones mediante información organizada y accesible en tiempo real.

El objetivo principal de este documento es establecer las estrategias técnicas y metodológicas que guiarán el desarrollo del software, incluyendo la arquitectura del sistema, los flujos de trabajo, los mecanismos de integración y entrega continua, así como los procesos de revisión y aprobación que aseguran la calidad y evolución controlada del sistema a lo largo del tiempo.

2. Arquitectura del proyecto

La arquitectura del Sistema Web de Gestión de Ventas se basa en un modelo de arquitectura en capas, que separa claramente las responsabilidades del sistema para facilitar su mantenimiento, escalabilidad y evolución.

El frontend, desarrollado utilizando React, se encarga de la presentación de la información y de la interacción directa con el usuario. Desde esta capa, los usuarios pueden gestionar productos, clientes y ventas mediante una interfaz web intuitiva.

El backend, implementado en Node.js, contiene la lógica de negocio del sistema, incluyendo validaciones, reglas de cálculo de ventas y control de accesos. Esta capa expone sus funcionalidades a través de una API REST, la cual es consumida por el frontend.

La base de datos, basada en PostgreSQL, almacena de forma persistente la información relacionada con usuarios, productos,



clientes y transacciones de venta. El backend es el único componente con acceso directo a la base de datos, garantizando la seguridad y consistencia de la información.

La comunicación entre los componentes se realiza de manera desacoplada, lo que permite modificar o ampliar cada módulo sin afectar directamente al resto del sistema.

3. Estrategia de Pipelines (CI/CD)

Para asegurar la calidad del software y reducir errores durante el desarrollo, el proyecto implementa una estrategia de Integración Continua (CI) y Entrega Continua (CD).

En el pipeline de Integración Continua, cada vez que se realiza un commit o un push a la rama develop, se ejecutan automáticamente las siguientes etapas:

- Análisis de estilo y formato del código mediante herramientas de linting.
- Ejecución de pruebas unitarias para verificar el correcto funcionamiento de los módulos del sistema.
- Generación automática del build del frontend y backend.
- Despliegue automático del sistema en un entorno de pruebas o staging.

En el pipeline de Entrega Continua, una vez que los cambios han sido validados y aprobados mediante un Pull Request, el sistema se despliega automáticamente en el entorno de producción. Este proceso garantiza que solo código revisado y probado forme parte de la versión final del sistema.

4. Estrategia de flujos de desarrollo

El desarrollo del sistema se gestiona mediante un modelo de control de versiones basado en ramas, lo que permite un trabajo colaborativo ordenado y seguro.

La rama main contiene la versión estable del sistema que se encuentra en producción.

La rama develop se utiliza como punto de integración de nuevas funcionalidades y mejoras.



Las ramas feature/* se crean para desarrollar funcionalidades específicas, como la gestión de productos o el registro de ventas.

Las ramas hotfix/* se utilizan para corregir errores críticos detectados en producción.

Cuando una funcionalidad ha sido completada, se crea un Pull Request desde la rama feature hacia develop, el cual debe ser revisado y aprobado antes de su integración.

5. Gestión de historias de usuario

Las funcionalidades del Sistema Web de Gestión de Ventas se definen mediante historias de usuario, siguiendo un enfoque ágil que permite priorizar el valor para el usuario final.

Las historias de usuario utilizan el siguiente formato:

Como [rol], quiero [funcionalidad], para [beneficio].

Un ejemplo de historia de usuario es:

Como administrador, quiero registrar y visualizar las ventas diarias, para tener control sobre los ingresos del negocio.

Las historias de usuario se gestionan en herramientas como Jira o Trello, donde se asignan prioridades, responsables y se organizan en sprints de dos semanas. Esto permite realizar un seguimiento claro del avance del proyecto y facilita la planificación de nuevas funcionalidades.

6. Estrategia de revisiones y aprobaciones

Con el fin de mantener un alto nivel de calidad en el código, todos los cambios realizados en el sistema deben ser gestionados mediante Pull Requests.

Cada Pull Request debe incluir una descripción clara de los cambios realizados y estar asociado a una historia de usuario específica. Antes de ser aprobado, el código es revisado por al menos un miembro del equipo, quien verifica el cumplimiento de los estándares de desarrollo.



Para apoyar este proceso, se utiliza un checklist de revisión que incluye:

- Cumplimiento de los estándares de estilo de código.
- Ejecución exitosa de pruebas unitarias.
- Actualización de la documentación correspondiente.

Solo después de cumplir con estos criterios, el Pull Request puede ser aprobado y fusionado en la rama correspondiente.

7. Herramientas y conexiones

El proyecto utiliza diversas herramientas para facilitar la gestión, desarrollo y comunicación del equipo.

La gestión de tareas y seguimiento de historias de usuario se realiza mediante Jira o Trello. El código fuente se almacena en repositorios GitHub o GitLab, donde se gestionan las ramas y Pull Requests.

La automatización de los procesos de integración y entrega continua se implementa mediante GitHub Actions o Jenkins. Para la comunicación del equipo, se utilizan herramientas como Slack o Microsoft Teams, las cuales permiten recibir notificaciones automáticas relacionadas con commits, revisiones de código y despliegues.

La integración de estas herramientas asegura una trazabilidad completa desde los requisitos definidos hasta el software desplegado en producción.

8. Conclusiones

Las estrategias descritas en este documento permiten una construcción y evolución controlada del Sistema Web de Gestión de Ventas, asegurando altos niveles de calidad, organización y trazabilidad.

La arquitectura en capas facilita el mantenimiento y la escalabilidad del sistema, mientras que los pipelines de CI/CD permiten detectar errores de forma temprana y garantizar la estabilidad de cada versión entregada.



Finalmente, el uso de flujos de desarrollo bien definidos, revisiones de código y herramientas de gestión contribuye a una evolución ordenada del software, alineada con los objetivos del proyecto y las necesidades del negocio.