

Desarrollo de un Sitio Web Transaccional para PYMES: Un Enfoque Práctico con Spring Boot

Brandon Josué Varga Moreira

Jonathan Cortés Castro

Elias Viquez Segura

Universidad Fidélitas

bvargas40854@ufide.ac.cr

Jcortes20382@ufide.ac.cr

jviquez20768@ufide.ac.cr

Abstract— This article presents the development of a transactional website for small and medium-sized enterprises, employing the use of Java and the Spring Boot framework. Conducted as part of a university course, the project aimed to address simulated client needs by applying the knowledge and skills acquired during the course. The methodology combined practical learning, guided by web design and patterns, with collaborative GitHub-based teamwork. The project's key functionalities include user authentication and role management, a shopping cart system, invoicing, and CRUD operations for product management. Despite not following a formal methodology, the iterative implementation process, supported by user stories and prototypes, ensured alignment with the initial requirements. Challenges such as front-end debugging and team coordination were addressed with manual reviews and version control strategies. The project demonstrates the potential of accessible, tailored technology solutions for SMEs while reflecting on lessons learned and offering recommendations for future work.

1. Resumen

En este artículo se presenta el desarrollo de un sitio web transaccional dirigido a PYMES, implementado con Java y Spring Boot. Este proyecto, realizado en el marco de un curso universitario como lo es Desarrollo de Aplicaciones Web y Patrones. Dicho proyecto tiene como objetivo satisfacer necesidades específicas de clientes, esto mediante la aplicación de varios temas y además de las herramientas vistas en clase. La metodología de trabajo estuvo basada en un aprendizaje práctico, guiado por diseño y patrones web, complementado con el uso colaborativo de GitHub. Entre las funcionalidades

destacadas se incluyen un sistema de autenticación y roles (log in / log out), carrito de compras, facturación y gestión CRUD de productos, un carrito y usuarios. Finalmente, se reflexiona sobre los aprendizajes adquiridos y se proponen recomendaciones para proyectos futuros.

2. Introducción

Según el Pacto Mundial de la ONU [6], las pequeñas y medianas empresas (PYMES) representan el 90% de las empresas y el 50% del empleo total a nivel mundial, lo que subraya su importancia en la economía global y la necesidad de adoptar herramientas digitales competitivas.

Las pequeñas y medianas empresas enfrentan el desafío de adaptarse a un entorno cada vez más digitalizado. La creación de soluciones tecnológicas accesibles y funcionales se vuelve crucial para fortalecer su competitividad. Este proyecto se desarrolló con el objetivo de diseñar un sitio web transaccional que permita a las PYMES gestionar productos, ventas y usuarios de manera eficiente y sencilla aparte de ser agradable e intuitiva para el usuario. La iniciativa también buscó consolidar conocimientos adquiridos en temas como patrones de diseño, desarrollo web y manejo de bases de datos, normalización de tablas, promoviendo el aprendizaje colaborativo y la aplicación práctica de tecnologías modernas las cuales nos permitieron dividir la carga de trabajo para de esta manera realizar una labor más eficiente.

3. Metodología

El desarrollo del proyecto se llevó a cabo en varias etapas clave:

1. Historias de Usuario y Diseño del Prototipo:

En la etapa inicial, se definieron historias de usuario específicas que alineaban los requerimientos del cliente con las funcionalidades deseadas. Estas historias sirvieron como guía para el diseño de un prototipo interactivo.

2. Implementación Iterativa:

- El avance del proyecto se realizó en etapas del 40%, 60% y entrega final.
- Cada etapa se enfocó en implementar funcionalidades clave siguiendo las historias de usuario.
- Se utilizó el patrón MVC para estructurar el sistema, junto con Spring Boot para el backend y Thymeleaf para la presentación. El patrón MVC, que separa las preocupaciones en tres componentes (Modelo, Vista y Controlador), facilita el mantenimiento y la escalabilidad del sistema. Según Rodríguez en *freeCodeCamp* [5], este patrón mejora la organización del código, permitiendo que cada parte del sistema se maneje de manera independiente, lo que es ideal para proyectos web de gran tamaño.

Al realizar el proyecto de esta manera, nos facilitó el manejo de versiones y archivos, implementando un tipo de metodología Cascada, ya que cada avance le sumaba al anterior. Este enfoque permitió un orden en el proceso de desarrollo, similar a lo que describen Gamma et al. [1], y facilitó la implementación de buenas prácticas de diseño.

Al realizar el proyecto de esta manera nos facilitó el manejo de versiones y archivos y de esta manera implementar un tipo de metodología cascada la cual cada avance le sumaba a el anterior, y de cierta manera se pudo seguir un orden junto a lo visto en la clase.

3. Uso de GitHub:

El control de versiones fue esencial para coordinar el trabajo en equipo. Se utilizaron ramas individuales para cada colaborador y Pull Requests para integrar cambios significativos. Esto fue un gran aporte al proyecto, ya que nos permitió revertir cambios erróneos a versiones previas de manera eficiente [4].

4. Enfoque Práctico:

Aunque no se utilizó una metodología formal, se adoptó un enfoque guiado por las prácticas vistas en clase, permitiendo una adaptación flexible y eficiente. Como se mencionó al principio, no se utilizó una metodología específica, pero se implementó un enfoque tipo Cascada, ya que no se avanzaba a la siguiente fase hasta no tener

un entregable mayormente finalizado, lo que refleja la flexibilidad del enfoque ágil sugerido por Beck & Andres [3]. Además, no se adoptó la metodología ágil de Scrum, ya que, como menciona Consulting Systems [7], Scrum requiere una planificación extensa que no se llevó a cabo en este proyecto. Resultados y Discusión

Funcionalidades Implementadas: El sistema desarrollado incluyó las siguientes funcionalidades clave:

- CRUD de productos: Los usuarios con los roles adecuados pueden agregar, editar y eliminar productos de la tienda. Esto así ya que los usuarios comunes sin roles específicos solo podrán realizar la visualización de estos.
- Carrito de compras: Integración de un sistema de carrito que permite gestionar los productos seleccionados para la compra, si como eliminar dicho producto o agregar una mayor cantidad del mismo.
- Autenticación y roles: Sistema de log in y log out con roles definidos para usuarios y administradores.
- Facturación: Generación de facturas automatizadas al finalizar una compra.
- Internacionalización: Implementación de soporte para múltiples idiomas, ampliando la accesibilidad del sitio.

5. Desafíos y Soluciones

- Coordinación de equipo: El uso de Git y ramas organizadas ayudó a resolver conflictos y garantizar la integración fluida de código. Esto se debe a que, como menciona Amazon Web Services, el uso de ramas permite que los equipos trabajen en tareas separadas sin interferir en el código principal, reduciendo así los conflictos. Además, se facilita la integración continua mediante pull requests y revisiones de código, lo que mantiene un flujo de trabajo organizado [1]. Sin embargo, también ocurrieron situaciones en las cuales hubo conflictos de versiones distintas por una mala comunicación y gestión del programa, y mediante el uso de un repositorio, este facilitó y agilizó el manejo de las soluciones [4].
- Errores en HTML: La falta de notificación de errores en el IDE fue mitigada mediante revisiones manuales y pruebas frecuentes. Según lo explicado en *DesarrolloWeb* [9], es importante entender la sintaxis de HTML para evitar errores comunes, como el mal cierre de etiquetas o el uso incorrecto de atributos, lo cual puede afectar la estructura y presentación de una página web. La revisión constante del código es clave para asegurar su correcto funcionamiento.

- Cumplimiento del prototipo: Aunque el diseño inicial se respetó, se realizaron ajustes para mejorar la experiencia del usuario. Esto ya que tras el interactuar con la aplicación final nos percatamos que quizás lo presentado al momento de realizar el prototipo no fue quizás lo más adecuado para el cliente y por ello se realizaron cambios significativos para de esta manera proveer una mejor experiencia.

6. Evaluación del Cliente Potencia

El proyecto fue evaluado por el docente como cliente simulado, quien destacó la alineación entre las funcionalidades y los requerimientos iniciales.

7. Conclusiones

El desarrollo de este proyecto permitió aplicar conceptos avanzados de programación y diseño web en un entorno práctico, ofreciendo una solución funcional y accesible para PYMES. Entre los aprendizajes clave se destacan la importancia de la coordinación del equipo, el manejo efectivo de versiones con GitHub y la planificación basada en historias de usuario. Como sugiere el uso de GitHub [4], la gestión de versiones y la colaboración en equipo son fundamentales para evitar conflictos de código. Asimismo, se identificaron áreas de mejora como la necesidad de adoptar metodologías formales en futuros proyectos para optimizar tiempos y procesos, como se menciona en la importancia de las metodologías ágiles [10]. El proyecto no solo reforzó habilidades técnicas, sino también fomentó el trabajo colaborativo y el pensamiento crítico para resolver problemas reales. Finalmente, se recomienda la incorporación de pruebas automatizadas y herramientas avanzadas de depuración en desarrollos futuros, así como explorar tecnologías emergentes para mejorar la escalabilidad y seguridad de las aplicaciones web.

8. Referencias

- [1] Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley.
- [2] "Spring Boot Documentation." Spring.io. Disponible en: <https://spring.io/projects/spring-boot>
- [3] Beck, K., & Andres, C. (2004). Extreme Programming Explained: Embrace Change. Addison-Wesley.
- [4] "GitHub Documentation." GitHub.com. Disponible en: <https://docs.github.com/>
- [5] M. Rodríguez, "El modelo de arquitectura View-Controller Pattern," freeCodeCamp Español, 2021. [Enlace]. Disponible en:

<https://www.freecodecamp.org/espanol/news/el-modelo-de-arquitectura-view-controller-pattern/>

- [6] Pacto Mundial de las Naciones Unidas. (s. f.). El papel clave de las PYMES en la sostenibilidad empresarial. Recuperado el 17 de diciembre de 2024, de <https://www.pactomundial.org/noticia/el-papel-clave-de-las-pymes-en-la-sostenibilidad-empresarial>
- [7] Consulting Systems, "Cascada vs Scrum," Consulting Systems, 2023. [Enlace]. Disponible en: <https://consulting-systems.tech/cascada-vs-scrum/>
- [8] Amazon Web Services, "Ventajas y desventajas de la estrategia GitHub Flow," AWS Prescriptive Guidance, 2024. [Enlace]. Disponible en: https://docs.aws.amazon.com/es_es/prescriptive-guidance/latest/choosing-git-branch-approach/advantages-and-disadvantages-of-the-git-hub-flow-strategy.html
- [9] J. Rodríguez, "Sintaxis de HTML: Estructura y reglas," DesarrolloWeb.com, 2024. [Enlace]. Disponible en: <https://desarrolloweb.com/articulos/535.php>
- [10] Beck, K., & Andres, C. (2004). Extreme Programming Explained: Embrace Change. Addison-Wesley.