



Tecnólogo en Informática  
UTU - UTEC - UdelaR  
San José de Mayo  
6/12/2023

## Documento Final Taller de Sistemas de Información .NET 2021

### Integrantes del Grupo

5.259.920-9 - Leonardo sellanes

4.999.935-3 - Gimena Deleón

5.856.776-9 - Federico Matías Díaz Cabrera

# Índice

## Contenidos

Índice.....	2
Resumen.....	3
Palabras Clave.....	3
Introducción.....	4
Contexto y Desafíos.....	4
Objetivos del Trabajo.....	4
Estructura del Documento.....	4
Marco Conceptual.....	5
Conceptos Clave.....	5
Descripción del Problema.....	7
Solución Planteada.....	8
Visión General.....	8
Estrategias de Venta.....	8
Seguridad y Escalabilidad.....	8
Integración y Colaboración.....	8
Sub-secciones Relevantes.....	8
Arquitectura del Sistema.....	9
Implementación.....	10
Herramientas y Tecnologías Destacadas.....	10
Productos y Herramientas.....	11
Problemas Encontrados.....	11
Evaluación de la Solución.....	12
Puntos Fuertes:.....	12
Áreas de Oportunidad:.....	12
Desarrollo del Proyecto.....	12
Cronograma Original:.....	12
Desviación y Desafíos Encontrados:.....	12
Dedicación y Esfuerzo:.....	13
Conclusión:.....	13
Conclusiones y Trabajo a Futuro.....	13
Trabajo a Futuro:.....	13
Referencias.....	13

# Resumen

Este proyecto, enmarcado dentro de la asignatura .NET, tiene como objetivo principal el desarrollo de un sistema innovador de gestión de E-Commerce, bajo la marca "Acme". La implementación se llevará a cabo utilizando tecnologías clave como .NET y Vue.js.

La motivación subyacente a este proyecto se fundamenta en la creciente relevancia de los emprendimientos basados en sistemas de E-Commerce. En este contexto, se propone el diseño e implementación de una plataforma informática que capitalice la potencia de .NET y Vue.js. Esta solución moderna facilitará a los administradores de "Acme" la supervisión y organización efectiva de las operaciones del mercado.

La contribución significativa de este trabajo se materializa en la implementación de un sistema robusto destinado a optimizar la gestión, brindando a la empresa "Acme" herramientas para una administración eficaz. Los resultados esperados incluyen una plataforma tecnológica avanzada que permita a la empresa destacarse en el competitivo mundo de los commerce, asegurando un manejo eficiente y rentable de las operaciones.

## Palabras Clave

**Vue.js:** Un marco de JavaScript utilizado para construir interfaces de usuario interactivas.

**Vuex:** Un estado de gestión de la biblioteca para aplicaciones Vue.js.

**Vite:** Un entorno de desarrollo rápido para aplicaciones web Vue.js.

**Vuex Store:** Almacenamiento centralizado para gestionar el estado en una aplicación Vue.js.

**Router:** En Vue.js, se refiere al enrutador de la aplicación para la navegación entre páginas.

**Startup.cs:** Archivo de configuración principal en una aplicación ASP.NET Core.

**Role-Based Access Control (RBAC):** Sistema de control de acceso basado en roles. **Ecommerce:** Actividad comercial en línea que involucra la compra y venta de productos y servicios a través de internet.

**Entity Framework:** Framework de mapeo objeto-relacional (ORM) utilizado en el desarrollo de aplicaciones .NET para interactuar con bases de datos.

**Back End o Back Office:** La parte de la aplicación que se ocupa del procesamiento y gestión de datos, operaciones de servidor y lógica empresarial.

**Front End o Front Office:** La parte de la aplicación con la que interactúan los usuarios finales, que se ocupa de la presentación y la interfaz de usuario.

**Up Selling y Cross Selling:** Estrategias de venta que implican persuadir a los clientes para que compren productos adicionales (up selling) o productos relacionados (cross selling).

**Criptomoneda:** Moneda digital que utiliza criptografía para garantizar transacciones seguras y controlar la creación de nuevas unidades.

**API Mock:** Una interfaz de programación de aplicaciones simulada utilizada para propósitos de prueba o desarrollo.

**Checkout:** Proceso final de compra en línea donde se confirman los productos seleccionados, se elige el método de pago y se completa la transacción.

**Tailwind CSS:** Framework de CSS utilizable con Vue.js para el diseño y estilización de la interfaz de usuario.

**Paypal:** Un servicio que te permite pagar, enviar dinero y aceptar pagos sin tener que introducir tus datos financieros continuamente.

**.NET:** Plataforma de desarrollo de software de Microsoft que incluye un entorno de ejecución, bibliotecas y herramientas para construir diferentes tipos de aplicaciones.

## Introducción

La investigación que se presenta se enmarca en la edición 2023 de la asignatura Taller de Sistemas de Información .NET Surge como respuesta a la necesidad imperante de afrontar desafíos específicos en el ámbito de los sistemas de información, en consonancia con la constante búsqueda de soluciones innovadoras y eficientes para problemáticas actuales en la informática y gestión de la información.

## Contexto y Desafíos

En un entorno caracterizado por rápidos avances tecnológicos, nos enfrentamos a la necesidad apremiante de abordar cuestiones particulares en el diseño y desarrollo de sistemas de información. La complejidad de las demandas de los usuarios y la gestión eficaz de grandes cantidades de datos imponen desafíos significativos. La solución propuesta no solo se orienta a resolver estos desafíos de manera práctica, sino también a establecer los fundamentos para futuras innovaciones en el campo.

## Objetivos del Trabajo

El propósito central de esta investigación es presentar una solución integral que aborde las particularidades del diseño de sistemas de información en el contexto actual. Buscamos proporcionar una solución que se distinga por su eficiencia, escalabilidad y capacidad de adaptación a las demandas cambiantes del entorno tecnológico.

## Estructura del Documento

Este documento está organizado para proporcionar una comprensión detallada de la propuesta y su implementación:

Sección 2: Ofrece un análisis detallado del contexto y los desafíos identificados.

Sección 3: Describe la solución propuesta, destacando aspectos clave y decisiones de diseño.

Sección 4: Analiza los resultados obtenidos durante la implementación y evaluación de la solución.

Sección 5: Aborda posibles áreas de mejora y futuras líneas de investigación.

Conclusiones: Sintetiza los hallazgos del trabajo y destaca las contribuciones significativas realizadas.

Este documento aspira a contribuir al avance del conocimiento en el campo de los sistemas de información y a proporcionar una guía práctica para abordar desafíos similares en el futuro.

# Marco Conceptual

## Conceptos Clave

En esta sección, se presentan resumidamente los conceptos clave que son fundamentales para la comprensión del proyecto "Acme Marketplace: Sistema de Gestión de E-Commerce". Estos conceptos proporcionarán un marco de referencia esencial para abordar la solución propuesta y entender su contexto tecnológico.

### 1 - Arquitectura Multi-Tenant

La arquitectura multi-tenant, en el contexto de sistemas de información, se refiere a un diseño que permite a una única instancia del software atender a múltiples clientes, conocidos como "tenants". Cada tenant comparte la misma aplicación subyacente, pero los datos y configuraciones son aislados para garantizar la privacidad y seguridad de cada cliente. En el proyecto, la implementación de una arquitectura multi-tenant es crucial para respaldar la gestión eficiente de diversos E-Commerce bajo la marca "Acme".

### 2 - Vue.js y .NET: Tecnologías Clave

Vue.js, un marco de JavaScript progresivo, se utiliza para construir interfaces de usuario interactivas y eficientes en el Front End. Por otro lado, .NET, un marco de desarrollo de software de Microsoft, ofrece un entorno robusto para la creación de aplicaciones escalables y seguras en el Back End. La combinación de estas tecnologías proporciona un marco sólido para la implementación exitosa del sistema de gestión de E-Commerce.

### 3 - Servicios Web (Web Services)

Los servicios web son componentes de software que permiten la comunicación entre sistemas a través de la red. En el contexto del proyecto, se utilizan servicios web para facilitar la integración fluida entre diversos módulos del sistema y garantizar la coherencia en el intercambio de datos. Esta

interoperabilidad es esencial para lograr una gestión eficiente de los E-Commerce y mejorar la experiencia del usuario.

#### **4 - Seguridad en Sistemas de Información en la Nube**

Dado que el proyecto se desarrolla en un entorno basado en la nube, la seguridad de la información es una consideración crítica. La implementación de prácticas de seguridad en la nube, como la autenticación y autorización adecuadas, garantiza la protección de los datos sensibles y la integridad del sistema.

#### **5 - Escalabilidad y Adaptabilidad**

La escalabilidad y adaptabilidad son conceptos centrales para asegurar que el sistema pueda manejar eficientemente el crecimiento de usuarios y la evolución de las demandas tecnológicas. El diseño del proyecto se enfoca en garantizar que la aplicación sea escalable y adaptable a fin de mantener un rendimiento óptimo a medida que evolucionan los E-Commerce y las necesidades del negocio.

#### **6 - Ecommerce**

Ecommerce, o comercio electrónico, se refiere a la compra y venta de bienes y servicios a través de internet. En el contexto del proyecto, el sistema se desarrolla con funcionalidades específicas de ecommerce para facilitar la operaciones en línea.

#### **7 - Entity Framework**

Entity Framework es un framework de mapeo objeto-relacional (ORM) utilizado en el desarrollo de aplicaciones .NET para interactuar con bases de datos. En el proyecto, Entity Framework contribuye a la gestión eficiente de datos y su interacción con la aplicación.

#### **8 - Back End o Back Office**

El Back End o Back Office representa la parte de la aplicación que se ocupa del procesamiento y gestión de datos, operaciones de servidor y lógica empresarial. En el proyecto, esta componente es esencial para asegurar el funcionamiento fluido y seguro de la plataforma.

#### **9 - Front End o Front Office**

El Front End o Front Office es la parte de la aplicación con la que interactúan los usuarios finales. Se refiere a la presentación y la interfaz de usuario. En el proyecto, el Front End se desarrolla utilizando Vue.js para ofrecer una experiencia de usuario interactiva y atractiva.

#### **10 - Up Selling y Cross Selling**

Up Selling y Cross Selling son estrategias de venta que implican persuadir a los clientes para que compren productos adicionales (Up Selling) o productos relacionados (Cross Selling). Estas estrategias están integradas en el proyecto para mejorar la experiencia de compra y aumentar las ventas.

#### **11 - Criptomoneda**

Una criptomoneda es una moneda digital que utiliza criptografía para garantizar transacciones seguras y controlar la creación de nuevas unidades. En el proyecto, se considera la posibilidad de incluir métodos de pago con criptomonedas para ofrecer opciones diversificadas a los clientes.

#### **12 - API Mock**

Una API Mock es una interfaz de programación de aplicaciones simulada utilizada para propósitos de prueba o desarrollo. En el proyecto, se emplea una API Mock para simular respuestas de servicios externos, como procesadores de pago, durante el desarrollo y las pruebas.

#### **13 - Checkout**

El proceso de Checkout se refiere a la fase final de la compra en línea donde se confirman los productos seleccionados, se elige el método de pago y se completa la transacción. En el proyecto, el Checkout es una parte crítica para garantizar una experiencia de usuario fluida y segura.

#### **14 - Vue y Vite**

Vue y Vite son tecnologías clave en el desarrollo del Front End. Vue.js se utiliza para construir interfaces de usuario interactivas, mientras que Vite proporciona un entorno de desarrollo rápido para aplicaciones web Vue.js, mejorando la eficiencia del desarrollo.

#### **15 - Tailwind CSS**

Tailwind CSS es un framework de CSS utilizable con Vue.js para el diseño y estilización de la interfaz de usuario. En el proyecto, Tailwind CSS contribuye a la creación de interfaces atractivas y responsivas.

#### **16 - .NET**

.NET es una plataforma de desarrollo de software de Microsoft que incluye un entorno de ejecución, bibliotecas y herramientas para construir diferentes tipos de aplicaciones. En el proyecto, .NET proporciona el marco de desarrollo para la implementación del Back End de la aplicación. Estos conceptos clave proporcionan una base sólida para la comprensión integral del proyecto "Acme Marketplace: Sistema de Gestión de E-Commerce" y su contexto tecnológico.

## Descripción del Problema

En esta sección, se aborda la problemática central que motiva el desarrollo del proyecto "Acme Marketplace: Sistema de Gestión de E-Commerce". El problema identificado consiste en la necesidad de diseñar y construir una solución integral que aborde los desafíos en la gestión de E-Commerce, específicamente en el ámbito del comercio electrónico.

Las principales funcionalidades requeridas para la solución incluyen:

**Registro y Configuración de Tiendas:** Capacidad para que las empresas registren y configuren sus tiendas de e-commerce, estableciendo el look & feel de su presencia online.

**Gestión de Productos:** Funcionalidades que permitan a las empresas registrar productos con detalles como título, descripción, foto, calificaciones, precio, tipo de IVA, y posiblemente documentos PDF asociados.

**Asociación de Productos:** La posibilidad de asociar productos relacionados (cross-selling) o productos complementarios (up-selling), ya sea a nivel de producto individual o categorías de productos.

**Proceso de Compra:** Implementación de un proceso de compra fluido para los clientes, que incluya búsqueda y selección de productos, un carrito de compras, y un proceso de checkout que permita elegir dirección de envío, método de pago, y confirmación de la compra.

**Diversidad de Métodos de Pago:** Soporte para diferentes métodos de pago, como tarjetas de crédito/débito y otros métodos integrados como MercadoPago o criptomonedas.

**Gestión de Envíos:** Integración con proveedores de envío, permitiendo a los clientes seleccionar direcciones de entrega o puntos de entrega en un mapa, con la posibilidad de gestionar costos adicionales y códigos de seguimiento mediante una API Mock.

**Notificaciones Automáticas:** Implementación de notificaciones automáticas por correo electrónico para recordar a los clientes sobre productos en sus carritos y enviar detalles de la orden de compra una vez confirmada.

**Opiniones y Valoraciones:** Funcionalidad que permita a los clientes dejar opiniones y valoraciones sobre los productos adquiridos.

La implementación de la solución se llevará a cabo utilizando la plataforma .NET 10 y Vue.

## Solución Planteada

En esta sección, proporcionamos una visión general de la solución propuesta para abordar la problemática central del proyecto "Acme Marketplace: Sistema de Gestión de E-Commerce ". A lo largo de la sección 3, hemos explorado detalladamente diversos aspectos, desde la arquitectura multi-tenant hasta la implementación de estrategias de venta como up-selling y cross-selling. Ahora, consolidamos estos elementos en una descripción integral de la solución alcanzada.

### Visión General

La solución diseñada se fundamenta en una arquitectura multi-tenant que permite gestionar eficientemente múltiples mercados de venta bajo la marca "Acme". La combinación de tecnologías clave, como Vue.js en el Front End y .NET en el Back End, asegura una aplicación web robusta y escalable. Se ha implementado un sistema de servicios web para facilitar la integración entre módulos, garantizando una experiencia de usuario coherente.

## Estrategias de Venta

Destacamos la incorporación exitosa de estrategias de venta como up-selling y cross-selling para potenciar la experiencia de compra de los usuarios. Estas estrategias se integran de manera fluida en el proceso de checkout, ofreciendo recomendaciones personalizadas y aumentando la satisfacción del cliente.

## Seguridad y Escalabilidad

Hemos puesto un énfasis significativo en la seguridad del sistema, implementando prácticas adecuadas en todas las capas, desde el Front End hasta el Back End. Además, la arquitectura escalable y adaptable garantiza que la aplicación pueda gestionar eficientemente el crecimiento de usuarios y las demandas cambiantes del entorno tecnológico.

## Integración y Colaboración

Un aspecto destacado de la solución es la exitosa integración con otros grupos, donde se ha logrado una colaboración efectiva para abordar desafíos específicos. La comunicación y coordinación entre equipos han sido clave para la implementación exitosa de ciertos aspectos novedosos y la resolución eficiente de requerimientos complejos.

### Figura 1 - Diagrama de la Solución

Incluimos la Figura 1, un diagrama que ofrece una visión global de la arquitectura y los componentes principales de la solución. Esta figura proporciona una representación visual de cómo se han integrado los diferentes elementos para construir un sistema de gestión de mercados de venta eficiente y efectivo.

## Sub-secciones Relevantes

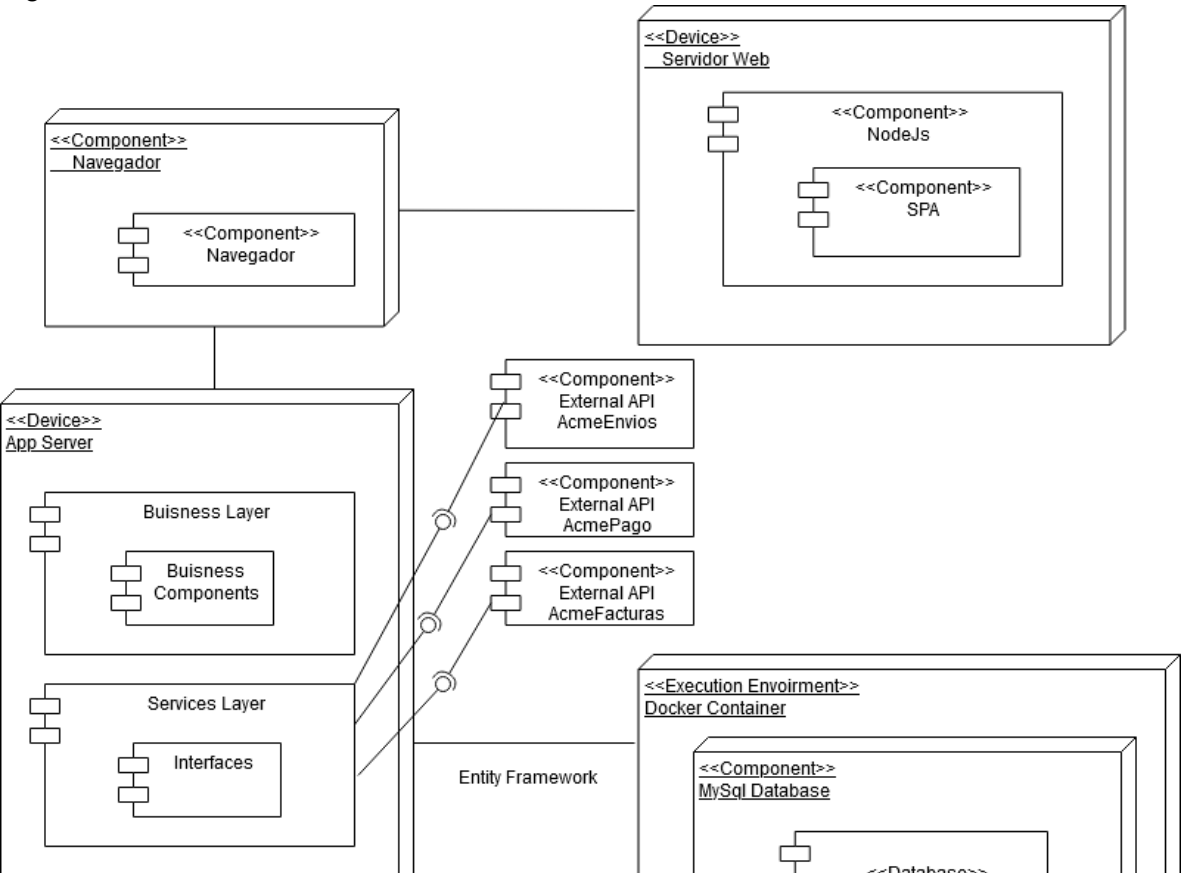
**Integración con Servicios Externos:** Detallamos cómo se logró la integración con servicios externos, como procesadores de pago y proveedores de envío, para asegurar un flujo de trabajo sin interrupciones.

**Optimización de Procesos de Checkout:** Describimos cómo se optimizaron los procesos de checkout para mejorar la eficiencia y simplicidad, contribuyendo a una experiencia de usuario más fluida.

Esta sección ofrece una visión clara y concisa de la solución implementada, resaltando los logros más significativos y proporcionando un marco comprensible para la continuación de la lectura.

## Arquitectura del Sistema

Figura 1.



# Implementación

## Herramientas y Tecnologías Destacadas

### **Vue.js - Framework Front End:**

**Descripción:** Vue.js ha sido la piedra angular en el desarrollo del Front End de la aplicación. Su enfoque progresivo y su capacidad para construir interfaces de usuario interactivas han mejorado significativamente la experiencia del usuario.

### **.NET - Framework Back End:**

**Descripción:** La elección de .NET para el desarrollo del Back End ha proporcionado un entorno sólido y seguro. Su capacidad para construir aplicaciones escalables en el lado del servidor ha sido fundamental para el éxito del proyecto.

### **Entity Framework - Mapeo Objeto-Relacional:**

**Descripción:** Entity Framework se ha utilizado para simplificar el acceso y manipulación de datos en la base de datos. Su capacidad de realizar operaciones CRUD de manera eficiente ha optimizado el manejo de información.

### **Vite - Entorno de Desarrollo Rápido:**

**Descripción:** Vite ha sido empleado como un entorno de desarrollo rápido para aplicaciones web Vue.js. Su capacidad para proporcionar tiempos de compilación rápidos y una experiencia de desarrollo fluida ha mejorado la eficiencia del equipo.

### **Tailwind CSS - Framework de Estilos Utilizado:**

**Descripción:** Tailwind CSS ha sido la elección para la gestión de estilos en la interfaz de usuario. Su enfoque utilitario y su configuración fácil de personalizar han facilitado la creación de una interfaz visual atractiva y coherente.

### **Visual Studio Code - Entorno de Desarrollo Integrado (Frontend):**

**Descripción:** Visual Studio Code ha sido la principal herramienta de desarrollo para el frontend, ofreciendo una interfaz ligera y funcional para la codificación y la gestión de proyectos.

### **API Mock - Simulador de Respuestas Externas:**

**Descripción:** Se ha implementado un API Mock independiente para simular respuestas externas, como las provenientes de procesadores de pago y proveedores de envío. Esto ha permitido probar y validar la integración de forma controlada.

### **Google Maps API - Integración de Mapas:**

**Descripción:** La integración de la API de Google Maps ha mejorado la experiencia del usuario al permitir la selección de puntos de entrega a través de un mapa interactivo.

### **Swagger - Documentación de API:**

**Descripción:** Swagger se ha utilizado para documentar y visualizar la API, facilitando la comprensión y prueba de los puntos de conexión expuestos.

### **Visual Studio - Entorno de Desarrollo Integrado (Backend):**

**Descripción:** Visual Studio ha sido la principal herramienta de desarrollo para el backend, proporcionando un entorno robusto para la codificación, la depuración y la gestión de proyectos.

### **Ant-Design-Vue - Kit ui:**

**Descripción:** Un kit de componentes bien documentado y fácil de usar, con una estética minimalista y actual



## Productos y Herramientas

En esta subsección, se presenta una evaluación de los productos y herramientas fundamentales utilizados en la implementación del proyecto "Acme Marketplace: Sistema de Gestión de E-Commerce

**Tabla 1. Evaluación de Productos y Herramientas**

Producto	Puntos Fuertes	Puntos Débiles	Evaluación General
<b>Vue.js - Framework Front End</b>	- Enfoque progresivo para construir interfaces interactivas.		9/10
<b>.NET - Framework Back End</b>	- Entorno sólido y seguro para el desarrollo del Back End.		10/10
<b>Entity Framework</b>	- Simplificación del acceso y manipulación de datos.		10/10
<b>Vite - Entorno de Desarrollo Rápido</b>	- Compilación rápida y experiencia de desarrollo fluida.		10/10
<b>Tailwind CSS - Framework de Estilos</b>	- Enfoque utilitario y fácil personalización de estilos.	-Diferente enfoque al css clasico	7/10
<b>Visual Studio Code - IDE (Frontend)</b>	- Interfaz ligera y funcional para el desarrollo Front End.		10/10
<b>API Mock - Simulador de Respuestas</b>	- Control en la prueba de integración con respuestas simuladas.	- No son Apis Reales	10/10
<b>Google Maps API - Integración de Mapas</b>	- Mejora la experiencia del usuario con mapas interactivos.	- Complejo al principio	7/10
<b>Swagger - Documentación de API</b>	- Facilita la documentación y visualización de la API.		9/10
<b>Visual Studio - IDE (Backend)</b>	- Entorno robusto para codificación, depuración y gestión.		10/10
<b>Ant - design-vue</b>	- Kit ui de muy fácil uso -Bien documentado	Algunos componentes no responden bien a algunas propiedades nativas	8/10

## Problemas Encontrados

**Gimena:** La magnitud de nuestra visión y la complejidad de las tareas nos llevaron por un camino lleno de desafíos temporales. Aunque empezamos con una planificación sólida y buenas intenciones, la realidad de la ejecución nos confrontó directamente. Tuvimos que ajustar nuestras estrategias y repensar nuestras expectativas en tiempo real.

**Leonardo:** Las fases iniciales del proyecto estuvieron marcadas por un desafío considerable en el ámbito de la comunicación interpersonal.

**Federico:** Durante el desarrollo de la interfaz de usuario utilizando Vue.js, me enfrenté a desafíos significativos debido a mi falta de experiencia previa con este framework. La curva de aprendizaje fue problematica y, en momentos, duplicó la complejidad de las tareas.

## Evaluación de la Solución

En esta sección, llevaremos a cabo una evaluación crítica de la solución propuesta, abordando tanto sus aspectos destacados como sus áreas de oportunidad. A continuación, se presentan de manera concisa los puntos fuertes y los puntos que consideramos susceptibles de mejora:

### Puntos Fuertes:

**Facilidad de Uso:** La interfaz de usuario proporciona una experiencia intuitiva, facilitando la interacción para los usuarios finales.

**Diseño atractivo:** La estética de la aplicación es agradable, contribuyendo a una experiencia visual positiva para los usuarios.

**Eficiencia en el Tiempo:** El desarrollo y despliegue del software se realizaron en un periodo relativamente corto, demostrando una eficiencia en términos de tiempo.

**Producto de respuesta rápida:** Dadas las tecnologías que utilizamos, el programa tiene cortos tiempos de carga y un rápido acceso a los datos.

### Áreas de Oportunidad:

**Mejorables Aspectos de Usabilidad:** Aunque la interfaz es fácil de usar, identificamos áreas que podrían mejorarse para optimizar aún más la experiencia del usuario.

**Potencial para Mejoras Estéticas:** Aunque el diseño es atractivo, se considera que existen oportunidades para afinar aspectos estéticos con el fin de alcanzar un nivel superior.

**Oportunidades de Optimización Temporal:** A pesar del desarrollo eficiente, existen áreas donde se podrían explorar oportunidades para reducir aún más los tiempos de implementación y despliegue.

Esta evaluación crítica proporciona una visión equilibrada de los logros obtenidos hasta ahora y plantea perspectivas para futuras mejoras en la solución propuesta.

**Mejoras de seguridad:** Somos conscientes de que nuestro software no es 100%, y hay áreas de seguridad que se pueden mejorar

## Desarrollo del Proyecto

En esta sección, exploramos el desarrollo de nuestro proyecto a lo largo del tiempo, analizando la planificación original y cómo se tradujo en la realidad.

### Cronograma Original:

- **Semana 1 (27-09-2023 al 04-10-2023):** Nos sumergimos en la creación del modelo de dominio y en la definición de la arquitectura del proyecto.
- **Semanas 2 y 3 (05-10-2023 al 25-10-2023):** Dedicamos tiempo a la implementación de clases, la creación de getters y setters, y avanzamos en la lógica y comunicación del sistema. La entrega de la arquitectura y diseño fue un hito crucial.
- **Semanas 4 y 5 (25-10-2023 al 08-11-2023):** Centramos nuestros esfuerzos en la lógica y comunicación, así como en el desarrollo de APIs. La entrega del prototipo de arquitectura marcó un paso importante.
- **Semanas 6 y 7 (08-11-2023 al 22-11-2023):** Nos volcamos al desarrollo del frontend, dando forma a la interfaz de usuario para una experiencia atractiva.
- **Semanas 8 y 9 (22-11-2023 al 06-12-2023):** Nos sumergimos en las etapas finales, dedicando tiempo a pruebas, correcciones y la elaboración de la documentación final.

### Desviación y Desafíos Encontrados:

El camino no estuvo exento de desafíos. La desviación con respecto al cronograma original fue de aproximadamente 1.5 semanas. La complejidad del proyecto, especialmente en las primeras etapas, dificultó la estimación precisa del tiempo necesario. La adaptabilidad y resiliencia de nuestro equipo fueron cruciales para superar obstáculos inesperados.

## Dedicación y Esfuerzo:

Cada integrante del equipo aportó en promedio 15 horas semanales. Esta dedicación constante refleja nuestro compromiso con el proyecto, incluso cuando nos enfrentamos a desafíos imprevistos.

## Conclusión:

A pesar de las desviaciones temporales, miramos hacia atrás con un sentido de logro. Cada desafío superado contribuyó al crecimiento individual y colectivo de nuestro equipo. Este proyecto no solo fue un ejercicio técnico, sino también una lección valiosa sobre la importancia de la adaptabilidad y la colaboración en entornos de desarrollo dinámicos.

## Conclusiones y Trabajo a Futuro

### Trabajo a Futuro:

1. **Fortalecimiento de la Seguridad:** La seguridad es nuestra principal preocupación para las próximas etapas. Queremos asegurarnos de que los datos estén a salvo y que nuestra aplicación sea lo más robusta posible.
2. **Optimización de Controladores:** La optimización de algunos aspectos técnicos puede mejorar la eficiencia y la respuesta de la aplicación. Queremos que la experiencia del usuario sea aún mejor, y eso es lo que buscaremos lograr.

En resumen: Este no es el final, sino el comienzo de algo más grande.

## Referencias

1. Laboratorio de Integración de Sistemas. Taller de Sistemas de Información 1 – Trabajo Obligatorio. Año 2014.
2. Microsoft .NET Framework  
<http://www.microsoft.com/net/> - [Consulta: Mayo 2010]
3. ACM Proceedings Templates  
<http://www.acm.org/chapters/policy/toolkit/template.html>  
[Consulta: Mayo 2010]