

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

**DESARROLLO DE UNA PLATAFORMA WEB PARA LA GESTIÓN
DE RECURSOS A CARGO DEL CENTRO DE EDUCACIÓN FÍSICA
DE LA EPN**

RESERVACIÓN DE ESPACIOS DEPORTIVOS PARA EVENTOS

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
SOFTWARE**

Dennis Augusto Chamba Posso

dennis.chamba@epn.edu.ec

DIRECTOR: PhD. Pamela Catherine Flores Naranjo

pamela.flores@epn.edu.ec

DMQ, Julio 2024

CERTIFICACIONES

Yo, Dennis Augusto Chamba Posso declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

Dennis Chamba

Certifico que el presente trabajo de integración curricular fue desarrollado por Dennis Augusto Chamba Posso, bajo mi supervisión.

PhD. Pamela Flores
DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

Dennis Chamba

PhD. Pamela Flores

Kevin López

Wellinton Peralta

DEDICATORIA

Dedico este trabajo a mis padres, cuya constante e incondicional guía y cariño han sido el pilar de mi formación. Su disciplina y amor me han forjado en la persona que hoy soy, alguien de quien estoy profundamente orgulloso.

A mi hermano, por su sabio consejo, orientación e incondicional apoyo durante esta etapa de mi educación. Sus palabras me brindaron fortaleza en los momentos más desafiantes de mi carrera.

Y a mi mejor amigo Kevin Anilema, cuya lealtad y apoyo han sido invaluable. Saber que siempre podía contar con su amistad ha sido un aliciente fundamental en la culminación de este trabajo.

AGRADECIMIENTO

Deseo expresar mi más profundo agradecimiento a mis docentes, quienes han invertido su tiempo y sabiduría en mi formación, guiándome para convertirme en un profesional digno de la ingeniería.

A mi familia, por su inquebrantable confianza y apoyo a lo largo de mi vida. Su presencia constante ha sido un faro de esperanza y motivación en cada paso de este camino.

A mis amigos Kevin y Wellington, cuya colaboración y apoyo hicieron posible la culminación de este proyecto. Lo que en un principio parecía una tarea interminable, se convirtió en un logro compartido gracias a su inestimable ayuda y compañerismo.

ÍNDICE DE CONTENIDO

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	VII
ABSTRACT.....	VIII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO	1
1.1 Objetivo general	2
1.2 Objetivos específicos	2
1.3 Alcance	3
Registro de eventos.....	3
Reservas recurrentes.....	3
Difusión de los eventos deportivos	3
2 ESTADO DEL ARTE	4
2.1 Contexto y antecedentes.....	4
Feature driven development (FDD).....	4
Scrum	5
Adaptive software development (ASD)	5
Modelo en cascada.....	5
Rational unified process (RUP)	5
2.2 Justificación metodológica	6
Criterios para la selección de una metodología.....	7
Selección de metodología	8
2.3 Comparativa de metodologías ágiles.....	9
2.4 Elección de la metodología.....	10
3 MARCO TEORICO.....	11
3.1 Microservicios	11
Características de los microservicios.....	11
Ventajas.....	11
Desventajas	12
3.2 Adaptative software development (ASD)	12
Principales características:	12

Ventajas:.....	12
Especulación.....	13
Colaboración	14
Aprendizaje	14
4 IMPLEMENTACIÓN	17
4.1 Fase de especulación.....	17
Requerimientos	17
Arquitectura	20
Selección de herramientas	23
Historias de usuario.....	24
Planificación de ciclos.....	26
4.2 Fase de colaboración.....	26
Actividades multidisciplinarias	27
Planificación de actividades.....	28
Primera iteración.....	29
Segunda iteración.....	31
4.3 Fase de aprendizaje	33
Entrega de resultados.....	33
Retroalimentación de los stakeholders	37
5 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES	38
5.1 Conclusiones.....	38
5.2 Recomendaciones	39
6 REFERENCIAS BIBLIOGRÁFICAS	40
7 ANEXOS	42

RESUMEN

La Escuela Politécnica Nacional (EPN) hasta la fecha no cuenta con un sistema que apoye al área de educación física la misma que esta encarga de realizar reservas de espacios deportivos y la gestión de implementos deportivos. En este sentido se propone realizar un proyecto de integración curricular (TIC) compuesto de tres componentes: a) Reservación de espacios deportivos b) Gestión de recursos deportivos, c) Reservación de espacios deportivos para eventos.

En este documento se presenta el desarrollo del componente c) el mismo que estuvo conducido por la metodología ASD la que se centra en 3 fases: a) Especulación, b) Colaboración, c) Aprendizaje

La fase de especulación se centró en el proceso de levantamiento de requerimientos, la redacción de historias de usuario, la elaboración de la arquitectura, y la selección de herramientas de desarrollo. Durante esta fase, se identificarán las necesidades y expectativas de los usuarios para asegurar que el sistema cumpla con los objetivos planteados.

La fase de colaboración abarco la programación de la interfaz de usuario y la lógica de negocio, garantizando una integración fluida y eficiente entre todos los componentes del sistema. Además, permitió gestionar la colaboración y conexión entre los integrantes del TIC, garantizando la entrega continua de las funcionalidades planificadas.

Finalmente, la fase de aprendizaje se enfocó en la integración de los módulos desarrollados, asegurando que funcionen de manera cohesiva y cumplan con los requerimientos definidos, permitiendo evaluar su efectividad y el impacto en la gestión de los recursos deportivos de la EPN.

PALABRAS CLAVE: Escuela Politécnica Nacional (EPN), Proyecto de Integración Curricular (TIC), gestión de eventos, implementos, requerimientos, escalable, planificación

ABSTRACT

The National Polytechnic School (EPN) currently lacks a system to support its physical education department, which is responsible for booking sports facilities and managing sports equipment. In this sense, it is proposed to carry out a curricular integration project (ICT) composed of three components: a) Booking of sports facilities, b) Management of sports resources, c) Booking of sports facilities for events.

This document presents the development of component c), which was guided by the ASD methodology, which focuses on 3 phases: a) Planning, b) Collaboration, c) Learning.

The planning phase focused on the requirements gathering process, the writing of user stories, the development of the architecture, and the selection of development tools. During this phase, the needs and expectations of users will be identified to ensure that the system meets the objectives set.

The collaboration phase encompassed the programming of the user interface and business logic, ensuring a smooth and efficient integration between all components of the system. Additionally, it allowed for the management of collaboration and connection between the members of the ICT, guaranteeing the continuous delivery of the planned functionalities.

Finally, the learning phase focused on the integration of the developed modules, ensuring that they work cohesively and meet the defined requirements, allowing for the evaluation of their effectiveness and the impact on the management of the EPN's sports resources.

KEYWORDS: National Polytechnic School (EPN), curricular integration Project (TIC), Event management, Equipment, Requirements, Scalable, Plann

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El componente de Reservas de Espacios Deportivos para Eventos se encarga de gestionar eficazmente las reservas de espacios e implementos para eventos deportivos, campeonatos y ferias que requieren planificación con anticipación, así como para reservaciones recurrentes de horarios de clase o clubes. Permite el registro detallado de eventos, facilita la reserva de espacios para actividades regulares y genera una página principal que sirve como la vitrina central del sistema, mostrando de manera visual y accesible los eventos programados.

Hoy en día, la tecnología y el desarrollo web ha avanzado a pasos agigantados permitiendo automatizar procesos que años atrás eran más manuales y tediosos. A partir de una necesidad latente, se elaboró un componente que permita gestionar de eventos deportivos dentro de la Escuela Politécnica Nacional (EPN) pensado para la comunidad politécnica, usando tecnologías a la vanguardia.

Además, el componente se enfoca en el desarrollo visual de una cartelera digital de control de eventos, que permite una fácil visualización y presentación de información relevante del área de deportes. Este panel de control integrará diversas métricas, proporcionando a los usuarios una herramienta interactiva y dinámica para monitorear y gestionar eventos deportivos

En la planificación del componente, se inicia con un levantamiento detallado de los requerimientos y necesidades, recolectando información de los usuarios finales, administradores y otros interesados. Esta fase asegura que el sistema cumpla con las expectativas y necesidades específicas de la comunidad politécnica.

La elaboración del documento de historias de usuario es un punto clave, debido a que permite detalla el valor que el sistema genera para el usuario final. Las historias de usuario guían el desarrollo y garantizan que todas las funcionalidades necesarias estén contempladas en el diseño del sistema.

En la planificación de la arquitectura, se seleccionan modelos arquitectónicos adecuados y se diseña una estructura modular que permita la escalabilidad y el mantenimiento. La arquitectura asegura que el sistema pueda expandirse y mejorar sin comprometer su estabilidad funcional.

El desarrollo e implementación abarcara tanto el frontend, backend y las bases de datos. El frontend se enfoca en crear una interfaz de usuario intuitiva y accesible teniendo en cuenta estilo y referencias de páginas oficiales de la institución, mientras que el backend

maneja la lógica de negocio, gestionando solicitudes y procesando datos de manera eficiente. Las bases de datos almacenan de forma segura toda la información relacionada con los implementos deportivos, las reservas de canchas y los reportes generados.

La integración de los módulos garantiza que todos los componentes del sistema funcionen de manera cohesiva, permitiendo una comunicación fluida entre los componentes del sistema. Las pruebas de integración son cruciales para identificar y corregir cualquier problema que surja durante la interacción de los diferentes módulos.

La realización de pruebas con los usuarios finales tanto con el personal administrativo y la comunidad politécnica, son vitales en esta etapa para realizar los ajustes necesarios y asegurar su satisfacción.

Este componente no está destinado a funcionar de manera aislada, sino que se integra en un sistema más grande de gestión de recursos y reservas de la Escuela Politécnica Nacional. Esta integración permite una visión general de la administración de recursos, facilitando la coordinación entre diferentes áreas y optimizando el uso de los recursos disponibles. La interoperabilidad entre los diferentes sistemas garantiza que la información fluya sin problemas, mejorando la eficiencia y efectividad de la gestión general de la institución.

El desarrollo de este componente para la gestión de reservas de eventos deportivos y sociales de la Escuela Politécnica Nacional. Al seguir un proceso de desarrollo bien definido y al integrar el componente en un sistema más amplio, se garantiza una solución robusta, eficiente y alineada con las necesidades de la comunidad politécnica. Este proyecto busca proporcionar una herramienta integral que optimice la administración y control de la reserva de eventos, contribuyendo así al mejor funcionamiento del área de deportes de la EPN.

1.1 Objetivo general

Desarrollar un módulo integral para la gestión eficiente de eventos deportivos, que facilite la planificación, reservación y gestión de espacios, implementos deportivos y servicios adicionales, optimizando la experiencia de los organizadores y participantes.

1.2 Objetivos específicos

- Diseñar y desarrollar una plataforma web interactiva que permita a los usuarios registrar eventos deportivos, campeonatos y ferias, así como realizar reservas de

espacios e implementos con anticipación, garantizando una gestión eficiente y organizada.

- Desarrollar un panel de control visual e interactivo para eventos deportivos que permita la fácil visualización, monitoreo y gestión de información relevante del área de deportes

1.3 Alcance

Registro de eventos

- Habilitar la inscripción de eventos deportivos, campeonatos y ferias que requieran reservación anticipada de espacios e implementos.
- Facilitar la gestión detallada de la información de cada evento, incluyendo fechas, horarios, ubicaciones e implementos deportivos.

Reservas recurrentes

- Permitir la programación de reservas periódicas para clases o clubes deportivos, optimizando la planificación de actividades deportivas regulares.

Difusión de los eventos deportivos

- Desarrollar una página principal dinámica que presente de forma clara y atractiva los eventos registrados, fomentando la participación de la comunidad politécnica.
- Incluir instrucciones detalladas sobre el proceso de registro de eventos y reservas de espacios, proporcionando una guía completa para los usuarios.

2 ESTADO DEL ARTE

Seleccionar la metodología de desarrollo de software es fundamental para el éxito de cualquier proyecto en el ámbito de las tecnologías de la información y comunicación (TIC). En años anteriores, las metodologías ágiles han ganado prominencia debido a su capacidad para entregar rápidamente software funcional y adaptarse continuamente a los cambios de requisitos. Estas metodologías promueven un enfoque flexible y colaborativo que permite mejoras constantes a lo largo del ciclo de vida del desarrollo del software [1].

Por otro lado, las metodologías tradicionales han sido fundamentales en el desarrollo de software durante décadas, caracterizándose por su enfoque estructurado y secuencial. Aunque menos ágiles en términos de adaptación a cambios, ofrecen una mayor previsibilidad y control en proyectos con requisitos estables y bien definidos. Según un artículo de Santander Open Academy, estas metodologías tradicionales se utilizan a menudo en proyectos donde la claridad y la estabilidad de los requisitos son cruciales [2].

Este estado del arte examina las principales metodologías ágiles, destacando sus ventajas y desventajas específicas en los sistemas de gestión de recursos, para ayudar en la selección de la metodología más adecuada al contexto del TIC. Además, se exploran las metodologías tradicionales para comprender su aplicación en contextos donde la estabilidad y la gestión son prioritarias.

2.1 Contexto y antecedentes

“**Las metodologías ágiles** se destacan por su enfoque en la entrega rápida de pequeñas piezas de software funcional, permitiendo una mayor satisfacción del cliente y una mejor capacidad de respuesta a los cambios [3]. Entre las metodologías ágiles más conocidas se encuentran Scrum, Feature-Driven Development (FDD) y Adaptive Software Development (ASD) [3][5].”

Feature driven development (FDD)

FDD es un enfoque ágil centrado en la entrega incremental de funcionalidades tangibles del software [3]. Este método se adapta especialmente bien a proyectos a largo plazo y equipos grandes, proporcionando un marco claro para la planificación y desarrollo de funciones mediante un ciclo de vida dividido en pasos clave:

- Desarrollo del modelo general: Definición del alcance y contexto del sistema mediante la colaboración directa con el arquitecto principal [3].

- Creación de una lista de funciones: Identificación y priorización de las funciones importantes para el cliente, expresadas como acciones concretas que el software debe realizar [3].
- Planificación por función: Asignación de funciones a equipos y programadores según capacidades y prioridades, con consideración de riesgos y dependencias [3]
- Diseño y compilación por función: Desarrollo y diseño iterativo de funciones, seguido de la implementación y prueba para su integración en la compilación principal del software [3].

Scrum

“Scrum es una metodología ágil para la gestión de proyectos complejos que se caracteriza por su agilidad, flexibilidad y capacidad de adaptación a cambios. Sus componentes principales incluyen roles definidos (Product Owner, Scrum Master, Equipo Scrum), eventos regulares (Sprint Planning, Daily Standup, Sprint Review, Sprint Retrospective) y artefactos clave (Product Backlog, Sprint Backlog, Incremento) [4].”

Adaptive software development (ASD)

ASD es otra metodología ágil basada en la adaptación continua del proceso de desarrollo según el aprendizaje adquirido durante el ciclo de vida [5]. Este enfoque promueve la colaboración efectiva y el desarrollo de componentes concurrentes, facilitando la respuesta a cambios y la mejora continua del producto [5].

Por otro lado, las metodologías tradicionales como el modelo en cascada, RUP (Rational Unified Process) y XP (Extreme Programming) han sido fundamentales en el desarrollo de software durante décadas, caracterizándose por su enfoque estructurado y secuencial.

Modelo en cascada

Descripción: El modelo en cascada sigue un enfoque lineal y secuencial en el desarrollo de software, en el que cada etapa debe finalizarse antes de comenzar la siguiente. Este modelo consta de fases como la recopilación de requisitos, el diseño, la implementación, la verificación y el mantenimiento.

Rational unified process (RUP)

Descripción: Es un proceso de desarrollo de software basado en la arquitectura, que define fases como Inicio, Elaboración, Construcción y Transición, con objetivos específicos y entregables.

2.2 Justificación metodológica

En la actualidad existen diferentes metodologías en el desarrollo de software, cada metodología tiene un contexto diferente de uso, en el apartado a continuación se evaluarán varias metodologías aplicadas en el enfoque de gestión de recursos u objetos.

Tabla 1. Investigación metodologías de desarrollo

Trabajo	Metodología y justificación	Equipo y división del trabajo	Tiempo
'Elaboración de un prototipo de una aplicación web y móvil para gestionar la reserva de canchas sintéticas de fútbol en Ecuador' [6].	SCRUM: Método ágil completo para diseñar la plataforma (página 32).	3 personas: Tutor (Scrum Master), Director de sistemas (Product Owner), Estudiante (Team Development).	107 días (3-4 meses)
'Diseño e implantación de un sistema web para la gestión y control de campeonatos deportivos de fútbol y básquet en FEDELIBAL' [7].	ICONIX: Uso de UML y verificación en todas las fases (página 7).	3 personas: 2 Estudiantes desarrollaron según los casos de uso definidos, Tutor supervisa.	No indica
'Desarrollo de un sistema web para gestionar los servicios del centro recreacional El Varadero' [8].	SCRUM: Levantamiento de requerimientos y enfoque ágil (página 4).	3 personas: Estudiante desarrolló la aplicación, Dueña del paradero aportó requisitos, Tutor supervisa.	4 meses
'Desarrollo de un sistema web y aplicación móvil para la reserva de canchas y oferta de cursos en la urbanización Los Retoños' [9].	SCRUM: Entrega de producto de alta calidad, mantenibilidad y comunicación continua (página 15).	3 personas: Estudiante desarrolló la aplicación, Dueña del paradero aportó requisitos, Tutor supervisa.	4 meses
'Plataforma web para gestionar reservas de horas de canchas de fútbol' [10].	Iterativa-incremental: Mitigación de riesgos y mejora continua del proceso de desarrollo (página 20).	2 personas: Ambos estudiantes desarrollaron la aplicación.	No indica
'Sistema de reservación de canchas sintéticas para la	XP: Desarrollo de la parte funcional mínima e	1 persona: Desarrolló toda la aplicación.	No indica

empresa Sports Césped' [11].	incremento de funciones conforme se avanza.		
'Sistema web para el proceso de reservas de canchas deportivas en Corporación Trescientos Sesenta Grados S.A.C. – Lima' [12].	RUP: Artefactos para mostrar objetos, procesos, desarrollo e implementación (página 5).	1 persona: Desarrolló toda la aplicación.	4 meses
'Gestión de reservas para canchas sintéticas en la ciudad de Pereira' [13].	Cascada: Integración de subsistemas como mayor riesgo identificado (página 19).	2 personas: Desarrollaron según los casos de uso.	4 meses
'Desarrollo de una aplicación móvil para reserva de canchas y organización de torneos en Santa Elena' [14].	SCRUM y Design Thinking: Soluciones creativas basadas en interacciones con el usuario y desarrollo incremental (página 7).	1 persona: Desarrolló toda la aplicación.	3 meses
'Sistema web para mejorar el proceso de reservas de canchas deportivas en El Encuentro S.A.C.' [15].	XP: Potencia relaciones interpersonales, desarrollo iterativo e incremental (página 17).	1 persona: Desarrolló toda la aplicación.	3 meses
'Diseño de una herramienta para la gestión de pistas deportivas' [16].	SCRUM y Kanban: Autonomía, responsabilidad y visualización del trabajo y flujo.	1 persona: Desarrolló toda la aplicación.	4 meses
'Desarrollo de aplicativo de reservas y gestión de la información para escenarios deportivos en Cartagena' [17].	RUP: No especifica la razón.	1 persona: Desarrolló toda la aplicación.	14 meses

Criterios para la selección de una metodología

Para seleccionar la metodología más adecuada, se consideraron los siguientes factores:

- Tiempo: Se evaluó la duración del proyecto y la necesidad de entregas rápidas y frecuentes de software funcional.

- **Recursos:** Se analizó la disponibilidad de personal calificado, el presupuesto asignado y las herramientas necesarias para implementar la metodología.
- **Tamaño del equipo:** Se tuvo en cuenta el número de desarrolladores disponibles y su experiencia en metodologías ágiles, así como la capacidad para formar equipos colaborativos y autoorganizados.
- **Enfoque del proyecto:** Se consideraron los objetivos del proyecto y su naturaleza cambiante, permitiendo adaptaciones rápidas a nuevos requisitos y prioridades del cliente.

Selección de metodología

Para la gestión de reservas de espacios deportivos, la selección de metodologías para el desarrollo de sistemas juega un importante papel para en el éxito del proyecto. Por medio de la revisión de diversos enfoques, se identificaron varias metodologías utilizadas por proyectos similares

- **Metodología Scrum:** Rodrigo Ayala en [6] destaca la elección de Scrum en la implementación de una aplicación web-móvil debido a su agilidad y capacidad para gestionar proyectos complejos. Scrum es considerada adecuada para el diseño de la plataforma, ya que proporciona una estructura clara con fases definidas, como la preliminar, la planificación, el desarrollo y la finalización.
- **Metodología iterativa e incremental:** El desarrollo de la plataforma web presentada en [7] utiliza una metodología iterativa e incremental, adaptándose a las necesidades del proyecto. Esta metodología se selecciona por su capacidad para mitigar riesgos y aprovechar el conocimiento adquirido en cada iteración para mejorar el proceso de desarrollo.
- **Metodología Scrum y Design Thinking:** En [8], se combinan SCRUM y Design Thinking satisfaciendo las necesidades del cliente y fomentar la creatividad en la generación de soluciones. Ambas metodologías se centran en el factor humano y se orientan hacia el cliente, lo que las hace adecuadas para proyectos donde la interacción con los usuarios es fundamental.
- **Metodología XP (Programación extrema):** La elección de XP en [9] se destaca por su capacidad para desarrollar la parte funcional mínima y agregar funciones adicionales de manera incremental. XP se centra en la programación en parejas, el

diseño simple y las pruebas continuas, lo que lo hace adecuado para proyectos que requieren entregas rápidas y frecuentes.

- **Adaptive software development (ASD):** Según Highsmith y Cockburn [10], ASD es una metodología ágil que promueve la adaptación continua del proceso de desarrollo según el aprendizaje adquirido durante el ciclo de vida del proyecto. ASD se destaca en su enfoque marcado por colaboración efectiva y el desarrollo de componentes concurrentes, facilitando una respuesta rápida a los cambios y la mejora continua del producto.

2.3 Comparativa de metodologías ágiles

Se seleccionaron tres metodologías ágiles que mejor se adaptan al desarrollo de un sistema de gestión de reservas. Luego, se compararon utilizando criterios específicos para determinar cuál de ellas es la más adecuada para el proyecto TIC.

Tabla 2. Comparativa metodologías ágiles

Metodología	Descripción	Tiempo y costo	Tamaño del equipo	Ventajas	Desventajas
Scrum	Equipos multifuncionales trabajan en iteraciones cortas (1-2 semanas). Al final de cada iteración, se debe haber completado trabajo de diseño, desarrollo y prueba.	Ideal para equipos y proyectos pequeños a medianos. Entrega rápida. Costos variables debido a la adaptabilidad a cambios y la priorización constante durante los sprints.	Equipos pequeños, cohesionados y autoorganizadas.	Adaptabilidad a cambios rápidos, entrega incremental y frecuente de valor, mejora la colaboración y comunicación, mayor visibilidad y transparencia del progreso.	Requiere un cambio cultural significativo, posible tendencia a la sobre programación (overcommitting), falta de enfoque en aspectos técnicos y de ingeniería.
FDD	Enfoque ágil centrado en la entrega incremental de características específicas y	Pensado para equipos y empresas grandes con proyectos de gran alcance.	Equipos grandes con varios roles necesarios.	Enfoque en características tangibles, modelado de dominio detallado,	Énfasis en el desarrollo técnico, menos énfasis en la comunicación

	tangibles, incluyendo funcionalidad, integración y pruebas.	Costos organizados en torno a equipos especializados responsables de conjuntos específicos de características.		organización por equipos especializados, flexibilidad para adaptarse a cambios.	constante, depende de que no exista gran incertidumbre.
ASD	Adaptable a cambios en los requisitos y fomenta la colaboración continua en las etapas del desarrollo de software. Se centra en el desarrollo modular, descomponiendo el sistema en componentes independientes y cohesionados.	ASD se adapta a presupuestos variables. La adaptabilidad a cambios puede afectar los costos, pero también permitira una gestión más eficiente de los recursos.	Flexible para equipos de diferentes tamaños, útil en entornos donde la comunicación eficaz y la capacidad de respuesta a cambios son esenciales.	Énfasis en la colaboración activa, desarrollo de componentes concurrentes, mejora la eficiencia y la capacidad de respuesta del sistema.	Requiere planificación detallada, la retroalimentación constante puede ser un desafío en entornos con limitada colaboración con los usuarios finales.

2.4 Elección de la metodología

Después de revisar los enfoques utilizados en proyectos similares, y considerando los criterios de tiempo, recursos, tamaño del equipo y enfoque del proyecto, se decidió adoptar metodologías ágiles. Dentro de las metodologías ágiles, se seleccionó el enfoque de Adaptive Software Development (ASD) por su capacidad de adaptación continua al proceso de desarrollo, colaboración efectiva y desarrollo de componentes concurrentes, lo que se alinea perfectamente con las necesidades del proyecto de gestión de reservas de canchas deportivas.

Estas características de ASD ofrecen la flexibilidad y adaptabilidad necesarias satisfaciendo las necesidades cambiantes en el cliente y el proyecto, garantizando así un desarrollo eficiente y exitoso.

3 MARCO TEORICO

En la actualidad existen muchas herramientas, metodologías, marcos de trabajo y arquitecturas de software, cada una con características específicas, para su aplicación es importante poder reconocer sus características principales y conocer sus conceptos para poder entender el enfoque y su aplicación.

3.1 Microservicios

“Los microservicios son una arquitectura de software que estructura una aplicación como un conjunto de servicios pequeños y autónomos, cada uno ejecutando un proceso único y comunicándose mediante mecanismos ligeros, usualmente una API HTTP. Cada microservicio es responsable de una funcionalidad específica y puede ser desarrollado, desplegado y escalado de manera independiente. Esta arquitectura promueve una mayor flexibilidad, escalabilidad y capacidad de mantenimiento en comparación con los enfoques monolíticos tradicionales [1].”

Características de los microservicios

- **Modularidad:** Cada microservicio es un módulo independiente que realiza una función específica.
- **Descentralización:** Promueve la descentralización de los datos y la lógica del negocio.
- **Escalabilidad Independiente:** Cada servicio puede ser escalado de manera independiente según las necesidades de carga.
- **Despliegue Independiente:** Permite desplegar y actualizar servicios sin afectar al resto del sistema.
- **Comunicación Ligera:** Los microservicios se comunican generalmente a través de HTTP/HTTPS usando APIs RESTful o mensajería asíncrona.

Ventajas

- **Flexibilidad en el desarrollo:** Los equipos pueden trabajar en diferentes servicios de manera simultánea.
- **Facilidad de mantenimiento y actualizaciones:** Cambios y actualizaciones pueden ser implementados sin afectar todo el sistema.
- **Resiliencia:** Fallos en un microservicio no afectan a toda la aplicación.

Desventajas

- **Complejidad operacional:** La gestión y orquestación de múltiples servicios puede ser compleja.
- **Latencia de comunicación:** Las comunicaciones entre servicios pueden añadir latencia.
- **Gestión de transacciones distribuidas:** Coordinación de transacciones a través de múltiples servicios puede ser complicada.

3.2 Adaptive software development (ASD)

La metodología ASD (Adaptive Software Development) es un enfoque ágil para la gestión de proyectos de software, destacando la colaboración, el aprendizaje y la adaptación a cambios rápidos. Propuesta por Jim Highsmith y Sam Bayer a inicios de la década de 1990, ASD enfatiza el desarrollo iterativo e incremental, con prototipado constante y flexibilidad para responder a cambios. Se basa en la idea de que el proceso de desarrollo debe ser adaptable y flexible durante el proyecto [18].

Highsmith creó esta metodología con el propósito de proporcionar una opción diferente para el desarrollo de software. Su enfoque se basa en la idea de que la optimización es la única respuesta viable ante problemas de creciente complejidad. Según Highsmith, los proyectos de software son sistemas complejos adaptativos, y la optimización solo sirve para sofocar la necesaria emergencia frente al cambio. Su metodología busca abordar la complejidad y la adaptabilidad de los proyectos de software de manera más flexible y dinámica que la optimización tradicional [19].

Principales características:

- Orientado a componentes de software.
- Ciclo de vida iterativo.
- Flexibilidad y capacidad de respuesta a cambios.
- Guiado por riesgos y funcionalidad.
- Enfatiza la colaboración y ciclos cortos de entrega para aprender de errores.

Ventajas:

- Adaptabilidad a cambios de requisitos.

- Entrega continua y rápida de componentes funcionales.
- Colaboración estrecha entre el cliente y el equipo de desarrollo.
- Reducción y eliminación de tareas innecesarias.
- Enfoque en la calidad técnica y el diseño eficaz.
- Optimización continua de los procesos y del equipo de desarrollo.

La metodología se organiza en tres etapas: especulación, colaboración y aprendizaje.

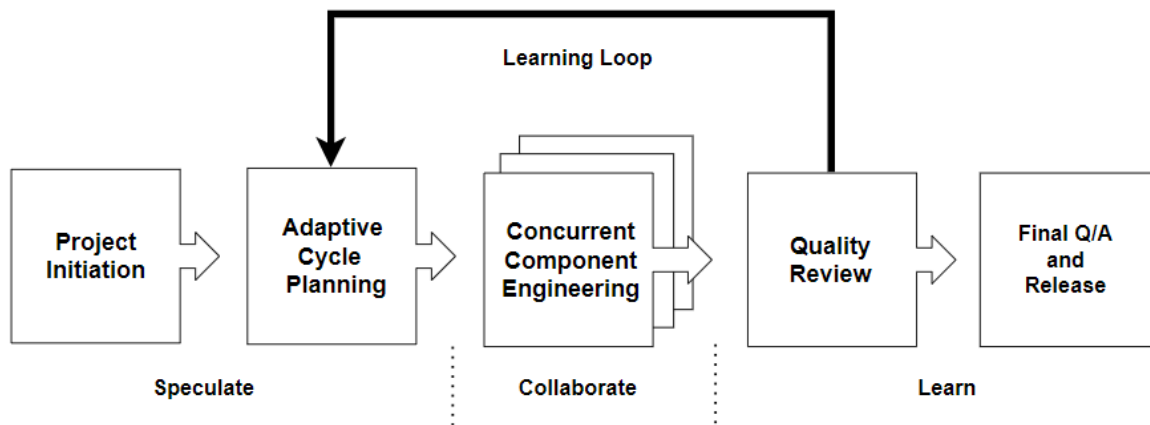


Figura 1. Flujo de trabajo ASD

Especulación

El proyecto inicia con una fase de especulación, en la que se realiza una planificación preliminar considerando las entregas futuras que se proporcionarán al usuario.

Independientemente de la exhaustividad del plan para el ciclo, siempre habrá necesidad de ajustes [18]. Así, en cada iteración, se descubrirán nuevas funcionalidades, se revisarán aspectos previos y se modificarán los requisitos.

En esta fase se definen los objetivos y metas principales del proyecto en general, además de determinar el número de ciclos o iteraciones tentativas necesarios para su desarrollo [19].

Según Rivero Jenny (Rivero, 2014) la etapa de especulación se subdivide en:

- **Inicio del proyecto:** Durante esta etapa se definen la función y el propósito del proyecto, junto con los requisitos de software y hardware. Se elabora un documento que proporciona una visión general del proyecto, incluyendo una hoja de datos, una descripción de la misión del producto y un esquema de los requisitos. Este

documento se presenta una vez que el equipo tiene una comprensión clara del sistema para iniciar el proyecto [20].

- **Planificación de los ciclos:** En esta fase se establece el número de iteraciones del proyecto, es decir, la cantidad de veces que se repetirá cada etapa (de acuerdo con el control de calidad), así como la duración de cada iteración. Además, se define el objetivo de cada iteración y se asigna la funcionalidad correspondiente a cada una de ellas [20].

Colaboración

Es en esta fase en la que el equipo liberara la funcionalidad planificada, se construye lo que se especuló en el paso anterior manteniendo un componente cíclico.

ASD no sugiere técnicas específicas ni establece tareas concretas para la fase de construcción, sino que señala que se preferirán las prácticas que fortalezcan la colaboración, alineándose con la filosofía de las metodologías ágiles en cuanto a la orientación hacia componentes.

Colaboración es un acto de creación compartida, se ve favorecida por la confianza y el respeto. Colaborar sin prejuicios y ayudar a los miembros del equipo que así lo requieran, puede llevar al éxito [18].

El punto focal se centra en las conexiones entre los individuos involucrados en el proyecto, como desarrolladores y usuarios. Es esencial que estas relaciones sean sólidas y se basen en el respeto mutuo para garantizar una comunicación y colaboración eficaces.

Según Rivero Jenny (Rivero, 2014) la etapa de la colaboración se subdivide en:

- **La ingeniería de componentes concurrente:** Consiste en integrar diversos métodos para reducir los tiempos de desarrollo de software. En esta etapa, se realiza la construcción del sistema, que abarca la codificación de la funcionalidad del proyecto y su diseño. Además, se lleva a cabo la gestión del producto, que implica la dirección, organización y coordinación general del producto [20].

Aprendizaje

En esta etapa, al final de cada ciclo se realiza una revisión de la calidad, en la cual Highsmith identifica cuatro tipos de aprendizaje:

- **Calidad del resultado desde el punto de vista del cliente:** En esta sección se propone un enfoque centrado en el cliente como una estrategia para evaluar la

calidad desde la perspectiva del cliente. Estos grupos permiten recopilar nuevos requerimientos o posibles cambios que el cliente pueda necesitar, lo cual es crucial para mantener un producto o servicio que satisfaga las necesidades y expectativas del cliente en constante evolución. Además, al involucrar a los clientes de esta manera, se fomenta una mayor interacción y participación en el proceso de mejora continua del producto o servicio.

- **Calidad del resultado desde el punto de vista técnico:** En esta sección implica analizar minuciosamente la calidad del producto, enfocándose en revisar el diseño, el código y las pruebas para aprender de los errores y desviaciones detectadas. Es esencial abordar estos problemas sin buscar culpables, ya que esto podría afectar negativamente las relaciones dentro del equipo de trabajo. Después de esta evaluación inicial, se procede a la fase práctica de la etapa, donde se profundiza en las exploraciones realizadas previamente. Esto nos permite realizar modificaciones tanto en el diseño del sistema como en los posibles cambios de requerimientos que el cliente pueda solicitar. Este enfoque facilita la resolución de problemas de manera colaborativa y orientada a mejorar continuamente la calidad del producto o servicio.
- **Operación del equipo de desarrollo y las prácticas empleadas:** En esta etapa de aprendizaje, se destaca la importancia de la interacción entre las partes involucradas, la dinámica del equipo y las técnicas acordadas para el proyecto. Al concluir cada ciclo de vida, se llevan a cabo reuniones breves para analizar y reflexionar sobre lo aprendido (lo que comúnmente se denomina una "autopsia"). Antes de iniciar una nueva iteración, se realizan discusiones sobre los procesos que contribuyen al desarrollo positivo del proyecto y se descartan aquellos que tienen influencia negativa. Estos últimos procesos incluyen aquellos que puedan haber afectado el trabajo del equipo y aquellos que no cumplen con los requerimientos establecidos. Este enfoque permite identificar y corregir los problemas de manera proactiva, fomentando así un ambiente de mejora continua y eficiencia en el desarrollo del proyecto.
- **Estatus del proyecto:** En esta etapa final, se lleva a cabo una revisión exhaustiva de todo lo realizado en el proyecto, tanto en términos técnicos como humanos, comparándolo con lo que se había planificado inicialmente. Es en este momento donde se identifican posibles diferencias que podrían influir en la dirección del proyecto. Después de explorar las tres fases anteriores, se inicia un ciclo de aprendizaje que incluye revisiones de calidad con la participación activa del cliente

como experto. Además, se complementa la presencia del cliente con sesiones de talleres, donde programadores y representantes del cliente se reúnen para discutir aspectos del producto desde una perspectiva no técnica, sino orientada a los aspectos de negocio. Esta combinación de evaluaciones técnicas y enfoque en las necesidades del cliente garantiza un producto final alineado con los objetivos y requisitos del proyecto.

4 IMPLEMENTACIÓN

La metodología ASD cuenta con 3 fases importantes en el desarrollo de software, en los apartados a continuación se evidencia la aplicación de la metodología a lo largo de un proyecto TIC.

4.1 Fase de especulación

En la fase a continuación se evidencia la planificación del proyecto, tomando en cuentas las futuras entregas y los puntos que agregan valor a la fase de especulación.

Requerimientos

La fase inicial del proyecto se enfoca en el levantamiento exhaustivo de requerimientos a través de técnicas estructuradas y participativas.

Primera reunión

El proceso conto con un primer acercamiento con el área de deportes, el propósito primordial de este encuentro fue obtener información crucial sobre los procesos en las reservas y prestamos de implementos deportivos, la estructura organizativa, las regulaciones vigentes y el funcionamiento integral del área de deportes en la EPN.

Para maximizar la efectividad y la interacción con los stakeholders, se empleó la técnica del Focus Group. La elección se basó en la necesidad de facilitar una discusión profunda y colaborativa entre los diversos miembros del área de deportes presentes en la reunión. Además, se elaboró una encuesta estructurada con preguntas abiertas que garanticen la captación de una gran cantidad de información de los procesos, que funcione como guía durante la sesión, asegurando que todos los puntos importantes a ser tratados fueran abordados de manera sistemática y que ninguna información crítica fuera pasada por alto.

Durante la fase de encuesta y Focus Group, se explora una serie de aspectos fundamentales para el desarrollo del sistema de gestión de recursos, brindando un primer acercamiento para conocer el proceso actual del área de deportes

- **Acerca del área**
 - ¿De qué funciones dentro de la universidad es responsable el área de deportes?
 - ¿Quiénes son los responsables del área de deportes?
 - ¿Cuáles son los roles dentro del área de deportes?
- **Implementos**

- ¿Qué datos de los implementos considera importantes de almacenar al momento de su registro (por ejemplo: marca, fecha de adquisición, valor, ¿etc)?
 - ¿Qué sucede si un implemento es regresado en mal estado o dañado?
 - ¿Qué medidas se toman cuando un estudiante no regresa un implemento deportivo que haya sido prestado?
 - ¿Llevar el control de los implementos deportivos ayuda en algo dentro del área de deportes?
 - ¿Existen requisitos para reservar implementos deportivos? (Como por ejemplo dejar la cedula o que otro tipo de identificación se puede usar para pedir prestados implementos)
- **Reservas**
 - ¿Quién es la persona encargada de realizar los préstamos?
 - ¿El registro de los préstamos de canchas e implementos son utilizados para algo dentro del área de deportes?
 - ¿Cuáles son los pasos clave que sigue para llevar a cabo un préstamo de implementos deportivos?
 - ¿Como organiza un evento? (happy path)

Este primer acercamiento permitirá obtener una comprensión profunda de los desafíos actuales enfrentados por el área de deportes y las necesidades específicas que el sistema de gestión de recursos debe abordar de manera efectiva. Estos hallazgos iniciales son cruciales para informar el diseño y desarrollo subsiguiente de la plataforma web, asegurando que la soluciones estén alineada con las expectativas y requerimientos reales los usuarios.

Segunda reunión

La segunda reunión tuvo el objetivo principal de refinar los requisitos a través de la metodología de Sketching. Esta técnica utiliza dibujos rápidos y esquemáticos para representar ideas y conceptos de manera visual y comprensible. Se empleó la herramienta Figma como plataforma para facilitar este proceso de diseño colaborativo, permitiendo la interacción entre el usuario y las representaciones.

Además, se implementó la técnica de Think Aloud, la cual promueve una interacción más efectiva con los usuarios al realizar actividades sobre el prototipo elaborado para el sketching, asegurando así una comprensión profunda de las necesidades y expectativas del área de deportes, permitiendo generar un mayor entendimiento sobre las necesidades reales a ser solventadas.

- **Actividades Propuestas**

Durante la reunión, se llevaron a cabo las siguientes actividades para recolectar de manera más eficiente los requerimientos:

- Reserva de Cancha e Implementos: Se realizó la reserva de la cancha 1 de fútbol para el horario de 13:00 a 14:00 del lunes 18, incluyendo la reserva del balón 1.
- Creación de Evento Recurrente: Se diseñó un evento de fútbol para el día martes a las 7:00 AM, programado de manera recurrente para facilitar la organización y gestión de actividades deportivas.

Para entender mejor cómo aplicar estas técnicas, es útil comprender el concepto de Sketching en el contexto del diseño de sistemas. Sketching utiliza dibujos rápidos y esquemáticos para representar visualmente ideas y conceptos, permitiendo representar el entendimiento funcional de los requerimientos y dando una primera aproximación de cómo será el comportamiento del sistema. Esto facilita la comunicación clara de ideas complejas, estimula la generación rápida de ideas creativas y permite ajustes ágiles durante el proceso de diseño.

En esta reunión, el uso de Sketching junto con Think Aloud permitió no solo comprender mejor los requisitos del sistema, adicional permitieron conocer nuevos requerimientos que en un principio no se había tomado en cuenta, permitiendo visualizar y proponer soluciones de manera colaborativa y efectiva con los usuarios finales del sistema.

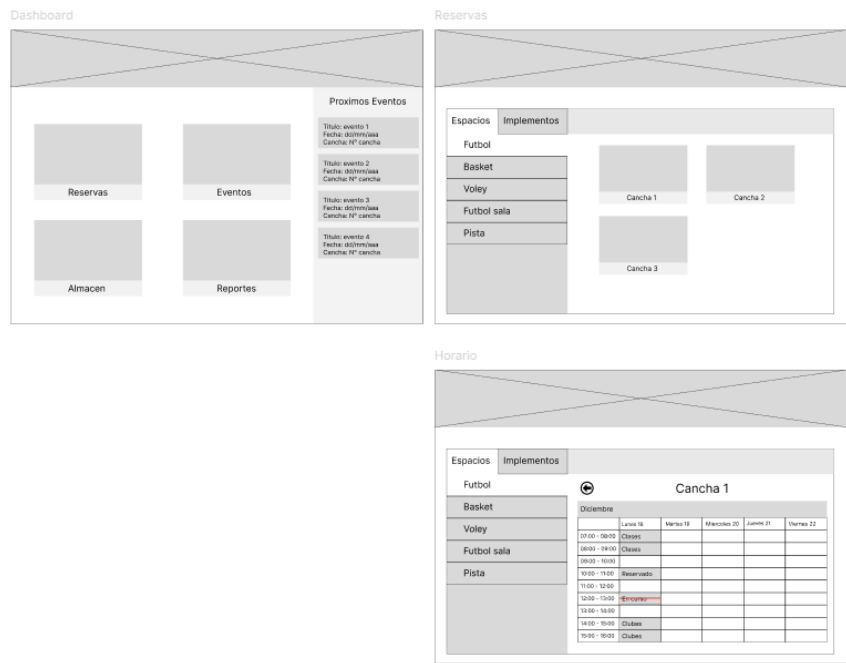


Figura 2.1. Sketching ilustraciones



Figura 2.2. Sketching ilustraciones

Arquitectura

Para el diseño de la arquitectura del sistema, se adoptó una arquitectura basada en microservicios, optimizando la flexibilidad y escalabilidad del sistema. Se desarrollaron tres

versiones de la arquitectura, cada una iterando sobre los principios de modularidad, independencia, desacoplamiento y escalabilidad característicos de los microservicios.

Primera versión

La primera versión de la arquitectura se centró en la segmentación de las funcionalidades principales de manera modular en forma de microservicios individuales. Cada microservicio se diseñó para manejar una tarea específica de un módulo, como la gestión de reservas de implementos, la administración de eventos deportivos, el seguimiento de inventario de canchas e implementos y la generación de reportes. Esta primera versión permitió una implementación rápida y una gestión eficiente de cada componente del sistema, permitiendo observar una visión general del comportamiento y las relaciones existentes entre los módulos a ser desarrollados.

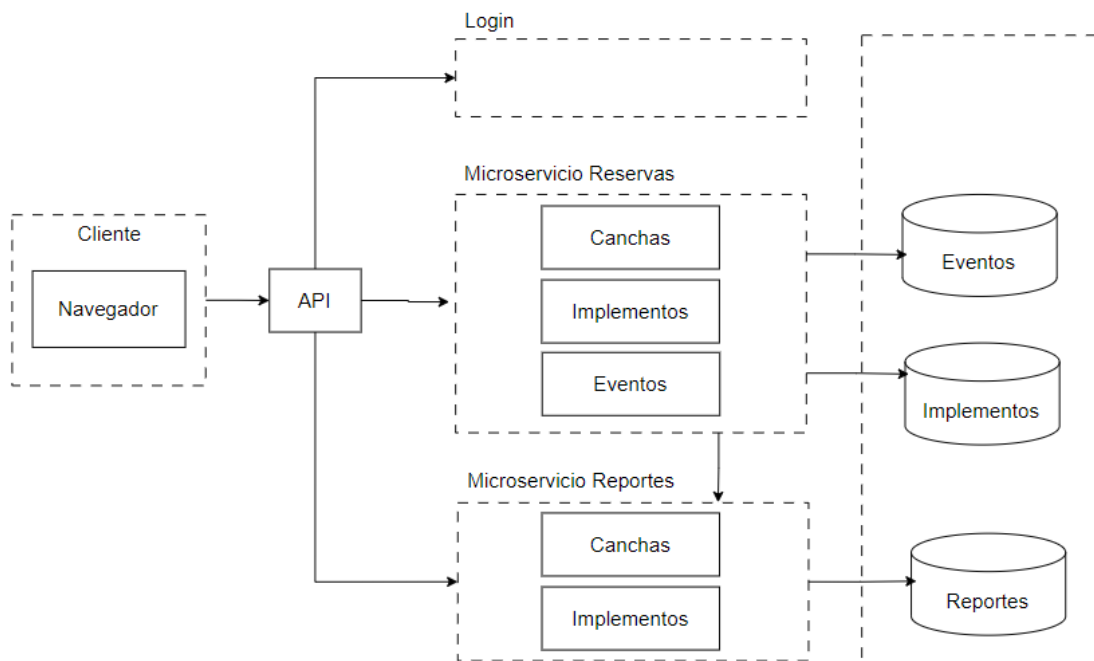


Figura 3. Diagrama arquitectura V1

Segunda versión

En la segunda iteración, se mejoró la interoperabilidad y la independencia de los microservicios, teniendo en cuenta el principio de responsabilidad única de cada microservicio, mediante la adopción de interfaces bien definidas y contratos de comunicación claros entre ellos, teniendo en cuenta una comunicación entre un microservicio y su base de datos. Se implementaron patrones de diseño como API Gateway para gestionar las solicitudes externas, exponiendo puntos de conexión para consumo y

conexión con otros microservicios facilitando el acceso unificado a los microservicios. Esta versión buscó mejorar la cohesión y la adaptabilidad del sistema, asegurando una mayor robustez frente a cambios y actualizaciones futuras.

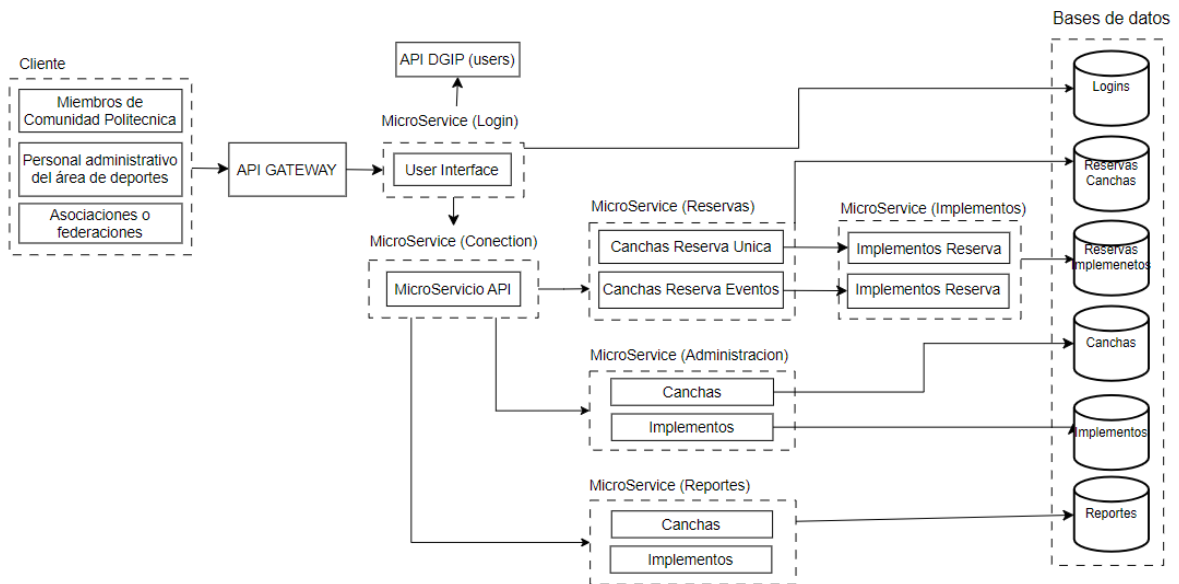


Figura 4. Diagrama arquitectura V2

Tercera versión

La última versión de la arquitectura se enfocó en la escalabilidad horizontal y la resiliencia del sistema. Se introdujeron técnicas como la implementación de contenedores Docker para facilitar el despliegue y la gestión de los microservicios en un entorno distribuido. Además, se incorporó un modelo de clases para organizar y relacionar la información dentro del sistema, mejorando la estructura y eficiencia del manejo de datos.

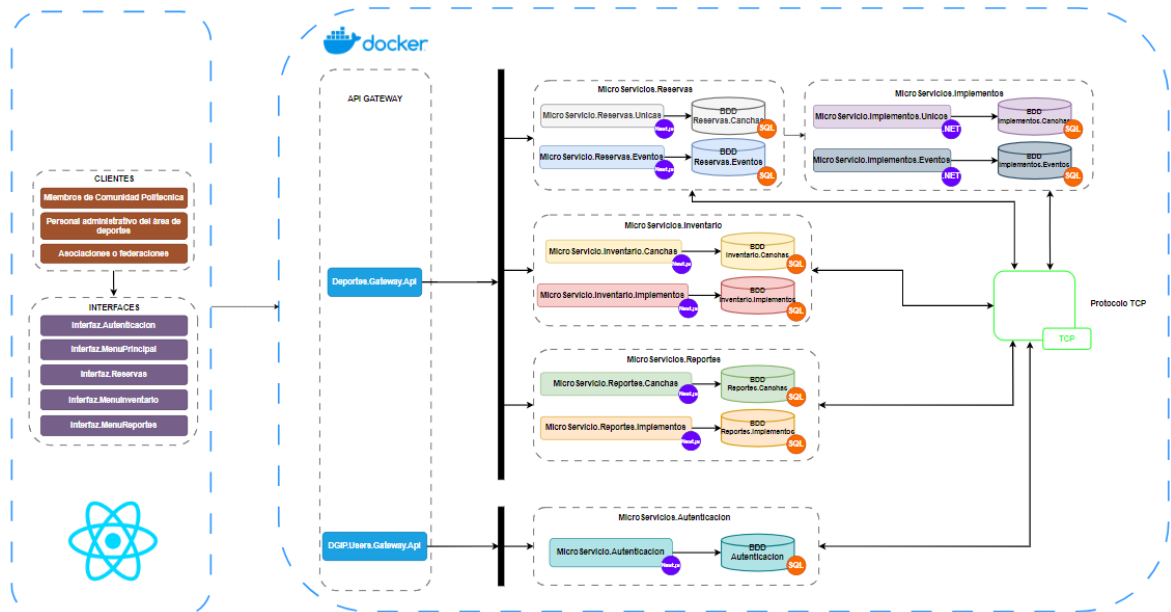


Figura 5. Diagrama arquitectura V3

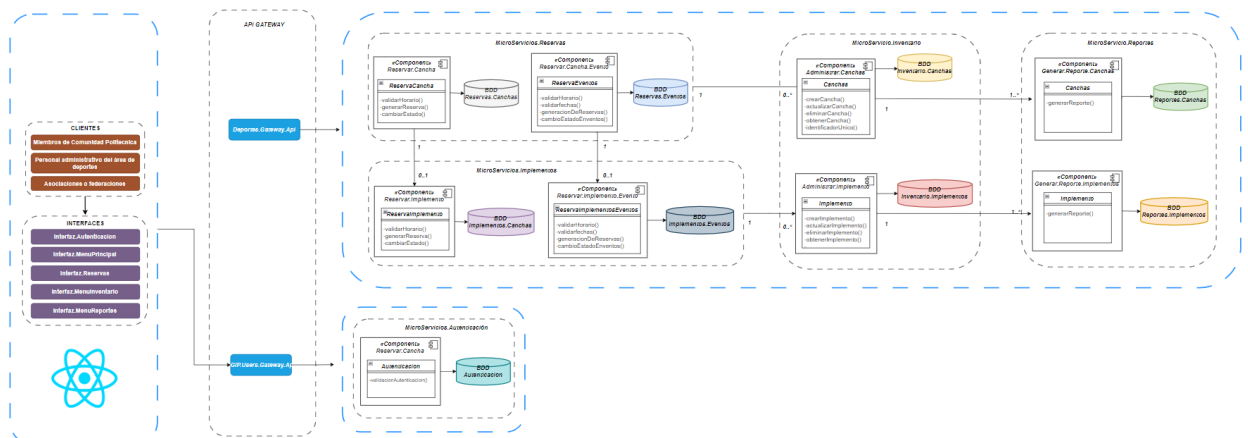


Figura 6. Diagrama de clases

Estas iteraciones permitieron no solo optimizar el rendimiento y la gestión del sistema de gestión de recursos deportivos, sino también prepararlo para crecer de manera sostenible y adaptarse a futuras necesidades y exigencias del usuario.

Selección de herramientas

GitHub y gestión de branches

Se optó por utilizar GitHub como plataforma principal de control de versiones y colaboración debido a su amplia adopción en proyectos de desarrollo de software y su robusta funcionalidad para la gestión de código fuente. GitHub permite a cada integrante del equipo trabajar de manera colaborativa en el mismo repositorio, manteniendo un historial detallado

de cambios a través de commits y facilitando la revisión y fusión de código mediante un pull requests. La gestión de branches en GitHub asegura que cada desarrollador pueda trabajar de manera independiente en sus funcionalidades asignadas (feature branches), facilitando la integración continua y minimizando conflictos durante el desarrollo, manteniendo el uso de buenas prácticas para evitar conflictos entre las versiones del código.

Nest.js

En la implementación del backend del sistema de gestión de recursos deportivos, se seleccionó Nest como framework de desarrollo debido a su arquitectura basada en módulos, su soporte para TypeScript y su enfoque en el desarrollo de microservicios que sean escalables y mantenibles en el tiempo. Nest proporciona una estructura organizada y modular que facilita la construcción de aplicaciones escalables y mantenibles. Su integración con librerías populares su enfoque en la creación de APIs RESTful o GraphQL son adecuados para las necesidades del proyecto, asegurando un desarrollo eficiente y estructurado del backend.

Next.js

Para el desarrollo del frontend de la plataforma web, se eligió Next.js por su capacidad para ofrecer renderización tanto del lado del servidor (SSR) como del lado del cliente (CSR), mejorando así la velocidad de carga y la experiencia del usuario final. Next.js, siendo un framework de React, simplifica el manejo de rutas, optimiza el rendimiento mediante la carga automática de código y proporciona funcionalidades avanzadas como la generación estática de sitios web (SSG). Estas características hacen que Next.js sea ideal para construir interfaces de usuario interactivas y dinámicas, alineándose perfectamente con los objetivos de diseño y funcionalidad del sistema de gestión de recursos deportivos.

Obsidian para la gestión del proyecto

En la gestión integral del proyecto, se implementó Obsidian como herramienta principal. Obsidian ofrece un entorno flexible y poderoso para la organización de ideas, notas y tareas, utilizando un sistema de enlaces bidireccionales y visualización de datos. Esto facilita la estructuración y conexión de información relevante para el proyecto, permitiendo una gestión efectiva del conocimiento y colaboración entre los miembros del equipo. Obsidian es especialmente útil para mantener un seguimiento detallado de requisitos, decisiones de diseño y progreso del proyecto de manera integrada y accesible.

Historias de usuario

Las historias de usuario fueron redactadas con el propósito de definir claramente las funcionalidades y requisitos de cada módulo dentro del sistema de gestión de recursos deportivos. Cada historia representa una necesidad específica del usuario final, proporcionando un marco detallado de lo que el software debe hacer y por qué es esencial desde la perspectiva del usuario.

Estas historias son cruciales para alinear el desarrollo del software con las expectativas y necesidades reales de los usuarios, asegurando que cada funcionalidad implementada agregue valor directo y resuelva problemas específicos del contexto deportivo. Al centrarse en las historias de usuario, el equipo puede priorizar las características más relevantes y críticas para la experiencia del usuario, mejorando así la calidad del producto final y la satisfacción del usuario.

Historias

Título	Gestión de Disponibilidad de Canchas Deportivas e Implementos	HU1
Como miembro de la comunidad politécnica, quiero hacer uso de las canchas e implementos deportivos en mi tiempo libre, para lo cual quisiera conocer la disponibilidad de las canchas e implementos deportivos en tiempo real, con el fin de gestionar eficientemente mi tiempo.		

Título	Generación de Reportes de Uso de Canchas Deportivas	HU2
Como personal administrativo del área de deportes, quiero acceder a la información relacionada al uso de las canchas e implementos deportivos, con el fin de dar seguimiento y gestionar eficientemente los recursos disponibles, así como generar informes operativos que faciliten una óptima gestión de los recursos.		

Título	Gestión de Eventos Deportivos	HU3
Como personal administrativo del área de deportes, deseo administrar eventos deportivos institucionales solicitados por asociaciones o federaciones a demanda, con el fin de gestionar eficientemente la disponibilidad de los recursos.		

Título	Exposición de Actividades del Área de Deportes	HU4
Como personal administrativo del área de deportes, deseo exponer las actividades realizadas por el área de deportes, con el fin de promover y dar visibilidad de los servicios prestados a la comunidad politécnica.		

Planificación de ciclos

La planificación de estos ciclos dentro del marco ASD permite adaptar continuamente las estrategias y funcionalidades para garantizar un sistema que no solo cumpla con los requisitos iniciales, sino que también evolucione y mejore en respuesta directa a las necesidades y expectativas de los usuarios finales.

Primera iteración

Durante el primer ciclo, tuvo el enfoque en dos áreas principales:

- **Desarrollo del frontend de los módulos:** Implementaremos la interfaz de usuario y la experiencia de usuario para los diferentes módulos del sistema. Esto incluirá la creación de vistas, interacciones y navegación que faciliten el uso intuitivo del sistema por parte de los usuarios finales.
- **Elaboración del backend del módulo de gestión de implementos deportivos:** Implementar la lógica y la funcionalidad necesaria para gestionar eficazmente los implementos deportivos, canchas deportivas.

Al finalizar esta fase inicial, presentaremos los avances al área de deportes para recibir retroalimentación y validación de los requisitos y funcionalidades implementadas.

Segunda iteración

Tras recibir la retroalimentación del primer ciclo, iniciaremos una fase de iteración para:

- **Desarrollo de correcciones en el frontend:** Implementaremos cualquier ajuste o mejora sugerida por los usuarios y stakeholders durante la presentación inicial.
- **Desarrollo de los módulos de reservas y reportes en el backend:** Extendemos la funcionalidad del sistema para permitir la gestión eficiente de reservas de instalaciones deportivas y otros recursos y la gestión de reportes.

Este ciclo repetirá la colaboración con el área de deportes para asegurar que las modificaciones y expansiones respondan directamente a las necesidades identificadas durante la primera fase de desarrollo.

4.2 Fase de colaboración

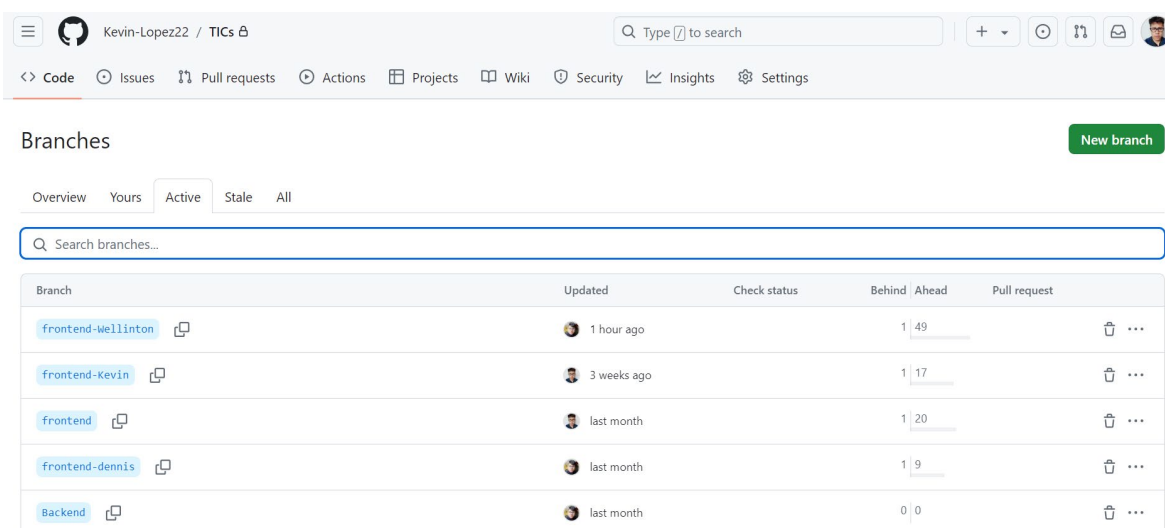
El desarrollo del TIC se organiza en entregables específicos para guiar las tareas y resultados esperados en cada etapa del proyecto. Las actividades se estructuran para cumplir con los objetivos y asegurar un seguimiento efectivo del progreso.

Actividades multidisciplinarias

Se facilitaron reuniones periódicas y sesiones de trabajo donde participan representantes del área de deportes, desarrolladores, diseñadores y otros stakeholders clave. Esta interacción multidisciplinaria permite alinear los objetivos del proyecto con las necesidades del usuario final, integrando diversas perspectivas que enriquecen la solución final.

- **Gestión del equipo de desarrollo**

Dentro del equipo, se gestionó el versionamiento de código utilizando un repositorio tanto para el desarrollo del backend como del frontend. Se implementaron buenas prácticas para evitar conflictos entre ramas, de manera que cada integrante del grupo manejaba una rama de frontend y backend, integrando luego los cambios mediante pull requests.



Branch	Updated	Check status	Behind	Ahead	Pull request
frontend-Wellinton	1 hour ago		1	49	
frontend-Kevin	3 weeks ago		1	17	
frontend	last month		1	20	
frontend-dennis	last month		1	9	
Backend	last month		0	0	

Figura 7. GitHub versionamiento

El equipo mantenía reuniones semanales para medir avances y resolver dudas adicionales que pudieran tener los integrantes. Además, se realizaban pequeñas charlas diarias, cuando era necesario, para solventar dudas a medida que avanzaba el desarrollo.

- **Iteración y retroalimentación continua**

Durante las fases de desarrollo, se implementan ciclos iterativos donde se presentan avances y prototipos al área de deportes para obtener retroalimentación temprana. Esta retroalimentación es esencial para validar y ajustar las funcionalidades del sistema según las necesidades reales del usuario y las expectativas del cliente.

- **Adaptabilidad y flexibilidad**

La metodología ASD permite adaptarse rápidamente a cambios en los requisitos y prioridades del proyecto. Mediante la colaboración continua y la comunicación abierta, el equipo puede responder ágilmente a nuevas oportunidades o desafíos, asegurando la entrega de un sistema que cumpla con las expectativas y requisitos cambiantes.

Planificación de actividades

Este apartado especifica los entregables planificados para el desarrollo del proyecto de integración curricular. Estos entregables están diseñados para proporcionar una guía clara sobre las tareas a completar y los resultados esperados en cada etapa del proyecto.

- **Periodicidad de las entregas**

Se ha decidido realizar entregas cada dos semanas. Esta periodicidad se ha seleccionado por considerarse un tiempo razonable para el desarrollo de entregables significativos, lo que permite un seguimiento efectivo del progreso del proyecto y una oportunidad para ajustar la planificación en caso necesario.

Tabla 3. Cronograma de actividades

Fecha	Actividades
01/05/2024	<ul style="list-style-type: none"> • Entregables para el proyecto • Arquitectura del sistema • Documento de requerimientos • Documento de planificación de actividades
10/05/2024	<ul style="list-style-type: none"> • Diagrama detallado (un solo diagrama de bajo nivel con los 3 módulos)
15/05/2024	<ul style="list-style-type: none"> • Frontend módulo de reservas • Desarrollo de mockups • Desarrollo de páginas utilizando React
29/05/2024	<ul style="list-style-type: none"> • Backend módulo de reservas • Desarrollo de lógica de módulo de reservas • Desarrollo de plan de pruebas • Informe de pruebas
12/06/2024	<ul style="list-style-type: none"> • Frontend módulo de implementos y reportes • Desarrollo de mockups • Desarrollo de páginas utilizando React
26/06/2024	<ul style="list-style-type: none"> • Backend módulo de implementos y reportes

	<ul style="list-style-type: none"> • Desarrollo de lógica de módulo de implementos y reportes • Desarrollo de plan de pruebas • Informe de pruebas
10/07/2024	<ul style="list-style-type: none"> • Frontend módulo de reservas Eventos • Desarrollo de mockups • Desarrollo de páginas utilizando React
24/07/2024	<ul style="list-style-type: none"> • Backend módulo de reservas Eventos • Desarrollo de lógica de módulo de reservas Eventos

Primera iteración

La primera interacción se centró en dos aspectos principales el desarrollo del frontend de la aplicación en general y el backend del módulo de bodega virtual.

- **Desarrollo del frontend**

Para el desarrollo del frontend, se buscó crear un estilo visual que se asemejara a las páginas de la Escuela Politécnica Nacional (EPN), de manera que los usuarios se sientan identificados al usar la aplicación. Durante este proceso, se utilizó la guía de los mockups elaborados en el proceso de levantamiento de requerimientos, lo que permitió tener una idea inicial para el posible diseño de la solución. Una vez alineado el diseño preliminar y establecidos los requisitos, se procedió a generar una interfaz responsiva, fácil de entender y cómoda de usar para el usuario.

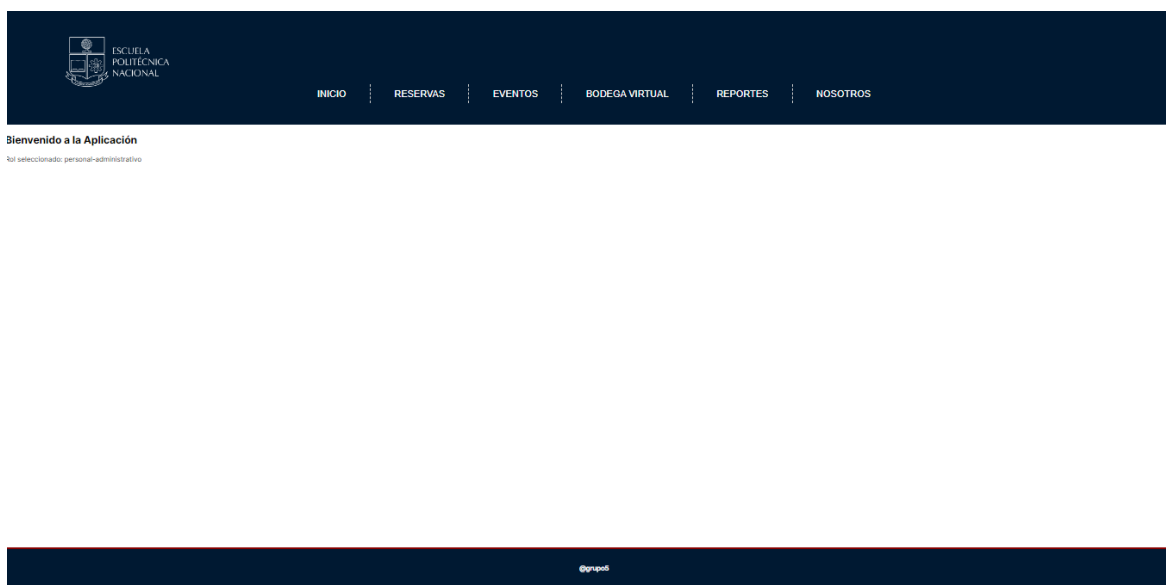


Figura 8. Página principal

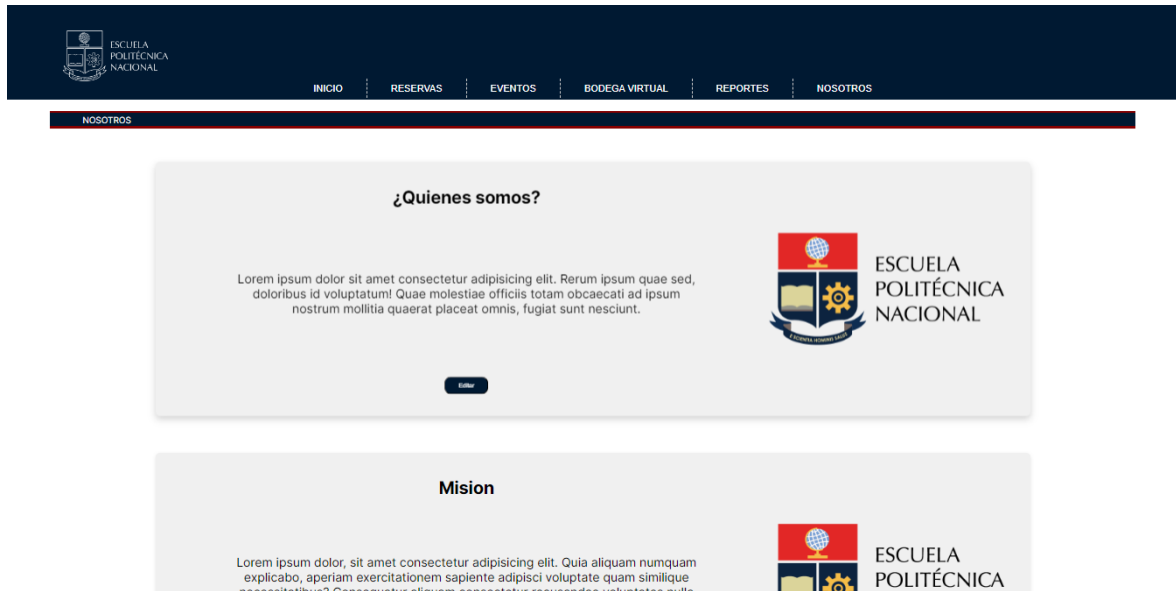


Figura 9. Página nosotros

- **Implementación técnica**

El uso de Next.js facilitó la creación y gestión de rutas, permitiendo así una navegación eficiente dentro de la aplicación. Además, se crearon componentes con responsabilidad única, lo que permite la reutilización de aquellos que sean similares o que manejen una lógica común. Esta modularidad en el diseño de los componentes no solo mejora la mantenibilidad del código, sino que también acelera el desarrollo al evitar la duplicación de esfuerzos, como parte del proceso para un diseño funcional se usaron APIs con información genérica para las pruebas.

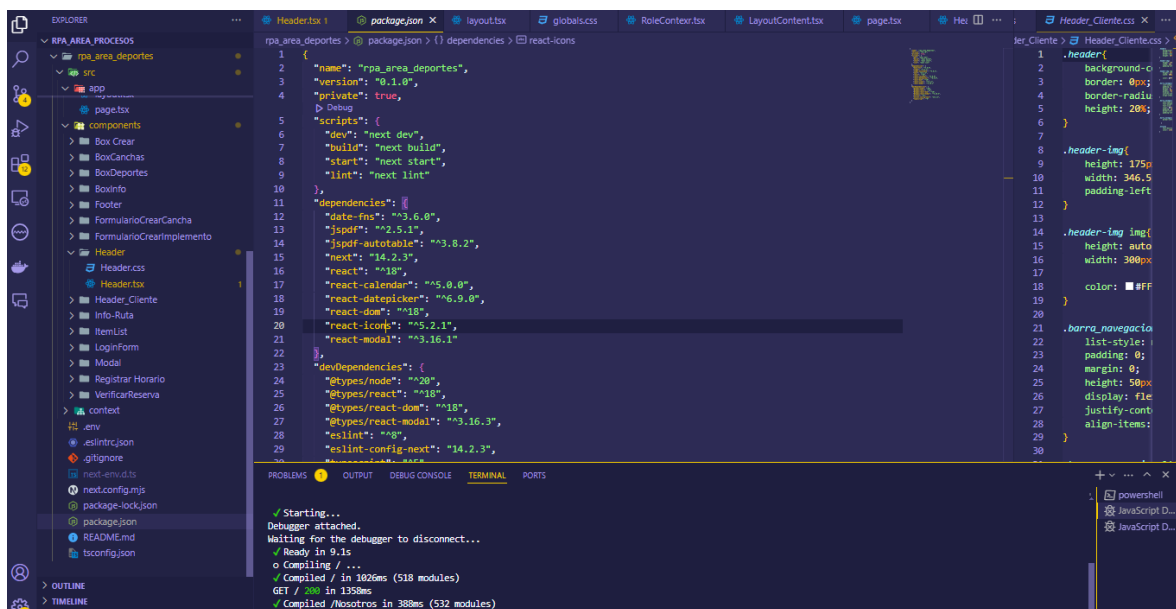


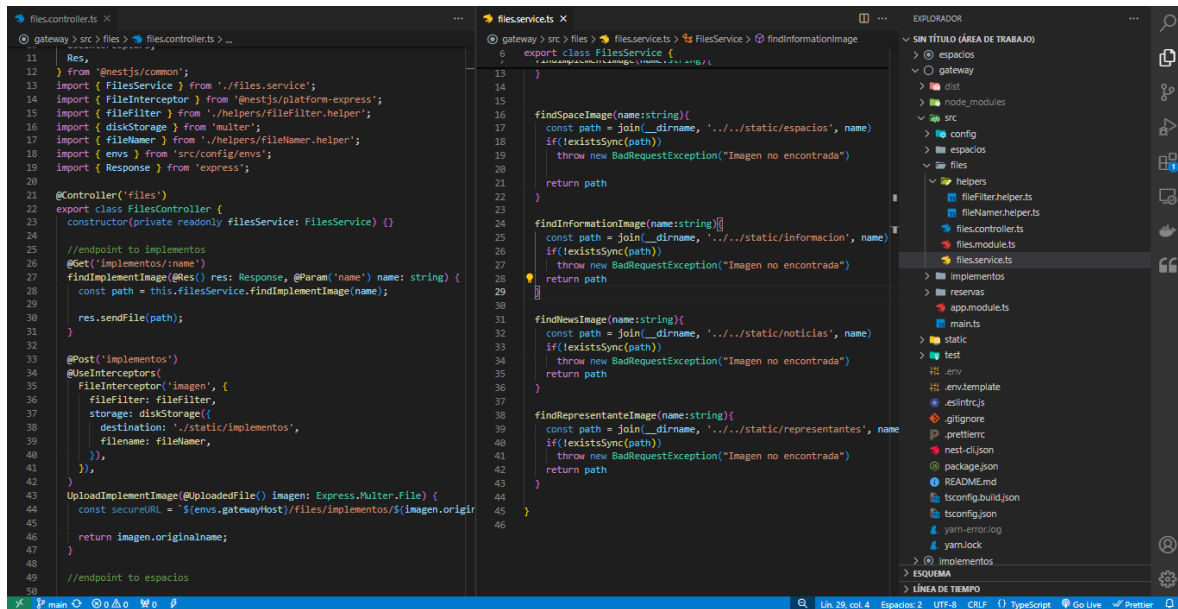
Figura 10. Gestión de componentes

- **Desarrollo del backend del módulo de bodega virtual**

En cuanto al backend del módulo de bodega virtual, se implementaron las funcionalidades necesarias para la gestión eficiente de los recursos almacenados. Esto incluye la creación de APIs seguras para el acceso y manipulación de datos, así como la integración con el frontend para asegurar una comunicación fluida entre ambos lados de la aplicación.

- **Microservicios y gestión de recursos deportivos**

Además, se desarrollaron microservicios específicos para la gestión de implementos y canchas deportivas. Para estos microservicios, se implementó un CRUD (Crear, Leer, Actualizar, Eliminar) que permite la gestión completa de estos recursos. Un aspecto destacado fue la gestión de imágenes, de manera que cuando un usuario realiza una reserva, pueda guiarse por las imágenes y saber exactamente qué cancha está reservando. Para esta implementación, se utilizó el framework NestJS, que permitió estructurar el backend de manera modular y eficiente.



```
files.controllers.ts
11 Res:
12 } from '@nestjs/common';
13 import { FilesService } from './files.service';
14 import { FileInterceptor } from '@nestjs/platform-express';
15 import { fileFilter } from './helpers/fileFilter.helper';
16 import { diskStorage } from 'multer';
17 import { fileNames } from './helpers/fileName.helper';
18 import { envs } from 'src/config/envs';
19 import { Response } from 'express';
20
21 @Controller('files')
22 export class FilesController {
23   constructor(private readonly filesService: FilesService) {}
24
25   //endpoint to implementos
26   @Get('implementos/:name')
27   findImplementImage(@Res() res: Response, @Param('name') name: string) {
28     const path = this.filesService.findImplementImage(name);
29
30     res.sendFile(path);
31   }
32
33   @Post('implementos')
34   @UseInterceptors(
35     FileInterceptor('imagen', {
36       fileFilter: fileFilter,
37       storage: diskStorage({
38         destination: './static/implementos',
39         filename: fileNames,
40       }),
41     }),
42   )
43   uploadImplementImage(@UploadedFile() imagen: Express.Multer.File) {
44     const secureURL = `${envs.gatewayHost}/files/implementos/${imagen.originalname}`;
45     return imagen.originalname;
46   }
47
48   //endpoint to espacios
49
50
files.services.ts
6 export class FilesService {
7
8
9
10
11
12
13
14
15
16 findSpaceImage(name:string){
17   const path = join(__dirname, '../../static/espacios', name)
18   if(!existsSync(path))
19     throw new BadRequestException("Imagen no encontrada")
20
21   return path
22 }
23
24
25 findInformationImage(name:string){
26   const path = join(__dirname, '../../static/informacion', name)
27   if(!existsSync(path))
28     throw new BadRequestException("Imagen no encontrada")
29   return path
30 }
31
32
33 findNewsImage(name:string){
34   const path = join(__dirname, '../../static/noticias', name)
35   if(!existsSync(path))
36     throw new BadRequestException("Imagen no encontrada")
37   return path
38 }
39
40
41 findRepresentanteImage(name:string){
42   const path = join(__dirname, '../../static/representantes', name)
43   if(!existsSync(path))
44     throw new BadRequestException("Imagen no encontrada")
45   return path
46 }
```

Figura 11. Modulo renderización imágenes

Segunda iteración

Esta segunda interacción se centró en el control de cambios el frontend según la retroalimentación recibida, ampliar las capacidades del backend mediante la implementación de herramientas avanzadas de generación de reportes, y en la gestión del cambio para garantizar la entrega de todas las funcionalidades pendientes. Esto contribuyó a una aplicación más completa y alineada con las necesidades y expectativas de los usuarios.

Refinamiento del frontend

Durante la segunda interacción, se realizaron correcciones en el frontend basadas en las conversaciones con los clientes. Se validaron los requisitos y se facilitó la resolución de nuevos requerimientos. Entre los principales comentarios recibidos, se destacaron:

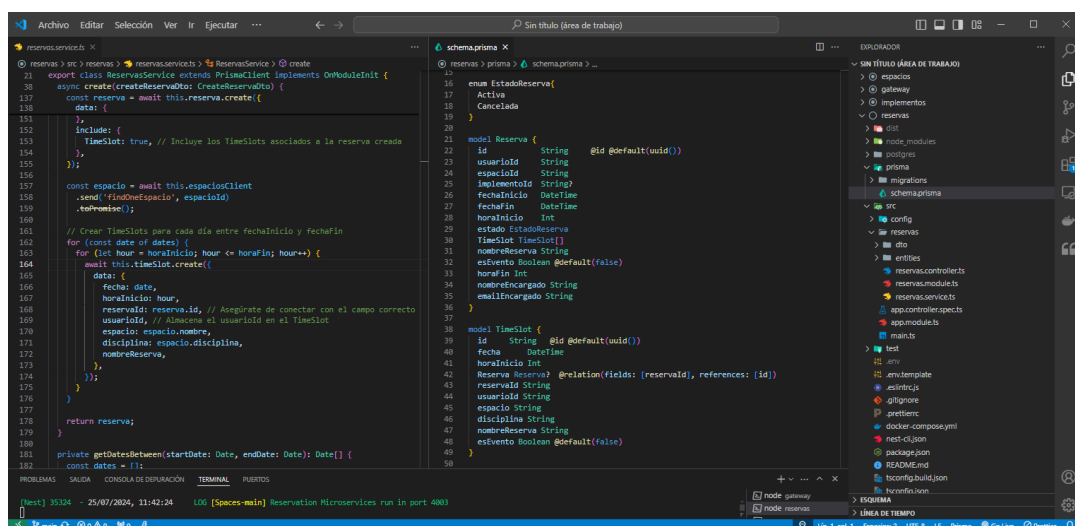
- **Mejoras en la selección de colores:** Se ajustaron los colores para que estuvieran más acorde con la paleta de colores de la Escuela Politécnica Nacional (EPN).
- **Validación del acceso a la información:** Se aseguró que los usuarios tuvieran acceso a la información necesaria de manera clara y eficiente.

Estas mejoras garantizaron que la aplicación no solo fuera funcional, sino también estéticamente coherente con la identidad de la EPN, mejorando la experiencia del usuario.

Desarrollo del backend

En la parte del backend, se desarrolló un modelo para la generación de reportes utilizando PowerBI. Este modelo permite a los administradores generar reportes detallados y visuales sobre el uso de los recursos deportivos y otros aspectos gestionados por la aplicación. La integración con PowerBI proporciona una poderosa herramienta de análisis, facilitando la toma de decisiones basada en datos precisos y actualizados.

Adicional se desarrollo el microservicio para la gestión de reservas, teniendo en cuenta que dentro del microservicio existen dos tipos de reservas únicas y reservas de eventos, se implementó la logica de creación de ranuras de tiempo (time slots) para mejorar la presentacion y el manejo de la disponibilidad de espacios deportivos.



```
reservas.service.ts
21 export class ReservasService extends PrismaClient implements OnModuleInit {
22   async create(createReservasDto: createReservaDto) {
23     const reserva = await this.reserva.create({
24       data: {
25         include: {
26           TimeSlot: true, // Incluye los TimeSlots asociados a la reserva creada
27         },
28       },
29     });
30     const espacio = await this.espacioClient
31       .send(findOneEspacio, espacioId)
32       .toPromise();
33     // Crear TimeSlots para cada día entre fechaInicio y fechaFin
34     for (const date of dates) {
35       for (let hour = horaInicio; hour <= horaFin; hour++) {
36         await this.timeSlot.create({
37           data: {
38             fecha: date,
39             horaInicio: hour,
40             reservaId: reserva.id, // Asegúrate de conectar con el campo correcto
41             usuarioId, // Almacena el usuarioId en el TimeSlot
42             espacio: espacio.nombre,
43             disciplina: espacio.disciplina,
44             nombreReserva,
45           },
46         });
47       }
48     }
49     return reserva;
50   }
51   private getDatesBetween(startDate: Date, endDate: Date): Date[] {
52     const dates = [];
53     let date = startDate;
54     while (date <= endDate) {
55       dates.push(date);
56       date = new Date(date.getFullYear(), date.getMonth(), date.getDate() + 1);
57     }
58     return dates;
59   }
60 }
```

```
schema.prisma
enum EstadoReserva {
  Activa
  Cancelada
}
model Reserva {
  id String @id @default(uuid())
  usuarioId String
  espacioId String
  implementado String?
  fechaInicio DateTime
  fechaFin DateTime
  horaInicio Int
  estado EstadoReserva
  TimeSlot TimeSlot[]
  nombreReserva String
  esEvento Boolean @default(false)
  horaFin Int
  nombreEncargado String
  emailEncargado String
}
model TimeSlot {
  id String @id @default(uuid())
  fecha DateTime
  horaInicio Int
  reserva Reserva? @relation(fields: [reservaId], references: [id])
  reservaId String
  usuarioId String
  espacio String
  disciplina String
  nombreReserva String
  esEvento Boolean @default(false)
}
```

Figura 12. Modelo y manejo de time slots en microservicio de reservas

4.3 Fase de aprendizaje

Se caracteriza por poder adquirir conocimiento sobre el proyecto, identificar retroalimentación riesgos y establecer buena comunicación. Involucra revisión de requerimientos, investigación tecnológica, creación de prototipos y planificación de riesgos.

Entrega de resultados

- **Primera iteración**

Durante este periodo se logró desarrollar una interfaz que refleja la identidad visual de la Escuela Politécnica Nacional (EPN). Los componentes del frontend se diseñaron para ser reutilizables y fáciles de mantener, se mantuvo un diseño similar al apartado de reservas para mantener la consistencia y estándares establecidos durante la fase de especulación.

En el manejo de la página web, los usuarios administradores podrán acceder a un apartado para la administración y gestión de eventos, la pantalla muestra la opción de crear nuevos eventos y bajo esta se propone mostrar los eventos existentes:

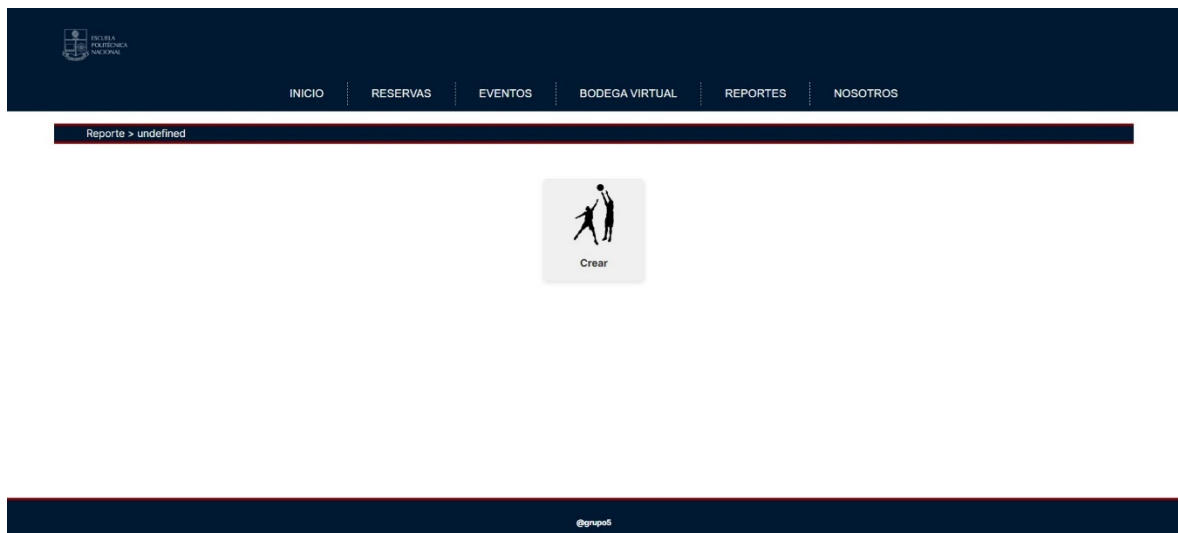


Figura 13. Página de eventos V1

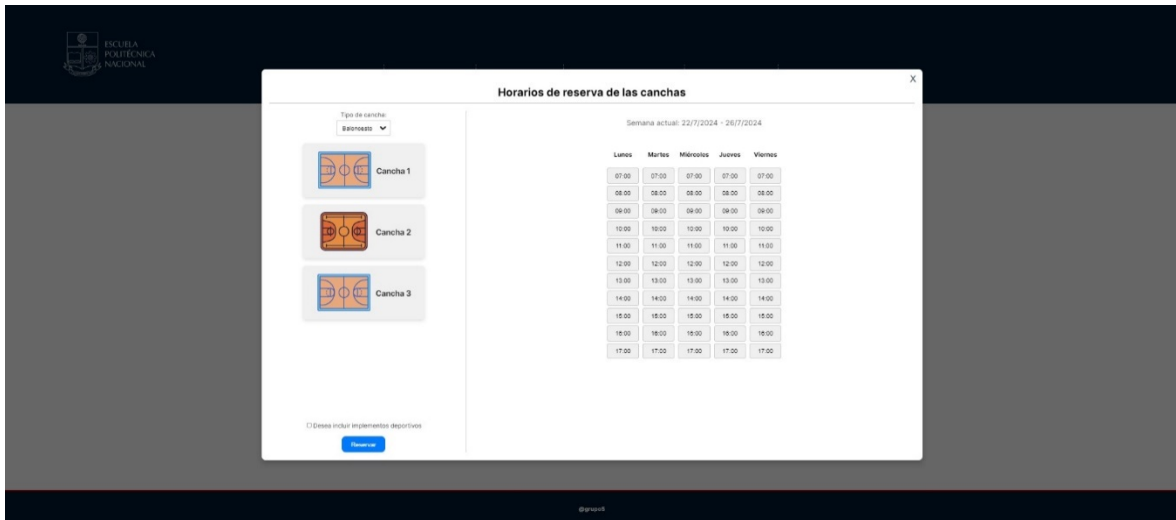


Figura 14. Creación de evento V1

Durante esta iteración se destinó los esfuerzos en la creación de la interfaz de usuario, motivo que orillo a la postergación de la implementación del backend para el módulo de eventos.

- **Segunda iteración**

Durante la segunda iteración, se ajustaron los colores y elementos visuales del frontend para alinearse con los estándares de la EPN, mejorando así la experiencia del usuario. Se validó que la interfaz fuera intuitiva y accesible, y se corrigieron errores basados en la retroalimentación de los clientes.

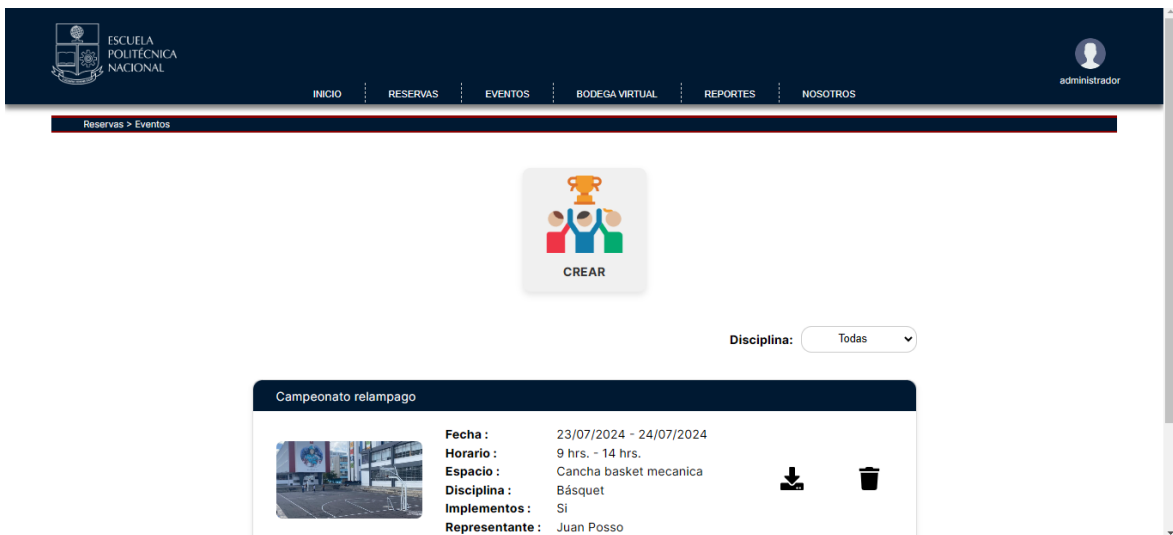


Figura 15. Página de eventos V2

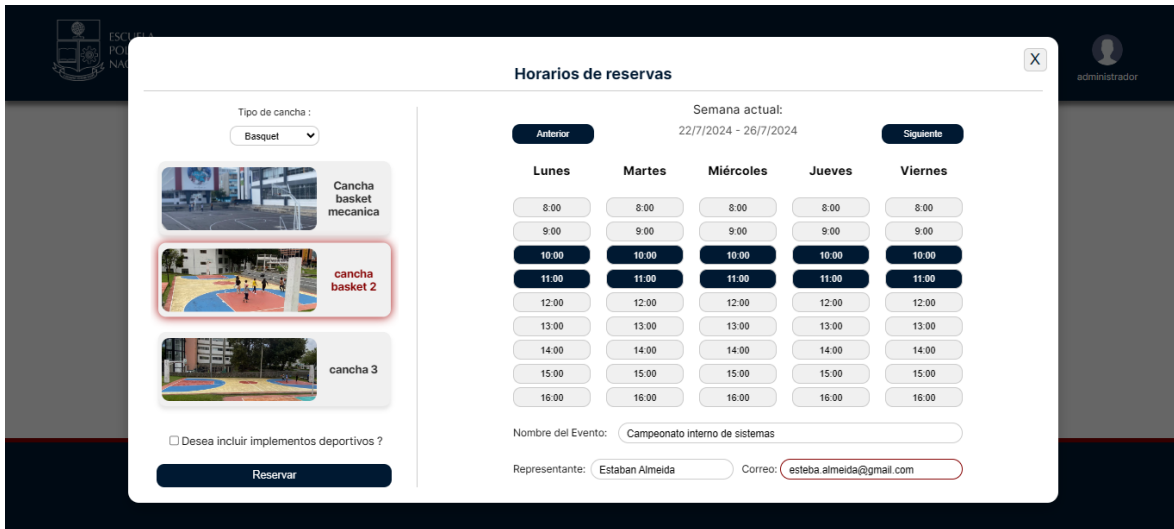


Figura 16. Creación de eventos V2

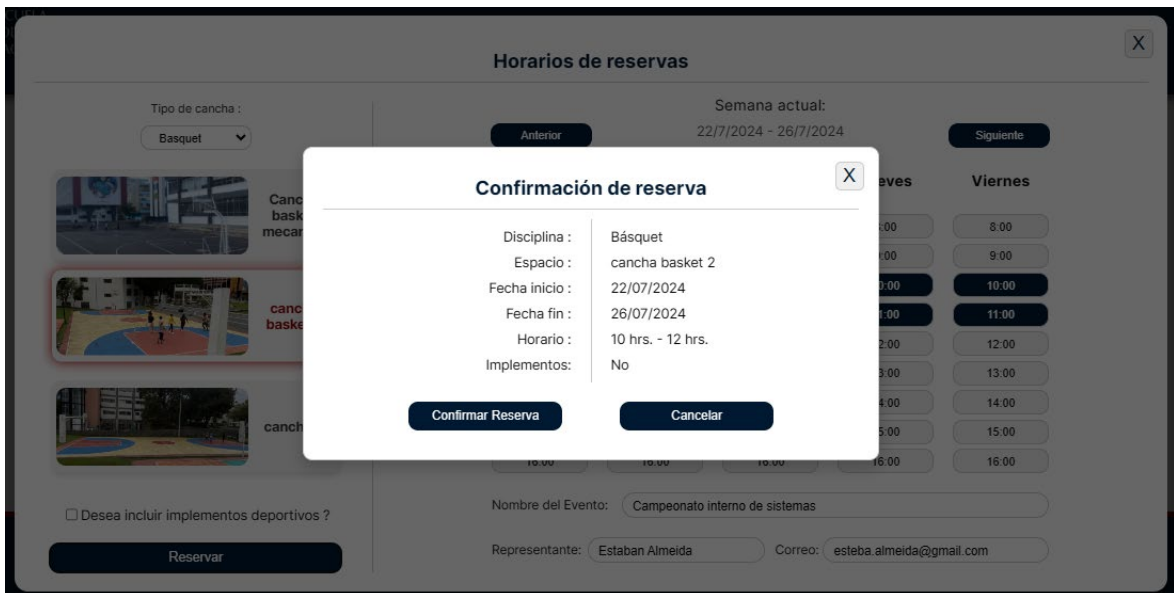


Figura 17. Confirmación de eventos V1

Al confirmarse la creación de un evento el sistema genera un documento el cual funge como comprobante del evento



CONFIRMACION DE RESERVA

Campeonato interno de sistemas

RESERVA

Responsable: Estaban Almeida
 Fecha: 22-07-2024 - 26-07-2024
 Horario: 10 hrs - 12 hrs

DETALLES DE CANCHA

Disciplina: Básquet
 Espacio: cancha basket 2



DETALLES DE IMPLEMENTOS

Implementos deportivos: No

Observaciones:.....

Estimado(a), para confirmar su reserva, presente este comprobante junto con su documento de mayor de edad en el Área de Deportes.

Figura 18. Documento de confirmación de eventos

En adición al manejo de reservas, también se desarrolló el apartado de noticias en la página de inicio y página de información sobre el área de deportes.



Figura 19. Página principal V2

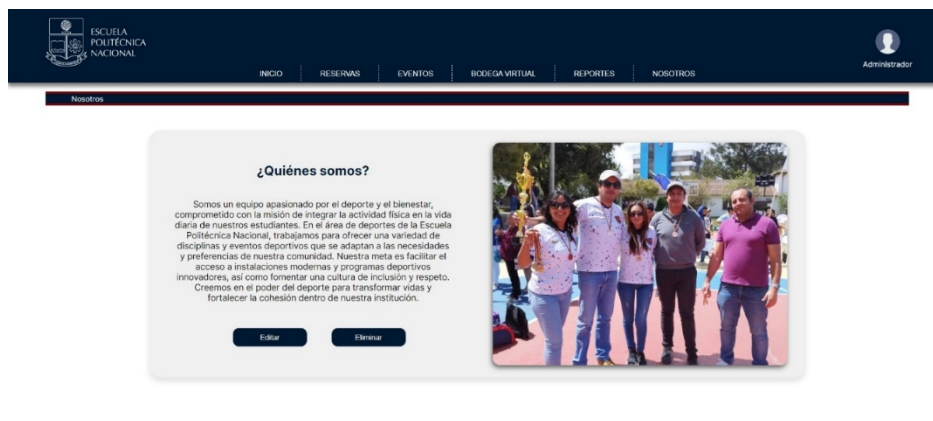


Figura 20. Página nosotros V2

Retroalimentación de los stakeholders

Primera iteración

Una vez finalizada la presentación con los stakeholders, se recibieron varias retroalimentaciones, muchas relacionadas con el color y las funcionalidades que se deberían considerar. Entre los principales cambios y sugerencias se encuentran:

- Añadir un botón para deshabilitar un botón en el apartado de la bodega virtual.
- Cambiar la paleta de colores para ciertos botones.
- Añadir botones para cerrar ventanas.
- Incluir mensajes informativos.
- En la parte visual, permitir filtrar por tipo de implemento o cancha en las búsquedas.
- En la parte de los reportes, consolidar los apartados de implementos y canchas en un único tipo de reporte en lugar de tenerlos por separado
- Se solicitó la adición de campos que permitan especificar al encargado de los eventos que se creen.

Segunda iteración

Después de socializar las actualizaciones con los clientes y recibir su retroalimentación, se corrigieron los errores detectados y se gestionó el control de cambios para asegurar que todas las modificaciones fueran correctamente implementadas. Además, se entregaron las funcionalidades pendientes, garantizando que la aplicación cumpliera con los requisitos y expectativas establecidos. Esta gestión del cambio fue crucial para asegurar una aplicación funcional y alineada con las necesidades del usuario

5 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

Luego de haber realizado el proyecto de integración curricular y en particular el componente “Nombre componente” podemos concluir que:

- Es vital asegurarse de que todas las partes interesadas, especialmente aquellas con roles críticos en el proceso, sean consultadas desde el principio. En particular, en el área de educación física la persona encargada de las reservas de las reservas de canchas e implementos deportivos no fue incluida en las discusiones iniciales. Al presentar la idea, esta persona se mostró en contra de la creación del software. Este incidente subraya la importancia de tener empatía y de dialogar desde el principio con todos los participantes e involucrados en el proyecto. Entender las necesidades de todos es esencial. Además, es crucial mantener una comunicación efectiva y continua con todas las partes interesadas durante todo el proceso de desarrollo. Es importante comunicar claramente los objetivos del proyecto y asegurar que nadie sienta que su trabajo está en riesgo, sino que, por el contrario, la aplicación mejorará la eficiencia de sus procesos.
- En el proceso de la selección de una arquitectura de software, es importante reconocer que a nivel teórico se pueda entender cómo aplicar una arquitectura de manera eficiente, en práctica puede revelar desafíos imprevistos. Inicialmente, para el desarrollo de TIC se planteó el desarrollo de microservicios con la intención de que cada módulo o funcionalidad tuviera responsabilidades únicas y específicas. Por ejemplo, se propuso tener un microservicio para reservas normales y otro para reservas de eventos, sin embargo, durante el desarrollo, nos dimos cuenta de que no era eficiente tener dos microservicios con funcionalidades tan similares. Aunque la teoría sugería esta división, en la práctica no resultó ser una solución óptima, finalmente se decidió combinar ambas funcionalidades en un solo microservicio que manejara ambos tipos de reservas. En conclusión, es crucial comprender que no siempre es posible ni conveniente adaptar o forzar una división estricta en microservicios. La modularidad excesiva puede complicar el desarrollo y afectar la eficiencia del sistema. Es importante ser flexible y evaluar continuamente la arquitectura propuesta durante el proceso de implementación para asegurarse de que realmente se está logrando la optimización y eficiencia deseadas.

5.2 Recomendaciones

En el proceso de desarrollo de software, es fundamental conocer cómo otras personas han planteado el desarrollo de soluciones similares. Este conocimiento nos permite tener una idea del éxito o fracaso de proyectos anteriores, sirviendo como una valiosa referencia para nuestros propios proyectos. Por esta razón, se recomienda realizar un estudio del estado del arte, el mismo que permite identificar y comprender las metodologías utilizadas por otros desarrolladores, así como los enfoques que adoptaron durante el proceso de desarrollo. Este análisis proporciona una visión clara y detallada que puede servir tanto para aplicar directamente metodologías exitosas como para adaptar y mejorar las mismas según las necesidades específicas de nuestro proyecto; incorporar este estudio como uno de los primeros pasos en el desarrollo de software es crucial. La investigación puede emplear diversas técnicas, como la revisión de literatura académica, análisis de casos de estudio y consulta de expertos en la materia. Estas técnicas ayudan a construir una base sólida de conocimiento que incrementa la probabilidad de éxito del proyecto.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] «Red Hat - We make open source technologies for the enterprise,» 19 Julio 2022. [En línea]. Available: <https://www.redhat.com/es/topics/devops/what-is-agile-methodology#:~:text=La%20metodolog%C3%ADa%20%C3%A1gil%20favorece%20un,lugar%20de%20resistirse%20a%20ellos..> [Último acceso: 07 2024].
- [2] S. Universidades, «Metodologías de desarrollo de software: ¿qué son?,» 21 12 2020. [En línea]. Available: <https://www.santanderopenacademy.com/es/blog/metodologias-desarrollo-software.html>. [Último acceso: 07 2024].
- [3] Lucidchart, «Desarrollo basado en funciones FDD: Por qué y cómo utilizarlo,» [En línea]. Available: <https://www.lucidchart.com/blog/es/utilizar-el-desarrollo-basado-en-funciones>. [Último acceso: 07 2024].
- [4] M. Á. D. DIOS, «Scrum: qué es y cómo funciona este marco de trabajo,» wearemarketing, 2022 05 2022. [En línea]. [Último acceso: <https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html#> 04 2024].
- [5] ingenieriadesoftware, «ASD Ingeniería de Software,» [En línea]. Available: http://ingenieriadesoftware.mex.tl/61154_asd.html#:~:text=El%20desarrollo%20de%20software%20adaptable,continua%20del%20proceso%20al%20trabajo.. [Último acceso: 04 2024].
- [6] R. J. Ayala Nazate, «Elaboración de un prototipo de una aplicación web y móvil para gestionar la reserva de canchas sintéticas de fútbol en el Ecuador,» UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA, Loja, 2018.
- [7] P. F. CASTILLO DUARTE y L. M. FIGUEROA ROBLES, «DISEÑO E IMPLANTACIÓN DE UN SISTEMA WEB PARA LA GESTIÓN Y CONTROL DE CAMPEONATOS DEPORTIVOS DE LAS DISCIPLINAS DE FÚTBOL Y BÁSQUET, ORGANIZADOS POR FEDELIBAL, UTILIZANDO UN MODELO MATEMÁTICO PARA LA GENERACIÓN DEL CALENDARIO DE ENFRENTAMIENTOS,» UNIVERSIDAD NACIONAL DE LOJA, Loja, 2011.
- [8] B. P. OLIVARES PILATAXI, «DESARROLLO DE UN SISTEMA WEB PARA GESTIONAR LOS SERVICIOS DEL CENTRO RECREACIONAL EL VARADERO,» ESCUELA POLITÉCNICA NACIONAL, Quito, 2021.
- [9] P. I. GUALA BRAVO, «DESARROLLO DE UN SISTEMA WEB Y APLICACIÓN MÓVIL PARA LA RESERVA DE CANCHAS Y OFERTA DE CURSOS EN LA URBANIZACIÓN LOS RETOÑOS,» ESCUELA DE FORMACIÓN DE TECNÓLOGOS, Quito, 2023.
- [10] P. O. Contreras Villegas y S. I. González Cruz, «Plataforma web para gestionar reservas de horas de canchas de fútbol.,» UNIVERSIDAD DEL BÍO-BÍO, Chillán, 2018.

- [11] J. I. CUEVA QUINTANA, «SISTEMA DE RESERVACIÓN DE CANCHAS SINTÉTICAS PARA LA EMPRESA SPORTS CÉSPED.,» UNIVERSIDAD TECNOLÓGICA ISRAEL, Quito, 2019.
- [12] P. P. Alex Andy , «Sistema Web para el proceso de reservas de Canchas Deportivas en la Empresa Corporación Trescientos Sesenta Grados S.A.C. – Lima,» Universidad César Vallejo, Lima, 2020.
- [13] S. J. EDWIN TAFURTH, «GESTION DE RESERVAS PARA CANCHAS SINTETICAS EN LA CIUDAD DE PEREIRA,» UNIVERSIDAD TECNOLÓGICA DE PEREIRA, Pereira, 2018.
- [14] D. . I. DEL PEZO RODRÍGUEZ, «DESARROLLO DE UNA APLICACIÓN MÓVIL PARA RESERVA DE CANCHAS Y ORGANIZACIÓN DE TORNEOS EN COMPLEJOS DEPORTIVOS DE LA PROVINCIA DE SANTA ELENA,» ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL, GUAYAQUIL, 2017.
- [15] J. M. Gutierrez Onofrio, «Sistema web para mejorar el proceso de reservas de canchas deportivas en la empresa El Encuentro S.A.C., La Convención 2022,» UNIVERSIDAD PRIVADA LÍDER PERUANA, CUSCO, 2022.
- [16] J. Y. Cabronero, «Diseño de una herramienta para la gestión de pistas deportivas,» UNIVERSIDAD AUTÓNOMA DE MADRID, MADRID, 2016.
- [17] E. A. GUERRERO ACOSTA y D. G. NEWBALL DE ÁVILA, «DESARROLLO DE APLICATIVO DE RESERVAS Y GESTIÓN DE LA INFORMACIÓN PARA ESCENARIOS DEPORTIVOS EN LA CIUDAD DE CARTAGENA,» UNIVERSIDAD DE CARTAGENA, CARTAGENA, 2023.
- [18] J. M. Arze, Artist, *Sistema Web Colaborativo Integrado de Control Administración y Seguimiento basado en gestión de Procesos de Negocios.Caso: Empresa EDUCOM-SRL..* [Art]. Universidad Mayor de San Andrés, 2016.
- [19] I. R. Licon, Artist, *Modelo de sistemas de información contable con ASD CASO:Asociación centro juvenil para el desarrollo humano CEJUPA..* [Art]. Universidad Mayor de San Andrés, 2011.
- [20] J. Rivero, Artist, *Desarrollo Adaptable de Software (ASD).* [Art]. Universidad Nacional Experimental De los Llanos Occidentales, 2014.

7 ANEXOS

ANEXO I

Repositorio Backend para Proyecto de Integración Curricular

El código fuente del backend para el proyecto de integración curricular se encuentra disponible en el siguiente repositorio de GitHub:

<https://github.com/BackendTic>

Este repositorio contiene la lógica del servidor, las conexiones a la base de datos, y las API endpoints necesarias para el funcionamiento del proyecto.

ANEXO II

Repositorio Frontend para Proyecto de Integración Curricular

El código fuente del frontend para el proyecto de integración curricular se encuentra disponible en el siguiente repositorio de GitHub:

<https://github.com/Kevin-Lopez22/TICs>

Este repositorio incluye la interfaz de usuario, la lógica de interacción con el backend, y los componentes visuales que componen la aplicación web.