

**Implementation of a Web Application for the Administration and
Coordination of Cultural Events**
**Implementación de una Aplicación Web para la Administración y
Coordinación de Eventos Culturales**

Autores:

Argudo-Piedra, Miguel Angel
UNIVERSIDAD CATÓLICA DE CUENCA
Estudiante de la Unidad Académica de Informática, Ciencias de la Computación, e
Innovación Tecnológica
Cuenca – Ecuador



maargudop30@est.ucacue.edu.ec



<https://orcid.org/0009-0000-6179-5087>

Sañay-Sañay, Segundo Isael
UNIVERSIDAD CATÓLICA DE CUENCA
Docente de la Unidad Académica de Informática, Ciencias de la Computación, e
Innovación Tecnológica
Cuenca – Ecuador



ssanay@ucacue.edu.ec



<https://orcid.org/0000-0003-4661-318X>

Fechas de recepción: 27-ENE-2025 aceptación: 27-FEB-2025 publicación: 15-MAR-2025



<https://orcid.org/0000-0002-8695-5005>

<http://mqrinvestigar.com/>



Resumen

Este documento presenta el desarrollo de “aplicación web para la administración y coordinación de eventos culturales” en Cuenca - Ecuador. La Casa Cultural es reconocida por su contribución al desarrollo cultural y social, sirviendo como un punto de encuentro para la comunidad y como un espacio de preservación del patrimonio cultural. Sin embargo, la gestión manual de procesos como la organización de eventos, la reserva de salas y el control de inventarios ha limitado su operatividad y eficiencia. El proyecto aplica FURPS+ que es un modelo de calidad de software que, bajo el acrónimo de su nombre, establece un conjunto de factores y atributos de calidad. En el proceso de desarrollo de software, se utilizó el marco de trabajo AUP [Proceso Unificado Ágil] de desarrollo de software.

El objetivo central se orienta a implementar una App Web que permita centralizar y automatizar las tareas, mejorando la planificación, reduciendo errores y promoviendo una administración más eficiente. El desarrollo de la herramienta tecnológica no solo facilitará la gestión operativa, sino que también posicionará a la Casa Cultural como un modelo de innovación cultural.

La implementación de la aplicación ha permitido una notable reducción en los tiempos de planificación y organización de eventos, aumentando la capacidad de respuesta ante solicitudes de reservas y consultas. Además, el control de inventarios se ha vuelto más preciso y accesible, disminuyendo errores relacionados con el registro manual. Por otro lado, la centralización de la información ha facilitado el acceso a datos en tiempo real, promoviendo una toma de decisiones más ágil. Estos avances posicionan a la Casa Cultural como un referente de innovación tecnológica en la gestión cultural de la ciudad.

Palabras clave: Administración de eventos culturales; Patrimonio cultural; Eficiencia operativa; Proceso Unificado Ágil



Abstract

This document presents the development of a web application for the administration and coordination of cultural events in Cuenca, Ecuador. The Cultural Center is recognized for its contribution to cultural and social development, serving as a meeting point for the community and a space for the preservation of cultural heritage. However, the manual management of processes such as event organization, room reservations, and inventory control has limited its operational capacity and efficiency.

The project applies FURPS+, a software quality model that establishes a set of factors and quality attributes. During the software development process, the Agile Unified Process (AUP) framework was used. The main objective is to implement a web application that centralizes and automates these tasks, improving planning, reducing errors, and promoting more efficient administration. The development of this technological tool will not only facilitate operational management but also position the Cultural Center as a model of cultural innovation.

The implementation of the application has significantly reduced event planning and organization times, increasing responsiveness to reservation requests and inquiries. Additionally, inventory control has become more precise and accessible, minimizing errors related to manual records. Moreover, the centralization of information has facilitated real-time data access, promoting more agile decision-making. These advancements position the Cultural Center as a benchmark for technological innovation in cultural management within the city.

Keywords: Cultural event management; Cultural heritage; Operational efficiency; Agile Unified Process

Introducción

El uso de las aplicaciones web [App Web], se define como herramientas que permiten a los usuarios acceder a servicios a través de la red mediante un navegador. Las aplicaciones se ejecutan en entornos intranet como en internet, permitiendo la interacción con servidores remotos (Lujan Mora, 2002). La Casa Cultural, busca innovar y aplicar tecnología para optimizar la gestión administrativa. Con este propósito, la propuesta ha sido desarrollar una aplicación web [Cliente - Servidor] que permita centralizar y automatizar procesos administrativos, como la organización de eventos, el alquiler de salas y el control de inventarios.

El documento se ha estructurado en cuatro secciones. De modo que, la primera sección corresponde al “Marco teórico”, en la que se realiza una revisión de la literatura orientado al entorno cultural y aspectos de arquitectura de software; mientras que en la sección 2 se aborda los “Métodos y herramientas de desarrollo de software”, aquí se describe el proceso de la investigación y las herramientas tecnológicas utilizadas en la misma. Asimismo, en la sección 3 se orienta a la “Discusión y resultados”, se describen las implicaciones del desarrollo del producto y los resultados obtenidos del mismo. Y para finalizar, se trata las “Conclusiones y Recomendaciones”, describiendo los resultados de la ejecución del proyecto.

Con la ejecución del proyecto en el espacio cultural, en proceso de restauración, se ha consolidado como un referente en la promoción de actividades artísticas y culturales que enriquecen la vida comunitaria.

El área de cultura busca, a la par de su proceso de restauración, promover actividades artísticas y culturales. Por ello la dependencia de metodologías manuales limita la eficiencia operativa, afectando el alcance necesario al público al que se direcciona la Casa Cultural.

Marco teórico

Administración de eventos culturales

En la gestión de eventos culturales en la actualidad se requiere el uso de plataformas que faciliten procesos, como un calendario de eventos, reserva de espacios y el control de inventarios. Existen herramientas especializadas, como calendarios interactivos y sistemas



de gestión ampliamente utilizados en diversos sectores. Sin embargo, estas soluciones presentan limitaciones cuando se aplican a espacios culturales, ya que no están diseñadas para atender a necesidades específicas como la personalización de roles o la integración de diferentes procesos operativos en un solo sistema.

Para (Castaño & Francisco, 2023), el uso de entornos digitales en la difusión cultural permite una mejora significativa en la comunicación y la interacción con los usuarios, contribuyendo así a la promoción de eventos y facilitando el acceso del público objetivo. Sin embargo, muchas plataformas no logran integrar de manera eficiente el manejo de usuarios, la gestión de inventarios y gestión de reservas en un sistema (SafetyCulture, 2024).

En Ecuador, no se encuentran App Web que integren en un sistema la gestión de procesos culturales. Una herramienta encontrada es el “Sistema de Información del Patrimonio Cultural Ecuatoriano” [SIPCE] centrada en la catalogación y consulta de bienes culturales, sin abordar de forma íntegra la gestión operativa de casas culturales.

Eventos Culturales

Son un mecanismo propio del manejo cultural de las diversas expresiones artísticas. Indica (Marujo, 2015), estos eventos, conocidos como festivales tradicionales, juegan un papel decisivo en la consolidación de la relación entre la comunidad y su entorno, promoviendo el diálogo y fortaleciendo el sentido de pertinencia de la comunidad.

Arquitectura Cliente-Servidor

Es un intercambio donde el cliente recibe la información y el servidor aloja todos los recursos. Según (Sepulveda, 2014) en este modelo, el cliente solicita los recursos mientras el servidor procesa aplicaciones y respuestas (Ilustración 1).

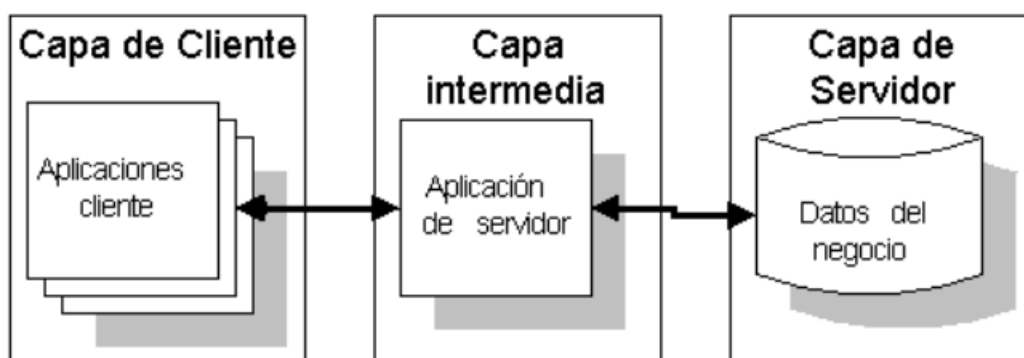


ILUSTRACIÓN 1.- Arquitectura Cliente/Servidor de tres capas (Gaviria & Buitrago, 2018), cada una de estas capas (presentación, aplicación, datos) forman un conjunto de esta arquitectura que es utilizado para el manejo de sistemas operativos como el de la presente investigación.

Métodos y Herramientas de desarrollo de software

Metodología

FURPS+

El modelo FURPS (ver Ilustración 2), creado por Hewlett-Packard (HP) en 1987, es un marco de calidad de software que analiza y controla la calidad de un producto en su ciclo de vida (Murquincho & Valla, 2024). FURPS+, versión extendida del modelo principal, estructurado para proporcionar una clasificación detallada de los requisitos. Esta extensión cubre el modelo básico con los factores de funcionalidad, facilidad de uso, confiabilidad, rendimiento y soporte; además incluye nuevas categorías como es diseño, implementación, interfaz y limitaciones de requisitos físicos (Loaiza-Barrios, 2024), mejorando la evaluación y gestión de los requisitos del software, asegurando un desarrollo estructurado y alineándose con las necesidades del usuario y del entorno tecnológico.

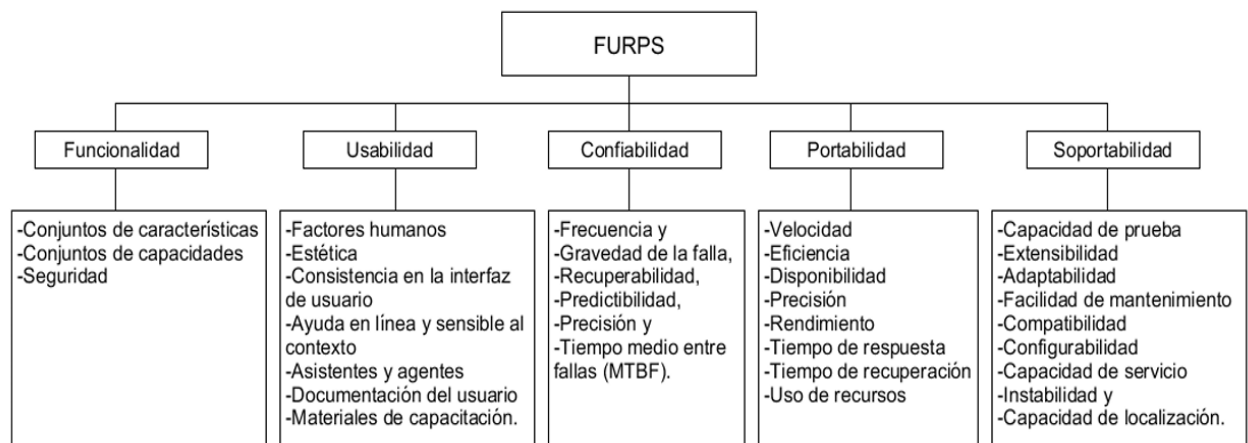


ILUSTRACIÓN 2.- Modelo FURPS (Dueñas Viviana, 2019) estructura que evalúa los factores del desarrollo, garantizando la calidad del software en función de requerimientos tanto funcionales como no funcionales.

AUP

Proceso Unificado Ágil [AUP, por sus siglas en inglés] la aplicabilidad del modelo de desarrollo, combina la flexibilidad y rapidez de los enfoques ágiles con la estructura del Proceso Unificado de Rational (RUP) como se muestra en la Ilustración 3. Ofrece un marco de trabajo equilibrado, permitiendo adaptarse a los cambios a su vez mejorando la eficiencia, sin perder la escalabilidad en el proceso. AUP mantiene los principios fundamentales de RUP, con un enfoque simple, permitiendo ciclos de desarrollo más efectivos y diligentes (Consulting, 2023).

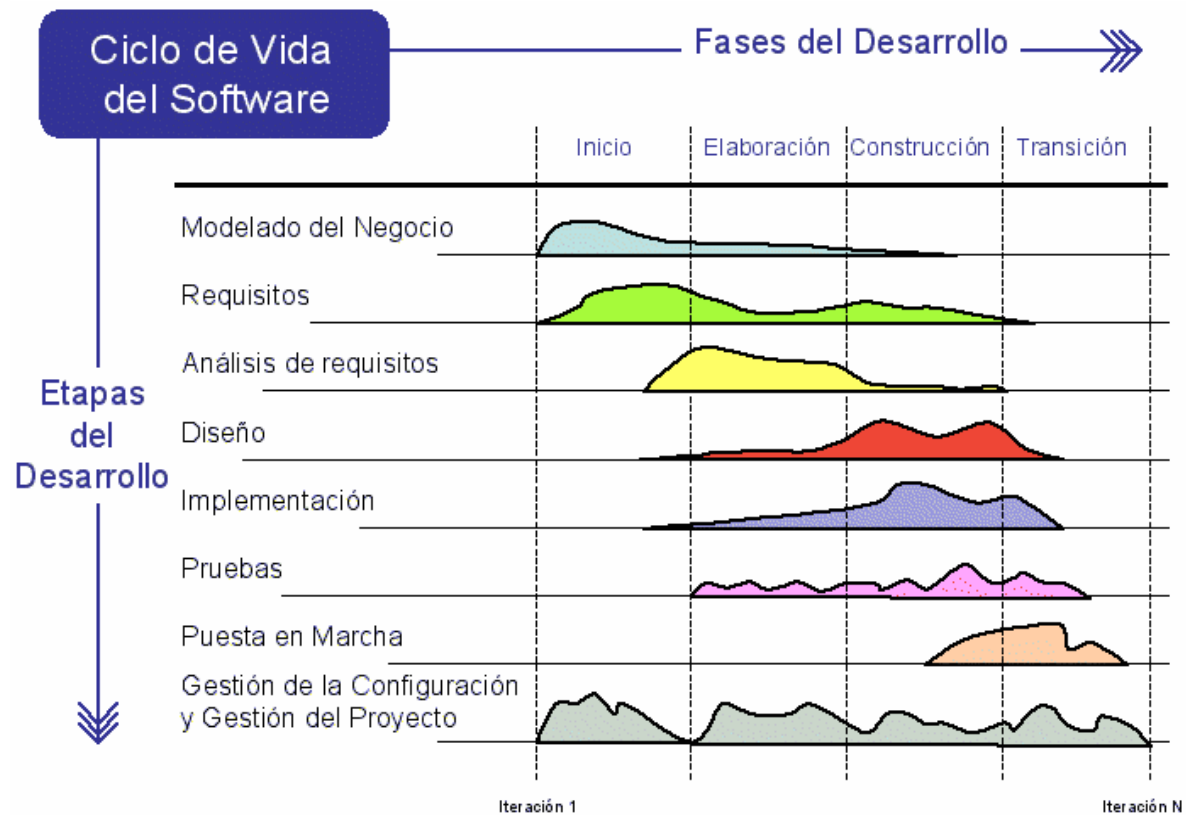


ILUSTRACIÓN 3.- Modelo conceptual de las fases que componen el Proceso Unificado Ágil [AUP] con los flujos de trabajo. Al finalizar la fase inicial, presenta los objetivos, definiendo el alcance del proyecto, al concluir la fase de desarrollo, se obtiene la arquitectura del sistema. Cada iteración de la construcción proporciona un producto que incluye las

funcionalidades previas junto con las nuevas actualizaciones de esa iteración. De este modo, finalizando la construcción tenemos todas las capacidades operacionales, es decir, con toda la funcionalidad requerida. Finalmente, al concluir la fase de transición, se entrega un producto totalmente funcional (Gomez, Cervantes, & González, 2019).

Herramientas de software

React

Librería de JavaScript empleado en la creación de interfaces de usuario (UI). Para (Degni, 2023), Desarrollado por Facebook permite el desarrollo de aplicaciones de alto rendimiento como Instagram y Airbnb, enfocado en utilizar componentes reutilizables ha permitido dar el correcto mantenimiento de las aplicaciones desarrolladas con el mismo.

Node. js

Es un entorno de ejecución de JavaScript basado en eventos, diseñado para construir aplicaciones de red escalables (OpenJS foundation, 2022). Su naturaleza asíncrona le permite controlar de manera efectiva las aplicaciones simultáneas, dando la capacidad de Node.js de introducir un cliente -servidor en un entorno cultural.

SQL Server

Sistema de gestión de base de datos relacional que permite almacenar y consultar información de forma estructurada. Según (West & Ray, 2025), este permite la interacción con la base de datos mediante consultas Transact-SQL (T-SQL).

CSS (Cascading Style Sheets)

Estándar utilizado en la definición de la presentación visual de los documentos web, las hojas de estilo en cascada controlan la apariencia de la página con la configuración de colores, fuentes y disposición de los elementos, asegurando una experiencia de usuario intuitiva y atractiva en plataformas digitales (Saavedra, 2023).

Discusión y Resultados

Discusión

Aplicación de metodología AUP y FURPS+ en el Desarrollo de software

En la primera etapa del proceso de desarrollo de software [AUP], se ejecuta el estándar de calidad FURPS+ encaminado a la elicitación de requisitos [Funcionalidad (f): Requisitos



funcionales], que se desglosa a continuación. A lo largo del proceso de desarrollo se controla la calidad del producto [requisitos No funcionales URPS+] de software.

Fase de Inicio. – Elicitación requisitos

Funcionalidad (F).

1. CU 001. - El sistema debe permitir la gestión de reservas en tiempo real, mostrando detalles de las salas disponibles. Estos datos se obtendrán desde una base de datos SQL a través de una API desarrollada en Node.js y SQL Server.
2. CU 002. - Los administradores tendrán acceso exclusivo a un módulo de inventarios para consultar, registrar y actualizar recursos como mobiliario y bienes históricos.
3. CU 003. - Registro y autenticación de usuarios con roles diferenciados: Administrador; Acceso total a los recursos de gestión de reservas, inventarios y calendarios. Usuario externo; Visualización del calendario de eventos culturales, consultar salas, realizar reservas y registrarse en la App.
4. CU 004. - Visualización gráfica de un croquis interactivo para identificar la ubicación de las salas dentro de la Casa Cultural.
5. CU 005. - Calendario de eventos editable con su respectivo CRUD.
6. CU 006. - Interfaz gráfica desarrollada en React, con componentes modulares que facilitan la navegación.

Usabilidad (U).

1. Integración con REST APIs para la gestión de datos y comunicación entre el Front End y el Back End.
2. Respuesta inmediata para confirmar reservas y en caso de haber errores notificarlos.
3. Facilidad en el registro de usuarios, visualización del croquis y reservas mejorando la experiencia de usuarios externos.

Confiabilidad (R):

1. Seguridad mediante Webpack [utilizado en el framework de desarrollo de React, para empaquetar sus archivos] y control de acceso según los roles definidos.
2. Recuperación de datos en caso de existir fallos en el sistema.

Rendimiento (P):

1. Eficiencia en el procesamiento de datos para soportar consultas simultáneas.



2. Sistema escalable para la creciente demanda que tenga la App.

Soporte (S):

1. Documentación del desarrollo del sistema para su mantenimiento.
2. Reunión con los departamentos encargados para visualización de su funcionamiento.

Requerimientos Adicionales (+):

1. Integración con plataformas externas a través de APIs RESTful en caso de ser requerido.
2. Incorporación de nuevas funcionalidades en versiones futuras dado la flexibilidad del sistema.

Con la especificación de requisitos funcionales y no funcionales se elabora en el lenguaje de programación [TypeScript y JavaScript] la arquitectura de la App Web, que se muestra en la Ilustración 4, misma que indica como los usuarios interactúan con la App Web [frontend, desarrollada con React]. Los usuarios externos acceden al sistema para utilizar los servicios, mientras que los administradores gestionan y supervisan el sistema.

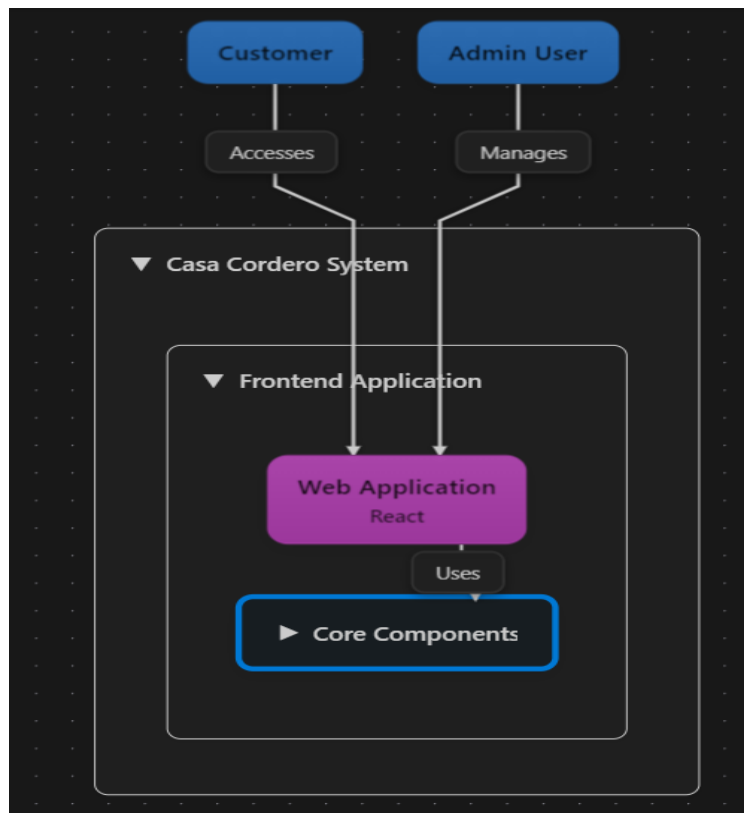


ILUSTRACIÓN 4.- Interacción de usuarios en la arquitectura cliente - servidor. El diagrama indica como todos los usuarios interactúan con la aplicación. Los usuarios externos acceden al sistema para utilizar los servicios, mientras que los administradores gestionan y supervisan el sistema. La aplicación frontend, desarrollada con React, da como ingreso a los diferentes tipos de usuarios apoyándose en componentes principales para el correcto funcionamiento.

Fase de Elaboración

En esta etapa se diseñó el sistema, la App Web adopta la arquitectura cliente – servidor permitiendo desglosar los módulos de los componentes, al lado del cliente [navegador web] y al lado del servidor [servidor web, base de datos y otros servicios propios http]. Como entregables, se realizó la integración de herramientas de desarrollo en el FrontEnd [React, TypeScript, JavaScript, CSS y HTML, json] y en el BackEnd [Node.js, Express, SQLServer, Postman, JMeter, vizcode] además; se elaboró el modelo de datos, se generó el bosquejo de las interfaces de usuarios, en la Ilustración 5, se muestra los entregables generados en la etapa de construcción.

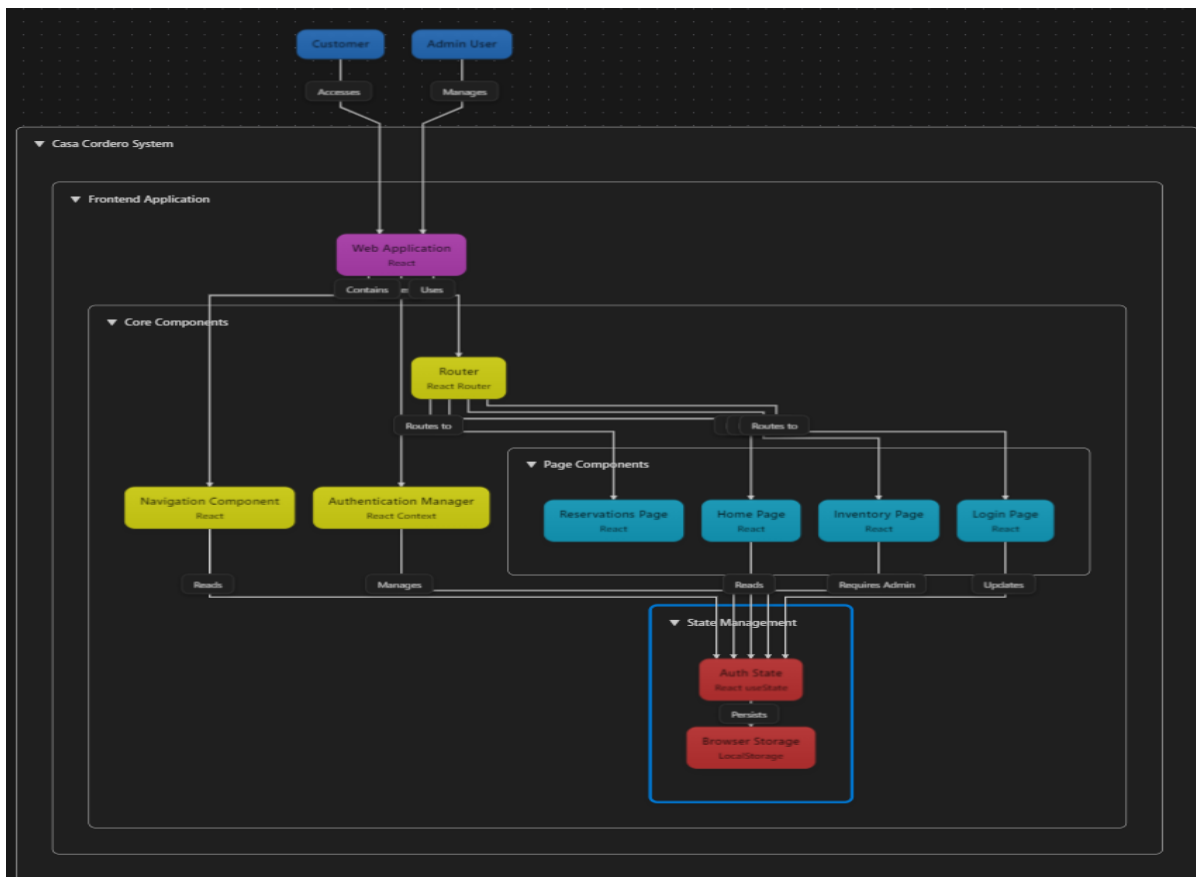


ILUSTRACIÓN 5.- Diagrama del sistema web implementado con React, incluyendo roles de usuario, autenticación, navegación dentro de las páginas. Los componentes centrales están estructurados para permitir la interacción fluida entre los usuarios, administradores y las funcionalidades del sistema.

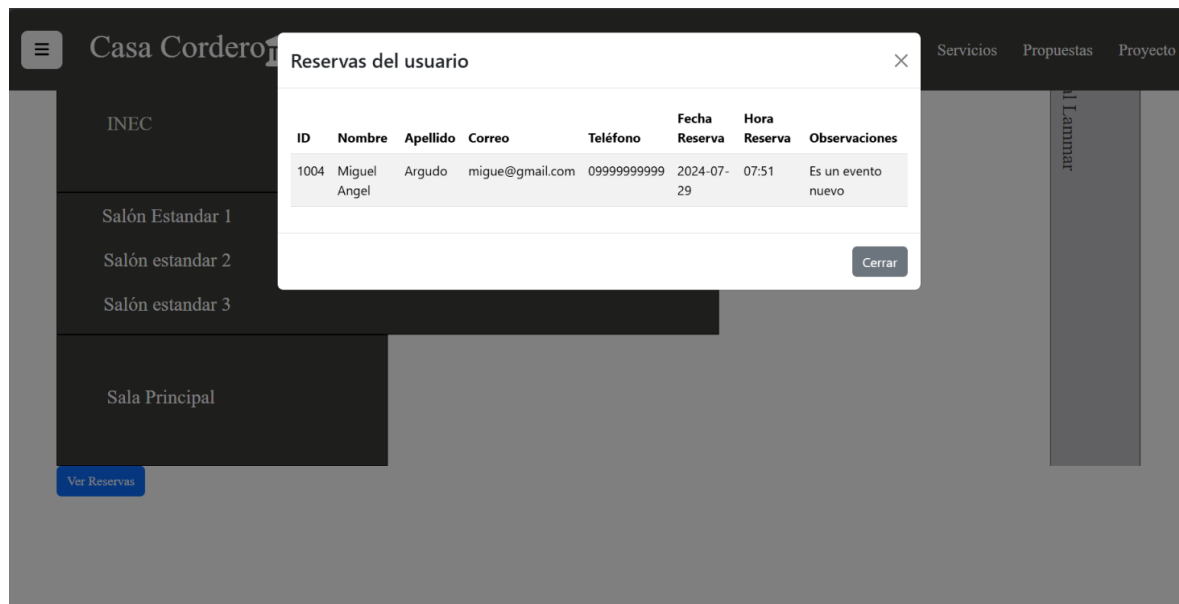
Se implementó una API con Node.js y Express para gestionar la lógica del servidor y conectarse a la base de datos SQL.

Fase de Construcción

Aquí se priorizaron la creación de interfaces dinámicas y accesibles (como se observa en Ilustración 5). El desarrollo se enfocó en iteraciones cortas, permitiendo entregas funcionales completas al final de cada ciclo.

Primera iteración. – Se desarrolla el FrontEnd utilizando herramientas [React, TypeScript, JavaScript, CSS y HTML] de creación de software, esto permitió la reutilización de los componentes. Como por ejemplo la página principal fue diseñada con una estructura basada en bloques organizados de acuerdo con los requisitos definidos. Para agilizar la gestión del material visual, se creó una carpeta para guardar imágenes y archivos multimedia dentro del proyecto.

Segunda iteración. - Se desarrolló el módulo de reservas, en el que los usuarios consultan la disponibilidad de las salas y a su vez realizan las reservas de estas. Los formularios de cada

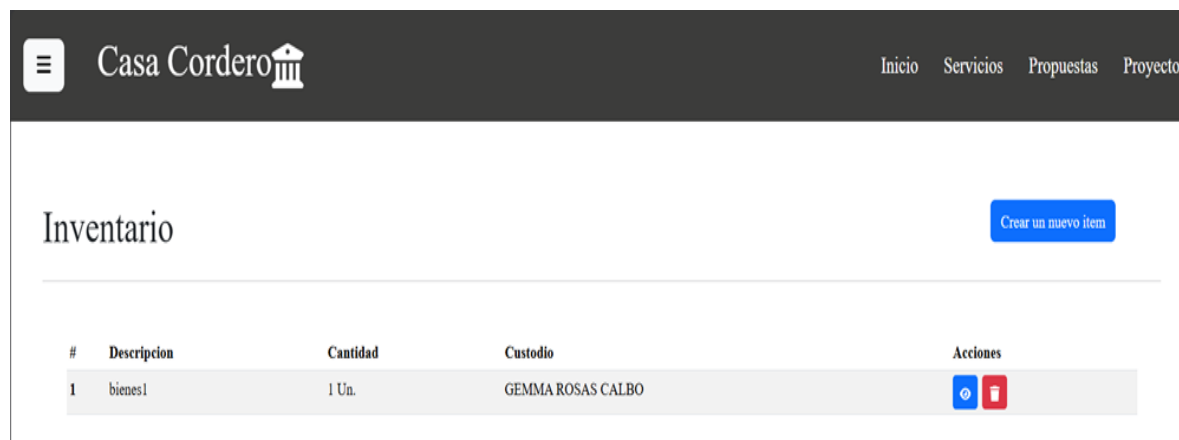


sala se enlazan a la base de datos con información detallada, facilitando la administración y consulta en tiempo (Ilustración 6).

ILUSTRACIÓN 6.- Visualización del modal de las reservas del usuario donde el usuario puede ver sus reservas activas dentro de la AppWeb al ingresar con sus datos.

Tercera iteración. - En esta iteración, se desarrolló el módulo de gestión de usuarios, estableciendo dos niveles de acceso: administrador y usuario externo.

Cuarta iteración. - Se creó el módulo de inventarios restringido sola a administradores, el sistema permite registrar, actualizar y eliminar los bienes patrimoniales como se observa en la Ilustración 7.



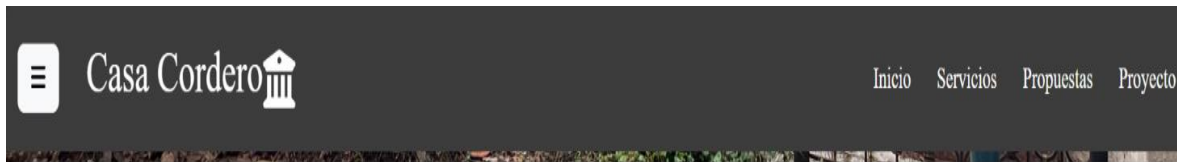
#	Descripción	Cantidad	Custodio	Acciones
1	bienes	1 Un.	GEMMA ROSAS CALBO	 

ILUSTRACIÓN 7.- El apartado de inventarios con el respectivo CRUD de los inventarios a manejar por parte de la Casa Cultural.

Quinta iteración. - En esta iteración, se implementó un calendario interactivo integrado en la interfaz de la página principal donde los administradores pueden realizar el CRUD de los eventos a desarrollarse en la Casa Cultural, teniendo un correcto direccionamiento a los usuarios externos que deseen asistir a eventos culturales de su interés (Ilustración 8).

ILUSTRACIÓN 8.- Apartado de calendario donde se mostrarán los eventos organizados dentro de la Casa Cultural, gestionado directamente por el administrador diferenciando del usuario externo que solo podrá visualizar los eventos a realizarse.

Fase de Transición



Calendario de eventos

+ Agregar evento

Hoy	Anterior	Siguiente	febrero 2025				Mes	Semana	Día	Agenda
lun.	mar.	mié.	jue.	vie.	sáb.	dom.				
27	28	29	30	31	01	02				
		examen								
03	04	05	06	07	08	09				
10	11	12	13	14	15	16				
17	18	19	20	21	22	23				
24	25	26	27	28	01	02				

Para el despliegue del producto [App Web: Casa Cordero], las iteraciones fueron sometidas a pruebas automatizadas [Pruebas Unitarias con Jest, Pruebas de Integración con Postman, Pruebas de Rendimiento JMeter y Pruebas de Seguridad con Webpack] logrando una aplicación robusta alineada con los requerimientos de la Casa Cultural, esto se observa en la Ilustración 9.



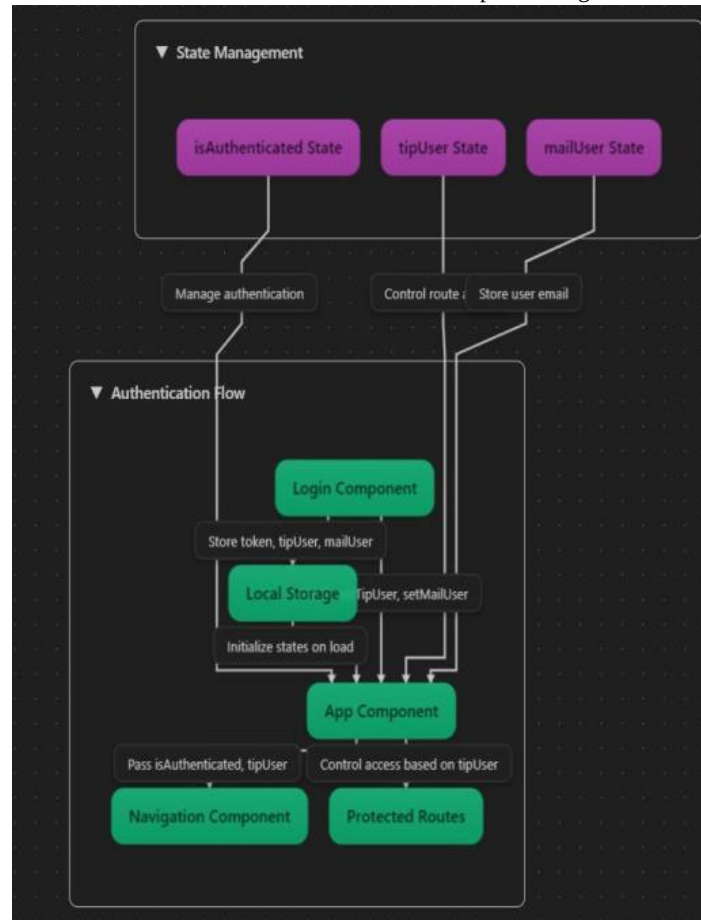


ILUSTRACIÓN 9.- Podemos visualizar el sistema de gestión de usuario, que realiza la autenticación respectiva para poder visualizar las funcionalidades direccionadas en cada rol. En el proceso del desarrollo del software se identificaron varios desafíos técnicos que requirieron una mayor precisión al momento de su resolución. Uno de ellos fue la integración del backend, este mismo fue desarrollado con Node.js, Express y una base de datos en SQL, estas herramientas garantizan una correcta comunicación entre la API y el frontend para visualizar los datos en tiempo real. Se implementaron técnicas de optimización en las consultas a la base de datos mejorando la respuesta del sistema.

La seguridad del sistema y la autenticación de usuarios fue otro desafío importante, ya que se debía asegurar que las funciones específicas estuviesen restringidas según los roles de usuario, para ello el uso de Webpack desempeñó un papel crucial para la configuración del frontend integrando la seguridad con la compilación de los módulos de JavaScript.

Se organizó una reunión con el departamento de TI para definir dónde se va a alojar la aplicación, en ello se acordó dar el acceso respectivo al servidor institucional donde está anexada la Casa Cultural entregándolo con la respectiva documentación para dar el mantenimiento y modificaciones pertinentes de ser el caso, permitiendo escalar el sistema a futuro.

En la entrega final del proyecto será administrada por el departamento de comunicación en conjunto con el departamento de TI, esto permitirá que las personas a cargo puedan modificar imágenes, información y todo lo que sea necesario que sea afín a las directrices de estos departamentos. Adicionalmente, cabe acotar que, el departamento al dar acceso al servidor con las respectivas credenciales, supo indicar que no cuentan con la licencia de SQL Server, por lo tanto, se procedió a migrar la base de datos a MariaDB, que es una base de datos libre que permite alojar en el servidor proporcionado.

Al ver que es un proyecto escalable, se sugiere que se permitan futuras versiones, mantenimiento y mejoras del sistema por estudiantes de la carrera en la que se originó el proyecto. Esto garantiza la incorporación gradual de nuevas tecnologías y mejoras para la evolución del sistema en función de las necesidades de la Casa Cultural.



ILUSTRACION 10.- Reunión con los departamentos encargados de administrar y proveer el servidor [Departamento de TI] donde se va alojar la App Web.

Resultados

La implementación de la aplicación web ha generado mejoras clave que integran los servicios del espacio cultural.

Los servicios de la Casa Cultural, como reservas, inventarios y programación de eventos, se administran ahora desde una plataforma centralizada.

Se automatizaron los procesos, reemplazando las actividades manuales como el inventario por flujos digitales, agilizando los procesos administrativos.

Los usuarios pueden registrarse en la App Web, lo que les permite participar en eventos interactuando con la comunidad cultural de una manera más sencilla.

El calendario permite informar a la comunidad sobre los eventos a realizarse, teniendo un mayor acercamiento y difusión entre los usuarios que accedan a la aplicación.

Como parte del proceso de validación de resultados obtenidos en las pruebas mencionadas, se realizó un proceso de refactorización, garantizando la corrección y la sostenibilidad de la aplicación

Optimización de consultas en la base de datos para mejorar el tiempo de respuesta al modificar, crear y eliminar información de reservas, inventarios y en el calendario.

Mejoras en el backend mediante la estructuración del código en capas de servicios y controladores, asegurando una asignación correcta de responsabilidades.

Conclusiones

1. Al inicio del proyecto, se llevaron a cabo reuniones y entrevistas con la directora de la Casa Cultural y el equipo administrativo, esto con el fin de obtener información detallada sobre los procesos, los requerimientos y las expectativas para mejorar la gestión del espacio cultural.
2. En las interacciones que se realizaron se definieron con precisión los requisitos funcionales y no funcionales, para poder proceder con el desarrollo del sistema. Adicionalmente, se recopiló la información como material gráfico, que incluyó la recopilación de imágenes de recorridos previos realizados por estudiantes y



documentación sobre futuras remodelaciones del espacio.

3. El desarrollo se realizó con un enfoque iterativo, donde se validó cada fase del proyecto para asegurar el correcto funcionamiento y su alineación con las necesidades de la Casa Cultural.
4. Se realizaron pruebas unitarias donde se aseguraron de que los módulos funcionarán correctamente, pruebas de integración para validar la interacción entre los componentes del sistema, y pruebas funcionales para comprobar que los requisitos se cumplieran de manera adecuada. Además, se expuso el sistema al equipo administrativo para validar la usabilidad, donde se constató que la interfaz sea intuitiva y amigable para el usuario.
5. Se detectaron errores en el módulo CRUD (Creación, Lectura, Actualización y Eliminación) de ciertos componentes, los cuales fueron corregidos en las siguientes iteraciones del desarrollo.

Recomendaciones

1. Se sugiere implementar evaluaciones del rendimiento de la aplicación una vez alojada en el servidor, esto permitirá detectar posibles problemas de rendimiento y realizar los ajustes necesarios conforme aumente el número de usuarios y solicitudes.
2. Aunque la plataforma fue diseñada para cubrir las necesidades actuales de la Casa Cultural, se recomienda planificar mejoras escalables. En este sentido, será necesario ajustar la estructura del sistema para gestionar nuevos requerimientos y funcionalidades.
3. Adicionalmente, la integración de herramientas de análisis de datos en la plataforma podría ser útil para proporcionar información detallada sobre eventos, reservas y actividades que se realicen en la Casa Cultural. Esto facilitará la toma de decisiones al momento de realizar nuevas actividades y el manejo de recursos.
4. Se sugiere la implementación de un diseño responsivo que garantice una óptima experiencia de usuario en dispositivos móviles.
5. Es recomendable desarrollar módulos adicionales, lo que permitirá recordar a los usuarios sobre eventos próximos y la disponibilidad de salas.
6. Por último, se propone la integración con sistemas externos, como herramientas de recorrido virtual y pasarelas de pago para la reserva de salas, con el fin de ampliar las funcionalidades de la plataforma.



Referencias

- Alhir, S. S. (2023). El proceso unificado ágil (AUP). *METHODS AND TOOLS*.
- Castaño, J. F., & Francisco, L. R. (2023). EL USO DE LAS PLATAFORMAS DIGITALES EN LA DIVULGACIÓN DE. *researchgate*.
- Consulting, D. (2023). *Dhrama Consulting*. Obtenido de <https://dharmacn.net/2023/10/18/encontrando-el-equilibrio-introduccion-al-proceso-unificado-agil/>
- Degni, R. (28 de Julio de 2023). *Una completa introducción a la librería React*. Obtenido de Codemotion: <https://www.codemotion.com/magazine/es/lenguajes-de-programacion/una-completa-introduccion-a-la-libreria-react/>
- Dueñas Viviana, P. B. (mayo de 2019). Metodología para la validación de arquitecturas cognitivas bio-inspiradas. Jalisco, Guadalajara, México: Universidad de Guadalajara.
- Gaviria, V. R., & Buitrago, A. A. (2018). *Implementación de infraestructuras tecnológicas (GIS) Geographic Information System, expresada en servicios como patrón de diseño*. Pereira: Universidad Libre Seccional Pereira.
- Gomez, F. M., Cervantes, O. J., & González, P. P. (2019). *Fundamentos de Ingeniería de Software*. México D.F.: Litoprocess S. A. de C.V.
- Loaiza-Barrios, A. G.-S.-R. (2024). Monitoreo del flujo de personas en tiempo real y vision de computadora.
- Lujan Mora, S. (2002). *Programación de aplicaciones web: historia, principios básicos y clientes web*. Alicante: Editorial Club Universitario.
- Manobanda Ushca, A. B. (2020). MODELO FURPS PARA EVALUAR EL SISTEMA DE RECAUDACIÓN DE PATENTES GADM PENIPE. RIOBAMBA: Universidad Nacional de Chimborazo.
- Marujo, N. (2015). EVENTOS CULTURALES Y MOTIVACIONES DE LOS TURISTAS La Fiesta de Fin de Año en Isla de Madeira - Portugal . *Estudios y Perspectivas en Turismo*, 24(1).
- Murquincho, L. F., & Valla, J. D. (2024). *Microservicios para el módulo de proyectos del sistema de Gestión de Investigación de la UNACH*. Riobamba.



- OpenJS fundation. (2022). *NodeJS*. Obtenido de <https://nodejs.org/es/about>
- Saavedra, J. A. (3 de Mayo de 2023). *Escuela Británica de artes creativas y tecnología*. Obtenido de EBRAC: <https://ebac.mx/blog/que-es-css>
- SafetyCulture. (10 de Febrero de 2024). *Safety Culture*. Obtenido de https://safetyculture.com/es/aplicacion/software-de-gestion-de-inventario/?utm_source=chatgpt.com
- Sepulveda, I. (2014). *Arquitectura Cliente/Servidor*. Costa Rica: Uniciencia.
- West, R., & Ray, M. (02 de 01 de 2025). *Microsoft*. Obtenido de Microsoft Learn: <https://learn.microsoft.com/es-es/sql/sql-server/what-is-sql-server?view=sql-server-ver16>

Conflicto de intereses:

Los autores declaran que no existe conflicto de interés posible.

Financiamiento:

No existió asistencia financiera de partes externas al presente artículo.

Agradecimiento:

N/A

Nota:

El artículo no es producto de una publicación anterior.

