



Comparación entre herramientas No-Code/Low-Code para desarrollar un frontend para un backend ya implementado

Juan Pablo Aragón Alzate

Anteproyecto presentado como requisito parcial para optar al título de:
Magister en Ingeniería de Software

Director(a):
MSc Juan Pablo García Cifuentes

Pontificia Universidad Javeriana Cali
Facultad de Ingeniería
Departamento de Electrónica y Ciencias de la Computación
Cali, Colombia
7 de agosto de 2024

Ficha Resumen

Anteproyecto de Trabajo de Grado

Posible Título: Comparación entre herramientas No-Code/Low-Code para desarrollar un frontend para un backend ya implementado

1. Área de trabajo: Ingeniería de Software
2. Tipo de proyecto: Aplicado
3. Estudiante: Juan Pablo Aragón Alzate
4. Correo electrónico: jpragon2021@gmail.com
5. Dirección y teléfono: calle 34 #96-79 3043393989
6. Director: Juan Pablo Garcia Cifuentes
7. Vinculación del director: 13/03/2024
8. Correo electrónico del director: jpgarcia@javerianacali.edu.co
9. Palabras clave: Low-Code, No-Code, Gestión de reservas, Desarrollo Agil, Backend, Front-end
10. Fecha de inicio: Una vez sea aprobado el anteproyecto
11. Duración estimada: 6 meses
12. Resumen: El trabajo aborda la necesidad de implementar un sistema de gestión de reservas para clubes de tenis debido al aumento en la demanda de canchas y los desafíos asociados con la gestión manual de reservas. Se describe cómo la falta de acceso en tiempo real a la información y la asignación manual de canchas afectan la experiencia del usuario y los ingresos del club. Se propone como alternativa el uso de herramientas Low-Code No-Code para el desarrollo frontend de la aplicación, con el objetivo de mejorar la accesibilidad, eficiencia

y experiencia de los usuarios, y de lanzar al mercado un producto robusto en poco tiempo y a bajo costo. Las preguntas planteadas incluyen la identificación de herramientas disponibles en el mercado, la evaluación de criterios para seleccionar una herramienta y cómo desarrollar un frontend para una aplicación con backend ya desarrollado. Los objetivos del proyecto son diseñar y comparar la implementación de dos interfaces de usuario (frontend) utilizando herramientas Low-Code No-Code, conectándolas con un sistema de gestión (backend) desarrollado de manera convencional. Se espera definir criterios para evaluar el desarrollo frontend, tomar una decisión sobre la mejor herramienta, crear un prototipo funcional de la aplicación y recopilar comentarios de usuarios finales. En resumen, el trabajo busca abordar los desafíos de gestión de reservas en clubes de tenis mediante el uso de herramientas Low-Code No-Code, con el objetivo de mejorar la experiencia del usuario y la eficiencia operativa del proceso, a su vez evaluando los beneficios y limitantes del uso de estas herramientas.

Índice

1. Introducción	6
2. Definición del problema	8
2.1. Planteamiento del problema	8
2.2. Formulación del problema	9
3. Objetivos del proyecto	10
3.1. Objetivo General	10
3.2. Objetivos específicos	10
3.3. Resultados esperados	10
4. Alcance	11
5. Justificación del trabajo de grado	12
6. Marco teórico de referencia y antecedentes	14
6.1. Bases Teóricas	14
6.2. Estado del Arte	17
7. Metodología de la investigación	23
8. Recursos a emplear	24
9. Cronograma de actividades	25
10. Referencias Bibliográficas	26

Índice de figuras

1. Planeación Sprints. Elaboración propia	24
2. Cronograma de actividades. Elaboración propia	25

Resumen

Las herramientas y métodos para la construcción de aplicaciones ha ido evolucionando a través de los años. Actualmente las herramientas Low-Code No-Code son tendencias debido a las facilidades que ofrecen para la construcción de aplicaciones robustas. En el presente proyecto se busca construir una aplicación para la administración de reservas de canchas de tenis haciendo uso de estas herramientas Low-Code No-Code y evaluarla bajo los criterios con la que se evalúa una aplicación desarrollada de manera convencional para ver el verdadero beneficio y limitantes de su uso.

Palabras Clave Low-Code, No-Code, Aplicación de reservas

Abstract

The tools and methods for building applications have evolved over the years. Currently, Low-Code No-Code tools are trending due to the ease they offer in constructing robust applications. In the present project, the aim is to build an application for managing tennis court reservations using these Low-Code No-Code tools and evaluate it against the criteria used to evaluate conventionally developed applications to assess the true benefits and limitations of their use.

Keywords Low-Code, No-Code, Appointment Manager app

1. Introducción

Hace no mucho tiempo atrás, para desarrollar una aplicación, por más sencilla que fuera, se requería no solo de contar con un equipo especializado en el tema, lo que de por sí es costoso, sino también de un tiempo mínimo para poder llevar la aplicación al mercado. Este tiempo mínimo podía generar que competidores sacaran una solución antes al mercado, impactando de forma negativa el posicionamiento y la participación de la aplicación. De igual forma, ante los elevados costos de llevar a cabo un desarrollo, muchas ideas podían quedar rezagadas por el simple hecho de no contar con los recursos suficientes para salir al mercado.

En este contexto, gracias a los avances tecnológicos, han aparecido herramientas que tienen como premisa que no es necesario tener conocimientos avanzados en ingeniería de software para desarrollar una aplicación. Haciendo uso de las tecnologías Low-Code y No-Code cualquier idea podría llevarse al mercado en un tiempo y conocimiento inferior al necesario de hacerlo de forma convencional, a código puro.

Cuando hablamos de Low-Code nos referimos a una tecnología que proporciona un entorno no solo para desarrollar si no también para desplegar aplicaciones escribiendo muy pocas líneas de código. Por otro lado, en el caso de No-Code, la tecnología va un poco más allá y nos proporciona de igual manera un entorno para desarrollar y desplegar una aplicación, pero sin requerir una sola línea de código.

El crecimiento de este tipo de tecnologías, que para el 2023 era de 26.9 billones de dólares según Gartner (2022), ha generado un gran impacto no solo en la industria tecnológica, cambiando la forma en la que se entiende el desarrollo de software, si no a nivel global, llegando incluso a pensar que podrían sustituir el proceso actual del desarrollo de software.

La simplicidad a la que estas herramientas han reducido el desarrollo del código genera múltiples beneficios, entre los que se encuentra la reducción del tiempo de desarrollo en casi un 90% según Redhat (2018), lo que permite llevar ideas más rápido y económicas al mercado, así como probar conceptos sin la carga de una gran inversión económica detrás.

De igual forma, la curva de aprendizaje para una tecnología Low-Code No-Code

es mucho mas corta comparada con la del desarrollo tradicional. Esta facilidad ha generado que cada vez más empresas pequeñas o también personas sin mucho presupuesto o conocimiento puedan sacar ideas disruptivas al mercado, algo que era casi imposible 10 años atrás.

Teniendo como base los beneficios ya mencionados y el crecimiento del uso de estas tecnologías, en el presente trabajo se busca utilizar las bondades de las herramientas Low-Code No-Code para desarrollar una aplicación de gestión de reservas de cancha de tenis. Se utilizarán dos tecnologías para desarrollar la interfaz de usuario, ambas conectándose a un Backend ya desarrollado, y se compararán bajo unos criterios específicos.

La aplicación de gestión de reservas de cancha de tenis tiene como objetivo primordial mejorar la experiencia de los usuarios del club al permitirles consultar la disponibilidad en tiempo real, eliminando la necesidad de procesos manuales como llamar al administrador del club. Además, facilitará la gestión integral de reservas al permitir a los usuarios crear, modificar y eliminar reservas de manera eficiente. Asimismo, se enfocará en asignar automáticamente las canchas a las reservas, con el objetivo de minimizar la división de una reserva entre múltiples canchas y optimizar así la experiencia de juego de los usuarios.

2. Definición del problema

2.1. Planteamiento del problema

Según El País (2023), al 2023 el tenis era el tercer deporte más popular a nivel mundial con alrededor de 1000 millones de espectadores y alrededor de 300 millones de practicantes. En Colombia, si bien es un deporte ampliamente practicado por una parte de la población, al no ser tan popular y económicamente accesible, en su mayoría son clubes privados los que ofrecen el servicio de alquiler de canchas, ya sea solo para socios o para el público en general.

Esta popularidad, creciente a nivel nacional desde la pandemia del 2021 según lo conversado con un administrador de un club en la ciudad de Cali, ha llevado a un aumento significativo en la demanda de canchas de tenis. Sin embargo, muchos de los clubes de tenis regionales, siguen administrando de forma manual las reservas de las instalaciones, generando así una serie de desafíos que afectan tanto a los usuarios como a la eficiencia operativa del proceso.

La necesidad de comunicarse con un funcionario para conocer la disponibilidad de las canchas ha creado un proceso burocrático y lento que en muchas ocasiones desincentiva a los usuarios de generar una reserva. Los usuarios deben consultar con tiempo porque al no saber en cuanto tiempo obtendrán respuesta puede darse el caso en donde el funcionario responda ya sobre la hora de la reservada deseada. De igual forma, si un usuario está interesado en una hora en particular que en cierto momento no se encuentra disponible y en algún momento se libera el espacio, no hay un mecanismo para notificar al usuario y asignarle el espacio. Como se puede observar, la falta de acceso a la información en tiempo real impacta de forma directa el número de reservas generadas en un club y por consiguiente afecta los ingresos de este.

Adicionalmente, la asignación manual de las canchas presenta desafíos logísticos que pueden afectar la experiencia de usuario, con la posibilidad de generar solapamiento de reservas o una distribución ineficiente de las reservas que limite la utilización de las instalaciones, un recurso valioso en el contexto de que es limitado y es

el que genera los ingresos representativos del club. Ante este escenario, es casi que indispensable la implementación de un sistema de gestión de reservas para mejorar la accesibilidad, eficiencia y experiencia de los usuarios en el club de tenis.

Llevar a cabo el desarrollo de la aplicación de forma nativa, requeriría de un amplio conocimiento en desarrollo frontend, backend y en otras áreas de la ingeniería de software como devOps. Al ser apenas una idea que se va a probar y ajustar al mercado, no se cuenta con los recursos necesarios para contratar un equipo para el desarrollo además que se busca salir cuanto antes a producción e ir realizando cambios conforme a los comentarios de los usuarios.

Como alternativa al desarrollo convencional y teniendo como base los objetivos mencionados anteriormente, se busca utilizar herramientas Low-Code No-Code para el desarrollo frontend de la aplicación, sometiéndolas a evaluación bajo criterios a los que usualmente se someten los códigos generados de forma convencional. De esta forma se busca salir al mercado con un producto robusto, desarrollado a bajo costo y en un tiempo muy corto, sin perder de vista el alcance funcional.

2.2. Formulación del problema

¿Cuáles son las herramientas disponibles en el mercado para realizar un desarrollo frontend con Low-Code o No-Code?

¿Bajo cuales criterios se debería evaluar una herramienta Low-Code No-Code para el desarrollo frontend de una aplicación con backend ya desarrollado?

¿Cómo desarrollar un frontend para una aplicación que ya cuenta con un backend desarrollado?

¿Son las herramientas Low-Code No-Code una buena alternativa para sacar aplicaciones robustas al mercado en poco tiempo y a bajo costo?

3. Objetivos del proyecto

3.1. Objetivo General

Comparar la implementación de dos interfaces de usuario (frontend) utilizando herramientas de desarrollo Low-Code No-Code, conectándolas con un sistema de gestión (backend) desarrollado de manera convencional.

3.2. Objetivos específicos

1. Identificar dos herramientas Low-Code No-Code presentes en el mercado para el desarrollo frontend de una aplicación
2. Identificar los criterios bajo los cuales comparar las herramientas para el desarrollo frontend
3. Diseñar y desarrollar dos aplicaciones frontend conectadas con un backend ya funcional
4. Evaluar las aplicaciones bajo los criterios definidos

3.3. Resultados esperados

1. Definición de los criterios para la evaluación de un desarrollo frontend
2. Una decisión sobre la mejor herramienta, entre las utilizadas, para desarrollo Low-Code No-Code de aplicaciones frontend
3. Un prototipo funcional de una aplicación para la asignación de reservas de canchas de tenis
4. Recopilación de los comentarios de un usuario final sobre el prototipo

4. Alcance

- Solo se realizará el desarrollo frontend mediante el uso de herramientas Low-Code No-Code.
- El prototipo sera de una aplicación Web.
- El prototipo de la aplicación debe:
 1. Mostrar a los usuarios las horas disponibles para reservar una cancha en tiempo real
 2. Permitir las siguientes operaciones con una reserva:
 - a) Crear una reserva
 - b) Eliminar una reserva
 - c) Modificar una reserva
 3. Asignarle una cancha a la reserva, buscando minimizar la asignación de una reserva a múltiple canchas
 4. Mostrarle al administrador del club las reservas y horas disponibles en tiempo real
 5. Soportar hasta mínimo 30 canchas
 6. Permitir al usuario configurar el rango de tiempo en que cada cancha esta disponible

5. Justificación del trabajo de grado

En Cali son pocos los clubes de tenis que cuentan con una herramienta digital por la cual los usuarios puedan gestionar reservas. Este “atraso” tecnológico genera varios inconvenientes tanto para los usuarios como para el mismo club. Entre los inconvenientes mas comunes expresados por los usuarios esta que el proceso para una reserva se vuelve lento y burocrático, para el usuario es imposible saber las horas disponibles al consultarle al club, se pierde tiempo esperando una respuesta e inclusive esperando la confirmación de una reserva.

Del lado del club también se presentan diferentes problemas, como la insatisfacción de los usuarios por el tiempo de respuesta, la doble asignación de una reserva, el perder reservas por responder muy tarde a una solicitud y por ultimo el no tener la capacidad de asignar una reserva cancelada con muy poco tiempo de anticipación.

Si bien hay ya algunas aplicaciones en el mercado que se pueden utilizar para la gestión de reservas, estas carecen de ciertas características propias del negocio del tenis, como por ejemplo que el usuario pueda reservar mas de una hora desde una sola solicitud y que sea la aplicación la que según la disponibilidad le asigne una cancha buscando evitar generar espacio muertos entre las reservas de una cancha. De igual forma, para los usuarios es muy valioso no tener que estar revisando múltiples veces la aplicación para ver si el espacio en que querían jugar e inicialmente no estaba disponible se libera en algún momento, si no que sean notificados y se les asigne la hora automáticamente, una funcionalidad que no se encontró en ninguna aplicación que opere localmente.

Ante estos inconvenientes nace la idea de desarrollar una aplicación que permita la gestión de las reservas en tiempo real con características que se asemejen mas al comportamiento actual de los clubes de tenis. Al ser un proyecto personal, no cuenta con un gran presupuesto y se busca contar con una versión inicial cuanto antes, probarla en el mercado y con base a los comentarios recopilados realizarle mejoras y así, cíclicamente hasta dar con un producto estable y ampliamente aceptado por la comunidad del tenis. Aquí es donde las herramientas Low-Code No-Code entran a jugar un papel primordial y es la de permitir llevar estas ideas, que cuentan aun

con mucha incertidumbre, al mercado sin la necesidad de contar con un equipo de TI especializado o sin un gran presupuesto.

El presente trabajo de grado busca no solo desarrollar la aplicación apoyándose en las herramientas Low-Code No-Code, sino que también busca analizar las herramientas que ofrece actualmente el mercado y entre dos de ellas generar una comparación bajo los criterios sobre los cuales usualmente se evaluaría una aplicación desarrollada de forma convencional, dejando claro los trade-off de utilizarlas.

6. Marco teórico de referencia y antecedentes

6.1. Bases Teóricas

Plataformas Low-Code y No-Code

Vincent et al. (2020) define a las plataformas Low-Code (LCP por sus siglas en ingles) como aplicaciones que soportan el desarrollo, despliegue, ejecución y gestión rápida de aplicaciones, utilizando abstracciones de programación declarativas y de alto nivel. Por otro lado, para Hylton et al. (2021) una plataforma No-Code permite desarrollar aplicaciones sin escribir una sola línea de Código.

A nivel general, tanto las herramientas No-Code como las Low-Code ofrecen una experiencia basada en técnicas de "drag-and-drop" por encima de los lenguajes de programación tradicionales Università et al. (2023)

La utilización de estas plataformas tiene como objetivo generar un aumento de la productividad y una reducción en el costo del desarrollo y mantenimiento de una aplicación, mejorando así la habilidad de las compañías para adaptar su sistema a los constantes cambios de requerimientos por parte del mercado Bock and Frank (2021b) . Así mismo Moskal (2021) indica que dentro de los beneficios de estas plataformas también se encuentra la flexibilidad y posibilidad de desarrollar herramientas con un esfuerzo de implementación y mantenimiento menor.

Transformación digital

Para George and Paul (2020) " la transformación digital lleva a las organizaciones a contemplar la implementación de diversas iniciativas tecnológicas. Sin embargo, también implica una serie de acciones en ámbitos sociales, culturales, políticos, económicos, ecológicos y normativos"

De acuerdo con Okano et al. (2021), estas transformaciones conllevan la reconfiguración del modelo de negocio y el surgimiento de nuevas empresas. Además, en general, dan lugar al establecimiento de formas innovadoras de comerciar y comunicarse.

Según BCG (2020) las compañías pueden esperar un aumento en sus márgenes de entre 12% al 20% al utilizar palancas digitales, las compañías que se resistan a la transformación digital poco a poco se verán apartadas del mercado.

Front-end y Back-end

Amazon Web Services (2023) define el front-end como aquello que ven los usuarios, en donde se incluyen elementos como los botones, casillas de verificación, gráficos y mensajes de textos. En otras palabras es el punto de contacto que permite a los usuarios interactuar con la aplicación. Hay tres lenguajes de computación principales para el desarrollo:

1. El lenguaje HTML que define la estructura del front-end
2. La hoja de estilos CSS que define el estilo y diseño de la aplicación.
3. Lenguaje de programación, por lo general JavaScript, para agregar una capa de funcionalidad dinámica

Por otro lado el back-end, a veces denominado servidor, son los datos y la infraestructura que permite que las aplicaciones funcionen. Cuando el usuario realiza una petición por medio del front-end, el backend procesa la solicitud y devuelve una respuesta, para esto usualmente el servidor interactúa con:

1. Base de datos para la recuperar o modificar información
2. Micro servicios para la realización de tareas específicas
3. APIs externos ya sea para recopilar información adicional o realizar funciones adicionales

API Rest

Para poder entender que es una API Rest primero entremos a definir que es un API. Las API, según la documentación de Amazon Web Services (2023), son mecanismos que permiten a dos componentes de software comunicarse entre sí mediante un

conjunto de definiciones y protocolos. Ahora, teniendo claro que es una API, pasemos a la segunda parte del termino. REST significa transferencia de estado representacional y es un estilo arquitectónico para diseñar sistema de software distribuidos que fue propuesto por Roy fielding. Este hace uso de operaciones HTTP estándar como GET,POST,PUT,DELETE, para realizar acciones en recursos identificables mediante URLs.

Teniendo claro ya la definición de los términos por separado, según REDHAT (2023) “una API Rest es una interfaz de programación de aplicaciones (API o API web) que se ajusta a los límites de la arquitectura REST y permite la interacción con los servicios web de RESTful”

Productividad en el desarrollo del software

Las herramientas Low-Code No-Code prometen mejorar la productividad en el desarrollo del software. Para Varajão et al. (2023) la productividad en el ámbito de las tecnologías de desarrollo de software se puede definir como la eficiencia lograda en la producción de bienes o servicios de software. Por lo general la productividad se expresa como una medida entre la relación de las salidas, las entradas y el tiempo. El diccionario de Cambridge tiene una definición bastante clara de la productividad, la define como ”La tasa a la que una empresa o país fabrica bienes, generalmente juzgada en relación con el número de personas y la cantidad de materiales necesarios para producir los bienes” .

En su artículo, Varajão et al. (2023) mediante un experimento concluye que las herramientas Low-Code No-Code muestran un aumento considerable de la productividad sobre las herramientas basadas en código. Este impacto se traduce en menores tiempo de entrega y por lo tanto en mayor utilidades a la hora de realizar un proyecto. Por ejemplo, en su artículo, Sanchis et al. (2020) señala que según una investigación realizada por Forrester, se comprobó que las plataformas Low-Code aceleran el desarrollo entre 5 y 10 veces más rápido. Asimismo, tras comparar herramientas Low-Code y basadas en código, Trigo et al. (2022) concluye que la productividad en las herramientas Low-Code es aproximadamente tres veces mayor que en las tecnologías

basadas en código (Code-based).

6.2. Estado del Arte

A continuación se presentara primero las herramientas No-Code y Low-Code para el desarrollo front-end presentes en el mercado actual, con un pequeño análisis de las características que ofrece cada una. Posteriormente, se investigará sobre trabajos ya realizados del impacto del uso de herramientas No-Code Low-Code en el proceso del desarrollo del software. Por ultimo se hará un análisis de las aplicaciones que se encuentran actualmente disponibles para la gestión de reservas de canchas de tenis y como la aplicación en desarrollo puede ganar terreno en el mercado.

Entre las herramientas mas populares actualmente para el desarrollo Low-Code No-Code de aplicaciones front-end se encuentran:

1. WeWeb
2. bubble
3. Adalo
4. Oracle Apex

Weweb (<https://www.weweb.io/>) es una plataforma No-Code para la construcción y desarrollo front-end que se puede utilizar sobre cualquier back-end. Esta aplicación asegura que haciendo uso de ella se pueden construir aplicaciones 10 veces mas rápido. Dentro de sus principales características se encuentran:

1. Diseño web mediante la técnica de drag-drop
2. Fácil despliegue de la aplicación desarrollada
3. Conexiones con cualquier tipo de API
4. Escalabilidad de la aplicación mediante diferentes paquetes
5. Trabajo en equipo sobre un mismo proyecto, pagando por cada usuario extra

6. Enfoque especializado en el desarrollo front-end

Bubble (<https://bubble.io/>) a diferencia de Weweb, no solo soporta el desarrollo front-end de una aplicación si no que también soporta el desarrollo back-end. Entre sus promesas se encuentra la de poder construir cualquier aplicación sin código, solo haciendo uso de drag-and-drop. Entre sus características principales se encuentra:

1. Permite la creación de aplicaciones multiplataformas
2. Permite la creación de aplicaciones multilenguaje de forma automática
3. Promete una escalabilidad sin cambios técnicos.
4. Se encarga del despliegue y el hosting de la aplicaciones.
5. Ofrece integraciones para facilitar procesos como el de pagos o envíos de correos
6. Diferentes personas trabajen al tiempo sobre un proyecto

Adalo (<https://es.adalo.com/>) , al igual que las otras aplicaciones, permite el desarrollo de aplicaciones robustas, tanto front-end como back-end. Entre sus principales características encontramos:

1. Permite publicar las aplicaciones en App stores
2. Diseño completamente visual y en tiempo real
3. Curvas de aprendizaje muy bajas
4. Una verdadera plataforma No-Code
5. Permitir realizar todo el desarrollo de la aplicación desde la plataforma.
6. Plantillas para un desarrollo mas ágil

Oracle Apex (<https://apex.oracle.com/es/>) es una herramienta Low-Code para crear aplicaciones web y móviles enfocadas en aplicaciones empresariales. Ofrece un editor visual de drag-and-drop para interfaces gráficas y se integra con otras tecnologías de oracle como la base de datos SQL de Oracle. Entre sus principales características encontramos:

1. Crear aplicaciones móviles y en la nube
2. Convertir hojas de calculo en aplicaciones
3. Formularios modernos
4. Desarrollo sencillo y seguro

Ante el gran abanico de plataformas Low-Code No-code presentes en el mercado, para tomar una decisión sobre cual utilizar se debe de evaluar cada uno bajo unos criterios específicos. En su artículo “Navigating the Low-Code Landscape: A Comparison of Development Platforms” Kirchhof et al. (2023) analiza y compara 21 diferentes plataformas Low-Code (LCDP) bajo ciertos criterios que considera importantes, con el fin de proporcionar orientación a los expertos en el dominio y a los investigadores sobre las características relevantes de estas plataformas.

De igual forma Bock and Frank (2021a) en su artículo “In Search of the Essence of Low-Code: An Exploratory Study of Seven Development Platforms” presenta un estudio exploratorio de siete plataformas de desarrollo Low-Code con el objetivo de entender su esencia y evaluarlas críticamente en relación con la investigación en desarrollo de sistemas de información. Utilizando un marco de análisis que cubre varios criterios relacionados con el desarrollo de sistemas de información profesional, se identifican las características comunes, ocasionales y poco frecuentes de estas plataformas

Entrando ya un poco sobre los beneficios e impactos del uso de tecnologías Low-Code No-Code, se encontraron los siguientes trabajos que ahondan sobre el tema.

El artículo “Benefits and limitations of using low-code development to support digitalization in the construction industry”, deMartinez and Pfister (2023), explora

cómo las herramientas No-Code pueden promover la digitalización en la construcción. Utilizando una metodología híbrida de investigación, el estudio analiza cómo estas herramientas pueden mejorar procesos específicos en el sector, destacando tanto sus ventajas como sus limitaciones. Se recomienda adoptar un enfoque estratégico para maximizar su potencial a nivel organizacional y se proponen áreas adicionales para futuras investigaciones, incluyendo la integración con otras tecnologías relevantes para la industria de la construcción.

Adicionalmente, en el artículo “Characteristics and Challenges of Low-Code Development: The Practitioner’s Perspective”, por Luo et al. (2021), examinan las características y desafíos del desarrollo con herramientas Low-Code (LCD por sus siglas en inglés) a través de datos de comunidades en línea como Stack Overflow y Reddit. Se recopilan y analizan publicaciones relacionadas con LCD, identificando sus ventajas, limitaciones y percepciones de los practicantes. Se concluye que el LCD proporciona una interfaz gráfica para desarrollar aplicaciones con poco o ningún código, siendo especialmente útil en dominios con necesidades de automatización. Sin embargo, existen opiniones divididas sobre sus ventajas y desventajas. Se destaca la importancia de definir claramente los términos relacionados con el LCD y evaluar su idoneidad para proyectos específicos

En cuanto a las aplicaciones que se encuentran en el mercado para la gestión de reservas de canchas de tenis, vamos a dividir las en nacionales e internacionales, recalando sus características y también las limitantes con las que cuentan.

Reservas de canchas alcaldía de barranquilla (<https://appbaq.barranquilla.gov.co:8989/canchas/>), es una plataforma un poco particular debido a que solo se usa para reservar canchas públicas de la ciudad de barranquilla Colombia. Dentro de sus funcionalidades encontramos:

1. Permite registrar una reserva
2. Permite consultar una reserva

Dentro de las limitaciones se encuentran

1. Solo permite el registro de una hora a la vez lo que puede generar que si el

usuario requiere de dos horas, ambas no cuenten con la misma cancha disponible.

2. El usuario es el que escoge la cancha, esto puede generar una asignación ineficiente de la ocupación de las canchas.
3. Solo esta enfocado en la ciudad de barranquilla
4. No cuenta con la posibilidad de la creación de usuarios, por lo que para cada reserva se deba ingresar información básica.
5. La experiencia de usuario para la creación de una reserva no es la mejor
6. No permite realizar la cancelación de una reserva desde la plataforma, toca comunicarse con un funcionario

Easycan[\(https://www.easycan.com/\)](https://www.easycan.com/) es quizás la aplicación mas fuerte en la gestión de canchas de tenis con presencia en colombia aunque va mas orientada a las canchas de padel pero poco a poco va ganando terreno en las canchas de tenis. La aplicación es chilena pero esta presente en 7 países entre los que se encuentra Brasil, Argentina, Colombia, Ecuador, Usa, México y Chile.

Dentro de sus funcionalidades para la reservas de canchas de tenis encontramos:

1. Permite la creación de usuarios en la aplicación
2. Es una aplicación multideporte
3. Cuenta con múltiples clubes inscritos, por lo que un usuario puede elegir entre diferentes instalaciones disponibles.
4. Permite que algunas instalaciones solo estén disponibles para socios
5. Tiene incorporado el proceso de pago de la reserva
6. La interfaz es amigable con el usuario
7. Entre los usuarios de la aplicación pueden cuadrar partidos

8. Cuenta con un ranking para los usuarios de la aplicación que jueguen partidos entre ellos

Dentro de las limitantes encontramos:

1. Solo se pueden realizar reservas de una hora definida, por lo que el proceso para reservar mas de una hora es engorroso.
2. El usuario es el que escoge la cancha, esto puede generar una asignación ineficiente de la ocupación de las canchas.

Courtserve (<https://courreserve.com/>) es la aplicación mas completa que se encuentra para la gestión de canchas de tenis pero solo cuenta con presencia en los estados unidos y sus procesos se encuentran muy enfocados al mercado americano. Entre sus funcionalidades mas importantes se encuentran

1. El sistema hace automáticamente la asignación de las reservas a las canchas de tenis
2. Permite a los usuarios establecer horarios predilectos para sus reservas
3. Los clubes pueden restringir las reservas en las horas picos
4. Los usuarios pueden reservas mas de una hora, si hay disponibilidad
5. Notifica a los usuarios tiempo antes de la reserva

Dentro de las limitantes encontramos

1. La configuración es complicada
2. La interfaz no es muy amigable con los usuarios
3. Usuarios reportan casos de doble asignación de canchas
4. No cuentan con presencia fuera de los estados unidos
5. La aplicación no esta optimizada para dispositivos móviles
6. Esta muy enfocado en la administración de un club y deja por fuera el mercado de las personas que reservan sin ser miembros de algún lugar

7. Metodología de la investigación

Para poder cumplir con los objetivos específicos y por consiguiente con el objetivo general de implementar dos interfaces de usuarios y compararlas, el trabajo se desarrollara bajo la metodología Ágil basada en Scrum, que mediante un desarrollo incremental nos permite generar pequeños avances funcionales que nos acercan cada vez mas a la solución planeada. El objetivo del desarrollo incremental es construir el software en etapas y en cada paso, agregar funcionalidades nuevas.

Basándonos en el marco de gestión de proyectos de metodología ágil, Scrum, aplicaremos las siguientes fases:

- Plantación del sprint: En esta etapa se definen los impactos funcionales que se van a realizar a lo largo del sprint.
- Sprint: Periodo del tiempo en donde se ejecutara lo planeado en la etapa inicial.
- Revisión del sprint: En esta etapa se revisa lo trabajado en el sprint para saber si se cumplió o no con lo planeado y cual fue el incremento funcional.
- Retrospectiva del Sprint: En esta etapa se levantan las lecciones aprendidas durante el sprint, al igual que los impedimentos que hubo y como se pueden mejorar.

Para poder cumplir con lo comprometido en el proyecto de grado se realizaran 6 Sprints de 3 semanas cada uno, en la se pueden observar lo planeado para cada uno de ellos.

Sprint 1	Selección de herramientas Low-Code No-code para el desarrollo del front	Semana 1	Semana 3
	Identificar los criterios con los cuales comparar una herramienta para el desarrollo front		
Sprint 2	Implementar inicio de sesión	Semana 4	Semana 6
	Implementar registro de usuarios		
Sprint 3	Implementar Dashboard para usuarios NO administradores	Semana 7	Semana 9
	Implementar registro de Reservas		
	Implementar cancelación de reservas		
Sprint 4	Implementar Dashboard para usuario administrador	Semana 10	Semana 12
	Implementar gestión de reservas para usuario administrador		
Sprint 5	Comparación entre ambos desarrollos front	Semana 13	Semana 15
	Pruebas a los desarrollos		
Sprint 6	Construcción del documento final	Semana 16	Semana 18

Figura 1: Planeación Sprints. Elaboración propia

8. Recursos a emplear

1. Recursos Humanos

- a) Director: Juan Pablo Garcia Cifuentes.
Magíster en Administración de la Universidad Icesi.
Ingeniero de Sistemas Y Computación de la Pontificia Universidad Javeriana.
Director del programa de Innovación por Diseño de red SUGAR.
- b) Asesor: Steve Agudelo Alzate Ingeniero de Sistemas Y Computación de la Pontificia Universidad Javeriana.
- c) Estudiante: Juan Pablo Aragón Álzate
Ingeniero Industrial de la Pontificia Universidad Javeriana.

2. Los recursos tecnológicos y económicos necesarios para poder llevar a cabo el trabajo de grado serán asumidos por el estudiante.

- a) Membresía de las aplicaciones No-Code Low-Code
- b) Computador

3. Para temas de investigación se hará uso de los recursos bibliográficos dispuestos por la Pontificia Universidad Javeriana

9. Cronograma de actividades

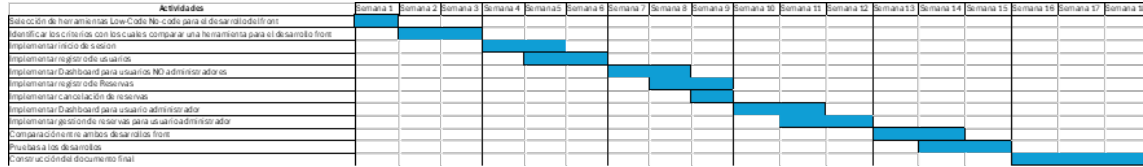


Figura 2: Cronograma de actividades. Elaboración propia

10. Referencias Bibliográficas

Referencias

Amazon Web Services (2023). Front End frente a back-end: diferencia entre el desarrollo de aplicaciones - AWS. <https://aws.amazon.com/es/compare/the-difference-between-frontend-and-backend/>.

BCG (2020). La transformación digital puede aumentar la rentabilidad de las empresas y construir resiliencia. <https://www.bcg.com/press/26november2020-digital-transformation-increase-business-profitability-build-resili>

Bock, A. C. and Frank, U. (2021a). In search of the essence of low-code: An exploratory study of seven development platforms. In *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, pages 57–66.

Bock, A. C. and Frank, U. (2021b). Low-code platform. *Business and Information Systems Engineering*, 63:733–740.

El País (2023). Descubra el top 5 de deportes más populares en 2023.

Gartner (2022). Gartner forecasts worldwide low-code development technologies market to grow 20% in 2023.

George, B. and Paul, J., editors (2020). *Digital transformation in business and society: Theory and cases*. Springer.

Hylton, D., Sung, S., and Xie, C. (2021). Adopting no-code methods to visualize computational thinking.

Kirchhof, J. C., Jansen, N., Rumpe, B., and Wortmann, A. (2023). Navigating the low-code landscape: A comparison of development platforms. pages 854–862. Institute of Electrical and Electronics Engineers Inc.

- Luo, Y., Liang, P., Wang, C., Shahin, M., and Zhan, J. (2021). Characteristics and challenges of low-code development: The practitioners perspective. IEEE Computer Society.
- Martinez, E. and Pfister, L. (2023). Benefits and limitations of using low-code development to support digitalization in the construction industry. *Automation in Construction*, 152:104909.
- Moskal, M. (2021). No-code application development on the example of logotec app studio platform. *Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Srodowiska*, 11:54–57.
- Okano, M. T., Inoue, P. K., Simões, E. A., and Batista, R. (2021). Business models in the digital transformation era. In Otolá, I. and Grabowska, M., editors, *Business models: Innovation, digital transformation, and analytics*, pages 13–31. CRC Press.
- Redhat (2018). Intelligent process automation and the emergence of digital automation platforms the transformation of application development and its emerging role to enable new competitive advantage f e b r u a r y 2 0 1 8 pat h f i n d e r r e p o r t.
- REDHAT (2023). ¿qué es una api rest?
- Sanchis, R., Óscar García-Perales, Fraile, F., and Poler, R. (2020). Low-code as enabler of digital transformation in manufacturing industry. *Applied Sciences (Switzerland)*, 10.
- Trigo, A., Varajao, J., and Almeida, M. (2022). Low-code versus code-based software development: Which wins the productivity game? *IT Professional*, 24:61–68.
- Università, G. M., Studi, D., Tre, R., and Masili, G. (2023). No-code development platforms: Breaking the boundaries between it and business experts.
- Varajão, J., Trigo, A., and Dias, M. F. (2023). Low-code development productivity. *ACM queue*, 21(5):87–107.

Vincent, P., Natis, Y., Iijima, K., Wong, J., Ray, S., Jain, A., and Leow, A. (2020).
Licensed for distribution magic quadrant for enterprise low-code application plat-
forms.