# **Título del proyecto.**

Aplicación web para préstamos de implementos deportivos

# **Pregunta problema.**

¿Cómo se puede abordar de manera efectiva la indisponibilidad de equipamiento y áreas destinadas a la práctica deportiva en Colombia?

# **Objetivo General.**

Desarrollar una aplicación que gestione los préstamos de equipamiento y espacios deportivos.

# **Objetivos específicos.**

1. Realizar un estudio de la población que incluya personas con recursos y personas sin recursos que deseen utilizar los servicios de préstamo de implementos deportivos.
2. Recolectar los requisitos funcionales y no funcionales del sistema de gestión de préstamos de equipamiento y espacios deportivos, para construir los diagramas UML (Lenguaje Unificado Modelado de datos), utilizando herramientas case.
3. Implementar un diseño detallado del software de gestión de préstamos de equipamiento y espacios deportivos, incluyendo la arquitectura del sistema, la interfaz de usuario, las funciones principales y la integración de características específicas.
4. Construir el software de gestión de préstamos de equipamiento y espacios deportivos, siguiendo el diseño previamente establecido, garantizando la funcionalidad completa de todas las características especificadas.

# **Definición de requisitos de calidad.**

## **Funcionalidad:**

* Permitir a los usuarios solicitar y administrar préstamos de equipos deportivos de manera eficiente, incluyendo la selección de los elementos deseados, fechas de préstamo y devolución, así como la consulta del historial de préstamos.
* Proporcionar una función de seguimiento del estado de los implementos deportivos, permitiendo a los usuarios verificar la disponibilidad, la condición y la ubicación de los equipos en tiempo real.
* Facilitar el proceso de devolución de equipos deportivos prestados, incluyendo la verificación del estado del equipo devuelto y la actualización de su disponibilidad en el sistema.
* Facilitar el registro inicial de usuarios en la plataforma, así como la capacidad de editar y actualizar la información de perfil, incluyendo detalles de contacto y preferencias de préstamo.
* Permitir a los usuarios anunciar los implementos deportivos que tienen disponibles para préstamo, especificando detalles como el tipo de equipo, disponibilidad de fechas y condiciones de préstamo.
* Registrar todas las transacciones realizadas en la plataforma, incluyendo préstamos, devoluciones, reservas de espacios deportivos y comunicaciones entre usuarios, para garantizar la trazabilidad y transparencia del sistema.
* Implementar un sistema de valoración y comentarios entre usuarios, permitiendo evaluar la experiencia de préstamo y facilitando la confianza y reputación dentro de la comunidad deportiva.

## **Rendimiento**

* Diseñar la aplicación para que sea capaz de manejar un aumento significativo en el tráfico de usuarios y transacciones sin comprometer su velocidad y eficiencia, mediante la implementación de una arquitectura escalable y el uso de tecnologías adecuadas para la gestión de recursos.
* Implementar estrategias de gestión de carga para distribuir equitativamente el tráfico de usuarios y evitar sobrecargas en el servidor, mediante el uso de técnicas como el balanceo de carga y la asignación dinámica de recursos.

## **Usabilidad**

* Desarrollar una interfaz de usuario intuitiva y fácil de usar, con una navegación clara y consistente, que permita a los usuarios acceder rápidamente a todas las funcionalidades del sistema sin confusiones ni complicaciones innecesarias.
* Incluir una guía interactiva o tutorial de bienvenida que oriente a los usuarios sobre cómo utilizar el sistema de manera efectiva, presentando las principales características y funcionalidades de forma didáctica y accesible.

## **Seguridad**

* El sistema debe velar por la seguridad de los usuarios Implementando medidas de seguridad robustas para proteger tanto a los usuarios como sus datos durante las transacciones y el intercambio de información.

## **Mantenibilidad**

* El sistema se debe implementar en una arquitectura limpia y modular, que facilite su mantenimiento a lo largo del tiempo.

## **Compatibilidad**

* El sistema se debe ser compatible con una amplia gama de dispositivos y navegadores web, garantizando una experiencia consistente y accesible para los usuarios, ya sea que accedan desde computadoras de escritorio, portátiles, tabletas o teléfonos inteligentes.

# **Establecimiento de Estándares de Calidad**

## **Codificación**

* Se adoptarán convenciones de nomenclatura consistentes y comprensibles para variables, funciones y archivos en el código del proyecto.
* Se privilegiarán las prácticas de codificación eficientes, como la modularización y la reutilización de código, para garantizar un desarrollo mantenible y escalable.
* Se evitarán prácticas de codificación perjudiciales, como la duplicación de código y el uso excesivo de variables globales, para mejorar la legibilidad y la mantenibilidad del código.

## **Diseño de interfaz de usuario**

* La interfaz de usuario se diseñará priorizando la intuitividad y la facilidad de uso, siguiendo principios de diseño centrados en la experiencia del usuario.
* Se aplicarán estándares reconocidos de diseño de interfaz de usuario, como Material Design o iOS Human Interface Guidelines, para asegurar una apariencia y una navegación coherentes y familiares para los usuarios.

## **Pruebas**

* Se llevarán a cabo pruebas exhaustivas en todas las etapas del desarrollo, incluyendo pruebas unitarias, de integración y de aceptación, con el fin de garantizar la calidad y la fiabilidad del sistema.
* Se establecerán criterios de cobertura de pruebas para asegurar que todas las funcionalidades críticas del sistema sean sometidas a pruebas rigurosas.

## **Seguridad**

* Se implementarán medidas de seguridad apropiadas, como autenticación de usuarios, cifrado de datos y protección contra vulnerabilidades conocidas, para salvaguardar la integridad y la confidencialidad de la información.
* Se realizarán pruebas de seguridad regulares para identificar y mitigar posibles vulnerabilidades en el sistema, garantizando así su robustez y su resistencia a ataques.

## **Documentación**

* Se mantendrá una documentación completa y actualizada del sistema, que incluirá especificaciones de diseño, guías de usuario y manuales técnicos, con el objetivo de facilitar la comprensión y el uso del sistema por parte de los desarrolladores y los usuarios finales.
* La documentación se redactará de manera clara y accesible para todos los miembros del equipo, promoviendo así una colaboración efectiva y una gestión eficiente del conocimiento.

## **Mantenibilidad**

* El código del sistema se organizará de forma clara y modular, para facilitar su comprensión y su mantenimiento por parte de otros desarrolladores en el futuro.
* Se establecerán procedimientos de gestión de cambios y versiones para garantizar la trazabilidad y la integridad del código, favoreciendo así la evolución continua y la mejora del sistema.

## **Compatibilidad**

* El sistema se diseñará y probará para ser compatible con una amplia variedad de navegadores web y dispositivos, asegurando una experiencia consistente y satisfactoria para todos los usuarios.
* Se llevarán a cabo pruebas de compatibilidad para verificar el rendimiento del sistema en diferentes entornos y configuraciones, garantizando así su funcionalidad óptima en cualquier situación de uso.

# **Planificación de pruebas**

## **Pruebas unitarias**

* Se simulará el proceso de solicitud de préstamo de un equipo deportivo específico por parte de un usuario. Se verificará que el sistema registre correctamente la solicitud, actualice el estado del equipo a "prestado" y proporcione al usuario información clara sobre los préstamos pendientes.
* Se realizarán solicitudes de carga de páginas y procesamiento de transacciones bajo diferentes condiciones de carga. Se medirán y registrarán los tiempos de respuesta del sistema para cada acción realizada. Se compararán los tiempos de respuesta con los umbrales definidos en los requisitos de rendimiento.
* Se utilizarán herramientas de accesibilidad para simular la experiencia de usuarios con discapacidades visuales. Se realizarán acciones en el sistema y se verificará que se proporcione retroalimentación adecuada, como mensajes de error y confirmaciones de éxito, de forma audible o mediante lectores de pantalla

## **Pruebas de integración**

* Se realizarán pruebas de integración para verificar la interoperabilidad entre los distintos módulos y componentes del sistema, garantizando que funcionen correctamente cuando se combinan.
* Se identificarán y resolverán posibles conflictos y dependencias entre los componentes durante el proceso de integración, asegurando la cohesión y el rendimiento del sistema completo.

## **Pruebas funcionales**

* Se llevarán a cabo pruebas funcionales para validar que el sistema cumple con todos los requisitos especificados en los documentos de requisitos del usuario.
* Se elaborarán casos de prueba basados en los escenarios de uso esperados por los usuarios finales, cubriendo todas las funcionalidades y características del sistema.

# **Herramientas y recurso**

* Entorno de Desarrollo Integrado (IDE): Un ambiente de programación donde puedas escribir, depurar y ejecutar tu código.
* Control de Versiones: Una forma de gestionar cambios en el código fuente y colaborar con otros desarrolladores de manera eficiente.
* Gestión de Proyectos: Herramientas para planificar, organizar y dar seguimiento al progreso del proyecto.
* Comunicación y Colaboración: Plataformas para facilitar la comunicación y colaboración entre los miembros del equipo de desarrollo.
* Desarrollo de Backend: Recursos para desarrollar la lógica de negocio y la funcionalidad del servidor.
* Desarrollo de Frontend: Herramientas para crear la interfaz de usuario con la que interactuarán los usuarios finales.
* Pruebas: Recursos para realizar pruebas de diferentes tipos, como pruebas unitarias, de integración y de aceptación.
* Despliegue y Hosting: Medios para poner en funcionamiento la aplicación en un entorno de producción accesible para los usuarios finales.
* Monitoreo y Registro: Soluciones para supervisar el rendimiento y el comportamiento de la aplicación en tiempo real.
* Documentación: Herramientas y recursos para documentar el código, el diseño y los procesos del proyecto de manera clara y comprensible para todos los miembros del equipo.