

**UNIVERSIDAD TECNOLOGICA DEL SUR DEL ESTADO DE MEXICO**

**“Sistema para la gestión de Biblioteca”**

Integrantes:

Osvaldo Florencio Jerónimo

Kevin Antonio Hernandez García

Contenido

[**2. Planteamiento del problema** 3](#_Toc210773316)

[**3. Justificación** 4](#_Toc210773317)

[**4. Objetivos** 5](#_Toc210773318)

[**Objetivo general** 5](#_Toc210773319)

[**Objetivos específicos** 5](#_Toc210773320)

[**5. MARCO TEÓRICO** 6](#_Toc210773321)

[**Ionic** 6](#_Toc210773322)

[**Visual Studio** 8](#_Toc210773323)

[**Spring boot** 11](#_Toc210773324)

[**Bootstrap** 13](#_Toc210773325)

[**Flag Icon** 16](#_Toc210773326)

[**HTML** 16](#_Toc210773327)

[**Java** 17](#_Toc210773328)

[**CSS** 17](#_Toc210773329)

[**GitHub** 18](#_Toc210773330)

[**Scrum** 19](#_Toc210773331)

[**6. METODOLOGÍA SCRUM** 21](#_Toc210773332)

[**7. CRONOGRAMA** 21](#_Toc210773333)

[**8. DELIMITACIÓN DEL ESTUDIO** 21](#_Toc210773334)

[**Alcance** 21](#_Toc210773335)

[**Limitantes** 22](#_Toc210773336)

[**10.** **Bibliografía** 24](#_Toc210773337)

# **2. Planteamiento del problema**

En la actualidad, muchas bibliotecas han comenzado a incorporar herramientas digitales para optimizar sus procesos de registro, control y seguimiento de libros. Sin embargo, la biblioteca de la Universidad Tecnológica del Sur del Estado de México continúa realizando estas actividades de forma manual, utilizando libros de registro o hojas de cálculo, lo que dificulta mantener un control eficiente del inventario y de los préstamos realizados.

Este método tradicional presenta diversas limitaciones, como la pérdida de información, errores humanos en los registros, dificultad para consultar la disponibilidad de ejemplares y falta de organización al momento de realizar préstamos o devoluciones. Además, la ausencia de un sistema digital impide llevar un historial confiable de los libros prestados, de los usuarios que los solicitan y de las fechas de devolución, lo que puede generar confusión y pérdida de materiales.

Frente a esta situación, surge la necesidad de desarrollar un sistema web que se aplicará en la biblioteca de la Universidad Tecnológica del Sur del Estado de México, el cual permitirá registrar libros y gestionar de manera eficiente los préstamos y devoluciones. Este sistema facilitará la administración del inventario, reducirá los errores en los registros y agilizará el proceso de préstamo, brindando un mayor control sobre el movimiento de los ejemplares.

El proyecto busca, en última instancia, modernizar la gestión interna de la biblioteca, mejorando la organización y trazabilidad de los libros, sin perder la atención personalizada que la caracteriza. De esta forma, la biblioteca podrá optimizar su funcionamiento diario y ofrecer un servicio más ordenado y confiable tanto para el personal como para los usuarios.

# **3. Justificación**

En la actualidad, las tecnologías de la información y la comunicación desempeñan un papel fundamental en la modernización de los procesos administrativos y en la optimización del manejo de la información. Sin embargo, muchas bibliotecas pequeñas continúan utilizando métodos manuales para registrar y controlar sus libros, lo que genera desorganización, pérdida de datos y dificultades para realizar un seguimiento adecuado de los préstamos y devoluciones.

La biblioteca de la Universidad Tecnológica del Sur del Estado de México enfrenta esta problemática, ya que actualmente no cuenta con un sistema que le permita llevar un registro automatizado de los libros disponibles ni de los préstamos realizados. Esta situación complica la gestión del inventario, retrasa las tareas administrativas y limita la capacidad de ofrecer un servicio eficiente y confiable a los usuarios.

El desarrollo de un sistema web para el registro y control de préstamos de libros se justifica como una herramienta tecnológica necesaria para mejorar la organización interna, reducir los errores humanos y agilizar los procesos de préstamo y devolución. Con esta solución, la biblioteca podrá disponer de un registro centralizado, actualizado y seguro de todos sus ejemplares, lo que facilitará la toma de decisiones y el seguimiento de las actividades diarias.

Además, este proyecto contribuye al fortalecimiento de la gestión educativa y cultural dentro de la universidad, ya que promueve el acceso ordenado a los materiales bibliográficos y fomenta la consulta académica entre la comunidad estudiantil y docente. De esta forma, la biblioteca podrá modernizar sus procesos internos sin perder su esencia de atención personalizada, combinando la eficiencia que ofrece la tecnología con un servicio de calidad.

En conclusión, la implementación de este sistema representa una mejora operativa significativa para la biblioteca, así como un avance hacia la digitalización y optimización de sus recursos, garantizando un control más efectivo y una administración más moderna y funcional.

# **4. Objetivos**

### **Objetivo general**

Desarrollar un sistema web que permita registrar y controlar de manera eficiente los libros y los préstamos realizados en la biblioteca de la Universidad Tecnológica del Sur del Estado de México, con el fin de optimizar la gestión del inventario, mejorar la organización interna y ofrecer un servicio más ágil y confiable.

### **Objetivos específicos**

1. Diseñar una herramienta digital que permita registrar, consultar y administrar la información de los libros de forma centralizada, evitando duplicidades y facilitando el control del inventario.
2. Desarrollar una funcionalidad que permita a los usuarios consultar la disponibilidad de libros, gestionar préstamos de manera rápida y sencilla.
3. Optimizar la gestión del inventario mediante reportes automatizados que reflejen el estado actual de los libros (prestados o disponibles).
4. Implementar un módulo de administración que facilite el registro, modificación y eliminación de libros dentro del sistema.

# **5. MARCO TEÓRICO**

## **Ionic**

Ionic Framework es un entorno de desarrollo de código abierto que permite la creación de aplicaciones móviles, web y de escritorio a partir de una única base de código. Este framework utiliza tecnologías web estándar como HTML, CSS y JavaScript/TypeScript, y puede integrarse fácilmente con frameworks modernos como Angular, React o Vue. Su principal objetivo es facilitar el desarrollo de aplicaciones multiplataforma, es decir, que puedan ejecutarse en diferentes sistemas operativos con un solo desarrollo base. (Wikipedia, 2019)

El proyecto Ionic fue creado en el año 2013 por Max Lynch, Ben Sperry y Adam Bradley, miembros de la empresa Drifty Co. Desde su lanzamiento, ha pasado por diversas actualizaciones y mejoras que lo han convertido en una de las herramientas más populares para el desarrollo de aplicaciones híbridas. En sus primeras versiones, Ionic se basaba completamente en AngularJS y utilizaba Apache Cordova para acceder a las funciones nativas de los dispositivos móviles. Sin embargo, a partir de Ionic 4, el framework adoptó el uso de Web Components, lo que permitió su compatibilidad con cualquier framework de frontend y una arquitectura más modular y flexible.

La filosofía de Ionic se basa en el concepto de desarrollo híbrido, el cual busca reducir los costos y el tiempo de desarrollo al crear aplicaciones que funcionen tanto en Android, iOS y la web, sin necesidad de escribir código nativo para cada plataforma. Esto se logra mediante el uso de tecnologías web encapsuladas dentro de un contenedor nativo, gracias a herramientas como Cordova o Capacitor, que actúan como puentes entre el código JavaScript y las funciones del dispositivo (como la cámara, el GPS o las notificaciones).

En el aspecto de la interfaz, Ionic proporciona una amplia colección de componentes UI prediseñados que imitan la apariencia y comportamiento de las aplicaciones nativas. Estos componentes incluyen botones, menús, formularios, listas, pestañas y elementos de navegación que se adaptan automáticamente al sistema operativo del usuario. Esto permite ofrecer una experiencia de usuario fluida y coherente en diferentes plataformas, sin necesidad de diseñar cada elemento desde cero.

Desde el punto de vista arquitectónico, Ionic utiliza Stencil.js para generar Web Components reutilizables, lo que lo hace independiente del framework de desarrollo utilizado. Además, incorpora funcionalidades como Lazy Loading (carga diferida de módulos), un CLI (Command Line Interface) que simplifica la creación y compilación de proyectos, y compatibilidad con PWA (Progressive Web Apps), permitiendo desarrollar aplicaciones que pueden funcionar incluso sin conexión a internet. (Dharmadhikari, 2017)

Entre las principales ventajas de Ionic se destacan su rapidez de desarrollo, la reutilización de código, la reducción de costos, su gran comunidad de soporte y la posibilidad de acceder a funciones nativas del dispositivo mediante plugins. Asimismo, ofrece un rendimiento adecuado para la mayoría de las aplicaciones empresariales y comerciales, y facilita la integración con servicios externos.

No obstante, también existen algunas limitaciones. El rendimiento puede verse afectado en aplicaciones que requieren procesamiento intensivo o gráficos avanzados, ya que Ionic utiliza un WebView para renderizar la interfaz. Además, la dependencia de plugins externos puede generar incompatibilidades o limitaciones cuando estos no están actualizados. A pesar de ello, las versiones más recientes, junto con el uso de Capacitor, han mejorado significativamente el rendimiento y la estabilidad general del framework.

Ionic Framework continúa evolucionando, consolidándose como una herramienta versátil y eficiente para el desarrollo de aplicaciones multiplataforma. Gracias a su arquitectura basada en componentes, soporte para PWA y compatibilidad con diversas tecnologías modernas, se ha convertido en una opción ideal para desarrolladores y empresas que buscan optimizar el tiempo y los recursos en sus procesos de desarrollo. (Ciberninjas, 2022)

## **Visual Studio**

Visual Studio es un entorno de desarrollo integrado (IDE) creado por Microsoft, diseñado para facilitar el proceso de creación, prueba y despliegue de aplicaciones en diversas plataformas, como escritorio, web y dispositivos móviles. Este entorno ofrece un conjunto de herramientas unificadas que incluyen editor de código, depurador, diseñador de interfaces, compilador y gestor de proyectos, lo que permite al desarrollador realizar todo el ciclo de desarrollo de software desde un mismo lugar.

Una de las principales características de Visual Studio es su compatibilidad con múltiples lenguajes de programación, como C#, C++, F# y Visual Basic, siendo este último uno de los lenguajes más representativos del entorno debido a su enfoque en la programación visual y orientada a eventos. Visual Basic surgió como una evolución del lenguaje BASIC, con la finalidad de simplificar el desarrollo de aplicaciones mediante una interfaz gráfica intuitiva, permitiendo a los usuarios crear programas de manera rápida utilizando herramientas visuales. (Microsoft, 2023)

Visual Basic se basa en el paradigma de la programación orientada a eventos, donde la ejecución del programa depende de las acciones del usuario, como hacer clic en un botón, escribir en un campo o seleccionar una opción. Este enfoque permite una mayor interacción entre el usuario y la aplicación, mejorando la experiencia de uso. En conjunto con Visual Studio, ofrece un entorno visual que permite diseñar interfaces gráficas mediante la técnica WYSIWYG (What You See Is What You Get), lo que significa que los desarrolladores pueden arrastrar y soltar controles (botones, cuadros de texto, etiquetas, etc.) sobre formularios y observar de inmediato cómo se verá la aplicación final.

Visual Studio está construido sobre la plataforma .NET Framework, la cual proporciona una infraestructura común para el desarrollo de aplicaciones, compuesta por el Common Language Runtime (CLR) y una amplia biblioteca de clases. El CLR se encarga de la ejecución del código intermedio (IL) y de tareas como la gestión de memoria, la seguridad y la interoperabilidad entre lenguajes. Gracias a esta arquitectura, las aplicaciones desarrolladas en Visual Studio pueden beneficiarse de una alta compatibilidad y un rendimiento optimizado dentro del ecosistema de Microsoft.

Entre las principales ventajas de Visual Studio destacan su entorno completo de trabajo, la integración de herramientas avanzadas de depuración, la facilidad para diseñar interfaces gráficas, el soporte para múltiples lenguajes y la amplia documentación disponible. Además, su compatibilidad con tecnologías modernas como ASP.NET, .NET Core y Azure lo convierten en una herramienta esencial para el desarrollo profesional de software.

Sin embargo, también presenta algunas limitaciones. Debido a su gran cantidad de herramientas y funcionalidades, Visual Studio puede consumir muchos recursos del sistema, lo que requiere equipos con especificaciones adecuadas. Además, su dependencia del entorno Windows puede representar una restricción para desarrolladores que utilizan otros sistemas operativos. A pesar de ello, las versiones más recientes, como Visual Studio Code, han permitido mayor flexibilidad y soporte multiplataforma, ampliando su alcance en la comunidad de desarrollo.

En conclusión, Visual Studio y Visual Basic representan una combinación poderosa dentro del desarrollo de software. La unión de un entorno visual intuitivo con un lenguaje accesible y un marco robusto como .NET proporciona una plataforma integral que facilita la creación de aplicaciones eficientes, escalables y visualmente atractivas, contribuyendo significativamente al avance de la programación moderna. (ProgVisualEstudio, 2024)

## **Spring boot**

Spring Boot es un framework de código abierto para la plataforma Java, desarrollado para facilitar la creación de aplicaciones independientes (“standalone”) de grado producción basadas en Spring. Su propósito principal es reducir la complejidad de configuración inicial y permitir que los desarrolladores inicien rápido proyectos robustos usando convenciones predeterminadas (“opinionated”) en lugar de configurar manualmente cada detalle. (Wikipedia, Wikipedia, 2019)

1. Origen y Evolución

Spring Boot fue desarrollado por Pivotal Software en 2013 y su primera versión estable se lanzó en 2014. Surgió como una respuesta a la necesidad de reducir la complejidad del desarrollo con Spring, que anteriormente requería múltiples archivos XML y configuraciones extensas. Con el tiempo, se ha consolidado como una herramienta clave para el desarrollo de microservicios y aplicaciones empresariales modernas. Las versiones más recientes han añadido soporte para nuevas versiones de Java, programación reactiva y compatibilidad con Jakarta EE. (GeeksforGeeks, 2014)

1. Principales Características

Entre sus características más destacadas se encuentran la autoconfiguración, que permite al framework detectar las dependencias y aplicar configuraciones por defecto; los servidores embebidos como Tomcat o Jetty, que eliminan la necesidad de instalar servidores externos; y los “starters”, que simplifican la inclusión de dependencias comunes. Además, incluye herramientas de monitoreo y métricas mediante Spring Boot Actuator, perfiles de configuración, manejo de propiedades externas y compatibilidad con contenedores como Docker. (Wikipedia, Wikipedia, 2019)

1. Ventajas

* Spring Boot ofrece múltiples beneficios:
* Acelera el desarrollo al eliminar la necesidad de configuraciones manuales extensas.
* Facilita la creación de microservicios, gracias a su naturaleza modular y ligera.
* Es altamente productivo, al incluir herramientas listas para producción.
* Posee una gran comunidad de desarrolladores y amplia documentación.
* Permite integrar fácilmente otras tecnologías del ecosistema Spring, como Spring Security o Spring Data. (Wikipedia, Wikipedia, 2019)

1. Desventajas

A pesar de sus ventajas, Spring Boot también presenta algunas limitaciones. Su configuración automatizada puede resultar restrictiva cuando se necesita un control más preciso sobre los componentes. Además, el tamaño del archivo ejecutable puede ser mayor debido a las dependencias incluidas por defecto. También puede haber una curva de aprendizaje para los nuevos desarrolladores que no estén familiarizados con Spring.

1. Aplicaciones Comunes

Spring Boot se utiliza ampliamente en el desarrollo de aplicaciones web, microservicios, APIs RESTful, sistemas empresariales escalables y aplicaciones en la nube. Su integración con Docker y Kubernetes lo convierte en una excelente opción para entornos de despliegue continuo y DevOps. (Wikipedia, Wikipedia, 2019)

## **Bootstrap**

Bootstrap es un framework de código abierto que facilita el desarrollo de interfaces web modernas y adaptables. Fue creado por desarrolladores de Twitter en 2011 con el propósito de unificar las herramientas utilizadas en el diseño de sus aplicaciones internas. Su enfoque “mobile-first” permite que los sitios web se adapten automáticamente a diferentes tamaños de pantalla, ofreciendo una experiencia de usuario consistente en dispositivos móviles y de escritorio.

Este framework utiliza una combinación de HTML, CSS y JavaScript para crear componentes reutilizables como menús, botones, formularios y tarjetas. Además, cuenta con un sistema de rejillas que ayuda a estructurar el contenido de manera flexible. Las versiones más recientes eliminan la dependencia de jQuery, mejorando el rendimiento y compatibilidad con navegadores modernos. (getbootstrap, 2022)

Bootstrap es un framework de código abierto utilizado para diseñar sitios web y aplicaciones web responsivas, es decir, que se adaptan automáticamente a diferentes tamaños de pantalla, como computadoras, tabletas o teléfonos móviles. Fue desarrollado originalmente por Mark Otto y Jacob Thornton, ingenieros de Twitter, y lanzado al público en el año 2011 bajo el nombre de Twitter Bootstrap.

Su principal objetivo es facilitar el diseño y desarrollo front-end, reduciendo el tiempo y esfuerzo que implica crear interfaces visualmente atractivas y coherentes. Bootstrap combina HTML, CSS y JavaScript para proporcionar componentes predefinidos y estilos consistentes en toda la aplicación.

1. Origen y Evolución

Bootstrap nació como una iniciativa interna de Twitter para unificar herramientas y estilos usados por los desarrolladores de la empresa. Antes de su creación, cada equipo usaba diferentes librerías y estándares, lo que generaba inconsistencias visuales.

Tras su lanzamiento público en 2011, Bootstrap se convirtió rápidamente en uno de los frameworks front-end más utilizados del mundo debido a su facilidad de uso, documentación clara y compatibilidad con la mayoría de los navegadores.

A lo largo de sus versiones, Bootstrap ha evolucionado:

* Bootstrap 2 (2012): introdujo el sistema de cuadrícula (grid system).
* Bootstrap 3 (2013): adoptó el enfoque mobile-first, priorizando el diseño para dispositivos móviles.
* Bootstrap 4 (2018): incorporó el uso de Flexbox, un modelo moderno para organizar el diseño.
* Bootstrap 5 (2021): eliminó la dependencia de jQuery, actualizó el diseño visual y agregó compatibilidad con CSS Grid y variables personalizables.

1. Estructura y Componentes

Bootstrap ofrece un conjunto de herramientas modulares y reutilizables que permiten desarrollar interfaces de manera más rápida:

1. Sistema de cuadrícula (Grid System):

Es uno de los pilares del framework. Divide la página en filas y columnas para organizar el contenido de forma flexible y adaptable al tamaño del dispositivo.

1. Componentes predefinidos:

Incluye botones, menús desplegables, formularios, barras de navegación, tarjetas, modales, alertas, carruseles, tablas y muchos otros elementos que se pueden integrar fácilmente mediante clases CSS.

1. Utilidades (Utilities):

Proporciona clases rápidas para ajustar márgenes, colores, alineación, tipografía y visibilidad, sin necesidad de escribir código CSS adicional.

1. JavaScript Plugins:

Bootstrap incorpora funciones interactivas como menús desplegables, ventanas modales, deslizadores y colapsables, que mejoran la experiencia del usuario. En las versiones recientes, estas funcionalidades están escritas en JavaScript puro, eliminando la necesidad de jQuery. (Mark Otto, 2020)

## **Flag Icon**

Flag Icon es una biblioteca de íconos de banderas desarrollada en formato SVG y CSS. Permite integrar fácilmente banderas de países en aplicaciones o sitios web mediante clases predefinidas. Su uso es común en plataformas multilingües o sistemas donde se requiere mostrar el idioma o país correspondiente.

El proyecto se encuentra disponible en repositorios públicos y ofrece un conjunto completo de banderas basadas en los códigos ISO de cada nación. La implementación es sencilla, utilizando clases CSS como .fi o .fi-xx para mostrar la bandera correspondiente.

## **HTML**

HTML, o Lenguaje de Marcado de Hipertexto, es el estándar utilizado para estructurar el contenido de las páginas web. Define los elementos básicos de una página, como encabezados, párrafos, imágenes, tablas, enlaces y formularios.

Fue creado por Tim Berners-Lee a principios de los años noventa y se ha convertido en la base de toda la web. Los navegadores interpretan el código HTML y muestran el contenido de acuerdo con las etiquetas y atributos definidos.

Aunque no es un lenguaje de programación, HTML es esencial para la creación de documentos web y trabaja junto con CSS y JavaScript para definir la apariencia y el comportamiento de los sitios.

## **Java**

Java es un lenguaje de programación orientado a objetos diseñado para ser portátil y seguro. Permite desarrollar aplicaciones que se ejecutan en diferentes plataformas mediante la Máquina Virtual de Java (JVM).

Fue lanzado por Sun Microsystems en 1995 y actualmente es mantenido por Oracle. Es ampliamente utilizado en el desarrollo de aplicaciones empresariales, móviles y de escritorio, así como en el backend de sistemas web.

Java destaca por su robustez, escalabilidad y comunidad activa de desarrolladores. Además, su ecosistema incluye una amplia gama de frameworks, como Spring y Hibernate, que facilitan el desarrollo de aplicaciones complejas.

## **CSS**

CSS, o Hojas de Estilo en Cascada, es el lenguaje utilizado para definir el diseño y la presentación visual de las páginas web. Permite controlar aspectos como los colores, fuentes, tamaños, márgenes, posiciones y animaciones de los elementos definidos en HTML.

El funcionamiento en cascada permite aplicar diferentes reglas de estilo según la jerarquía de los elementos, haciendo que la apariencia de un sitio sea flexible y personalizable.

CSS trabaja junto con HTML y frameworks como Bootstrap para lograr sitios web modernos, atractivos y responsivos.

## **GitHub**

GitHub es una plataforma en línea basada en el sistema de control de versiones Git. Permite a los desarrolladores almacenar, compartir y colaborar en proyectos de software de manera eficiente.

Fue fundada en 2008 y rápidamente se convirtió en una de las principales herramientas para el desarrollo colaborativo. En GitHub se pueden crear repositorios públicos o privados, realizar revisiones de código, administrar versiones y coordinar equipos de trabajo.

Además de su función como repositorio, GitHub es un espacio esencial para el software libre, donde se alojan proyectos, frameworks y bibliotecas utilizadas globalmente en múltiples lenguajes de programación. (github, 2012)

El propósito principal de GitHub es facilitar el trabajo en equipo y la colaboración en proyectos de programación, permitiendo que varios desarrolladores trabajen simultáneamente sobre el mismo código fuente, sin interferir en el trabajo de los demás. Esto se logra mediante el sistema de control de versiones Git, creado por Linus Torvalds (el mismo creador de Linux) en 2005.

1. Características Principales

* Repositorios (Repositories):

Son espacios donde se almacena todo el contenido de un proyecto, incluyendo el código fuente, documentación, archivos multimedia y el historial de versiones. Los repositorios pueden ser públicos (visibles para todos) o privados (solo para colaboradores autorizados).

* Commits y Versiones:

Cada cambio realizado en un archivo se guarda mediante un commit, que incluye una descripción del cambio y un identificador único. Esto permite rastrear quién hizo qué y cuándo.

* Ramas (Branches):

Permiten trabajar en diferentes versiones o funciones del mismo proyecto sin afectar la versión principal. Por ejemplo, un desarrollador puede crear una rama para agregar una nueva característica y luego fusionarla con la versión principal mediante un merge.

* Pull Requests:

Son solicitudes que permiten a los colaboradores proponer cambios en el código. Otros miembros del equipo pueden revisar, discutir y aprobar (o rechazar) las modificaciones antes de integrarlas. (freecodecamp, 2020)

## **Scrum**

Scrum es un marco de trabajo ágil utilizado para la gestión de proyectos complejos, especialmente en el desarrollo de software. Se caracteriza por su enfoque iterativo e incremental, promoviendo la autoorganización de equipos y la adaptación constante a los cambios. Originalmente concebido para entornos de software, Scrum ha sido adoptado en diversas industrias debido a su eficacia en la entrega de valor continuo y la mejora de procesos. (asana, 2023)

1. Principios y Estructura

Scrum se basa en tres pilares fundamentales: transparencia, inspección y adaptación. Su estructura incluye roles definidos como el Scrum Master, el Product Owner y el Equipo de Desarrollo. El trabajo se organiza en ciclos denominados "sprints", que suelen durar entre 2 y 4 semanas, al final de los cuales se entrega un incremento del producto. Las reuniones clave en Scrum son la planificación del sprint, las reuniones diarias (Daily Scrum), la revisión del sprint y la retrospectiva del sprint. (aws.amazon, 2023)

1. Aplicaciones y Beneficios

Además de su uso en desarrollo de software, Scrum ha demostrado ser eficaz en áreas como la educación, la gestión de procesos crediticios y la manufactura. Por ejemplo, en el ámbito educativo, se ha implementado para fomentar un ambiente de trabajo colaborativo y mejorar la productividad estudiantil. En el sector financiero, su aplicación ha resultado en una mayor eficiencia operativa y satisfacción del cliente. En la manufactura, ha contribuido a la mejora continua y a la reducción de costos.

1. Desafíos y Consideraciones

La implementación de Scrum puede enfrentar desafíos como la resistencia al cambio, la falta de formación adecuada y la adaptación a contextos específicos. Es crucial contar con un liderazgo comprometido y una cultura organizacional que apoye la agilidad para que Scrum sea efectivo. Además, es importante reconocer que las variaciones en la práctica de Scrum pueden ser necesarias y contextuales, no necesariamente indicativas de un mal uso. (Ulloa-Ulloa, 2012)

# **6. METODOLOGÍA SCRUM**

1. Roles y responsabilidades

* Product Owner (PO): Representa a la biblioteca/universidad; prioriza el Product Backlog y valida entregables.
* Scrum Master (SM): Facilita el proceso Scrum, remueve impedimentos, organiza reuniones.
* Equipo de desarrollo: Devs + QA + (si aplica) UX. Trabajan en las historias del sprint y entregan el Incremento.

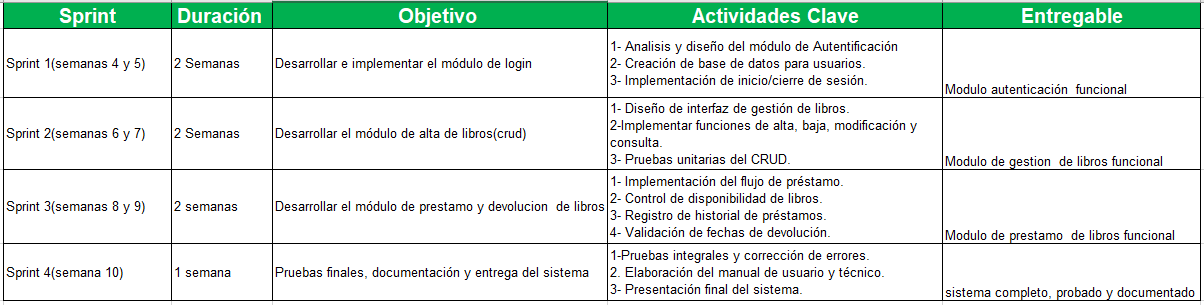
2. Ceremonias

|  |  |  |
| --- | --- | --- |
| **CEREMONIA** | **FRECUENCIA** | **OBJETIVO** |
| Sprint Planning | Inicio de cada Sprint | Elegir qué historias se desarrollarán |
| Daily Scrum | Diario (15 min) | Sincronizar avances y detectar bloqueos |
| Sprint Review | Fin del Sprint | Mostrar lo desarrollado al cliente o profesor |
| Sprint Retrospective | Fin del Sprint | Analizar como mejorar el proceso en el próximo ciclo |

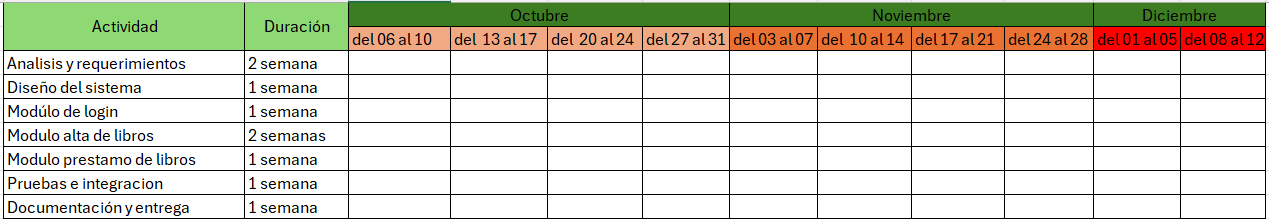
3. Artefactos

* Product Backlog: Lista priorizada de historias de usuario, bugs y tareas técnicas.
* Sprint Backlog: Historias seleccionadas + tareas desglosadas.
* Incremento: Software listo y potencialmente desplegable al final de cada sprint.
* Definition of Done (DoD) (ejemplo): Código en repositorio, pruebas unitarias mínimas, pruebas manuales pasadas, documentación mínima (README / endpoints), build exitoso en CI, desplegado en ambiente de staging.

4.Plan de Trabajo



# **7. CRONOGRAMA**



# **8. DELIMITACIÓN DEL ESTUDIO**

## **Alcance**

El presente proyecto tiene como alcance el diseño y desarrollo de un sistema web para la biblioteca de la Universidad Tecnológica del Sur del Estado de México, que permita registrar, consultar y controlar los libros disponibles de forma centralizada y eficiente. El sistema estará enfocado en mejorar la administración del inventario, facilitando la gestión de préstamos y devoluciones tanto para los administradores como para los usuarios.

El proyecto incluirá el desarrollo funcional para el registro, edición y eliminación de libros; la generación de reportes automáticos sobre el estado del inventario (libros disponibles o prestados). Además, se implementará una interfaz web intuitiva y accesible, adaptable a diferentes dispositivos, que permita una navegación sencilla y una experiencia de usuario óptima.

Asimismo, se contemplará la seguridad de la información mediante la aplicación de buenas prácticas de desarrollo web, garantizando la integridad y confidencialidad de los datos almacenados. El sistema estará dirigido principalmente al personal administrativo de la librería y a los usuarios registrados, sin contemplar por el momento integración con plataformas externas o funciones de venta en línea.

## **Limitantes**

* El sistema estará enfocado únicamente en el registro de libros y el control de préstamos y devoluciones, por lo que no incluirá funciones de compra o venta en línea.
* Solo podrá ser utilizado por el personal autorizado de la librería, sin acceso directo para los clientes externos.
* La gestión de usuarios se limitará al registro interno de quienes soliciten préstamos, sin implementación de perfiles avanzados o roles múltiples.
* El sistema no contará con integración a plataformas externas ni con un módulo de notificaciones automáticas (correo electrónico o mensajes).
* Requerirá de conexión a internet para su uso, ya que será una aplicación web y no funcionará en modo offline.

# **10.** **Bibliografía**

asana. (10 de 02 de 2023). *asana*. Obtenido de asana: https://asana.com/es/resources/what-is-scrum?utm\_source

aws.amazon. (13 de 09 de 2023). *aws.amazon*. Obtenido de aws.amazon: https://aws.amazon.com/es/what-is/scrum/?utm\_source

Ciberninjas. (22 de 02 de 2022). *Ciberninjas*. Obtenido de Ciberninjas: https://ciberninjas.com/ionic-framework/

Dharmadhikari, C. S. (23 de 12 de 2017). *Chaitali Sonar & Mugdha Dharmadhikari*. Obtenido de Chaitali Sonar & Mugdha Dharmadhikari: https://ijarcce.com/papers/hybrid-mobile-app-development-using-ionic-framework/

freecodecamp. (06 de 10 de 2020). *freecodecamp*. Obtenido de freecodecamp: https://www.freecodecamp.org/news/what-is-github-and-how-to-use-it/

GeeksforGeeks. (20 de 12 de 2014). *GeeksforGeeks*. Obtenido de GeeksforGeeks: https://www.geeksforgeeks.org/springboot/introduction-to-spring-boot/

getbootstrap. (12 de 09 de 2022). *getbootstrap*. Obtenido de getbootstrap: https://getbootstrap.com/?utm\_source=chatgpt.com

github. (01 de 01 de 2012). *github*. Obtenido de github: https://github.com/twbs/bootstrap?utm\_source

Mark Otto, J. T. (12 de 12 de 2020). *Mark Otto, Jacob Thornton*. Obtenido de Mark Otto, Jacob Thornton: https://en.wikipedia.org/wiki/Bootstrap\_(front-end\_framework)?utm\_source=chatgpt.com

Microsoft. (19 de 02 de 2023). *Microsoft*. Obtenido de Microsoft: https://visualstudio.microsoft.com/es/?utm\_source

ProgVisualEstudio. (22 de 07 de 2024). *ProgVisualEstudio*. Obtenido de ProgVisualEstudio: https://progvisualestudio.wordpress.com/marco-teorico/

Ulloa-Ulloa, D. (12 de 12 de 2012). *researchgate*. Obtenido de researchgate: https://www.researchgate.net/publication/393381452\_Metodologias\_agiles\_Scrum\_para\_la\_innovacion\_en\_los\_procesos\_crediticios

Wikipedia. (22 de 08 de 2019). *Wikipedia*. Obtenido de Wikipedia: https://en.wikipedia.org/wiki/Ionic\_(mobile\_app\_framework)

Wikipedia. (30 de 08 de 2019). *Wikipedia*. Obtenido de Wikipedia: https://en.wikipedia.org/wiki/Spring\_Boot?utm\_source