**MODULO:**

**Aplicación de metodologías ágiles y control de calidad de software**

**DOCENTE:**

**Heber Ernesto Mijango Vásquez.**

**ESTUDIANTES:**

**Manuel Antonio Abrego Alfaro.**

**Yudith Edith Hernández Monterroza.**

**GRUPO:**

**DSW 21” B”**

**TEMA:**

**Avance 2**

**FECHA DE ENTREGA:**

**12 de septiembre de 2024**

**Índice**

[**Introducción** 3](#_Toc180605876)

[**Descripción del proyecto** 4](#_Toc180605877)

[**Objetivo General:** 5](#_Toc180605878)

[**Objetivo Específico:** 5](#_Toc180605879)

[**Alcances:** 6](#_Toc180605880)

[**Limitaciones:** 7](#_Toc180605881)

[**Diseño de bases de datos** 8](#_Toc180605882)

[**Planificación del proyecto** 10](#_Toc180605883)

[**Los requerimientos** 13](#_Toc180605884)

[**Metodología seleccionada por el equipo de desarrollo** 14](#_Toc180605885)

[**Historias de usuario con sus criterios de aceptación** 14](#_Toc180605886)

[**Planificación por cada sprint (Sprint planning)** 18](#_Toc180605887)

[**Cronograma de actividades** 21](#_Toc180605888)

# **Introducción**

Este informe documenta el ciclo de vida del software para el proyecto "Registro de entradas y salidas de artículos", un sistema diseñado para optimizar la gestión de inventarios a través del control preciso de entradas y salidas de artículos. El sistema permite registrar artículos con información completa como nombre, descripción y cantidad, realizar un seguimiento en tiempo real del flujo de inventario, y generar informes y reportes en formato PDF para análisis y gestión.

El diseño de la base de datos que soporta este sistema incluye tablas para gestionar los artículos, proveedores, usuarios, movimientos de inventario y la generación de informes. Cada tabla está interrelacionada para asegurar un control eficiente del inventario, permitiendo el registro de entradas y salidas, el seguimiento de stock, y la trazabilidad de los movimientos. Este diseño facilita la generación de reportes detallados y el análisis del historial de inventarios.

Desarrollado bajo metodologías ágiles, el proyecto se basa en la colaboración continua, con entregas incrementales y pruebas aplicadas en cada fase para garantizar que el software cumpla con los estándares de calidad. Este enfoque asegura una solución tecnológica innovadora que optimiza la gestión de inventarios y se adapta a las necesidades específicas del cliente.

# **Descripción del proyecto**

Registro de entradas y salidas de artículos

. Descripción: Sistema para gestionar el inventario de artículos con control de entradas y

salidas.

Funcionalidades:

Registro detallado de artículos incluyendo nombre, descripción y cantidad.

Seguimiento de entradas y salidas de artículos.

Generacion de informes de inventario para análisis y gestión.

Generar reportes en pdf

# **Objetivo General:**

Desarrollar un sistema de "Registro de entradas y salidas de artículos" que optimice la gestión de inventarios mediante una solución tecnológica innovadora, aplicando metodologías ágiles de desarrollo para asegurar un enfoque colaborativo y eficaz.

# **Objetivo Específico:**

1. Documentar las etapas del ciclo de vida del software, desde la concepción hasta la entrega final.

2. Implementar metodologías ágiles para la entrega continua de productos incrementales y adaptación a cambios.

3. Utilizar herramientas colaborativas para mejorar la comunicación y eficiencia del equipo de desarrollo.

4. Aplicar pruebas específicas en cada etapa para asegurar la calidad del software.

# **Alcances:**

1. Registro Detallado de Artículos:

El sistema permitirá registrar artículos con detalles como nombre, descripción, y cantidad disponible.

Facilitará la actualización y modificación de los datos de los artículos de forma eficiente.

2. Seguimiento de Entradas y Salidas:

Controlará en tiempo real las entradas y salidas de los artículos, permitiendo un seguimiento detallado del flujo de inventario.

Mantendrá un historial de todas las transacciones realizadas, lo que ayudará a identificar patrones de consumo y necesidades de reabastecimiento.

3. Generación de Informes de Inventario:

Proporcionará informes de inventario actualizados para facilitar el análisis y la gestión de los artículos almacenados.

Permitirá a los usuarios exportar estos informes en varios formatos, como PDF, para uso interno y auditorías.

4. Generación de Reportes en PDF:

Generará reportes personalizados en formato PDF que incluyen información sobre el inventario, entradas y salidas, y otros detalles relevantes.

Permitirá la distribución de reportes a través de correo electrónico o impresión para su análisis.

# **Limitaciones:**

1. Dependencia del Usuario para la Precisión de Datos:

La efectividad del sistema depende de la precisión de los datos ingresados por los usuarios. Errores en el registro de artículos o transacciones pueden afectar la exactitud del inventario.

2. Capacitación del Usuario:

Es necesario que los usuarios tengan conocimientos básicos en el manejo del sistema, lo que podría requerir capacitación inicial.

3. Limitaciones Técnicas:

Dependencia de una infraestructura tecnológica adecuada (servidores, dispositivos de entrada de datos, etc.) para su funcionamiento óptimo.

El rendimiento del sistema podría verse afectado por un alto volumen de datos si no se optimiza correctamente la base de datos.

4. Alcance de los Reportes:

Los reportes generados en PDF pueden no ser completamente personalizables según necesidades específicas de cada usuario o empresa.

No se incluyen funcionalidades avanzadas de análisis como gráficos dinámicos o integración directa con herramientas de análisis de datos externas.

# **Diseño de bases de datos**

Una captura de pantalla de un celular

Descripción generada automáticamente con confianza media

De proveedor sale a la entrada de artículo

De entrada, de artículo sale a categoría

De artículos vamos salida de artículo

De artículo vamos a historia de inventario

De artículo vamos a movimiento inventario

De movimiento de inventario vamos a informe

Y de movimiento inventario vamos a usuario

CREATE TABLE articulo (

id int NOT NULL AUTO\_INCREMENT PRIMARY KEY,

id\_categoria int,

nombre varchar(255),

descripcion varchar(255),

precio\_unitario double,

stock\_actual int,

fecha\_creacion date DEFAULT NULL

);

ALTER TABLE articulo ADD CONSTRAINT articulo\_id\_fk FOREIGN KEY (id\_categoria) REFERENCES categoria (id);

# **Planificación del proyecto**

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Organización del equipo de trabajo para el levantamiento de requerimientos**

* **propietario**
* **Tareas**
* Planificación del proyectó
* Organizar cronograma de actividad de proyecto (trello)
* Comunicación entre el equipo encargado
* Requerimientos analiza
* El realizar entrevistas los jefes y a los administradores
* Requerimientos de documentos de caso uso o manual de usuario
* La validación de información recolectada
* **Desarrolladores**
* **Planificación**
* Coincidir en las reuniones previas de los requerimientos
* El desarrollo del sistema según lo implementado validos
* La creación de un diagrama de diseño técnico
* **Pruebas**
* **Actividades**
* Mantener los requerimientos y la claridad de testabilidad
* Hacer pruebas para su funcionalidad
* Crear reporte y errores a
* encontrar mediante las pruebas
* **Documentación**
* Registro de activos fijos nombre, descripción y valor.
* Seguimiento de depreciación de activos fijos.
* Generación de informes de inventario y depreciación.
* Generar reportes en pdf
* **Revisión**
* **Acciones**
* Crear actividades para la revisión de equipo
* Retroalimentación y las opciones del equipo
* Hora sesión 1-2
* **Aprobación**
* Verificar la presentación final de los requerimientos
* Alcanzar la aprobación de los procesos de desarrollo
* **Herramientas a utilizar**
* **Documentación**
* Word para documentos de requerimiento del prototipo
* **Gestión de las tareas**
* La configuración de trello para las tareas
* **Comunicación**
* Teamviwer para hacer la comunicación en tiempo real y resolver dudas
* **Prototipos**
* La creación de los prototipos visuales de las interfaces

Requerimientos

**Configuración**

* Instalar Laravel: composer create-project --prefer-dist laravel/laravel nombre proyecto.
* Configurar el servidor de desarrollo con Homestead o Docker.
* Configurar la base de datos en el archivo. env.
* **La estructura de proyecto**
* Hacer migraciones de nuestra tabla
* Desarrollo de los modelos y sus relaciones para manejo de datos
* Se implementará los modelos mediante comando de laravel
* **Desarrollo de las funcionalidades**
* Registro del artículo se creará un formulario del cual se hará guardado de articulo y actualización
* Gestión de administrador se le implementara y autenticara la autorización
* Salida de articulo se crear un formulario el cual permitirá el registro de salida
* Inventario el cual tendrá el registro de nuestro producto de entrada y salida
* **Integración**
* Se usará Blade templates para las vistas
* Se aplicará estilo de boostrap css para el diseño y lo resposivo
* **Pruebas**
* Se desarrollar la prueba unitaria con comandos de laravel hacia para ´probar la funcionalidad
* Realizar pruebas manuales y automatizadas para la aseguración que el sistema funcione adecuadamente

* **Entregables**
* Documentación completa requerimientos funcionales y no
* Diagrama la visualización del diseño de la interfaz
* Aplicaciones desplegables el sistema funcional y sus características
* Manuales y guías la documentación para administrador y usuario finales de desarrolladores
* **Seguimiento y control**
* Revisión de los avances proyecto y los ajustes
* Los procesos específicos un manejo de documentos de cambios en lo implementado

# **Los requerimientos**

Para el desarrollo del proyecto de 'Registro de entradas y salidas de artículos', se requieren una serie de especificaciones técnicas tanto de software como de hardware, para asegurar un entorno de desarrollo eficiente y estable.

Laravel

Es un framework de php el cual nos facilita un desarrollo de aplicaciones web funciona para desarrollar aplicaciones web.

Se usa el modelo mvc wue modelo controlador-vista del cual se proporciona un código basado en una estructura que se empieza a desarrollar.

Podemos usarlos utilizando código de php y aplicando comandos

* **Software**
* **Sistemas operativos** 
  + Windows 10/11 para desarrollar un entorno web en laravel
* **Servidor web local** 
  + XAMPP para los levantamientos de nuestros servidores locales
* **Leguajes de programación** 
  + Laravel el cual es requerido php 8.01 así asegurado un buen desarrollo
* **Base de datos** 
  + MySQL el entorno en el cual trabajaremos nuestros datos a desarrollar
* **Editor de código** 
  + En estas utilizaremos un editor de texto como visual studio code para desarrollar vuestro proyecto
* **Hardware**
* **Procesadore**
* Un Intel Core i5/i7 AMD con una Ryzen 5 es una buena opción para el desarrollo del proyecto
* **Memoria RAM**
  + 8 GB RAM mínimo para que el desarrollo pueda ser fluido aun que si un proyecto es más grande sería recomendable una de 16 GB o ser necesario mas
* **Almacenamiento** 
  + Un 512 seria lo más preferible para trabajar con más confianza
* **Tarjeta grafica** 
  + El CPU no es tan necesario que sea una tan avanzada

# **Metodología seleccionada por el equipo de desarrollo**

Scrum

Estés es un marco de trabajo que nos ayudara con la planificación de tiempo de nuestro proyecto el cual nos organizara en tarea de semanas así gestionando nuestras tareas de equipo en plazo definido

# **Historias de usuario con sus criterios de aceptación**

Primera pantalla

Registro de usuario

Yo: Yudith usuario final registro de entrada

Quiero: un software que me permita el registro de entrada de salida de articulo

Para: para realizar la entrada de articulo para el inventario

Criterio de Aceptación

El registro de entrada del artículo

El sistema le pedirás los datos, nombre, artículo, fecha entrada, costo usuario

El sistema permitirá el registro a la base de datos del artículo

Debe estar conectado a la base de datos de sistema

El sistema mostrar un mensaje a la hora de quede un campo vacío

Genera reporte generar de la entrada de articulo

Debe generar un informe de entrada del articulo

Cerrar

Segunda pantalla

Registro de usuario

Yo: Antonio usuario final registro de salida

Quiero: un software que me permita el registro de entrada de salida de artículo

Para: para realizar la salida de articulo para el inventario

Criterio de Aceptación

El registro de salida del artículo

El sistema le pedirás los datos, nombre, artículo, fecha salida, costo usuario

El sistema permitirá el registro a la base de datos del artículo

Debe estar conectado a la base de datos de sistema

Genera reporte generar de la salida de articulo

Debe generar un informe de salida del articulo

El sistema mostrar un mensaje a la hora de quede un campo vacío

Cerrar

# **Planificación por cada sprint (Sprint planning)**

Sprint 1: Análisis del Proyecto

Duración: 1 semana

1. Reunir requisitos del proyecto
   * Entrevistar a los interesados.
   * Documentar requisitos funcionales y no funcionales.
   * Aprobación de requisitos con las partes relevantes.
2. Definir especificaciones funcionales y no funcionales
   * Redactar especificaciones funcionales.
   * Redactar especificaciones no funcionales.
3. Análisis de la base de datos
   * Identificar entidades y atributos.
   * Establecer relaciones entre las tablas.
4. Establecer criterios de éxito y alcance del proyecto
   * Definir los criterios de éxito.
   * Acordar el alcance con las partes relevantes.

Sprint 2: Diseño del Sistema

Duración: 1 semana

1. Diseño de la base de datos
   * Crear diagrama entidad-relación.
   * Revisar y validar el diseño con el equipo.
2. Diseño de la interfaz de usuario
   * Crear wireframes de las pantallas principales.
   * Validar el diseño con usuarios o stakeholders.
3. Diseño de la arquitectura del sistema
   * Definir la arquitectura del sistema (Backend y Frontend).
   * Seleccionar herramientas y tecnologías.
   * Crear un plan de desarrollo.

Sprint 3: Implementación Backend Inicial

Duración: 2 semanas

1. Desarrollo del Backend
   * Implementar CRUD de artículos.
   * Implementar lógica para el registro de entradas y salidas de artículos.
   * Crear APIs para gestionar datos de inventario.
2. Pruebas iniciales
   * Realizar pruebas unitarias.
   * Realizar pruebas de integración.
   * Documentar y corregir errores iniciales.

Sprint 4: Implementación Frontend y Reportes

Duración: 2 semanas

1. Desarrollo del Frontend
   * Implementar interfaz para el registro de artículos.
   * Implementar interfaz para el registro de entradas y salidas.
   * Conectar frontend con backend.
2. Implementación de generación de reportes en PDF
   * Diseñar plantillas para informes de inventario.
   * Implementar la generación de reportes en PDF.
   * Validar la precisión de los reportes generados.

Sprint 5: Pruebas Completas

Duración: 1 semana

1. Pruebas funcionales
   * Probar el correcto funcionamiento del registro de artículos y movimientos.
   * Validar la generación de informes en PDF.
2. Pruebas de usabilidad
   * Realizar pruebas con usuarios finales para evaluar la facilidad de uso.
   * Recopilar comentarios de los usuarios.
3. Pruebas de rendimiento y seguridad
   * Realizar pruebas de carga y optimizar el rendimiento.
   * Probar la seguridad del sistema y proteger datos sensibles.

Sprint 6: Implantación

Duración: 1 semana

1. Preparación del entorno de producción
   * Configurar el servidor y migrar la base de datos a producción.
2. Despliegue del sistema
   * Desplegar el sistema en producción.
   * Verificar su correcto funcionamiento en producción.
3. Capacitación de usuarios
   * Crear material de capacitación.
   * Realizar sesiones de formación para los usuarios.

Sprint 7: Mantenimiento y Soporte

Duración: 1 semana (y continuo)

1. Monitoreo y soporte
   * Configurar herramientas de monitoreo para el sistema.
   * Establecer un canal de soporte para resolver problemas técnicos.
2. Actualizaciones y mejoras
   * Recopilar comentarios de usuarios.
   * Planificar y aplicar mejoras según las necesidades.
3. Corrección de errores
   * Identificar y solucionar errores detectados en el sistema.
   * Implementar parches y actualizaciones.

# Product backlog de todo el proyecto

* Investigación sobre el tema
* Objetivos de nuestro proyecto
* Alcances y competencia
* Sistema de la base de datos
* Estructura y planificación del proyecto
* Desarrollo del proyecto
* Implantación

# Estimación de tiempo por cada actividad

|  |  |  |
| --- | --- | --- |
| Actividades | Descripción | Tiempo estimado |
| **Requisitos del sistema** | Reunir y documentar los requisitos funcionales y no funcionales. | 1 semana |
| **actores del sistema** | Identificar todos los usuarios y roles del sistema | 3 días |
| **Estructura de la base de datos** | Modelar la base de datos, definir tablas y relaciones. | |  | | --- | | 1 semana | |
| **Crear los modelos para el sistema** | Desarrollo del modelo de datos para gestionar préstamos. y otros | 2 semanas |
| **Desarrollo autenticación de usuarios** | Implementar el sistema de registro y autenticación. | 2 días |
| **Crear los controladore y desarrollar su funcionamiento** | Crear la lógica de negocio para manejar préstamos. | 3 semanas |
| **Implementar relaciones entre usuarios y administrador de entrada y salida de articulo** | Establecer relaciones en la base de datos y en el código | 4 semanas |
| **crear diseño de usuario** | Diseñar y desarrollar la interfaz para gestión de equipos y préstamos | 3 días |
| **Verificar la funcionalidad de los formularios** | Realizar pruebas de aceptación y funcionalidad. | 3 semanas |
| **Comprobar las migraciones de la base de datos** | Asegurarse de que las migraciones estén correctas. | 2 días |
| **Implementación de la funcionalidad de roles** | Establecer roles y permisos de acceso. | 4 semanas |
| **Completo el diseño del menú de navegación** | Crear y probar el menú de navegación del sistema. | 3 semanas |
| **Pruebas finales y revisión del sistema** | Establecer roles y permisos de acceso. | 2 semanas |

# **Cronograma de actividades**

**1. Análisis (Duración: 2 semanas)**

* **Reunir requisitos del proyecto** (3 días)
  + Entrevistar a los interesados del proyecto.
  + Documentar los requisitos.
  + Aprobar los requisitos con todas las partes relevantes.
* **Definir especificaciones funcionales y no funcionales** (3 días)
  + Redactar especificaciones funcionales.
  + Redactar especificaciones no funcionales.
* **Análisis de la base de datos** (2 días)
  + Identificar entidades y atributos.
  + Establecer relaciones entre tablas.
* **Establecer criterios de éxito y alcance del proyecto** (2 días)
  + Definir criterios de éxito.
  + Acordar el alcance del proyecto con todas las partes relevantes.

**2. Diseño (Duración: 1 semana)**

* **Diseño de la base de datos** (3 días)
  + Crear diagrama entidad-relación.
  + Revisar y validar el diseño de la base de datos.
* **Diseño de la interfaz de usuario** (2 días)
  + Crear wireframes.
  + Validar diseño con las personas involucradas.
* **Diseño de la arquitectura del sistema** (2 días)
  + Definir la arquitectura del sistema.
  + Seleccionar herramientas y tecnologías.
  + Crear un plan de desarrollo.

**3. Implementación (Duración: 3 semanas)**

* **Desarrollo del backend** (1 semana)
  + Implementar CRUD de artículos.
  + Implementar registro de entradas y salidas.
  + Crear APIs para el manejo de datos.
* **Desarrollo del frontend** (1 semana)
  + Implementar interfaz de registro de artículos.
  + Implementar interfaz de entradas y salidas.
  + Conectar frontend con backend.
* **Implementación de generación de reportes en PDF** (4 días)
  + Diseñar plantillas de reportes.
  + Implementar generación de PDF.
  + Validar la precisión de los reportes.
* **Pruebas iniciales** (3 días)
  + Realizar pruebas unitarias.
  + Realizar pruebas de integración.
  + Documentar y corregir errores encontrados.

**4. Prueba (Duración: 1 semana)**

* **Pruebas funcionales** (2 días)
  + Probar registro de artículos.
  + Probar registro de entradas y salidas.
  + Validar generación de informes.
* **Pruebas de usabilidad** (1 día)
  + Realizar pruebas con usuarios finales.
  + Recopilar comentarios de los usuarios.
* **Pruebas de rendimiento** (1 día)
  + Realizar pruebas de carga.
  + Optimizar rendimiento según resultados.
* **Pruebas de seguridad** (1 día)
  + Realizar pruebas de seguridad.
  + Implementar medidas de seguridad necesarias.
* **Pruebas de aceptación del usuario** (2 días)
  + Realizar demostraciones a usuarios.
  + Recibir aprobación final.

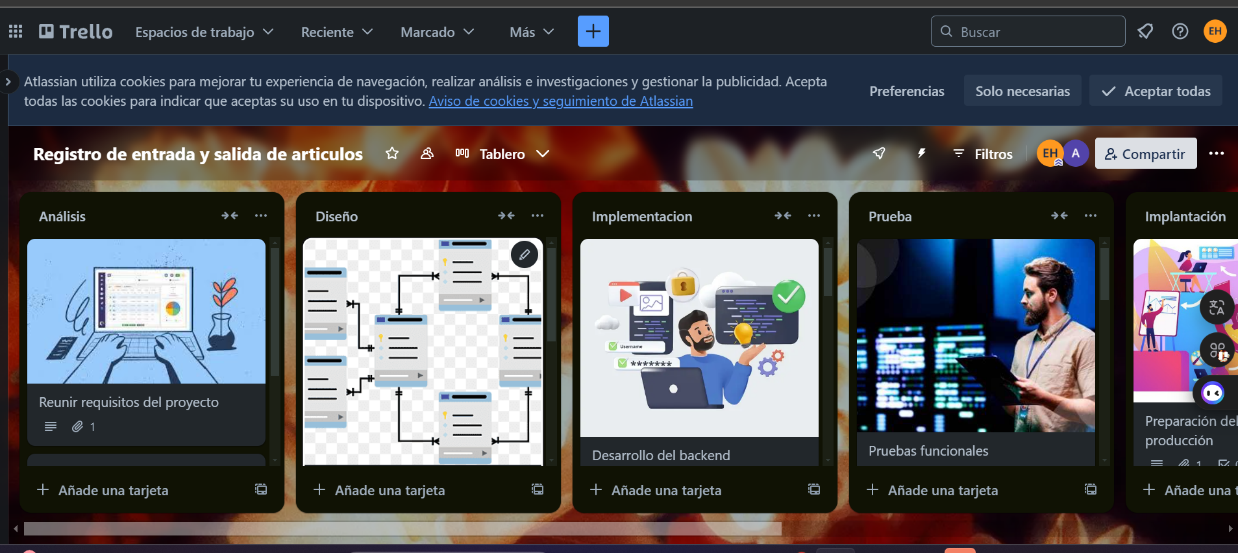
**5. Implantación (Duración: 1 semana)**

* **Preparación del entorno de producción** (2 días)
  + Configurar servidor.
  + Migrar base de datos a producción.
* **Despliegue del sistema** (2 días)
  + Realizar despliegue.
  + Verificar funcionamiento en producción.
* **Capacitación de usuarios** (2 días)
  + Crear material de capacitación.
  + Realizar sesiones de formación.
* **Documentación** (1 día)
  + Documentar código y funcionalidades.
  + Crear manual de usuario.

**6. Mantenimiento (Duración: indefinida)**

* **Monitoreo y soporte**
  + Configurar herramientas de monitoreo.
  + Establecer canal de soporte.
* **Actualizaciones y mejoras**
  + Recopilar comentarios de usuarios.
  + Planificar y aplicar mejoras.
* **Corrección de errores**
  + Identificar y documentar errores.
  + Implementar correcciones.

# Organización de actividades según tablero de trello



# Roles del equpo de desarrollo actividades asignadas por cada sprint

**Maestro Scrum (Yudith Hernández)**

* El trabajo que facilita reuniones y una planificación de los sprint para el equipo y el desarrollo de las funcionalidades del sistema de artículos de entrada y salida
* Estuvo al tanto de los miembros del informe del equipo sobre los avances y los impedimentos que se llevaron acabó en el sistema.
* Comprobar los sprint para que el equipo estes al tanto del sistema de artículos entrada y salida

**Propietario del producto (Antonio abrego)**

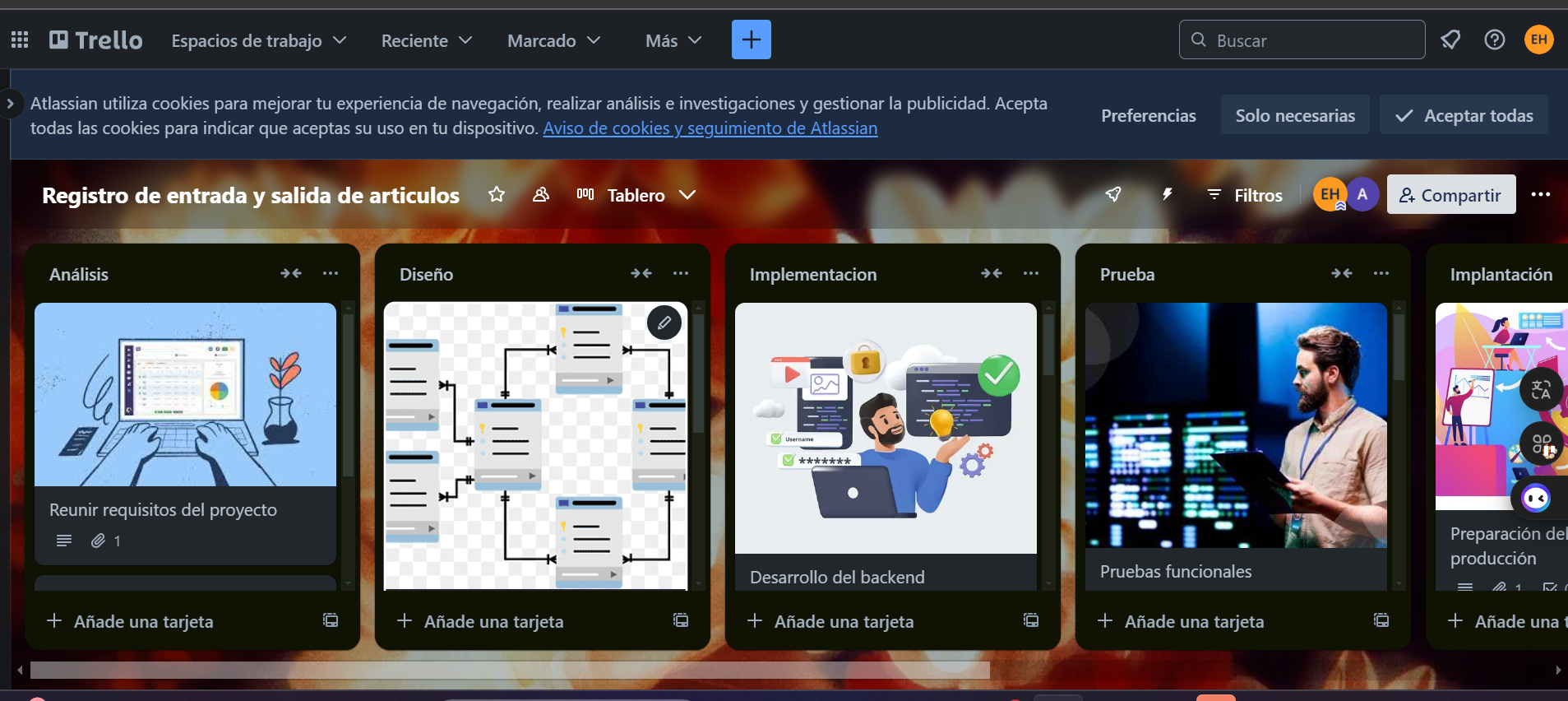
* Estructurar los requerimientos del producto de entrada y salida de articulo así verificando su funcionamiento
* Verificación de las funcionalidades y mejoras así interactuando con la interfaz de usuario
* Se garantiza que las funcionalidades se cumplan y sea entendible.

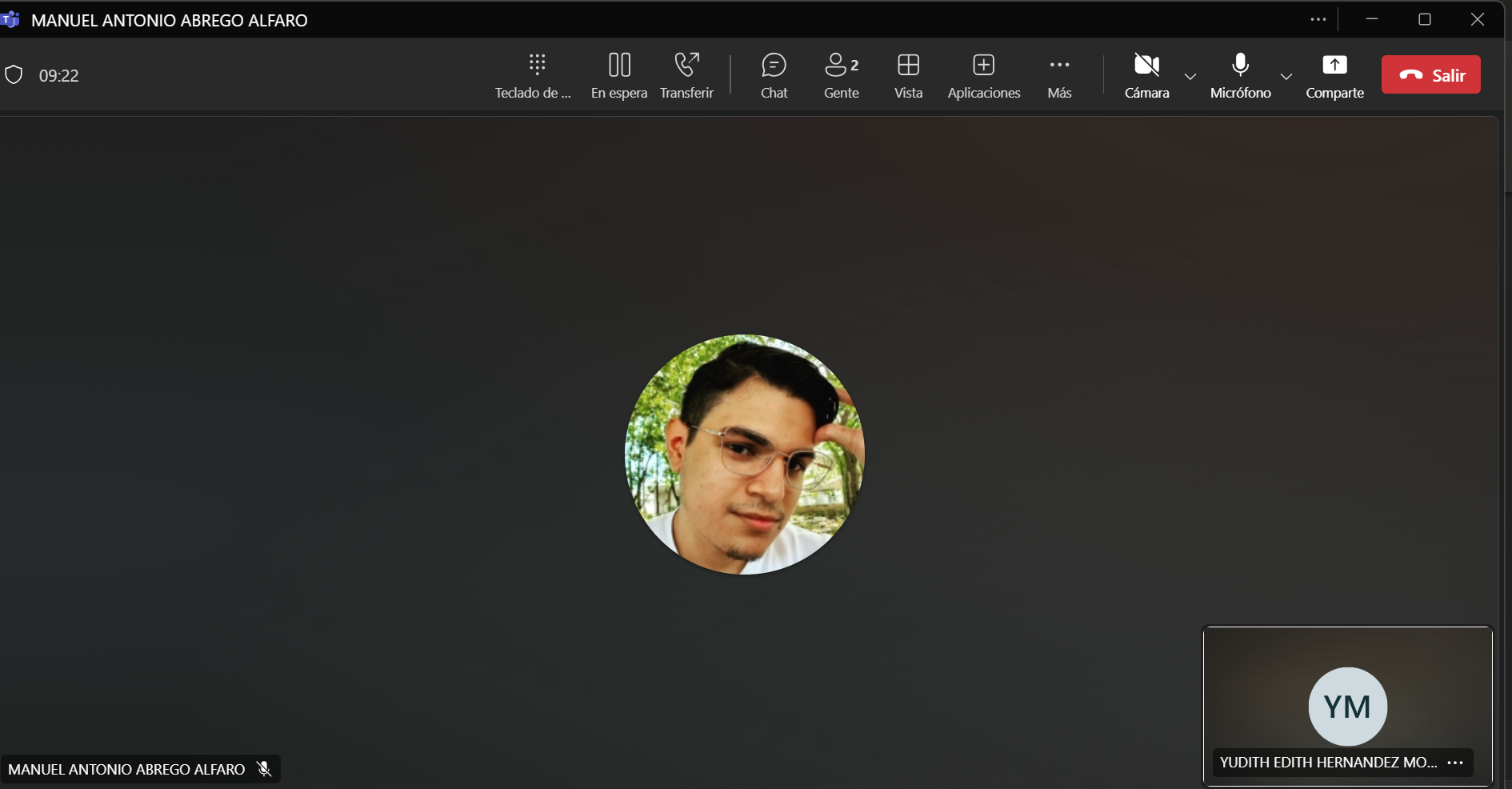
**Equipo de scrum**

**Product Owner (Dueño del Producto)**:

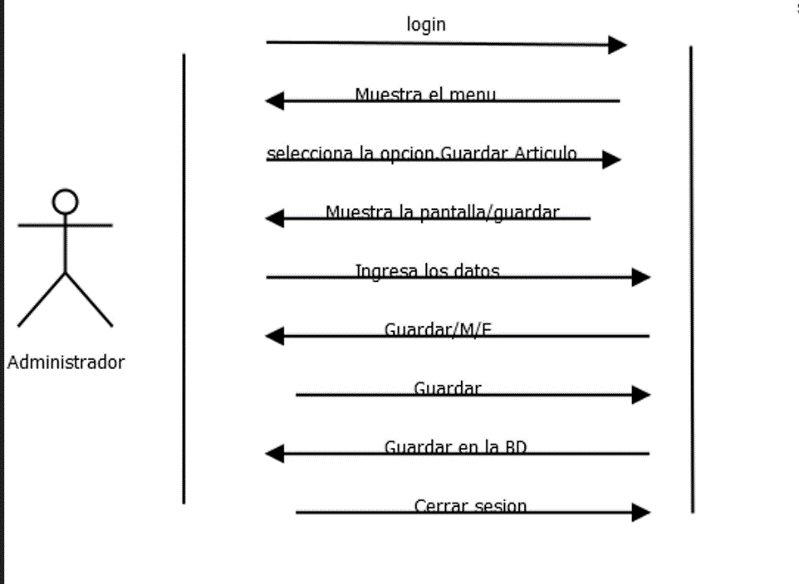
* Es el encargado de definir y priorizar las características del sistema de entrada y salida de articulo para mejor desarrollo de lo que se necesite para la empresa
* **Scrum Máster**:
* Estés facilita los procesos de sprint asi eliminado los obstáculos para el equipo desarrollo del sistema de entrada y salida de articulo

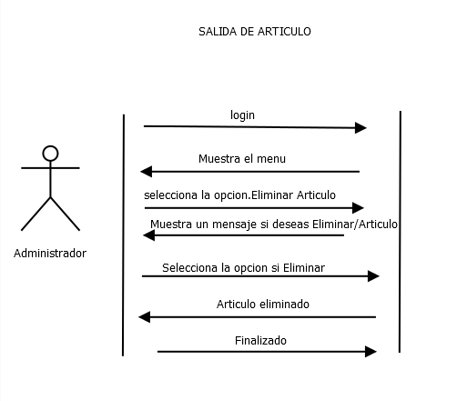
# Uso de herramientas colaborativas



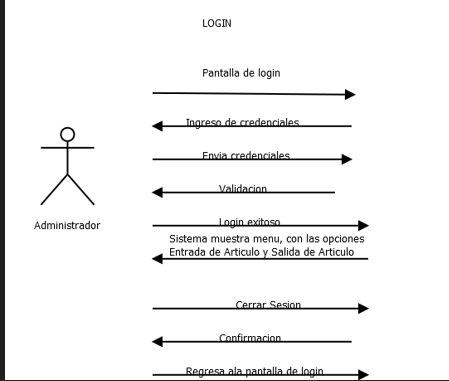


# Diagrama de casos de uso y diccionario









|  |  |
| --- | --- |
| Nombre | Entrada de Artículo |
| Autor | Autor Ejemplo |
| Fecha | 17/10/2024 |
| Descripción | El administrador puede ingresar un nuevo artículo al sistema con todos sus detalles. |
| Actores | Administrador |
| Pre-condiciones | El administrador debe estar logueado en el sistema. |
| Flujo Normal | 1. El administrador accede al sistema. 2. El sistema muestra el menú principal. 3. El administrador selecciona "Entrada de Artículos". 4. El sistema muestra la pantalla para ingresar datos del artículo. 5. El administrador ingresa los datos del artículo. 6. El sistema guarda los datos en la base de datos. 7. El sistema confirma el guardado. 8. El administrador cierra sesión. |
| Flujo Alternativo | Si el artículo ya existe en la base de datos, el sistema notifica al administrador y no permite duplicados. |
| Excepciones | Si el administrador no completa todos los campos obligatorios, el sistema muestra un mensaje de error. |
| Post-condiciones | El artículo es almacenado correctamente en la base de datos. |

|  |  |
| --- | --- |
| Nombre | Salida de Artículo |
| Autor | Administrador |
| Fecha | 17/10/2024 |
| Descripción | El administrador elimina artículos del sistema. |
| Actores | Administrador |
| Pre-condiciones | El artículo debe estar registrado en el sistema. |
| Flujo Normal | 1. Login 2. El sistema muestra el menú. 3. Selecciona la opción 'Eliminar'. 4. El sistema pide confirmación. 5. El administrador confirma la eliminación. 6. El artículo es eliminado. 7. Finaliza y cierra sesión. |
| Flujo Alternativo | |  | | --- | | **Error en la eliminación** |  |  |  | | --- | --- | |  | 1. Login. |  |  |  | | --- | --- | |  | 2. El sistema muestra el menú. |  |  |  | | --- | --- | |  | 3. Selecciona la opción "Salida de Artículos". |  |  |  | | --- | --- | |  | 4. El administrador confirma la eliminación del artículo. |  |  |  | | --- | --- | |  | 5. El sistema intenta eliminar el artículo. |  |  |  | | --- | --- | |  | 6. Ocurre un error al intentar eliminar el artículo. |  |  |  | | --- | --- | |  | 7. El sistema muestra un mensaje de error. |  |  |  | | --- | --- | |  | 8. El administrador puede optar por: |  |  |  | | --- | --- | |  | - Reintentar la eliminación. |  |  |  | | --- | --- | |  | - Regresar al menú principal. | |
| Excepciones | Si el artículo no existe, se muestra un mensaje de error. |
| Post-condiciones | |  | | --- | | **Flujo Normal:** El artículo ha sido eliminado correctamente. |  |  |  | | --- | --- | |  | **Flujos Alternativo:** |  |  |  | | --- | --- | |  | - Si hay un error, el artículo permanece en el sistema. | |

|  |  |
| --- | --- |
| Nombre | Informe |
| Autor | Autor Ejemplo |
| Fecha | 17/10/2024 |
| Descripción | El administrador puede generar un informe del inventario de artículos. |
| Actores | Administrador |
| Pre-condiciones | El administrador debe tener acceso a los datos del inventario. |
| Flujo Normal | 1. El administrador accede al sistema. 2. El sistema muestra el menú principal. 3. El administrador selecciona "Generar Informe". 4. El sistema muestra opciones de filtros para generar el informe. 5. El administrador selecciona los filtros y genera el informe. 6. El sistema genera el informe y lo muestra al administrador. 7. El administrador revisa y guarda el informe. |
| Flujo Alternativo | Si no hay artículos en el inventario, el sistema genera un informe vacío con un mensaje correspondiente. |
| Excepciones | Si el sistema encuentra un error al generar el informe, muestra un mensaje de error al administrador. |
| Post-condiciones | El informe es generado y guardado correctamente. |

|  |  |
| --- | --- |
| Nombre | Login |
| Autor | Autor Ejemplo |
| Fecha | 17/10/2024 |
| Descripción | Permite al administrador ingresar al sistema a través de sus credenciales. |
| Actores | Administrador |
| Pre-condiciones | El administrador debe tener credenciales válidas. |
| Flujo Normal | 1. El administrador accede a la pantalla de login. 2. Ingresa sus credenciales (usuario y contraseña). 3. El sistema envía las credenciales para validación. 4. El sistema valida las credenciales. 5. Si el login es exitoso, el sistema muestra el menú con las opciones: Entrada de artículo, Salida de artículo, Informe. 6. El administrador puede seleccionar una opción. 7. Al finalizar, el administrador puede cerrar sesión. |
| Flujo Alternativo | Si las credenciales son incorrectas, el sistema muestra un mensaje de error y permite volver a intentar. |
| Excepciones | Si hay un problema con la conexión a la base de datos, se muestra un mensaje de error. |
| Post-condiciones | El administrador ha iniciado sesión y tiene acceso al sistema. |