

UNIVERSIDAD MARIANO GALVEZ DE GUATEMALA

FACULTAD DE INGENIERIA EN SISTEMAS

ANALISIS DE SISTEMAS

ING. ROMAN CANCINOS

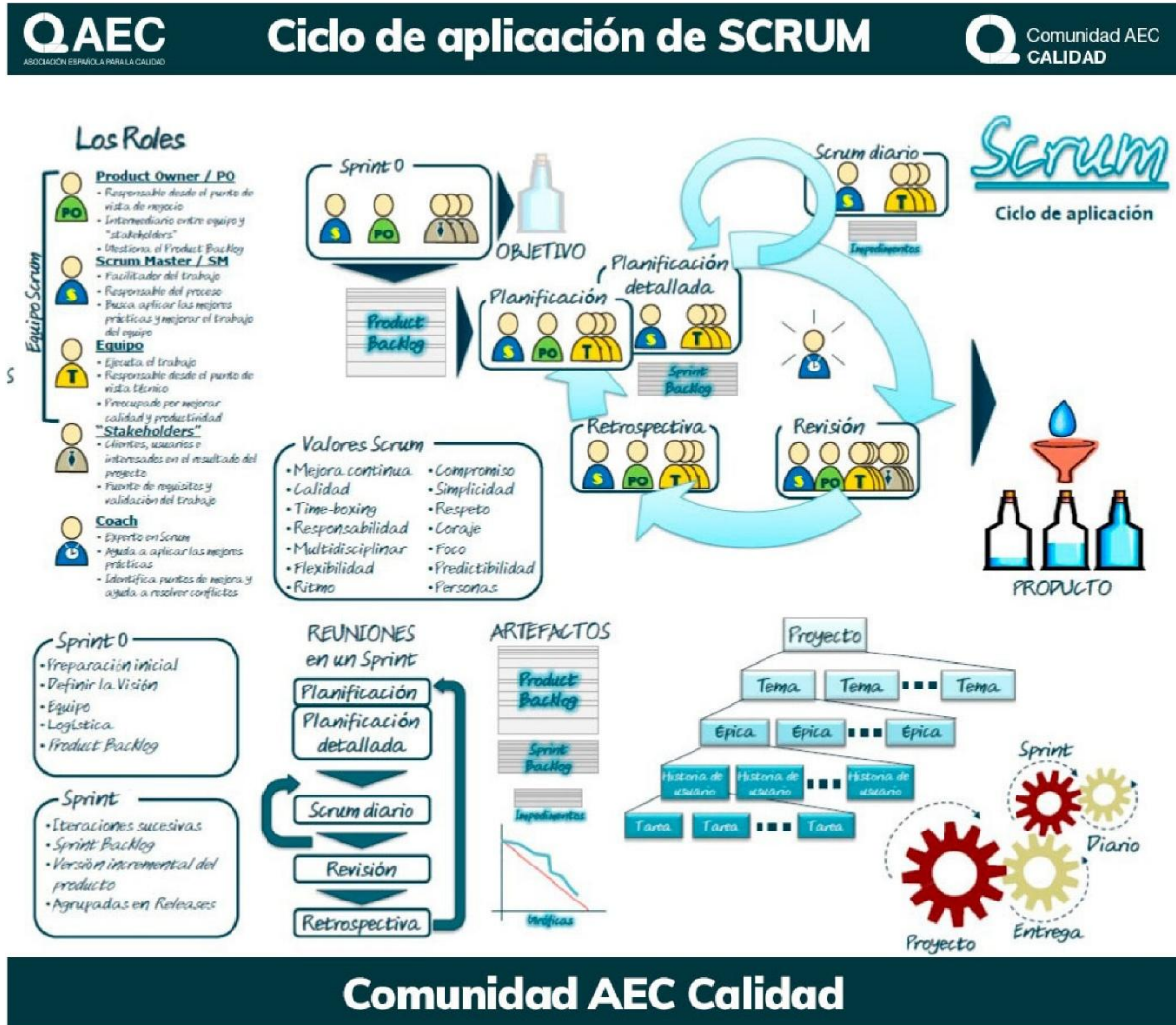


NOMBRE	NUMERO DE CARNE
Silvia Izabel Cotzajay Alvarez	7590-22-14192
Pablo Miguel Chávez Chocojay	7590-22-3566
Emerson Antonio Sec Prera	7590-22-7782
Gabriel Bejanmín Valdez De León	7590-22 11271

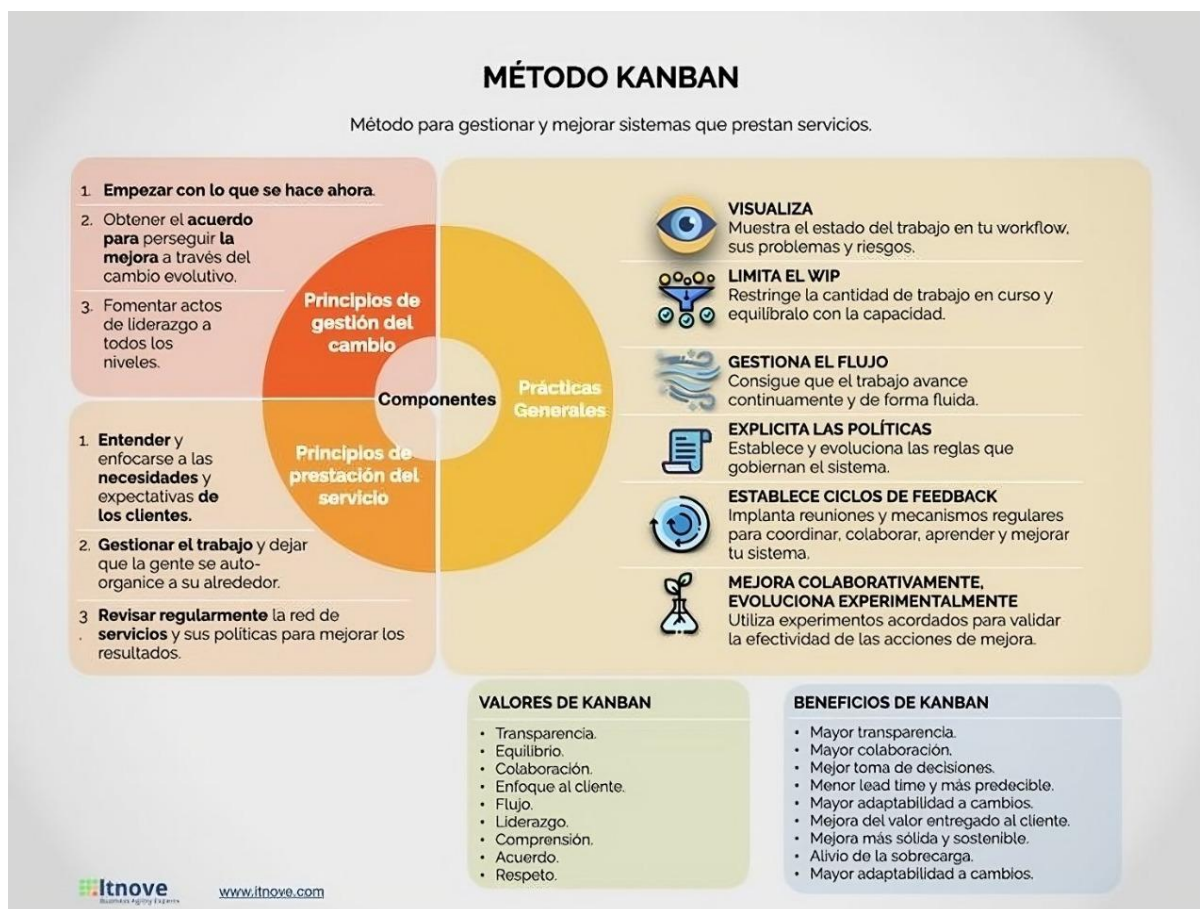
SCRUM

Guatemala, 10/05/2025

Fase 0: Preparación



Comunidad AEC Calidad



1 Fase 1: Inicio del Proyecto y Formación de Equipos

1.1 Introducción al Laboratorio

Introducción al Laboratorio: Metodologías Ágiles - Scrum y Kanban

El presente laboratorio tiene como finalidad comprender y aplicar los conceptos fundamentales de las metodologías ágiles Scrum y Kanban en el desarrollo de un proyecto práctico: "Tickers - Helpdesk". A través de una simulación colaborativa, se experimentarán los roles, artefactos y flujos de trabajo característicos de cada metodología, permitiendo identificar sus ventajas y adaptaciones en entornos reales.

Objetivos del Laboratorio:

Conocer los principios fundamentales de Scrum y Kanban.

Aplicar los conceptos de roles, eventos y artefactos en un entorno simulado.

Optimizar la gestión de tareas y el seguimiento del proyecto "Tickers - Helpdesk".

Identificar las diferencias y similitudes entre ambas metodologías ágiles.

Fomentar el trabajo en equipo y la colaboración efectiva en un entorno de proyecto ágil.

Dinámica del Laboratorio:

La dinámica se llevará a cabo de la siguiente forma:

Nos juntaremos en grupos de trabajo para facilitar la comprensión y aplicación de los conceptos.

Cada grupo asumirá roles definidos en Scrum: Product Owner, Scrum Master y Equipo de Desarrollo.

Se diseñará un Product Backlog para "Tickers - Helpdesk" y se planificarán Sprints semanales.

Se realizarán reuniones diarias de Daily Scrum para revisar avances y obstáculos.

Para la parte de Kanban, se montará un Tablero Kanban que refleje el flujo de tareas: Pendiente, En Proceso y Finalizado.

Proyecto a Desarrollar/Simular:

El proyecto principal de este laboratorio será "Tickers - Helpdesk", un sistema de gestión de tickets orientado al soporte técnico y atención al cliente. Durante el laboratorio se estructurará el proyecto, se organizarán los entregables y se optimizarán los procesos utilizando Scrum para la planificación iterativa y Kanban para el seguimiento visual del flujo de trabajo.

1.2 Formación de Equipos SCRUM:

Rol	Nombre(s)	Responsabilidades Principales
Product Owner (PO)	Silvia Cotzajay	- Definir y priorizar las funcionalidades del producto en el Product Backlog.
		- Asegurarse de que el equipo Scrum entienda claramente los requerimientos y expectativas.
		- Maximizar el valor del producto y gestionar la visión del proyecto.
		- Colaborar estrechamente con los stakeholders para reflejar sus necesidades en el backlog.
Scrum Master (SM)	Emerson Sec	- Facilitar las ceremonias Scrum: Sprint Planning, Daily Scrum, Sprint Review y Sprint Retrospective.
		- Eliminar impedimentos que

Rol	Nombre(s)	Responsabilidades Principales
		bloqueen el progreso del equipo.
		- Garantizar el cumplimiento de las prácticas ágiles y el marco Scrum.
		- Promover la autoorganización y el aprendizaje continuo en el equipo.
Development Team (DT)	Pablo Chavez, Gabriel Valdez Jaquelin Navichoc Sindy Acabal	- Desarrollar y entregar el incremento del producto en cada Sprint.
		- Participar activamente en las ceremonias Scrum y aportar ideas para la mejora del producto.
		- Colaborar con el Product Owner para clarificar requisitos técnicos.
		- Mantener la calidad del código y cumplir con los criterios de aceptación.

1.3 Visión del Producto y Creación del Product Backlog Inicial:

Proyecto: "Tickers - Helpdesk"

Visión del Producto:

Visión del Producto:

Proyecto: "Tickers - Helpdesk"

"Tickers - Helpdesk" tiene como objetivo desarrollar un sistema de gestión de tickets para un servicio de atención al cliente eficiente y organizado. Este sistema permitirá a los usuarios registrar, rastrear, y cerrar problemas de manera ágil, proporcionando a los agentes de soporte las herramientas necesarias para gestionar las solicitudes de forma efectiva.

El sistema contará con una interfaz sencilla e intuitiva, donde se podrán crear tickets de soporte, asignarlos a los agentes adecuados, hacer seguimiento del estado de cada ticket y asegurar una resolución satisfactoria para los clientes. Además, se implementará una función de notificaciones que permitirá a los clientes recibir actualizaciones sobre el progreso de sus solicitudes, garantizando una comunicación fluida y continua.

Objetivo principal:

Proveer una solución que facilite la gestión de incidencias, optimice la resolución de problemas y mejore la experiencia del cliente, asegurando que todos los tickets se resuelvan de manera eficiente y en el menor tiempo posible.

Beneficios esperados:

- Mejora en la eficiencia operativa del equipo de soporte.
- Reducción del tiempo de resolución de tickets.
- Mejor comunicación y transparencia para los clientes.
- Optimización en la asignación de tickets según especialización.

Público objetivo:

- Clientes que necesitan asistencia técnica o soporte.
- Agentes de soporte y administradores del sistema.

Product Backlog Inicial:**Historias de Usuario:****1. Prioridad Alta**

- **Historia de Usuario 1:**

Como agente de soporte, quiero poder registrar un nuevo ticket para poder gestionar las solicitudes de los clientes de manera eficiente.

Justificación: Este ítem es esencial para el flujo de trabajo del sistema y tiene un alto impacto en la productividad del equipo de soporte.

2. Prioridad Alta

- **Historia de Usuario 2:**

Como administrador del sistema, quiero poder asignar tickets a los agentes según

su especialización, para que cada problema sea tratado por el experto adecuado.

Justificación: La asignación eficiente de tickets mejora la atención al cliente y organiza el flujo de trabajo.

3. Prioridad Media

- **Historia de Usuario 3:**

Como cliente, quiero poder consultar el estado de mi ticket en tiempo real para saber si mi problema está siendo resuelto.

Justificación: Este ítem mejora la experiencia del cliente al ofrecer visibilidad y control sobre su solicitud.

4. Prioridad Media

- **Historia de Usuario 4:**

Como agente de soporte, quiero agregar notas internas a un ticket para dejar registros de mis intervenciones y facilitar la resolución.

Justificación: Ayuda en la documentación de los procesos de soporte y facilita el seguimiento de problemas complejos.

5. Prioridad Baja

- **Historia de Usuario 5:**

Como usuario, quiero recibir notificaciones por correo electrónico cuando mi ticket sea actualizado para estar al tanto del progreso.

Justificación: Mejora la comunicación con el usuario, pero es menos urgente que otras funciones.

Resultado esperado:

Un **Product Backlog** inicial priorizado para cada equipo, con las historias de usuario claramente definidas y categorizadas según su importancia. Las funcionalidades críticas (registrar tickets, asignación de tickets) se priorizan en la parte superior del backlog, mientras que las funciones secundarias (notificaciones y notas internas) quedan en la parte inferior.

2 Fase 2: Sprint Planning y Comienzo del Sprint

2.1 Sprint Planning:

Historia de Usuario

"Como agente de soporte, quiero poder registrar un nuevo ticket para poder gestionar las solicitudes de los clientes de manera eficiente."

Justificación:

Este ítem es esencial para el flujo de trabajo del sistema y tiene un alto impacto en la productividad del equipo de soporte.

Objetivo del Sprint

En este Sprint, nuestro equipo puede completar la historia de usuario:

"Como agente de soporte, quiero poder registrar un nuevo ticket para poder gestionar las solicitudes de los clientes de manera eficiente."

Consideramos que podemos terminar esta historia porque contamos con:

- El tiempo suficiente para el desarrollo y pruebas.
- El conocimiento técnico necesario en Frontend, Backend y manejo de bases de datos.
- La definición clara de los requisitos, realizada en conjunto con el Product Owner.

Hemos priorizado esta funcionalidad porque es la base del sistema Tickers - Helpdesk y es fundamental para optimizar la gestión de solicitudes de soporte, mejorando la experiencia tanto del cliente como del agente.

Tareas Planificadas

Para completar esta historia de usuario, la dividimos en las siguientes tareas:

1. Diseñar el formulario de creación de ticket (Frontend)
 - Creación de la interfaz amigable para el registro del ticket.
 - Definición de campos necesarios: Nombre del cliente, Descripción del problema,

Prioridad, Fecha, Estado, entre otros.

2. Crear la base de datos y tabla correspondiente (Backend)
 - Definición del modelo de datos para almacenar los tickets.
 - Configuración de relaciones en la base de datos si es necesario.
3. Programar el almacenamiento de datos del ticket (Backend)
 - Implementación del servicio para guardar los datos en la base de datos.
 - Manejo de errores en el guardado y confirmación de éxito.
4. Validar los campos del formulario (Frontend y Backend)
 - Verificación de campos requeridos, tipo de datos y formatos.
 - Mensajes de error para datos inválidos o faltantes.
5. Hacer pruebas funcionales de creación de ticket
 - Pruebas unitarias y de integración para asegurar el correcto funcionamiento.
 - Validación de datos y almacenamiento efectivo.
6. Documentar brevemente el flujo de creación
 - Explicación paso a paso de cómo funciona el proceso de creación del ticket.
 - Guía para el usuario y el equipo de soporte técnico.

Cada tarea será asignada a un integrante del equipo, y al final del Sprint entregaremos un prototipo funcional que permita:

- Registrar un nuevo ticket.
- Almacenar correctamente los datos en la base de datos.
- Validar los campos ingresados.
- Generar un ticket de soporte accesible para el seguimiento.

2.2 Definición de "Terminado" (Definition of Done - DoD) :

Para la historia de usuario:

"Como agente de soporte, quiero poder registrar un nuevo ticket para poder gestionar las solicitudes de los clientes de manera eficiente."

El equipo define que esta historia estará "Terminada" cuando se cumplan todos los siguientes criterios:

El formulario de registro de tickets está diseñado con los campos requeridos: nombre del cliente, descripción del problema, prioridad y fecha.

La información ingresada se almacena correctamente en la base de datos.

Las validaciones del formulario funcionan (campos obligatorios, formato correcto, control de errores).

El código fue probado manualmente, asegurando que el flujo completo funciona (de entrada de datos a almacenamiento).

El equipo agregó comentarios en el código explicando las secciones principales.

Se creó una pequeña guía visual o demo funcional para explicar cómo usar la nueva función.

El Product Owner revisó y aceptó el trabajo realizado, sin observaciones pendientes.

El incremento está listo para ser presentado en el Sprint Review.

Descripción del inicio del Sprint con base en esta historia:

- El equipo inicia el Sprint trabajando en el formulario de registro de tickets.
- Se asignan tareas específicas:
 - o Un miembro diseña el formulario.
 - o Otro crea la tabla tickets en la base de datos.
 - o Otro desarrolla la lógica para guardar los datos desde el formulario.
- El Scrum Master supervisa la fluidez del trabajo y documenta cualquier obstáculo.
- El Product Owner está presente para resolver dudas, como el significado de los campos o el flujo deseado.
- Se simulan situaciones reales: por ejemplo, se introduce un “impedimento” (falta de conexión entre frontend y backend) y el equipo debe reorganizar sus tareas o buscar soluciones.
- Se actualiza constantemente un tablero Kanban simulado, donde se mueven las tareas de "To Do" → "In Progress" → "Done".

2.3 Inicio del Sprint (Simulado):

Simulación del Sprint para la Historia de Usuario 1

Historia de Usuario 1:

Como agente de soporte, quiero poder registrar un nuevo ticket para poder gestionar las solicitudes de los clientes de manera eficiente.

Sprint Backlog:

Las tareas del **Sprint Backlog** deben desglosar los pasos necesarios para implementar la funcionalidad de registrar un ticket. Las tareas deben ser pequeñas y específicas, asignables a los miembros del equipo para completar durante el Sprint.

Tareas del Sprint Backlog:

1. Diseño de la base de datos para tickets

- **Descripción:** Crear la estructura de la base de datos para almacenar los tickets, incluyendo campos como: ID, título, descripción, fecha de creación, estado, prioridad, cliente asociado, etc.
- **Responsable:** Desarrollador Backend
- **Duración estimada:** 1 día

2. Diseñar la interfaz de usuario para registrar un nuevo ticket

- **Descripción:** Crear el diseño de la página de registro de tickets donde los agentes de soporte pueden ingresar la información necesaria para crear un ticket (título, descripción, cliente, prioridad, etc.).
- **Responsable:** Diseñador UI/UX
- **Duración estimada:** 2 días

3. Implementar la lógica de backend para crear un ticket

- **Descripción:** Desarrollar la lógica que permitirá a los agentes registrar un ticket en la base de datos a través de un formulario en la interfaz.
- **Responsable:** Desarrollador Backend
- **Duración estimada:** 2 días

4. Desarrollar la validación de datos del ticket

- **Descripción:** Implementar validaciones para asegurarse de que todos los campos requeridos estén completos y que los datos ingresados sean correctos antes de enviar el formulario para crear un ticket.
- **Responsable:** Desarrollador Backend
- **Duración estimada:** 1 día

5. Implementar el envío de notificación al cliente tras la creación del ticket

- **Descripción:** Desarrollar la funcionalidad que enviará una notificación al cliente cuando se haya creado un nuevo ticket, informándole sobre su solicitud.
- **Responsable:** Desarrollador Backend
- **Duración estimada:** 1 día

6. Pruebas unitarias y de integración

- **Descripción:** Realizar pruebas para verificar que la creación de tickets funcione correctamente, que los datos se registren adecuadamente y que las notificaciones se envíen como se espera.
- **Responsable:** Tester QA
- **Duración estimada:** 2 días

7. Revisión de código y ajustes finales

- **Descripción:** Revisar el código implementado para garantizar que siga las mejores prácticas y hacer cualquier ajuste necesario.
- **Responsable:** Desarrollador Backend y Frontend
- **Duración estimada:** 1 día

Plan de trabajo del Sprint:

Duración del Sprint: 1 semana (por ejemplo)

- **Día 1-2:**
 - Realización de tareas 1 (Diseño de la base de datos para tickets) y 2 (Diseñar la interfaz de usuario para registrar un nuevo ticket).
- **Día 3-4:**
 - Desarrollo de tareas 3 (Implementar la lógica de backend para crear un ticket) y 4 (Desarrollar la validación de datos del ticket).

- **Día 5:**
 - Implementación de la tarea 5 (Enviar notificación al cliente tras la creación del ticket).
- **Día 6:**
 - Realización de tareas de prueba (Pruebas unitarias y de integración).
- **Día 7:**
 - Revisión de código y ajustes finales (tarea 7).

Resultados esperados del Sprint:

- **Funcionalidad completada:** Los agentes de soporte podrán registrar un nuevo ticket en el sistema, la base de datos será capaz de almacenar la información del ticket correctamente, y los clientes recibirán notificaciones sobre la creación de sus tickets.
- **Criterios de aceptación:**
 1. El ticket debe ser registrado correctamente en la base de datos.
 2. El sistema debe validar los datos antes de aceptar el ticket.
 3. El cliente debe recibir una notificación por correo electrónico tras la creación del ticket.

Revisión del Sprint:

Al final del Sprint, el equipo realizará una **revisión del Sprint** para evaluar si todas las tareas del Sprint Backlog fueron completadas según lo planeado. Se presentarán los resultados a los stakeholders y se recogerán comentarios para mejorar en el siguiente Sprint.

3 Fase 3: Ejecución del Sprint y Daily Scrums

3.1 Daily Scrum (15 min al inicio de cada "día simulado"):

El Daily Scrum es una reunión diaria de 15 minutos que se realiza al inicio de cada "día simulado" del Sprint. El objetivo principal es sincronizar las actividades del equipo, evaluar el progreso hacia el objetivo del Sprint y detectar posibles impedimentos que puedan afectar el desarrollo.

Estructura del Daily Scrum:

El evento es facilitado por el Scrum Master (SM), quien se encarga de mantener el enfoque y el tiempo de la reunión.

Los miembros del Development Team (DT) responden a tres preguntas clave:

¿Qué hice ayer que ayudó al equipo a cumplir con el objetivo del Sprint?

¿Qué voy a hacer hoy para ayudar al equipo a cumplir con el objetivo del Sprint?

¿Hay algún impedimento que me esté bloqueando para avanzar?

Actualización del Sprint Backlog:

Durante la reunión, se actualiza el Sprint Backlog para reflejar el progreso y los cambios en las tareas.

Esto puede incluir:

Mover post-its en un tablero Kanban físico o digital, para indicar si una tarea está en "Pendiente", "En Proceso" o "Finalizado".

Agregar notas sobre avances, problemas encontrados o cambios en la estimación de tiempos.

Identificación de Impedimentos:

El Scrum Master (SM) se encarga de identificar y anotar los impedimentos que surgen durante el Daily Scrum.

Estos impedimentos pueden incluir:

Falta de recursos o permisos.

Problemas de comunicación entre los miembros del equipo.

Dependencias no resueltas con otros equipos o áreas.

Fallos en herramientas o tecnologías necesarias para avanzar.

El objetivo del SM es trabajar en la eliminación de estos impedimentos para que el equipo pueda continuar sin bloqueos y avanzar hacia la meta del Sprint de manera efectiva.

3.2 Trabajo en el Sprint (Simulado) (Resto del tiempo):

Simulación del Sprint - Resolución de Problemas, Colaboración y Eliminación de Impedimentos

Integrantes del equipo:

- **Silvia - Product Owner (PO)**
- **Emerson - Scrum Master (SM)**
- **Pablo - Desarrollador**
- **Gabriel - Desarrollador**
- **Jaquelin - Desarrolladora**
- **Sindi - Desarrolladora**

Fase 1: Preparación y Asignación de Tareas

En el **Sprint Planning**, el equipo se reúne para planificar el trabajo del sprint. El **Product Owner (PO)**, Silvia, presenta la **Historia de Usuario** y clarifica los requisitos. El equipo se asigna las tareas del Sprint Backlog según las prioridades establecidas y las habilidades de cada miembro.

Tareas Asignadas:

- **Pablo:** Diseño de la base de datos para los tickets.
- **Gabriel:** Diseño de la interfaz de usuario para registrar un nuevo ticket.
- **Jaquelin:** Implementación de la lógica de backend para crear un ticket.

- **Sindi:** Desarrollo de las pruebas unitarias para la funcionalidad de creación de tickets.

Impedimento: Falta de Jaquelin y Sindi

Solución: Emerson (Scrum Master) debe gestionar este impedimento para que los miembros restantes (Pablo y Gabriel) puedan avanzar mientras esperan la disponibilidad de Jaquelin y Sindi. Emerson se comunica con el equipo para conocer la situación y ajustar el plan de trabajo.

Simulación de Daily Scrums - Día 1

Emerson (Scrum Master) inicia la primera **Daily Scrum**:

Emerson (SM):

- **Pregunta:** "¿Qué hiciste ayer, qué harás hoy y si hay algún impedimento?"

Pablo:

- **Ayer:** Terminé el diseño de la base de datos y ya está lista para su implementación.
- **Hoy:** Empiezo a configurar la conexión entre el backend y la base de datos para que los tickets se registren correctamente.
- **Impedimento:** No he tenido impedimentos.

Gabriel:

- **Ayer:** Estuve trabajando en el diseño de la interfaz de usuario para registrar un ticket, ya tengo los prototipos listos.
- **Hoy:** Comenzaré a integrar la interfaz de usuario con el backend.

- **Impedimento:** Sin problemas por ahora, pero necesito aclaraciones sobre el formato de los datos del ticket.

Emerson (SM):

- **Ayer:** Había un impedimento por la falta de Jaquelin y Sindi, lo que ha retrasado algunas tareas, pero he hablado con ellos y estarán disponibles mañana.
- **Hoy:** El equipo puede continuar trabajando en las tareas pendientes, pero vamos a asegurarnos de estar listos para cuando Jaquelin y Sindi se reincorporen.
- **Impedimento:** La falta de personal, ya que Jaquelin y Sindi no están disponibles para contribuir.

Silvia (PO):

- **Aclaración:** "Para Gabriel, la interfaz debe manejar todos los campos que ya definimos en la base de datos. Además, asegúrate de que haya validación de datos antes de enviar la información al backend."

Simulación de Daily Scrums - Día 2

Emerson (Scrum Master) inicia la segunda **Daily Scrum**:

Emerson (SM):

- **Pregunta:** "¿Qué hiciste ayer, qué harás hoy y si hay algún impedimento?"

Pablo:

- **Ayer:** Avancé en la conexión entre el backend y la base de datos, ahora los tickets pueden ser creados y almacenados correctamente.
- **Hoy:** Realizaré pruebas para asegurarme de que el registro de tickets se haga sin errores y se almacenen correctamente.
- **Impedimento:** Ningún impedimento, todo va según lo planeado.

Gabriel:

- **Ayer:** Integré la interfaz con el backend, la funcionalidad de registro de tickets ya está en su lugar.
- **Hoy:** Terminaré de ajustar algunos detalles de la interfaz y realizaré algunas pruebas con datos de ejemplo.
- **Impedimento:** Sin impedimentos.

Emerson (SM):

- **Ayer:** Jaquelin y Sindi ya han podido unirse al equipo, por lo que ahora podemos avanzar más rápido.
- **Hoy:** El equipo debería enfocarse en las pruebas unitarias, que están en espera para que las validaciones sean correctas antes de la entrega final.
- **Impedimento:** No tengo impedimentos en este momento.

Silvia (PO):

- **Aclaración:** "Recuerden que el ticket debe ser validado completamente antes de ser enviado. Asegúrense de que todos los campos sean obligatorios y correctos."

Simulación de Daily Scrums - Día 3 (evento inesperado)

Emerson (Scrum Master) inicia la tercera **Daily Scrum**:

Emerson (SM):

- **Pregunta:** "¿Qué hiciste ayer, qué harás hoy y si hay algún impedimento?"

Pablo:

- **Ayer:** Realicé las pruebas unitarias para la conexión de la base de datos, todo funciona bien.
- **Hoy:** Continuaré con las pruebas de integración para asegurarme de que la creación de tickets esté bien implementada.
- **Impedimento:** Durante las pruebas, me di cuenta de que la base de datos no está validando correctamente algunos de los datos, lo que impide completar las pruebas correctamente.

Gabriel:

- **Ayer:** Ajusté la interfaz de usuario según las últimas especificaciones de Silvia. Ahora estoy trabajando en el diseño para la visualización de los tickets registrados.
- **Hoy:** Continuaré con el diseño de la pantalla de gestión de tickets.

- **Impedimento:** No tengo impedimentos por ahora.

Jaquelin:

- **Ayer:** Inicié la implementación de la lógica de backend para la validación de los tickets.
- **Hoy:** Finalizaré la validación y aseguraré que todos los campos sean correctos.
- **Impedimento:** No he encontrado impedimentos hasta ahora.

Sindi:

- **Ayer:** Revisé el código para las pruebas unitarias, pero he encontrado que las validaciones de datos no cubren todos los casos posibles.
- **Hoy:** Ajustaré las pruebas unitarias para cubrir estos casos.
- **Impedimento:** El problema con la base de datos de Pablo me está impidiendo avanzar, ya que las validaciones son fundamentales.

Resolución del Impedimento:

Emerson (SM) se da cuenta de que el **impedimento** de las validaciones de datos de la base de datos está afectando a varias tareas del equipo. **Emerson** se comunica con el equipo para encontrar una solución.

- **Emerson (SM):** "Pablo, ¿puedes revisar la configuración de validaciones en la base de datos? Parece que algo está fallando allí. Jaquelin y Sindi necesitan estas validaciones para continuar con sus pruebas y trabajo."

- **Pablo:** "¡Entendido! Revisaré la configuración y corregiré los errores de validación lo más rápido posible."

Conclusión del Sprint:

Al final del Sprint, el equipo realiza una **revisión de lo que se completó**. En este caso:

- **Tareas completadas:** El equipo pudo integrar la creación de tickets y las validaciones necesarias, y la funcionalidad fue probada correctamente.
- **Impedimentos gestionados:** El equipo manejó correctamente los impedimentos relacionados con la falta de miembros en el equipo y las validaciones de base de datos.

Emerson (SM): "El equipo ha trabajado bien en la gestión de problemas y la colaboración. La comunicación fue efectiva, aunque un par de impedimentos no fueron completamente resueltos desde el principio. Vamos a reflexionar sobre cómo mejorar en los próximos Sprints."

4 Fase 4: Sprint Review y Sprint Retrospective

4.1 Preparación para el Sprint Review:

Para este Sprint, el equipo ha trabajado en la creación de tickets en el sistema de gestión de soporte. La **demonstración del Incremento** incluirá lo siguiente:

- **Prototipo:** Mostrar la interfaz de usuario diseñada para el registro de tickets, donde los agentes de soporte pueden llenar los campos necesarios (título, descripción, cliente, prioridad, etc.).
- **Mockups:** Presentación de la interfaz de usuario donde se puede observar el flujo de trabajo del agente para crear un ticket de soporte.
- **Backend:** Demostración del proceso en el que el sistema guarda correctamente los tickets en la base de datos, con la validación de los campos antes de enviar el formulario.

Equipo: El equipo debe demostrar cómo un agente de soporte puede crear un ticket sin errores, asegurándose de que todos los campos sean obligatorios y que los datos ingresados sean correctos.

4.2 Sprint Review:

Durante el **Sprint Review**, el equipo presenta lo siguiente:

- **Incremento presentado:**
 - **Interfaz de usuario:** Los agentes ahora pueden registrar tickets a través de una interfaz limpia y sencilla.
 - **Funcionalidad del sistema:** El sistema registra los tickets en la base de datos, valida que los campos sean correctos y asegura que la información esté correctamente almacenada.
 - **Notificación:** Al crear un ticket, el cliente recibe un correo electrónico notificándole que su solicitud ha sido recibida.

- **Product Backlog:**

- **Ítems completados:**

- Implementación de la funcionalidad para registrar un ticket en la base de datos.
 - Diseño e implementación de la interfaz de usuario para el registro de tickets.
 - Notificación por correo electrónico al cliente al crear un ticket.

- **Ítems no completados:**

- Integración de los tickets con el sistema de seguimiento (visualización y gestión de tickets en una página centralizada).
 - Algunas pruebas de integración no se completaron debido a restricciones de tiempo.

Feedback:

Los stakeholders pueden dar retroalimentación. Algunos puntos importantes podrían incluir:

- Mejorar la validación de los campos en la interfaz para proporcionar más detalles a los agentes de soporte sobre los errores de entrada.
- Sugerir una mejor forma de visualizar los tickets creados, en lugar de solo mostrar la lista de tickets en la base de datos.

Actualización del Product Backlog:

- Se podrían agregar ítems para mejorar la visualización y gestión de los tickets en la interfaz de usuario.
- Se prioriza la mejora de la validación y la integración con el sistema de seguimiento de tickets.

4.3 Sprint Retrospective:

El equipo reflexiona sobre el Sprint, utilizando las preguntas y técnicas para identificar áreas de mejora y los éxitos del Sprint.

¿Qué salió bien?

- El equipo logró completar la historia de usuario 1 (registro de tickets) dentro del plazo.
- La interfaz de usuario fue diseñada de manera efectiva y es fácil de usar.
- La comunicación dentro del equipo fue efectiva, especialmente cuando surgieron obstáculos o impedimentos.

¿Qué podría mejorarse?

- La validación de los campos en la interfaz podría haberse implementado con más profundidad, ya que algunos errores no fueron capturados a tiempo.
- La integración con el sistema de gestión de tickets no se completó, lo que limita la capacidad de ver el estado de cada ticket.

¿Qué acciones concretas se tomarán para mejorar en el próximo Sprint?

- Mejorar la validación de los campos en la interfaz de usuario, asegurando que todos los datos sean correctos antes de permitir el registro de un ticket.
- Completar la integración con el sistema de seguimiento de tickets, para que los agentes puedan ver los tickets registrados y su progreso en tiempo real.
- Realizar más pruebas de integración para asegurar que el sistema de tickets funciona correctamente con otros módulos.

Técnicas usadas para la Sprint Retrospective:

- **Start, Stop, Continue:**
 - **Start:** Empezar a implementar más pruebas unitarias e integraciones para evitar errores no detectados.
 - **Stop:** Dejar de depender de validaciones manuales en lugar de automatizar la validación de datos.
 - **Continue:** Continuar con la buena comunicación dentro del equipo y la colaboración entre los desarrolladores.
- **Mad, Sad, Glad:**
 - **Mad:** Algunos errores no se detectaron debido a una validación incompleta en la interfaz de usuario.
 - **Sad:** No pudimos completar la integración de los tickets con el sistema de seguimiento, lo que afecta la funcionalidad del producto.

- **Glad:** El equipo completó la historia de usuario dentro del tiempo establecido, y la notificación por correo electrónico fue un éxito.
- **4 Ls (Liked, Learned, Lacked, Longed for):**
 - **Liked:** El diseño de la interfaz fue bien recibido, y el proceso de creación de tickets funciona correctamente.
 - **Learned:** Aprendimos que es importante realizar pruebas más exhaustivas antes de lanzar nuevas funcionalidades.
 - **Lacked:** Faltó tiempo para integrar completamente el sistema de gestión de tickets.
 - **Longed for:** Nos gustaría haber tenido más tiempo para integrar otras funcionalidades, como la visualización de tickets.

Resultado:

Se ha identificado un plan de acción para mejorar la validación de los datos y para completar la integración del sistema de gestión de tickets en el próximo Sprint.

5 Fase 5: Conclusiones y Discusión (Opcional, o integrada en la Sesión 4)

5.1 Reflexión General sobre la Experiencia:

Trabajar con la metodología Scrum ha sido una experiencia muy positiva y eficaz, siempre y cuando se implemente de la manera correcta. A través de las reuniones diarias (Daily Scrum) y los sprints, el equipo puede mantenerse enfocado en los objetivos inmediatos, ajustarse rápidamente a cualquier cambio y mejorar constantemente en cada iteración. Este enfoque

organizado y estructurado permite una mayor colaboración y comunicación entre los miembros, lo cual facilita el trabajo en equipo. Sin embargo, también hemos aprendido que la clave para que Scrum sea exitoso radica en tener roles claros y un buen seguimiento de las tareas durante el proceso.

5.2 Comparación de Cómo los Diferentes Equipos Abordaron los Desafíos:

Al observar cómo otros equipos enfrentaron los desafíos, nos dimos cuenta de que la planificación inicial juega un papel fundamental en el éxito de los sprints. Algunos equipos desglosaron las tareas en pequeñas acciones específicas y fueron más eficientes al completarlas, mientras que otros no tuvieron una visión tan clara desde el inicio y eso retrasó su progreso. En cuanto a los problemas que surgieron durante el Sprint, algunos equipos lograron resolverlos rápidamente gracias a la intervención activa del Scrum Master, mientras que otros tuvieron más dificultades para solucionarlos y eso les afectó a nivel de tiempos.

5.3 Discusión sobre la Aplicabilidad de SCRUM en Diferentes Contextos:

Aunque Scrum es ampliamente utilizado en el desarrollo de software, nos dimos cuenta de que puede aplicarse a otros contextos también. Por ejemplo, en proyectos de gestión de tareas o incluso en el ámbito educativo, donde se pueden dividir los trabajos en sprints y hacer un seguimiento constante. Esto permite que los equipos sean más flexibles y estén más enfocados en sus objetivos.

Sin embargo, no siempre es fácil adaptarlo a todos los contextos. En proyectos donde los plazos son fijos y los procesos muy estructurados, como en ciertos entornos regulatorios o de

investigación, Scrum podría no ser tan efectivo sin hacer ajustes. A pesar de eso, el enfoque de trabajo continuo y la mejora en cada ciclo puede aportar beneficios a casi cualquier equipo que busque optimizar su manera de trabajar y mejorar los resultados.