

**Prueba técnica – Dev Lead
Double V Partners**

Santiago Henao Galeano

**Medellín, Colombia
2023**

Contenido

<i>Descripción del problema.....</i>	3
1. <i>Estrategia de desarrollo</i>	4
2. <i>Propuesta técnica</i>	6
3. <i>Riesgos técnicos</i>	10
4 <i>Gestión del equipo.....</i>	11
5 <i>Estrategias y mecanismos para lograr el objetivo.....</i>	¡Error! Marcador no definido.
6 <i>Gestión del proceso, prácticas, deuda técnica .</i>	¡Error! Marcador no definido.

Descripción del problema

La compañía ha sido contratada para realizar un proyecto en el cual se plantea el desarrollo de un producto digital que debe ser implementado en conjunto con el equipo de producto. Para el desarrollo del producto te han informado que el producto a liberar contempla varias funcionalidades. Este será un producto enfocado en usuario masivo que a3 partir de la solución podrá:

1. Acceder a un marketplace de eventos artísticos (musicales, conciertos, fiestas) que son promovidos desde redes sociales pero que invitan al usuario a adquirir sus tickets en línea usando la aplicación de la compañía.
2. El usuario podrá comprar boletos usando algún medio de pago (tarjeta de crédito, tarjeta débito).
3. La compañía pretende conocer a sus usuarios y sus preferencias, con lo cual necesita poder generar campañas enfocadas en las preferencias de sus usuarios.
4. La compañía sigue un proceso de publicación de los eventos en el que intervienen varias áreas:
 - a) En particular la publicación de un evento comienza por un acuerdo comercial con el artista.
 - b) Sobre este acuerdo se generan todos los copies de las campañas, la descripción del evento, los espacios disponibles, el número de cupos, los valores de los tiquetes, fechas de apertura, fecha de cierre de ventas, promociones, códigos de redención, descuentos, productos relacionados.
 - c) Luego se generan unas propuestas de cuál debería ser la publicación más adecuada, la cual es aprobada por un equipo de publicación. Cada propuesta puede ser vista por el equipo y aprobada o puede solicitar cambios específicos al equipo que genera la publicación.
 - d) Una vez aprobada la publicación esta es publicada en un marketplace.
 - e) Para cada producto o promoción disponible, el sistema genera estadísticas de uso, visualización, permite recibir comentarios,,
 - f) El sistema debe proveer un dashboard de visualización de las estadísticas de cada producto o evento ofrecido.
 - g) El sistema debe permitir persistir los carritos de compra en los cuales el usuario realiza el proceso de compra y permitir al administrador conocer las estadísticas de churn del carrito para establecer mecanismos de retoma del proceso.

- h) El usuario tendrá la posibilidad de comprar sus tickets sin necesidad de crear cuentas en el sistema, pero podrá registrarse y contar con una cuenta que le permite acceder a beneficios.
- i) Una vez realizada la compra el sistema genera un recibo y un tiquete con una clave que debe ser configurada por el usuario de forma que pueda descargar sus tiquetes o visualizarlos entregando dicha clave y un código de compra.

Requerimientos:

1. Describir la estrategia de desarrollo del proyecto. Establecer un mecanismo de estimación del esfuerzo y plantear un modelo a seguir para el desarrollo.
2. Generar una propuesta técnica de orden arquitectónico no detallada de cómo resolver este problema y generar una estructura base (Diseño arquitectónico de alto nivel) que describa la solución. Casos de uso, características sistémicas, escenarios de calidad.
3. Describir los riesgos técnicos del proyecto.
4. Describir cómo va a gestionar al equipo desde el punto de vista técnico, cómo va a medir y controlar el proceso y gobernar el alcance técnico.
5. Establecer mecanismos y estrategias para lograr el objetivo de forma segura.
6. Describir cómo va a gestionar el proceso, prácticas, deuda técnica.

1. Estrategia de desarrollo

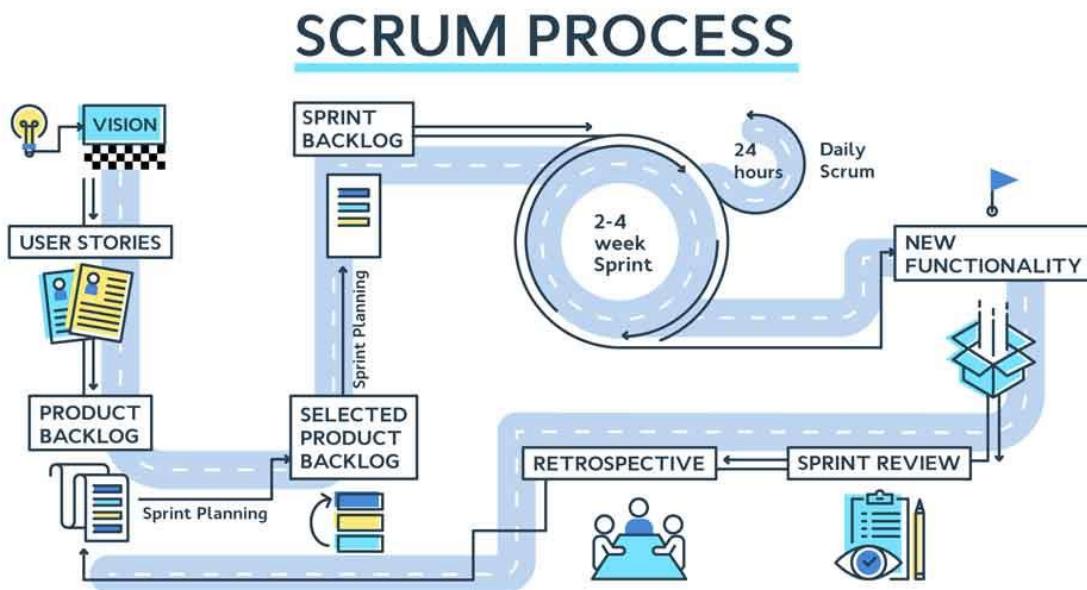
Para definir una estrategia de desarrollo del proyecto es fundamental tener una definición clara de las necesidades, alcance, tiempo, calidad, interesados y estrategia de comunicación. Para este ejercicio se supone que, dentro de la definición del proyecto, se han establecido previamente los documentos de visión, procesos fundamentales, casos de uso, prototipos, entre otros aspectos que corresponden a requerimientos funcionales y no funcionales, y se ha acordado prioridad de cada uno de los requerimientos.

Una vez establecida la etapa de definición, se lleva a cabo las etapas de planificación, diseño, ejecución y cierre. En este escenario, se puede sugerir una metodología ágil, como scrum, que facilite una iteración continua y una respuesta rápida frente a los posibles cambios. Es usual una duración de *sprints* de dos semanas a cuatro semanas, para priorizar y planificar las funcionalidades; sin embargo, es un aspecto que puede ajustarse de la mano del *Product Owner*, encargado de gestionar los requerimientos, el *Scrum Master*, quien asegura la

correcta implementación del proceso, la complejidad del proyecto y la estimación de tiempo de las actividades y de la retroalimentación del equipo de desarrollo.

Para la correcta ejecución del proyecto bajo esta metodología, se supone que la cultura de la empresa está alineada con la filosofía del agilismo, y se proponen 4 ceremonias, que incluyen el *sprint planning*, *sprint dailies*, *sprint review* y *retrospective*. El flujo de este desarrollo se presenta en la Figura 1.

Figura 1. Gestión de proyectos bajo metodología Scrum.



Nota. Obtenido de <https://www.itconsultors.com/metodologia-scrum>

Para la estimación de esfuerzo se propone el *planning poker*, debido a que desde la experiencia con dicha herramienta se puede fomentar la colaboración activa de todo el equipo y su inclusión en la planificación del proyecto. Emplear esta herramienta, permite mejorar la precisión en las estimaciones al contar con diferentes perspectivas y conocimientos, lo que facilita el intercambio de saberes, la identificación de riesgos, desafíos y consideraciones, así como la eliminación de sesgos individuales a la hora de estimar.

Al ser una técnica que involucra a todo el equipo y que requiere llegar a un consenso, es una herramienta valiosa, no solamente desde el punto de vista técnico, sino desde la comunicación, la colaboración, la apropiación y el sentido de pertenencia por el proyecto por parte del equipo de desarrollo.

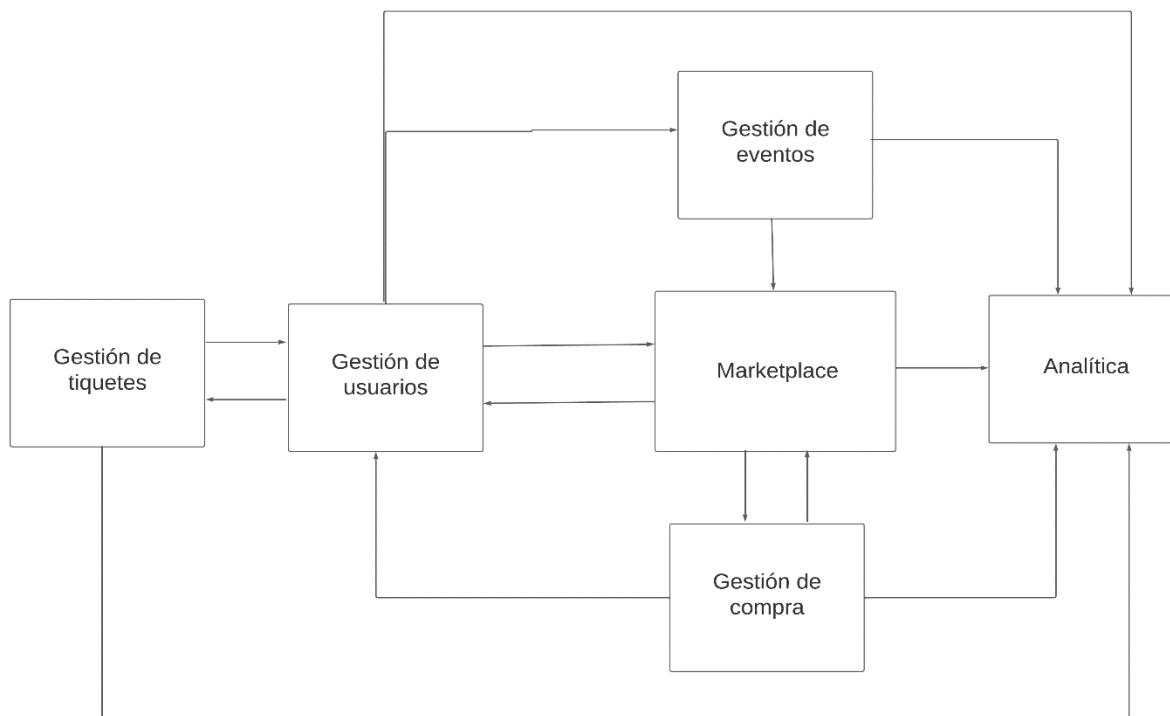
2. Propuesta técnica

Generar una propuesta técnica de orden arquitectónico no detallada de cómo resolver este problema y generar una estructura base (Diseño arquitectónico de alto nivel) que describa la solución. Casos de uso, características sistémicas, escenarios de calidad.

2.1 Diseño arquitectónico

En la Figura 2 se presenta el diseño arquitectónico de alto nivel para la problemática descrita. Cada uno de los bloques representa un servicio. Al ser un diseño de alto nivel no se consideran integraciones con herramientas de terceros, procesos internos de los servicios, ni selección de tecnologías.

Figura 2. Diseño arquitectónico de alto nivel.



Nota. Elaboración propia

Para el diseño de la arquitectura base se plantean los siguientes servicios:

- **Marketplace:** Este servicio permite a los usuarios acceder y explorar eventos artísticos como conciertos, etc. Proporciona una interfaz para que los usuarios adquieran boletos en línea utilizando medios de pago como tarjetas de crédito y débito.
- **Gestión de usuarios:** Este servicio se encarga de la gestión de usuarios y sus preferencias. Permite a los usuarios registrarse y acceder

a beneficios adicionales. Captura y almacena información sobre las preferencias de los usuarios para futuras campañas de marketing y recomendaciones personalizadas.

- **Gestión de eventos:** Este componente maneja el proceso de publicación de eventos, incluyendo la disponibilidad de espacios, precios de boletos, promociones y productos relacionados.
- **Gestión de compra:** Permite gestionar el proceso de compra de un usuario registrado y no registrado, administra el pago de tiquetes para los eventos y su flujo.
- **Gestión de tiquetes:** Gestiona los tiquetes adquiridos por los usuarios. Permite al usuario configurar una clave para acceder a sus tiquetes y descargarlos o visualizarlos utilizando esa clave y un código de compra
- **Analítica:** Recopila los datos obtenidos sobre estadísticas de uso, visualización, comentarios, procesos de compra, facturación etc. Proporciona un dashboard para análisis y toma de decisiones.

2.2 Casos de uso

Algunos de los principales casos de uso son:

Usuario accede al Marketplace y explora eventos artísticos:

- El usuario ingresa al sistema y navega por el Marketplace de eventos.
- El usuario puede buscar eventos por categoría, fecha, ubicación u otros filtros.
- El usuario visualiza información detallada de los eventos, como descripción, artistas, ubicación, fechas y precios.
- El usuario puede ver las promociones y productos relacionados disponibles para cada evento.

Usuario realiza una compra de boletos utilizando tarjeta de crédito o débito:

- El usuario selecciona un evento y elige los boletos deseados.
- El usuario proporciona la información de pago.
- El sistema procesa la transacción y genera un recibo y un tiquete con una clave y código de compra.
- El usuario puede descargar o visualizar los tiquetes utilizando la clave y el código de compra.

Compañía genera campañas personalizadas basadas en las preferencias de los usuarios:

- La compañía recopila y analiza las preferencias de los usuarios registrados.

- La compañía utiliza la información para crear campañas de marketing y promociones dirigidas a segmentos específicos de usuarios.
- Las campañas se generan en base a los eventos relevantes para cada segmento de usuarios, sus preferencias artísticas y otras características relevantes.

Equipo de publicación crea y aprueba la publicación de eventos:

- El equipo de publicación inicia el proceso de publicación de un evento después de alcanzar un acuerdo comercial con el artista o promotor.
- El equipo de publicación genera los copies de las campañas, la descripción del evento, los precios de los boletos, las fechas, las promociones, entre otros.
- Propone diferentes opciones de publicación y las presenta al equipo de aprobación.
- El equipo de publicación y los *stakeholders* revisan las propuestas y aprueban o solicitan cambios en los diseños.

Usuario persiste su carrito de compra

- El usuario tiene la opción de guardar su carrito de compra y retomar el proceso de compra en una sesión posterior.
- El sistema almacena el carrito de compra y permite al usuario acceder a él nuevamente.
- El administrador del sistema puede monitorear las estadísticas de los carritos de compra para identificar puntos problemáticos y tomar medidas para los procesos de compra.

Usuario se registra para acceder a beneficios adicionales:

- El usuario tiene la opción de registrarse en el sistema para crear una cuenta personal, proporcionando su información personal.
- El registro le permite acceder a beneficios adicionales, como ofertas exclusivas, descuentos especiales o acceso a eventos privilegiados.

2.3 Características sistémicas

Para las características sistémicas del diseño arquitectónico, se deben considerar siete elementos, los cuales hacen referencia a escalabilidad, disponibilidad, seguridad, rendimiento, mantenibilidad, integración y usabilidad. A continuación, se enlistan algunas de los vitales de cada uno de estos aspectos

- **Escalabilidad:** La creación de la arquitectura pensada en microservicios, se realiza con la capacidad de escalar horizontalmente los componentes, mediante replicación y balanceo de cargas, con el objetivo de responder a un alto volumen de usuarios y una demanda creciente.

- **Disponibilidad:** El sistema debe presentar una disponibilidad continua, con un porcentaje de disponibilidad del 99% implementando técnicas de redundancia y tolerancia a fallos para aumentar su resiliencia. Para el porcentaje de disponibilidad se supone que el proyecto cuenta con los recursos suficientes para infraestructura, redundancia, monitoreo y mantenimiento, de no ser así, se puede trabajar sobre un porcentaje menor.
- **Seguridad:** Se debe garantizar la seguridad de los datos personales de los usuarios y sus transacciones. Se debe contar con protocolos seguros en la comunicación y técnicas de cifrado para mantener la integridad de los datos y los inicios de sesión.
- **Rendimiento:** El sistema debe garantizar la fluidez en la experiencia de usuario. Se deben realizar pruebas de carga y rendimiento para identificar cuellos de botella y optimizar los componentes clave
- **Mantenibilidad:** Se deben garantizar buenas prácticas de desarrollo para facilitar el mantenimiento y los nuevos requerimientos de los servicios. Se debe garantizar la separación de responsabilidades y la cohesión de los componentes para facilitar la identificación y solución de problemas. Adicionalmente, se debe realizar el esfuerzo en documentar, para favorecer la transmisión de conocimientos y la adaptación rápida al proyecto desde el punto de vista técnico.
- **Integración:** El sistema debe ser capaz de integrarse con servicios externos, como pasarelas de pago, por ejemplo.
- **Usabilidad:** El sistema debe ser intuitivo para los usuarios, brindando una buena experiencia de usuario.

2.4 Escenarios de calidad

Con base en las características sistémicas y los escenarios planteados en las mismas, se proponen unas métricas que determinan la calidad de la implementación.

- **Escalabilidad:** Capacidad para escalar horizontalmente, capacidad de procesamiento, escalabilidad del almacenamiento de datos, escalabilidad de la infraestructura.
- **Disponibilidad:** Porcentaje de disponibilidad, tiempo de inactividad no planificado, tiempo medio entre fallas, tiempo medio de reparación.
- **Seguridad:** Cumplimiento de estándares de seguridad, encriptación de datos, protección contra ataques cibernéticos, integridad y confidencialidad de los datos.
- **Rendimiento:** Tiempo de respuesta promedio, tiempo de respuesta máximo, capacidad de carga, rendimiento bajo cargas pesadas.

- **Mantenibilidad:** Modularidad y separación de responsabilidades, tiempo promedio para corregir errores y tasa de errores recurrentes, métricas de deuda técnica, cobertura de documentación.
- **Integración:** Tiempo de respuesta de las integraciones, disponibilidad de los servicios externos, intercambio de datos sin problemas
- **Usabilidad:** Facilidad de navegación, tiempos de carga de páginas, claridad de la interfaz de usuario, tasas de conversión, tasa de abandono del carrito de compra.

3. Riesgos técnicos

Para la descripción de los riesgos técnicos se supone que se cuenta con un equipo técnico competente; sin embargo, no se está exento de retos técnicos, nuevas tecnologías y un proceso de adaptación y formación en estas que puede generar retrasos. Sin más, se presentan algunos riesgos técnicos asociados a las características sistémicas descritas previamente.

- **Integración compleja de servicios de terceros:** La falta de conocimientos, de documentación clara, cambios en la API o incompatibilidad con las tecnologías implementadas en el desarrollo pueden representar un desafío al momento de incluir servicios externos.
- **Escalabilidad insuficiente:** Si no se realiza un correcto diseño de la arquitectura, pueden existir cuellos de botella difícil de resolver, que sacrifiquen los tiempos de respuesta y la disponibilidad del sistema.
- **Problemas de seguridad:** La vulnerabilidad de software puede dar paso a filtración de datos y ataques cibernéticos que comprometan la confidencialidad de los datos y la plataforma.
- **Falta de robustez en la gestión de calidad:** Si no se manejan adecuadamente los errores y excepciones, el sistema puede volverse inestable o presentar comportamientos inesperados. La falta de mecanismos de recuperación de errores y pruebas exhaustivas puede aumentar el riesgo de fallas del sistema.
- **Mantenibilidad del sistema:** Si el código no se mantiene adecuadamente y no se siguen buenas prácticas de desarrollo, podría haber dificultades para agregar nuevas funcionalidades, corregir errores o realizar actualizaciones en el futuro
- **Mala gestión de equipo:** Si bien no corresponde a un riesgo técnico por sí, la mala gestión, puede conllevar al desaprovechamiento de las capacidades de los miembros del equipo, problemas internos, descoordinación, dependencias no resueltas, que se traduce en retrasos, baja motivación y el no alcanzar los objetivos.

- **Dependencias externas y control del cambio:** Al igual que el ítem anterior, el tema de dependencias de recursos externos, o solicitudes de cambios que afecten el correcto desarrollo del proyecto, puede acarrear problemas en el alcance del proyecto y una mayor presión sobre el equipo de desarrollo, lo que impacta el ambiente laboral y por ende se afecta el rendimiento.

4. Gestión del equipo.

Lo fundamental desde el rol de líder, en especial en un escenario de la construcción de un equipo de alto desempeño, está orientado a brindar las condiciones para que cada uno de los miembros pueda ofrecer lo mejor de sí mismo y crecer personal y profesionalmente, al ritmo de crecimiento de la compañía y de los proyectos en los que se encuentra involucrado. Por ende, se describen estrategias centradas principalmente en el capital humano que favorecen el éxito del proyecto.

- **Comunicación y colaboración efectiva:** Establecer canales de comunicación claros y frecuentes con el equipo, para asegurar una buena colaboración. Fomentar un ambiente de trabajo abierto donde los miembros del equipo puedan plantear dudas, compartir ideas y resolver problemas técnicos de manera conjunta.
- **Definición clara de roles y responsabilidades:** Establecer roles claros dentro del equipo técnico, como arquitectos, desarrolladores, analistas de pruebas, etc. Cada miembro debe tener una comprensión clara de sus responsabilidades y áreas de competencia.
- **Establecimiento de estándares y buenas prácticas:** Definir de manera conjunta estándares de codificación, arquitectura y documentación para mantener la consistencia y calidad del código. Fomentar el uso de buenas prácticas de desarrollo, como pruebas unitarias, revisión de código y control de versiones.
- **Capacitación y desarrollo del equipo:** Proporcionar oportunidades de formación y desarrollo profesional para mantener al equipo técnico actualizado con las últimas tecnologías.
- **Medición y seguimiento del proceso:** Establecer métricas y KPIs relevantes para medir el progreso y la calidad del proceso de desarrollo. Esto puede incluir métricas de velocidad, calidad del código, tiempo de entrega y satisfacción del cliente. Utilizar herramientas de seguimiento y gestión de proyectos como Jira para tener una visión clara del estado del proyecto y el avance del equipo

- **Gobernanza del alcance técnico:** Establecer un proceso de revisión y aprobación para cualquier cambio o adición al alcance técnico del proyecto. Esto implica evaluar el impacto en términos de recursos, tiempo y complejidad técnica antes de aceptar nuevos requisitos o funcionalidades. Es importante mantener una comunicación constante con los stakeholders y el equipo de producto para asegurar la viabilidad y coherencia técnica del proyecto.

5. Estrategias y mecanismos para lograr el objetivo de forma segura

En concordancia con lo mencionado en apartados anteriores, y teniendo presente que el cumplimiento de los objetivos requiere de una sinergia entre los encargados de producto, los stakeholders y el equipo de desarrollo, se plantean algunas estrategias y mecanismos que complementan lo ya dicho, y que incluyen otros aspectos relevantes, enmarcados dentro de una metodología ágil.

- **Planificación y organización:** Definir una estructura clara de roles y responsabilidades dentro del equipo, estableciendo claramente las tareas y entregables en cada sprint.
- **Comunicación efectiva:** Establecer canales de comunicación claros y frecuentes dentro del equipo y con los stakeholders del proyecto. Fomentar la colaboración y la retroalimentación para asegurar una comprensión común de los requisitos y expectativas.
- **Estimación y gestión del esfuerzo:** Utilizar técnicas de estimación, como *Planning Poker* para evaluar el esfuerzo necesario para cada tarea, y monitorear el progreso del equipo para asegurarte de que se están cumpliendo los plazos establecidos dentro de las dailies.
- **Gestión del alcance:** Definir claramente el alcance del proyecto, identificando los requisitos y funcionalidades clave desde las primeras etapas, pero establecer un proceso formal de cambio de alcance para evaluar y aprobar los cambios solicitados, evitando desviaciones no planificadas y asegurando una gestión efectiva del alcance.
- **Control de calidad:** Establecer estándares de calidad para el desarrollo del producto, incluyendo buenas prácticas, pruebas y revisión de código.
- **Gestión de riesgos:** Identificar y evaluar los riesgos potenciales que podrían impactar el desarrollo del proyecto antes y durante la etapa de desarrollo, y crear planes de mitigación y contingencia que permita abordar los riesgos identificados y minimizar su impacto en el proyecto.
- **Gestión de recursos:** Contar con los recursos necesarios, tanto humanos como tecnológicos, para el desarrollo del proyecto. Asignar adecuadamente

las tareas y supervisar el rendimiento del equipo para garantizar una asignación eficiente de los recursos disponibles.

- **Aprendizaje y mejora continua:** Promover un ambiente de aprendizaje y mejora continua, fomentando la retroalimentación y la revisión retrospectiva. Utiliza los resultados de estas revisiones para implementar cambios y mejoras.

6. Gestión del proceso, prácticas, deuda técnica.

Previamente se abordaron estrategias para el desarrollo exitoso de los objetivos, que incluyen la gestión de los procesos asociados a la construcción del proyecto; Sin embargo, uno de los elementos más importantes que se pretende abordar en esta sección hace referencia a la gestión de la deuda técnica, ya que dejarla acumular afecta la calidad del software y la productividad del equipo a mediano y largo plazo.

Algunos aspectos para considerar en la gestión de la deuda técnica son:

- **Reconocimiento de la deuda técnica:** Para disminuir la deuda técnica, es fundamental identificar los aspectos específicos donde está presente y su impacto en el proyecto.
- **Priorización:** Con base en el impacto que tiene la deuda en términos de estabilidad, rendimiento, seguridad y mantenibilidad del software se puede realizar un ejercicio de priorización para determinar qué aspectos deben ser atendidos de manera urgente.
- **Planificación:** Una vez establecida la prioridad para abordar la deuda técnica, se deben establecer metas claras y realistas, asignando el tiempo y los recursos necesarios para su solución dentro de los sprints
- **Refactorización:** El mejoramiento de código es una actividad de mejora continua, que se debe considerar dentro de los sprints, y que debe ir acompañado del fortalecimiento de buenas prácticas de desarrollo.
- **Pruebas:** Las pruebas son parte fundamental de la prevención de deuda técnica. Se debe realizar un trabajo de pruebas para garantizar la estabilidad del sistema.
- **Capacitación:** Una de las estrategias que no solo favorece la disminución de la deuda, sino que también favorece el trabajo colaborativo, es el espacio de formación e intercambio de conocimientos dentro del equipo, que puede realizarse a través de espacios comunes y formación específica.
- **Seguimiento y control:** Como todas las etapas, es necesario realizar un seguimiento a los avances en la gestión de la deuda técnicas para asegurarse que los problemas asociados a la deuda se puedan identificar,

priorizar, planear y ejecutar sin afectar de manera crítica el desarrollo del proyecto.