**GUÍA PARA DESARROLLAR EL TEXTO GUÍA**

PROGRAMA:

PROCESOS AGILES DE SOFTWARE

(Texto Guía)

DEPARTAMENTO OFERENTE:

INGENIERÍA

AUTOR(A):

JORGE ALEJANDRO AGUIRRE GUTIERREZ

FORMACIÓN:

ESPECIALIZACION DE INGENIERÍA DE SOFTWARE CON ÉNFASIS EN PRUEBAS

Manizales, 2024

**Tabla de Contenidos**

**UNIDAD 2: PRACTICA DE LA METODOLOGÍA SCRUM.**

* Conformación de equipos y roles.
* Creación de organización y épicas con herramienta Jira o Azure Devops.
* Creación de Backlog e historias de usuario con herramienta Jira o Azure Devops.
* Practica de realización de planning.
* Practica de realización de daily.
* Practica de realización de review.
* Practica de realización de retrospective.

***Antes de iniciar el texto guía, por favor lea con atención estas recomendaciones generales para el escrito:***

*El texto guía es un escrito académico elaborado por el autor del curso y tiene como principal función “presentar de forma didáctica el conocimiento de las distintas disciplinas…Por su diseño, son textos mixtos que reúnen información textual con otra de tipo visual (uso de imágenes, gráficas, tablas, etc.). También contienen muchos tipos de ayudas didácticas, especialmente seleccionadas y diseñadas, con el propósito de favorecer el aprendizaje (Díaz & Hernández, 2010, p. 158), tales como: videos, simuladores, animaciones, cuestionarios de verificación de lectura, ejercicios interactivos (opción múltiple, falso y verdadero, emparejamiento, clasificar conceptos) o juegos interactivos de Educaplay (*[*Ver tipo de actividades*](https://es.educaplay.com/tipos-de-actividades/)*) con recursos multimedia (video, audio, imagen, texto)*

*Siguiendo a Díaz y Hernández (2010), los textos considerados accesibles se caracterizan por:*

* *Tienen un arreglo estructurado y sistemático de las ideas.*
* *Poseen un buen nivel de coherencia de las ideas expuestas*
* *Son concisos y contienen poca información distractora o irrelevante*
* *Toman en cuenta el conocimiento previo del lector*
* *Presentan al lector la estructura del texto de forma clara (con señalizaciones) para beneficiar su comprensión profunda*

*Por favor tenga en cuenta las siguientes recomendaciones:*

* *Utilizar Verdana tamaño 12, interlineado sencillo.*
* *Las fuentes consultadas para desarrollar el contenido no deben ser de más de cinco años de antigüedad. Si se requiere utilizar fuentes de más años, se debe justificar en las referencias al final de la unidad.*
* *Todas las gráficas, ilustraciones o figuras, deben estar enumeradas con la herramienta de Word REFERENCIAS: INSERTAR TÍTULO. Debe registrarse la FUENTE: Autor, fecha, o sitio web. Cuando la gráfica es de autoría del docente, se debe especificar al final del título de la gráfica: FUENTE: ELABORACIÓN PROPIA.*
* *En el desarrollo del contenido, cuando se trate de ideas fundamentadas en otros autores, se debe hacer la cita correspondiente. Del mismo modo diferenciar claramente las opiniones del docente autor del módulo.*
* *Las citas deben referenciarse con Normas VANCOUVER (PARA CIENCIAS DE LA SALUD) – APA (CIENCIAS SOCIALES E INGENIERÍAS)*

**TÍTULO DE LA UNIDAD:**

**TEORIA SOBRE METODOLOGIAS AGILES**

**INTRODUCCIÓN A LA UNIDAD**

**Indicador de desempeño**

Al finalizar el estudio de esta unidad y desarrollar las actividades propuestas, el estudiante deberá estar en capacidad de: Conocer sobre la implementación del SCRUM dentro de un equipo de desarrollo y la participación efectiva de cada uno de sus eventos.

**-----------**

**TEMA 1: CONFORMACION DE EQUIPOS Y ROLES**

De acuerdo a la cantidad de estudiantes se conformarán equipos que en un formato de simulación se asignaran los siguientes roles:

* Líder técnico.
* Scrum Master.
* Product Owner.
* Desarrolladores.
* Desarrolladores de pruebas.

Harán una lectura de retroalimentación de cada uno de los roles dentro del SCRUM:

**Líder Tecnico:** En el marco de la metodología Scrum, juega un papel fundamental para el éxito del equipo de desarrollo. A diferencia de un rol jerárquico tradicional, el Líder Técnico en Scrum se caracteriza por ser un guía, mentor y facilitador que aporta su experiencia y conocimiento técnico para optimizar el proceso de desarrollo y asegurar la calidad del producto final.

Responsabilidades del Líder Técnico en Scrum:

* Guía al equipo en la toma de decisiones técnicas: El Líder Técnico aporta su experiencia y conocimiento para ayudar al equipo a elegir las mejores soluciones técnicas para cada desafío que se presente. Esto incluye la selección de tecnologías, herramientas y frameworks adecuados, así como la evaluación de la viabilidad de las propuestas.
* Mentoriza a los miembros del equipo: El Líder Técnico actúa como mentor para los miembros del equipo, compartiendo su conocimiento y experiencia para ayudarlos a desarrollar sus habilidades técnicas y profesionales. Esto puede incluir la realización de sesiones de capacitación, la revisión de código y feedback constructivo.
* Facilita la comunicación y la colaboración: El Líder Técnico facilita la comunicación y la colaboración entre los diferentes miembros del equipo, así como con stakeholders externos. Esto implica organizar reuniones, mediar en conflictos y garantizar que todos estén alineados con los objetivos del proyecto.
* Optimiza el proceso de desarrollo: El Líder Técnico busca continuamente formas de optimizar el proceso de desarrollo para hacerlo más eficiente y efectivo. Esto puede incluir la implementación de metodologías ágiles, la automatización de tareas y la mejora de las prácticas de ingeniería de software.
* Asegura la calidad del producto final: El Líder Técnico es responsable de asegurar que el producto final cumpla con los estándares de calidad establecidos. Esto implica la realización de pruebas, la revisión de código y la identificación y corrección de errores.

**SCRUM Master:** Juega un rol fundamental como facilitador, líder de servicio y coach para el equipo de desarrollo. A diferencia de un rol jerárquico tradicional, el Scrum Master no tiene autoridad sobre los miembros del equipo, sino que se enfoca en crear un ambiente propicio para que el equipo pueda trabajar de manera autónoma y autoorganizada.

Las principales responsabilidades del Scrum Master incluyen:

* Facilitar las ceremonias Scrum: El Scrum Master es responsable de organizar y facilitar las ceremonias Scrum, como son las reuniones de sprint planning, las revisiones de sprint y las retrospectivas. Estas ceremonias son esenciales para el buen funcionamiento del equipo y para mantener el progreso del proyecto.
* Eliminar impedimentos: El Scrum Master actúa como un removedor de impedimentos, identificando y eliminando cualquier obstáculo que pueda impedir el avance del equipo. Esto puede incluir problemas técnicos, dependencias externas o conflictos entre los miembros del equipo.
* Promover la autoorganización del equipo: El Scrum Master fomenta la autoorganización del equipo, empoderándolos para que tomen sus propias decisiones y gestionen su propio trabajo. Esto implica ayudar al equipo a definir sus objetivos, priorizar tareas y resolver problemas de manera autónoma.
* Proteger el tiempo del equipo: El Scrum Master protege el tiempo del equipo para que puedan enfocarse en las tareas más importantes del proyecto. Esto implica gestionar las interrupciones, negociar con stakeholders y defender al equipo ante presiones externas.
* Promover la colaboración y la comunicación: El Scrum Master fomenta la colaboración y la comunicación entre los miembros del equipo, así como con stakeholders externos. Esto implica crear un ambiente de trabajo abierto y transparente donde todos se sientan cómodos para compartir ideas y expresar sus preocupaciones.
* Servir al Product Owner: El Scrum Master sirve al Product Owner, ayudándolo a definir y priorizar el backlog del producto. También facilita la comunicación entre el Product Owner y el equipo de desarrollo.
* Ser un coach para el equipo: El Scrum Master actúa como un coach para el equipo, ayudándolos a mejorar sus habilidades y a crecer como profesionales. Esto puede incluir brindar feedback constructivo, ofrecer capacitación y fomentar el aprendizaje continuo.

**Product Owner:** Juega un rol fundamental como visionario del producto, representante del cliente y responsable del backlog del producto. A diferencia de un gerente de producto tradicional, el Product Owner en Scrum no se enfoca en la gestión del día a día del equipo de desarrollo, sino que se concentra en definir la visión del producto, priorizar las funcionalidades y maximizar el valor para el cliente.

Las principales responsabilidades del Product Owner incluyen:

* Definir la visión del producto: El Product Owner es responsable de definir la visión del producto, es decir, la imagen a largo plazo de lo que se quiere lograr con el producto. Esta visión debe ser clara, concisa e inspiradora, y debe servir como guía para el desarrollo del producto en todas sus etapas.
* Crear y mantener el backlog del producto: El Product Owner crea y mantiene el backlog del producto, que es una lista ordenada de todas las funcionalidades que se deben implementar en el producto. El backlog debe ser priorizado en función del valor que cada funcionalidad aporta al cliente.
* Representar al cliente: El Product Owner representa al cliente en el equipo de desarrollo. Esto significa que debe recopilar y analizar los requisitos del cliente, traducirlos en funcionalidades del producto y defender los intereses del cliente en las decisiones que se toman durante el desarrollo.
* Colaborar con el equipo de desarrollo: El Product Owner colabora con el equipo de desarrollo para garantizar que el producto se esté desarrollando de acuerdo con la visión y las necesidades del cliente. Esto implica participar en las reuniones de sprint planning, las revisiones de sprint y las retrospectivas.
* Priorizar las funcionalidades: El Product Owner es responsable de priorizar las funcionalidades del backlog del producto en función del valor que cada una aporta al cliente. Esto implica considerar factores como la necesidad del cliente, el impacto comercial y la viabilidad técnica.
* Aceptar o rechazar el trabajo del equipo de desarrollo: El Product Owner tiene la autoridad para aceptar o rechazar el trabajo del equipo de desarrollo. Esto significa que debe evaluar si el trabajo realizado cumple con los requisitos del producto y si aporta valor al cliente.
* Comunicar el progreso del producto a los stakeholders: El Product Owner es responsable de comunicar el progreso del producto a los stakeholders, como son los clientes, la gerencia y los inversores. Esto implica preparar informes, realizar presentaciones y responder preguntas.

Desarrolladores: Son un equipo autoorganizado y multifuncional responsable de la construcción del producto. A diferencia de los roles tradicionales de desarrollo de software, los desarrolladores en Scrum no tienen un líder jerárquico, sino que trabajan en conjunto para planificar, diseñar, implementar y probar las funcionalidades del producto.

Las principales responsabilidades de los desarrolladores en Scrum incluyen:

* Estimar el trabajo: Los desarrolladores son responsables de estimar el tiempo y el esfuerzo que se requiere para completar cada tarea del backlog del producto. Estas estimaciones se utilizan para planificar los sprints y para medir el progreso del equipo.
* Desarrollar el producto: Los desarrolladores diseñan, implementan y prueban las funcionalidades del producto. Esto implica escribir código, realizar pruebas unitarias, integrar el código y corregir errores.
* Colaborar entre sí: Los desarrolladores trabajan en conjunto para resolver problemas, compartir conocimientos y ayudarse mutuamente a alcanzar sus objetivos.
* Mejorar continuamente el proceso de desarrollo: Los desarrolladores buscan continuamente formas de mejorar el proceso de desarrollo, identificando y eliminando impedimentos, adoptando nuevas prácticas y aprendiendo de sus experiencias.

**Desarrollador de pruebas:** El rol de desarrollador de pruebas (tester) es el equipo encargado de escribir las pruebas unitarias, de integración, de seguridad y de rendimiento en compañía del equipo de desarrollo, líder técnico y Product Owner. El Scrum promueve un enfoque más colaborativo e integrado donde las pruebas son responsabilidad de todo el equipo.

¿Cómo se integran las pruebas en Scrum?

* Pruebas unitarias: Los desarrolladores son responsables de escribir y ejecutar pruebas unitarias para asegurar la calidad de su propio código. Estas pruebas se ejecutan de forma automática y verifican el correcto funcionamiento de las unidades de código individuales.
* Pruebas de integración: El equipo puede realizar pruebas de integración para verificar cómo interactúan las diferentes partes del código entre sí. Estas pruebas pueden ser automatizadas o manuales, y se ejecutan con mayor frecuencia que las pruebas unitarias.
* Pruebas de aceptación: El equipo, en conjunto con el Product Owner, define las pruebas de aceptación que se deben realizar para validar que el producto cumple con los requisitos del cliente. Estas pruebas pueden ser manuales o automatizadas, y se ejecutan al final de cada sprint para determinar si el sprint ha sido completado exitosamente.

¿Qué habilidades se esperan de los desarrolladores en Scrum en cuanto a pruebas?

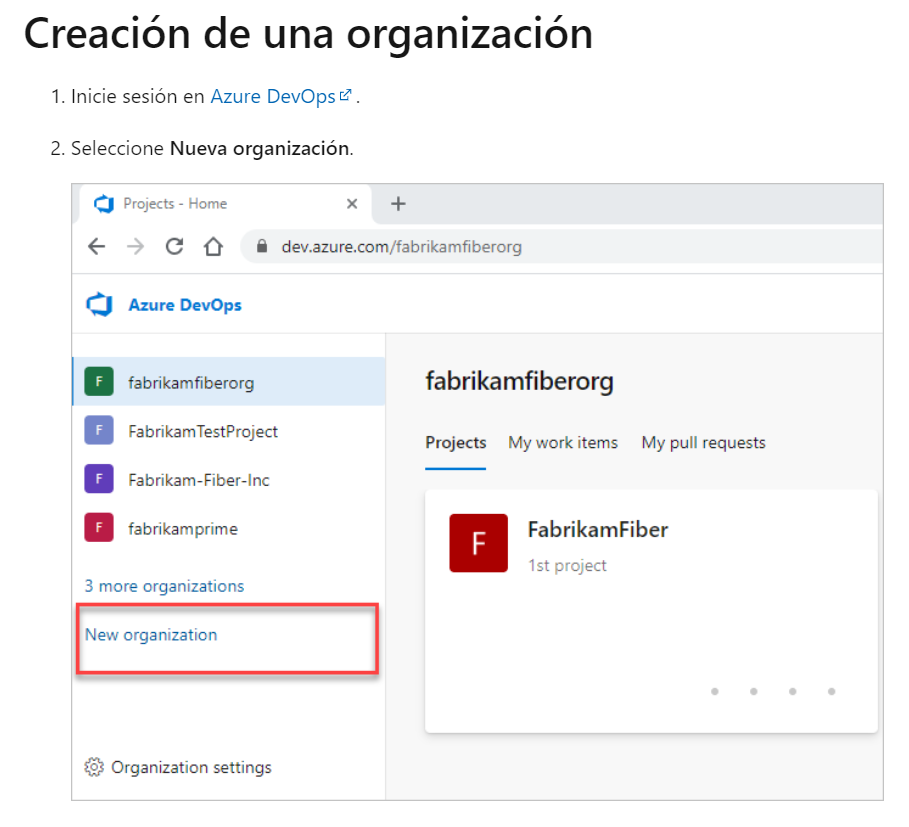
1. Conocimiento de técnicas de pruebas: Los desarrolladores deben tener un conocimiento básico de las técnicas de pruebas, como son las pruebas unitarias, las pruebas de integración y las pruebas de aceptación.
2. Habilidad para escribir pruebas automatizadas: Es deseable que los desarrolladores tengan la capacidad de escribir pruebas automatizadas, ya que esto permite agilizar el proceso de testing y reducir la cantidad de errores.
3. Mentalidad de tester: Los desarrolladores deben tener una mentalidad de tester, es decir, deben ser capaces de pensar como usuarios y anticipar posibles problemas o errores en el producto.
4. Colaboración con el equipo: Los desarrolladores deben colaborar con el equipo para definir las estrategias de testing, ejecutar las pruebas y analizar los resultados.

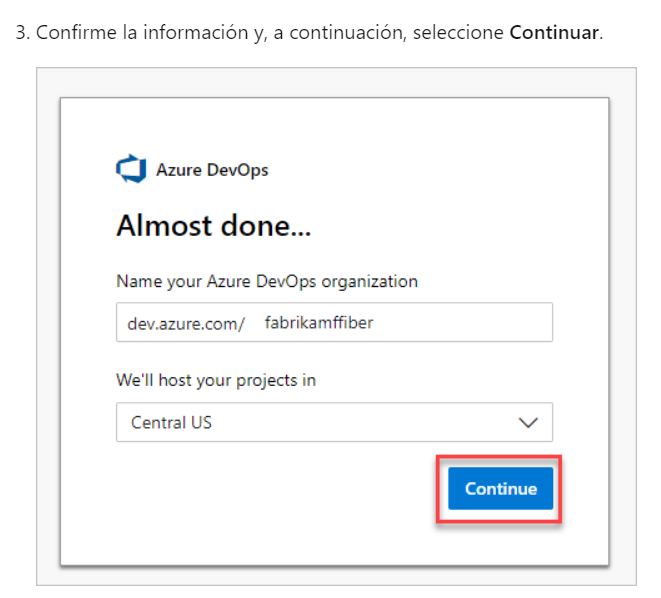
¿Beneficios de integrar las pruebas en Scrum?

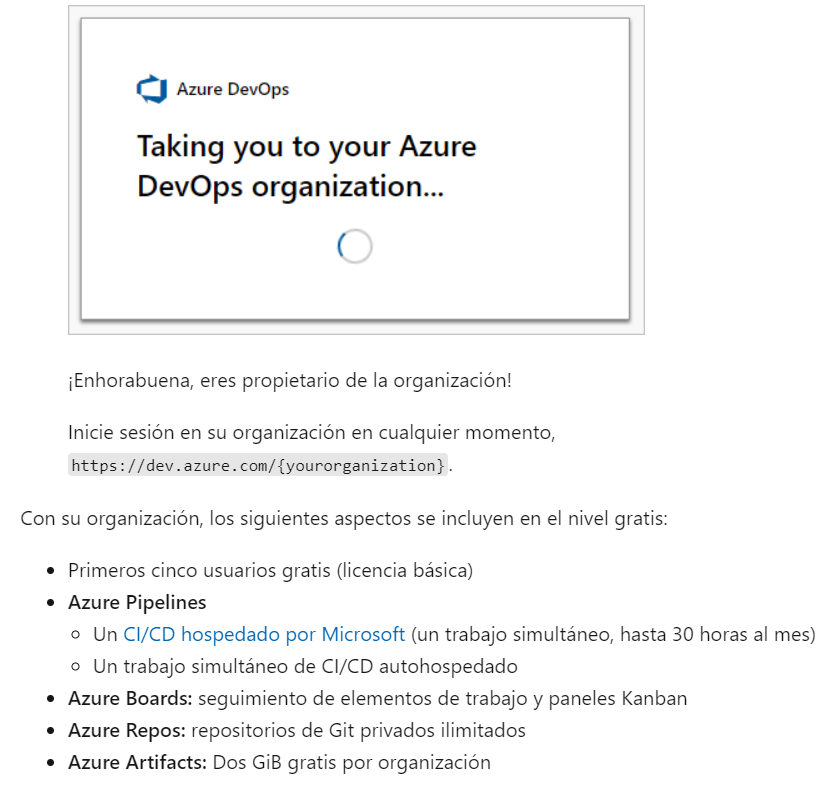
* Mayor calidad del producto: Al integrar las pruebas en el proceso de desarrollo, se pueden identificar y corregir errores de manera temprana, lo que resulta en un producto final de mayor calidad.
* Mayor velocidad de desarrollo: Las pruebas automatizadas permiten agilizar el proceso de desarrollo, ya que automatizan tareas repetitivas y liberan tiempo a los desarrolladores para que se enfoquen en otras actividades.
* Mayor satisfacción del cliente: Un producto de alta calidad que cumple con los requisitos del cliente conduce a una mayor satisfacción del cliente.

**TEMA 2: CREACIÓN DE ORGANIZACIÓN Y ÉPICAS CON HERRAMIENTA JIRA O AZURE DEVOPS.**

Se les hará entrega del nombre de la organización (Viajes S.A), nombre de la aplicación (ViajaYa), además de un proyecto en fase de predesarrollo, los estudiantes tendrán que crear la organización en Azure Devops, el proyecto a través de un repositorio, hacer configuración de Git y crear las épicas pertinentes después de ejecutar el evento de BrainStorming. Se hará esta actividad de la siguiente manera.







**Desarrollar cada uno de los temas propuestos**

***INDICACIONES:***

*Presentación del contenido utilizando estrategias que cumplan con las siguientes funciones:*

* *Detección de la información principal;*
* *Conceptualización de contenidos;*
* *Delimitación de la organización, estructura interna e interrelaciones entre dichos contenidos;*
* *y mantenimiento de la atención y motivación.*

*A lo largo del escrito haga uso de estrategias tales como: ilustraciones, esquemas, gráficas o resúmenes (al iniciar la unidad, al finalizar la unidad, o al finalizar un tema), preguntas intercaladas, analogías, pistas tipográficas (resaltar ideas centrales, conceptos, mapas conceptuales), ejemplificaciones para aclarar conceptos o mostrar aplicaciones, ejercicios.*

*Utilice recursos multimedia como material complementario: audios, videos, animaciones, simuladores, imágenes, entre otros.*

*Recuerde hacer las citas correspondientes con normas VANCOUVER O APA.*

*Recuerde proponer ejercicios interactivos a lo largo del texto para reforzar conocimientos o de verificación de comprensión de la lectura.*

*Recuerde titular y enumerar cada figura, tabla, imagen, gráfica, ilustración y registrar la fuente. Especificar cuando se trate de una gráfica de su propia autoría.*

**CIERRE DE LA UNIDAD**

***INDICACIONES:***

*Una vez terminado el desarrollo del contenido, procure formar una visión sintética, integradora e incluso crítica de la unidad, a través de estrategias como: resumen, mapa conceptual, preguntas o conclusiones generales de los temas presentados en el escrito.*

*Proponga material o lecturas complementarias si lo considera necesario.*

**REFERENCIAS BIBLIOGRÁFICAS**

***INDICACIONES:***

*Registre la documentación (NORMAS APA O VANCOUVER) consultada para desarrollar el contenido.*

*Utilice bibliografía con una antigüedad no superior a los cinco años.*