PLAN DE PRUEBAS



OBJETIVOS DE LAS PRUEBAS

El propósito principal de las pruebas es asegurar que proporcionamos una buena experiencia a los usuarios de la plataforma ymcbrand.com, manteniendo al mismo tiempo la funcionalidad, la seguridad, y un rendimiento óptimo.

Los siguientes objetivos guiarán las actividades de prueba y ayudarán a priorizar las áreas críticas de enfoque para asegurar que ymcbrand.com no solo cumple con las expectativas técnicas sino que también ofrece una experiencia de usuario superior, fomentando la satisfacción y la lealtad de sus clientes.

#### Verificar la Funcionalidad Completa

* Asegurar que todos los procesos de la tienda virtual, desde la selección y personalización de prendas hasta el proceso de pago y gestión de pedidos, funcionen según los requisitos especificados.

#### Optimizar la Experiencia del Usuario

* Garantizar una navegación intuitiva y accesible en toda la plataforma, permitiendo a los usuarios diseñar y comprar sus prendas personalizadas con facilidad.

#### Asegurar la Compatibilidad

* Comprobar que la tienda virtual es plenamente funcional en una gama representativa de dispositivos, navegadores y tamaños de pantalla, ofreciendo una experiencia uniforme en todas las plataformas.

#### Validar la Seguridad

* Confirmar que todos los datos de usuario, incluidos los detalles de pago, están seguros y protegidos contra vulnerabilidades y ataques cibernéticos.

#### Evaluar el Rendimiento

* Asegurar que la tienda virtual maneja eficientemente diferentes cargas de tráfico y realiza todas las operaciones dentro de los tiempos de respuesta esperados, especialmente durante periodos de mayor demanda.

#### Pruebas de Personalización y Configuración de Productos

* Validar las herramientas de personalización y configuración de camisetas para asegurar que los usuarios puedan diseñar sus productos sin problemas y que el resultado final coincida con sus expectativas.

#### Integración con Terceros

* Verificar las integraciones con servicios externos, como geolocalización, pasarelas de pago, servicios de envío, y proveedores de servicios de email, para garantizar que son confiables y funcionan como se espera.

#### Conformidad con Regulaciones

* Asegurar que la tienda cumple con todas las leyes y regulaciones aplicables, incluidas las relacionadas con la privacidad de datos (como el GDPR) y las transacciones comerciales electrónicas.

ALCANCE DE LAS PRUEBAS

El alcance de las pruebas define específicamente qué características y funciones serán objeto de prueba, así como aquellas que quedarán fuera de este proceso. Este enfoque ayuda a poner los límites de las actividades de prueba y asegura que los recursos se asignen eficazmente a las áreas de mayor impacto o importancia.

Abarca todas las funcionalidades críticas que impactan tanto en la operación efectiva de la plataforma como en la experiencia del usuario. Esto incluye:

#### Interfaz de Usuario y Experiencia del Usuario (UI/UX)

* Todas las páginas de la tienda, incluyendo la página de inicio, páginas de categorías, páginas de productos, y de personalización de prendas.
* Navegación y búsqueda dentro del sitio, asegurando que los usuarios pueden encontrar, fácilmente y de manera intuitiva, los productos o la información que buscan.
* Diseño responsivo y adaptabilidad a diferentes dispositivos y resoluciones de pantalla.

#### Proceso de Personalización de Productos

* Herramientas y opciones disponibles para la personalización de camisetas, incluyendo selección de colores, tamaños, y adición de textos o imágenes.
* Visualización previa de productos personalizados y exactitud de la representación del producto final.

#### Carrito de Compras y Proceso de Pago

* Funcionalidad del carrito de compras, incluyendo adición, modificación, y eliminación de productos. Comprobación del flujo completo.
* Integración y funcionamiento de pasarelas de pago, y seguridad de las transacciones.

#### Registro de Usuarios y Gestión de Perfiles

* Creación de cuentas, inicio de sesión, recuperación de contraseñas, y gestión de información de perfil de usuario.

#### Gestión de Pedidos y Funcionalidad de Envío

* Procesos de checkout, confirmación de pedidos, y seguimiento del estado del envío.
* Integración con proveedores de servicios de envío.

#### Funcionalidades Administrativas

* Herramientas de backend para la gestión y coordinación de los sistemas de inventario, pedidos, catálogo, así como generación de reportes de ventas.

#### Seguridad

* Pruebas de seguridad enfocadas en proteger los datos de los usuarios y transacciones contra accesos no autorizados, y otros tipos de ataque.

#### Rendimiento

* Tiempos de carga de página y capacidad de la plataforma para manejar cargas de usuario elevadas, especialmente durante promociones o eventos de ventas.

#### Fuera de Alcance

* **Contenido de Marketing y Blogs**
* Aunque el contenido es importante para la experiencia del usuario, las pruebas detalladas de contenido específico, como entradas de blog o materiales de marketing, están fuera del alcance principal.
* **Integraciones de Terceros no Críticas**
* Servicios de terceros que no afectan directamente las operaciones de compra, como herramientas de análisis de tráfico web o servicios de marketing por correo electrónico, están fuera del alcance de las pruebas detalladas.

RECURSOS Y HERRAMIENTAS DE PRUEBA

#### Herramientas de Automatización de Pruebas

* **Selenium Webdriver:** Utilizado para automatizar de manera más robusta las pruebas en navegadores web, permitiendo simular interacciones del usuario con la tienda virtual. Será clave para pruebas funcionales, especialmente para pruebas UI/UX y de regresión.
* **Selenium IDE:** Utilizado para procesos sencillos de automatización de pruebas en navegadores web. También será clave para las pruebas UI/UX y de regresión.
* **PyCharm:** Entorno de desarrollo integrado (IDE) seleccionado para desarrollar y ejecutar los scripts de pruebas automatizados en Python.
* **PyTest:** Framework de testing para Python que permite escribir pruebas de software de manera simple y escalable.
* **Postman**: Herramienta utilizada para el API Testing.

#### Herramientas de Gestión de Pruebas y Seguimiento de Defectos

* **Azure DevOps o Jira:** Herramienta para la gestión de tareas, seguimiento de errores, y coordinación entre los diferentes roles. Facilitará el seguimiento del progreso de las pruebas y la trazabilidad de las pruebas.
* **TestRail + Azure Testplans:** Para la gestión de casos de prueba, planificación de pruebas, y reporte de resultados.

#### Entorno de Pruebas

* En un primer momento se va a probar en el entorno de producción directamente, ya que la plataforma está desplegando de manera continua sus actualizaciones.
* Conforme se vaya avanzando en el proyecto será necesario probar en diferentes entornos que se determinarán en su momento.

#### Herramientas de Integración Continua y Entrega Continua (CI/CD)

* **GitHub Actions y Azure Pipelines:** Para automatizar la integración de código y despliegue de versiones de prueba, permitiendo ejecuciones automáticas de tests cada vez que se realiza una actualización.

#### Herramientas de Pruebas de Rendimiento y Carga

* **PyTest-Benchmark:** Para realizar pruebas de rendimiento.

#### Herramientas de Seguridad

* **OWASP ZAP:** Para realizar pruebas de seguridad automáticas y manuales, identificando vulnerabilidades comunes en aplicaciones web.
* **SQLMap:** Herramienta de prueba de penetración que automatiza el proceso de detección y explotación de vulnerabilidades de inyección SQL.

#### Herramientas de Pruebas de API

* **Postman o similar:** Para probar las APIs que la tienda virtual consume o expone, asegurando que se comportan según lo esperado en diferentes condiciones.

#### Control de versiones

* **Git:** Se utilizará el sistema de versiones Git y los repositorios remotos ymcbrand de GitHub y Azure DevOps.

ROLES Y RESPONSABILIDADES

Clarifica las responsabilidades de cada rol dentro del proyecto, asegurando una colaboración efectiva y enfocada en alcanzar los objetivos de calidad, funcionalidad, y satisfacción del usuario.

#### **QA Tester (Jorge Roncal)**

* **Diseño y Ejecución de Pruebas**
  + Desarrollar y ejecutar casos de prueba manuales y automatizados, enfocándose en las áreas de funcionalidad, usabilidad, seguridad y rendimiento.
* **Automatización de Pruebas**
  + Utilizar PyCharm para crear y mantener scripts de pruebas automatizadas, mejorando la eficiencia y cobertura de las pruebas.
* **Gestión de Defectos**
  + Registrar y seguir el progreso de los defectos identificados, trabajando en colaboración con el desarrollador externo para su resolución.
* **Comunicación de Resultados:**
  + Informar regularmente al product owner (PO) sobre los resultados de las pruebas, el estado de los defectos, y cualquier recomendación para mejorar la calidad del producto.

#### **Desarrollador Externo (odoo)**

* **Desarrollo de Funcionalidades**
  + Implementar nuevas funcionalidades y realizar el mantenimiento correctivo y evolutivo de la plataforma según los requerimientos proporcionados por el PO y el feedback de las pruebas.
* **Colaboración en la Definición de Requerimientos**
  + Trabajar de cerca con el PO para entender los requisitos detallados y las prioridades del negocio, asegurando una implementación alineada con las expectativas.
* **Integración y Despliegue**
  + Realizar la integración continua y el despliegue de nuevas versiones en el entorno de pruebas para su evaluación.

#### **Product Owner (Salvador Soriano)**

* **Definición de Requerimientos y Priorización:**
  + Establecer los requisitos del proyecto y priorizar el trabajo del desarrollador externo para alinear con los objetivos comerciales y las expectativas de los usuarios.
* **Validación y Aceptación de Funcionalidades:**
  + Revisar y validar las funcionalidades desarrolladas antes de su lanzamiento, asegurando que cumplan con los criterios de aceptación.
* **Gestión de Stakeholders:**
  + Actuar como el enlace principal entre el equipo de proyecto y cualquier otra parte interesada, incluyendo clientes, proveedores, y socios comerciales.
* **Supervisión del Proyecto:**
  + Monitorear el avance del proyecto, ajustar planes según sea necesario, y tomar decisiones clave para guiar el proyecto.

ESTRATEGIA Y METODOLOGÍA DE PRUEBA

La estrategia y la metodología de prueba están diseñadas para maximizar la calidad del producto y la eficiencia del proceso de pruebas, aprovechando la automatización y una comunicación efectiva entre los miembros del equipo.

Esta estrategia y metodología de prueba está diseñadas para ser dinámica y adaptativa, permitiéndonos responder rápidamente a cambios en el proyecto y asegurando que la tienda virtual no solo cumpla con los requisitos técnicos y de negocio, sino que también ofrezca una experiencia segura, eficiente y agradable para los usuarios.

Se priorizarán áreas críticas y áreas de mayor importancia para el posterior desarrollo de nuevas funcionalidades. Los siguientes aspectos guiarán las pruebas:

#### Automatización Primero

* **Priorizaremos la automatización de pruebas utilizando Selenium para las pruebas de UI/UX y de integración, lo cual permitirá realizar pruebas repetibles y consistentes a gran escala. PyCharm servirá como el IDE principal para desarrollar, depurar y mantener los scripts de pruebas automatizados.**

#### Pruebas Continuas e Integración Continua (CI)

* **Integraremos nuestras pruebas automatizadas en un proceso de CI utilizando GitHub Actions. Esto asegura que cada cambio en el código sea validado automáticamente, permitiendo detectar y corregir problemas de forma temprana.**

#### Enfoque en Calidad de Usuario y Seguridad

* **Las pruebas manuales se centrarán en explorar aspectos de la UX que son difíciles de automatizar y en realizar pruebas de seguridad específicas. Esto incluye pruebas de accesibilidad, pruebas exploratorias para identificar problemas de usabilidad, y pruebas de penetración para validar la seguridad.**

#### Pruebas de Rendimiento

* **Simularemos condiciones de carga elevada y evaluaremos el rendimiento de la plataforma.**

#### Gestión de Defectos Eficiente

* **Los defectos identificados a través de las pruebas serán documentados y rastreados. La comunicación efectiva entre los miembros del equipo para la rápida resolución de defectos será una prioridad.**

#### Feedback y Mejora Continua

* **El feedback del Product Owner será crucial después de cada ciclo de pruebas para ajustar prioridades y enfocar esfuerzos en áreas críticas. Además, revisaremos regularmente nuestra estrategia y metodología de pruebas para identificar oportunidades de mejora.**

#### Documentación y Transparencia

* **Mantendremos una documentación detallada de los casos de prueba, resultados, y defectos para asegurar la transparencia y facilitar la toma de decisiones basada en datos.**

CRONOGRAMA DE PRUEBAS

Dado que la plataforma está en un estado avanzado de construcción, durante las primeras semanas se irán probando y automatizando todas las funcionalidades terminadas. Al final de cada semana se realizará una reunión entre PO y Tester, para transmitir los fallos detectados y el feedback de mejora. Conforme se vayan terminando las funcionalidades pendientes, se irán incorporando al cronograma para ser probadas.

#### Puntos Clave del Cronograma

* **Flexibilidad:** El cronograma es una guía y debe ser adaptable a los cambios en el proyecto, incluyendo retrasos en el desarrollo o descubrimiento de defectos críticos.
* **Iteraciones:** Cada ciclo de desarrollo y pruebas es iterativo, permitiendo ajustes continuos basados en aprendizajes y feedback.
* **Comunicación Continua:** Las reuniones regulares con el product owner y el desarrollador externo son esenciales para mantener a todos alineados y ajustar el plan de pruebas según sea necesario.

#### Sprint 1: Inicio y Preparación

* Reunión inicial con el product owner (Salvador Soriano) y el desarrollador externo para confirmar el alcance de las funcionalidades a ser implementadas en el ciclo.
* Configuración y verificación de los entornos de pruebas.
* Diseño de casos de prueba para cubrir las funcionalidades planificadas.

#### Sprint 2: Ejecución Intensiva de Pruebas

* Ejecución y automatización de pruebas de humo en el entorno de pruebas para validar la estabilidad básica del sistema.
* Inicio de pruebas manuales en áreas críticas, especialmente en la personalización de camisetas.
* Ejecución de pruebas automatizadas según el desarrollador externo va completando módulos.

#### *Sprint 3: Pruebas de sistema e integración*

* Pruebas de API.
* Pruebas de rendimiento para evaluar la capacidad del sistema bajo carga.
* Pruebas de seguridad del sistema.

#### Sprint 4: Revisión Final y Ajustes

* Revisión de defectos con el desarrollador externo y pruebas de confirmación para verificar correcciones.
* Pruebas de regresión para asegurar que las nuevas funcionalidades no han afectado áreas existentes.
* Reunión de cierre con Salvador Soriano para revisar los resultados de las pruebas y obtener la aprobación final.

Este cronograma de pruebas asegura que haya tiempo suficiente no solo para la ejecución de pruebas, sino también para la planificación, revisión y ajustes basados en el feedback, lo cual es crucial para el éxito del proyecto. La estructura iterativa facilita la mejora continua y la adaptación a los desafíos que surjan durante el desarrollo.

CRITERIOS DE INICIO Y FINALIZACIÓN

Estos criterios aseguran que las pruebas comiencen solo cuando el sistema esté adecuadamente preparado y finalicen solo cuando los objetivos de calidad se hayan cumplido satisfactoriamente.

#### Criterios de Inicio

* Revisión de requisitos completada y aprobada por Salvador Soriano.
* Ambiente de pruebas configurado.
* El desarrollador externo ha integrado la primera versión de las nuevas funcionalidades en el entorno de pruebas.

#### Criterios de Finalización

* **Cobertura de Casos de Prueba**
  + Todos los casos de prueba planificados han sido ejecutados y documentados.
* **Resolución de Defectos**
  + Todos los defectos críticos y de alta prioridad han sido corregidos y verificados. Los defectos de menor prioridad han sido registrados para seguimiento futuro.
* **Cumplimiento de Criterios de Aceptación**
  + Las funcionalidades han sido validadas contra los criterios de aceptación definidos por el product owner.
* **Pruebas de Regresión**
  + Se han completado satisfactoriamente las pruebas de regresión para asegurar que cambios recientes no hayan introducido nuevos defectos en funcionalidades previamente probadas.
* **Aprobación del Product Owner**
  + El product owner ha revisado y está satisfecho con los resultados de las pruebas, otorgando la aprobación final para el lanzamiento.
* **Documentación Completa**
  + La documentación de pruebas, incluidos los resultados y los informes de defectos, está completa y ha sido entregada al equipo de desarrollo para referencia futura.

GESTIÓN DE RIESGOS

Consideraremos una gama de riesgos potenciales que ofrecen un marco más robusto y nos ayudará a prepararnos mejor para prevenir y responder a los desafíos específicos que podrían surgir durante el proyecto. La clave es mantener una actitud proactiva y estar preparado para adaptar las estrategias conforme evolucionan las circunstancias del proyecto.

#### Riesgos Técnicos

1. **Dependencia de Terceros:** El funcionamiento de la plataforma y la personalización de prendas depende de librerías y de un catálogo, que podría dejar de ser mantenidos.

* **Mitigación:** Evaluar alternativas y considerar una solución interna.

1. **Riesgos de Compatibilidad:** La tienda podría no funcionar correctamente en versiones antiguas de navegadores populares, afectando a una porción de la audiencia.

* **Mitigación:** Establecer políticas claras de navegadores soportados y realizar pruebas de compatibilidad tempranas.

#### Riesgos de Recursos

1. **Capacidad de Respuesta del Desarrollador Externo:** El desarrollador enfrenta sus propios retrasos o problemas de capacidad, impactando el cronograma del proyecto.

* **Mitigación:** Mantener una comunicación fluida y regular, establecer hitos claros y tener un plan B para las tareas críticas.

1. **Experiencia con Herramientas de Pruebas:** Descubrimiento de que las pruebas automatizadas o de cualquier otro tipo son más complejas de lo previsto debido a la naturaleza dinámica de la personalización de productos.

* **Mitigación:** Programar sesiones de capacitación y asesoramiento a lo largo del proyecto según surjan estas necesidades.

#### Riesgos de Proyecto

1. **Retrasos en el Cronograma Debido a Pruebas y Correcciones:** La identificación de un número significativo de defectos críticos podría requerir más tiempo del previsto para su corrección y verificación.

* **Mitigación:** Implementar pruebas continuas y de integración desde las primeras etapas, y reservar tiempo adicional en la planificación para la gestión de defectos.

#### Riesgos de Seguridad

1. **Vulnerabilidades de Seguridad en la Aplicación:** Exposición a inyecciones SQL a través de las diversas funcionalidades del sistema.

* **Mitigación:** Realizar pruebas de seguridad y penetración regularmente, aplicar principios de seguridad desde el diseño, y mantenerse actualizado con las prácticas recomendadas de seguridad.

1. **Riesgos de Exposición de Datos de Usuarios:** Brechas de seguridad que podrían comprometer los datos personales y de pago de los usuarios.

* **Mitigación:** Considerar la implementación de encriptación de datos en tránsito y reposo, seguir normativas de protección de datos como GDPR, y realizar auditorías de seguridad.

PROCEDIMIENTO DE MANEJO DE DEFECTOS

#### Identificación de Defectos

* Durante las sesiones de prueba, tanto automatizadas como manuales, los defectos se identificarán y documentarán. Cada defecto incluirá una descripción detallada, trazabilidad, pasos para reproducirlo, severidad, y capturas de pantalla o videos cuando sea posible.

#### Clasificación de Defectos

* Los defectos se clasificarán por severidad (Crítico, Alto, Medio, Bajo) y tipo (Funcionalidad, UI, Rendimiento, Seguridad). Esto facilitará la priorización y resolución.

#### Registro y Seguimiento

* Se registrará cada defecto, asignándole un identificador único que mantendrá la trazabilidad con los productos de trabajo relacionados. Azure DevOps o Jira serán los medios para el seguimiento del ciclo de vida del defecto, desde su identificación hasta su resolución y verificación.

#### Priorización y Asignación

* En las reuniones de sprint planning, y basándose en la clasificación de los defectos, el equipo priorizará y asignará los defectos para su corrección. Los defectos críticos y altos tendrán prioridad y deberán resolverse en el sprint actual siempre que sea posible.

#### Resolución de Defectos

* El desarrollador externo trabajará en la corrección de los defectos asignados, siguiendo las prácticas de desarrollo establecidas y manteniendo comunicación constante.

#### Verificación

* Una vez que un defecto ha sido corregido, el tester será responsable de verificar la resolución, realizando las pruebas de confirmación pertinentes para confirmar que el defecto ya no se reproduce y que no se han introducido nuevos problemas.

#### Cierre de Defectos

* Después de la verificación exitosa, el defecto puede ser marcado como resuelto/cerrado. Si un defecto no se puede resolver en el sprint actual, se revisará y reasignará en la próxima planificación de sprint.

#### Comunicación y Revisión

* Durante las reuniones de revisión de sprint, se discutirá el estado de los defectos abiertos y resueltos para mantener a todos los miembros del equipo informados. Las retrospectivas de cada sprint incluirán discusiones sobre los defectos para identificar áreas de mejora en el proceso de desarrollo y pruebas.

#### Documentación

* Todos los defectos y las acciones tomadas para su resolución serán documentadas meticulosamente para futuras referencias.

REQUERIMIENTOS DE DATOS DE PRUEBA

Este enfoque estructurado y detallado para los requerimientos de datos de prueba garantiza que se disponga de datos de alta calidad y representativos para las pruebas.

#### Definición de Datos de Prueba

* Los datos de prueba serán representativos de los escenarios reales de uso de la tienda por parte de los usuarios, incluyendo la navegación, selección de productos, personalización de camisetas, y procesos de compra.

#### Fuentes de Datos de Prueba

* **Datos Simulados:** Se generarán datos simulados para usuarios, transacciones, y personalizaciones de camisetas, asegurando una amplia cobertura de las posibilidades de uso.
* **Datos Reales Anonimizados:** Cuando sea posible y ético, se utilizarán datos reales anonimizados para pruebas.

#### Creación y Mantenimiento

* **Automatización de la Creación de Datos:** Utilizaremos scripts para automatizar la creación de datos de prueba, lo que permite generar rápidamente grandes volúmenes de datos representativos.
* **Mantenimiento de Datos:** Estableceremos procedimientos para asegurar que los datos de prueba sigan siendo relevantes y útiles a lo largo del tiempo.

#### Datos para Pruebas Específicas

* **Pruebas Funcionales y de UI:** Datos detallados para cada tipo de personalización disponible en la tienda, incluyendo combinaciones de colores, tamaños, y diseños.
* **Pruebas de Rendimiento:** Datos para simular diversos niveles de carga en el sistema, desde el uso normal hasta el pico de demanda durante, por ejemplo, promociones o campañas estacionales.
* **Pruebas de Seguridad:** Datos específicos para pruebas de inyección, cross-site scripting (XSS), y otros ataques, para evaluar la robustez de la plataforma frente a posibles vulnerabilidades.

#### Cumplimiento y Ética

* **Protección de Datos:** Aseguraremos que todos los datos de prueba cumplan con las normativas de protección de datos aplicables, como GDPR, especialmente cuando se utilicen datos reales anonimizados.
* **Ética de Uso:** Nos comprometeremos a un uso ético de los datos, evitando cualquier forma de discriminación o sesgo en la selección y utilización de los mismos.

ENTORNOS DE PRUEBA

#### Definición y Propósito de los Entornos

* **Entorno de Desarrollo Local:** Utilizado por el desarrollador externo para realizar cambios y pruebas iniciales antes de la integración. Este entorno ayuda a identificar errores tempranos en el ciclo de desarrollo.
* **Entorno de Pruebas Integradas (QA):** **Entorno de Producción (QA):** Donde el producto está en vivo y accesible para los usuarios finales. Dedicado a ejecutar pruebas automatizadas y manuales completas. Éste será el entorno idóneo para realizar las pruebas, aunque no es un entorno de pruebas per se, dadas las circunstancias del proyecto y el avanzado estado de su desarrollo.

#### Herramientas y Acceso

* **Herramientas de Integración y Despliegue Continuo (CI/CD):** Utilizaremos Azure o GitHub Actions como herramientas para automatizar el despliegue de aplicaciones en los diferentes entornos, facilitando pruebas rápidas y eficientes.
* **Control de Acceso:** Estableceremos políticas de control de acceso para los diferentes entornos, asegurando que solo el personal autorizado pueda realizar cambios o acceder a datos sensibles.

#### Monitoreo y Mantenimiento

* **Monitoreo Continuo:** Implementaremos soluciones de monitoreo en todos los entornos para detectar y responder rápidamente a problemas de rendimiento o errores.
* **Mantenimiento Regular:** Programaremos revisiones y mantenimiento regular de los entornos para asegurar que se mantengan actualizados y funcionando de manera óptima.

#### Cumplimiento y Seguridad

* **Seguridad:** Aplicaremos medidas de seguridad consistentes en todos los entornos para proteger contra vulnerabilidades y asegurar que la manipulación de datos cumpla con las regulaciones pertinentes, como GDPR.
* **Documentación:** Mantendremos una documentación detallada sobre la configuración, el uso y las políticas de cada entorno, facilitando la comprensión y el seguimiento de las prácticas de prueba.

PLAN DE COMUNICACIÓN

Establece cómo se comunicarán los avances, problemas, y resultados de las pruebas entre el equipo de pruebas y otros stakeholders del proyecto.

#### Objetivos del Plan de Comunicación

* Facilitar una comunicación clara y oportuna entre los miembros del equipo.
* Asegurar que la información importante sea compartida de manera eficiente para la toma de decisiones y la resolución de problemas.
* Mantener a todos los involucrados informados sobre el progreso, los cambios, y los resultados de las pruebas.

#### Herramientas de Comunicación

* **Herramientas Colaborativas:** Utilizaremos plataformas de comunicación como Microsoft Teams para la comunicación frecuente, permitiendo intercambios rápidos.
* **Gestión de Proyectos y Seguimiento de Tareas:** Utilizaremos herramientas como JIRA o Azure DevOps para el seguimiento de tareas, defectos y el progreso general del proyecto.
* **Videoconferencias:** Programaremos reuniones regulares a través de Zoom o Google Meet para discusiones más profundas o sesiones de planificación.

#### Estrategias de Comunicación

* **Reuniones de Planificación de Sprint:** Al inicio de cada sprint, llevaremos a cabo una reunión en persona para definir el trabajo a realizar, ajustar el plan de pruebas según las prioridades del sprint, y discutir cualquier cambio en los requisitos o en el enfoque de las pruebas.
* **Revisión de Sprint y Retrospectivas:** Al final de cada sprint, organizaremos una revisión para presentar los resultados de las pruebas al product owner, seguida de una retrospectiva para evaluar lo que funcionó bien, lo que se puede mejorar y planificar mejoras para el siguiente sprint.

#### Documentación y Registro

* **Centralización de la Documentación:** Mantener toda la documentación del proyecto, incluidos planes de pruebas, resultados, actas de reuniones y decisiones importantes, en un repositorio central accesible.
* **Registro de Comunicaciones:** Llevar un registro de las decisiones clave y los cambios acordados en reuniones para asegurar la transparencia y la trazabilidad.