

**1. Realizar las consultas que considere pertinentes al cliente, que le permitan acotar correctamente el alcance.**

- ¿Qué plataforma o sistema está utilizando actualmente para alojar sus cursos en línea?
- ¿Qué tipo de cursos ofrece Choucair Academy?
- ¿Hay algún software o herramienta específica que desea integrar con la plataforma de cursos en línea?
- ¿Hay algún requisito específico para matricularse en un curso?
- ¿Está buscando una solución de software escalable para manejar un creciente número de estudiantes y cursos?
- ¿Cómo se pueden consultar los cursos existentes en la plataforma?
- ¿Está buscando una solución de software personalizable para adaptarse a las necesidades específicas de Choucair Academy?
- ¿El proceso de matriculación es fácil de entender y realizar?
- ¿Hay algún requisito de seguridad específico que deba ser considerado en la plataforma de cursos en línea?
- ¿Qué tipo de recursos están disponibles para los estudiantes mientras realizan el curso?
- ¿Está buscando una solución de software fácil de usar para sus estudiantes y profesores?
- ¿Existe algún sistema de retroalimentación o evaluación para los estudiantes?
- ¿Está buscando una solución de software que brinde una experiencia de usuario intuitiva y atractiva para los estudiantes?
- ¿Hay algún soporte disponible para los estudiantes que tienen dificultades durante el curso?
- ¿Hay algún requisito de compatibilidad con dispositivos móviles o accesibilidad para personas con discapacidad?
- ¿Está buscando una solución de software que permita la integración con otros sistemas y plataformas externas?
- ¿Hay algún sistema de certificación disponible para los estudiantes que completan un curso exitosamente?
- ¿Se puede acceder a los cursos desde cualquier dispositivo y en cualquier momento?
- ¿Qué tipo de experiencia de usuario está buscando Choucair Academy para sus estudiantes?
- ¿Está buscando una solución de software con una amplia gama de funciones, como seguimiento de progreso, retroalimentación y evaluación?

**2. HU1 Búsqueda de cursos en línea**

COMO Un estudiante en línea de Choucair Academy

QUIERO poder buscar cursos específicos

PARA encontrar fácilmente los cursos que me interesen.

Descripción:

Como estudiante en línea, me gustaría tener la capacidad de buscar cursos específicos en la plataforma de educación en línea de Choucair Academy. Me gustaría tener un cuadro de búsqueda en la página principal donde pueda ingresar palabras clave o frases relacionadas con el curso que deseo tomar. Una vez que ingrese mi búsqueda, me gustaría que la plataforma muestre una lista de cursos relacionados con mi búsqueda. También me gustaría poder aplicar diferentes filtros, como nivel de dificultad, duración del curso para refinar mi búsqueda aún más.

### **Criterios de aceptación**

- El buscador debe estar ubicado en la página principal de la plataforma de educación en línea de Choucair Academy.
- El cuadro de búsqueda debe ser fácil de encontrar y usar.
- Los resultados de la búsqueda deben ser relevantes y precisos.
- Los filtros deben estar disponibles para refinar aún más los resultados de la búsqueda.
- Los resultados de la búsqueda deben actualizarse en tiempo real.
- El buscador debe ser accesible y fácil de usar en todos los dispositivos.

### **Criterios finales**

A. ¿Cree usted que puede diseñar casos de prueba sin tener la aplicación? ¿por qué?

Sí, es posible diseñar casos de prueba sin tener acceso a la aplicación. En general, el diseño de casos de prueba se basa en los requisitos o especificaciones del sistema o la aplicación que se está desarrollando.

Por lo tanto, se puede diseñar casos de prueba utilizando la información disponible, como la documentación de requisitos, diseños de interfaz de usuario, diagramas de flujo de proceso, diagramas de clases, entre otros. Además, los casos de prueba también pueden diseñarse a partir de conocimientos previos de la aplicación o de la experiencia en pruebas de software.

Es importante tener en cuenta que, aunque se pueden diseñar casos de prueba sin la aplicación, la prueba real de la aplicación es necesaria para verificar si los casos de prueba diseñados son efectivos y detectan los errores y problemas de la aplicación. Por lo tanto, una vez que la aplicación esté disponible, es importante probarla y ajustar los casos de prueba si es necesario para garantizar que se cubran adecuadamente todas las funcionalidades y posibles problemas.

B. ¿Qué información o documentación considera importante para el inicio de las pruebas?

- Documentación de requisitos: Esta documentación es fundamental para el diseño de casos de prueba. Los requisitos describen qué se espera que haga el software, y los casos de prueba se diseñan para verificar que se cumplan esos requisitos. Los requisitos pueden incluir especificaciones de funcionalidad, requisitos de rendimiento, requisitos de seguridad y otros.
- Diseños de interfaz de usuario: La documentación que describe el diseño de la interfaz de usuario puede ser útil para diseñar casos de prueba que aseguren que la aplicación es fácil de usar y que la navegación entre las pantallas funciona correctamente.

- Diagramas de flujo de procesos: Los diagramas de flujo pueden ayudar a entender cómo se procesan los datos en la aplicación, lo que es útil para diseñar casos de prueba que verifiquen la lógica de los procesos.
- Diagramas de clases: Los diagramas de clases pueden ser útiles para entender cómo se estructura el software y cómo se relacionan las diferentes clases y objetos, lo que puede ayudar en el diseño de casos de prueba.
- Documentación de casos de uso: La documentación de casos de uso describe los diferentes escenarios en los que se utilizará la aplicación. Esta información puede ser útil para diseñar casos de prueba que cubran los diferentes casos de uso de la aplicación.

C. ¿Si la documentación no es clara usted que haría?

Comunicarse con el equipo de desarrollo: Si la documentación no es clara, el tester puede comunicarse con el equipo de desarrollo para obtener más información. Esto puede incluir discutir los requisitos con el equipo de desarrollo, pedir aclaraciones sobre la documentación existente y solicitar que se proporcione documentación adicional.

Realizar pruebas exploratorias: Si la documentación es insuficiente o no está clara, el tester puede realizar pruebas exploratorias. Las pruebas exploratorias permiten al tester explorar el sistema y encontrar problemas que de otra manera podrían haber pasado desapercibidos. El tester puede utilizar su conocimiento y experiencia para probar diferentes escenarios y descubrir posibles problemas.

Hacer preguntas y solicitar retroalimentación: El tester puede hacer preguntas y solicitar retroalimentación a los usuarios finales o a los responsables del proyecto para entender mejor los requisitos de la aplicación. Esto puede ayudar al tester a aclarar la documentación y diseñar casos de prueba más efectivos.

Documentar problemas: Si el tester encuentra problemas o ambigüedades en la documentación, es importante documentarlos y comunicarlos al equipo de desarrollo para que puedan ser abordados.

D. ¿Quién cree que es el responsable de que usted conozca el contexto de negocio y la aplicación a probar?

En este caso, el gerente de servicio debería proporcionar la información necesaria a los testers sobre el contexto del negocio, los requisitos y objetivos de la aplicación, así como los riesgos y limitaciones. El gerente de servicio también debería asegurarse de que los testers tengan acceso a la documentación relevante y que tengan la oportunidad de hacer preguntas y aclaraciones sobre el contexto y los requisitos.

Es importante destacar que la colaboración y comunicación efectiva entre el gerente de servicio y los testers es fundamental para garantizar que el equipo de pruebas tenga la información necesaria para diseñar y ejecutar casos de prueba efectivos. De esta manera, el gerente de servicio puede contribuir a mejorar la calidad del software y garantizar la satisfacción del cliente.

E. ¿Si entregara un informe de avance de sus pruebas que información relacionaría?

Cobertura de pruebas: es importante informar sobre la cantidad de casos de prueba diseñados, ejecutados y pendientes, así como la cobertura de pruebas en términos de los módulos o

funcionalidades de la aplicación que se han probado y los que quedan por probar.

Resultados de las pruebas: es necesario informar sobre los resultados de las pruebas realizadas, incluyendo los defectos encontrados, su criticidad y su estado actual. También se pueden incluir estadísticas sobre la tasa de éxito de las pruebas realizadas.

Problemas y riesgos: es importante informar sobre cualquier problema o riesgo que se haya identificado durante las pruebas, como problemas con la configuración de la prueba, problemas con el entorno de prueba o problemas con los datos de prueba.

Fechas y plazos: es necesario informar sobre las fechas y plazos de entrega, así como el progreso actual en relación con estos plazos.

Recomendaciones: se pueden incluir recomendaciones para mejorar la calidad del software, como sugerencias para mejorar los casos de prueba, sugerencias para mejorar la documentación de requisitos o sugerencias para mejorar el proceso de pruebas.

F. ¿si tuviera que entregar indicadores del proceso de pruebas que incluiría?

Cobertura de pruebas: este indicador mide la cantidad de casos de prueba diseñados y ejecutados en comparación con los casos de prueba planificados. La cobertura de pruebas puede ayudar a determinar si se han realizado suficientes pruebas en todos los módulos y funcionalidades de la aplicación.

Tasa de defectos: este indicador mide la cantidad de defectos encontrados en relación con el número total de casos de prueba ejecutados. La tasa de defectos puede ayudar a determinar la calidad del software y si se están encontrando suficientes defectos durante el proceso de pruebas.

Tiempo de resolución de defectos: este indicador mide el tiempo que se tarda en resolver un defecto desde el momento en que se informa hasta el momento en que se cierra. Este indicador puede ayudar a identificar cuellos de botella y retrasos en el proceso de corrección de defectos.

Tasa de defectos críticos: este indicador mide la cantidad de defectos críticos encontrados en relación con el número total de defectos. Los defectos críticos son aquellos que pueden tener un impacto significativo en la funcionalidad o la seguridad del software. La tasa de defectos críticos puede ayudar a identificar áreas de mayor riesgo en el software.

Tiempo promedio de ciclo de pruebas: este indicador mide el tiempo que tarda el equipo de pruebas en diseñar, ejecutar y evaluar un conjunto de pruebas. El tiempo promedio de ciclo de pruebas puede ayudar a identificar áreas donde se puede mejorar la eficiencia del proceso de pruebas.

Índice de satisfacción del cliente: este indicador mide la satisfacción del cliente con la calidad del software. El índice de satisfacción del cliente puede ayudar a identificar áreas en las que se puede mejorar la calidad del software y la experiencia del usuario.

