



GESTIÓN DEL PROCESO DE DESARROLLO DE SOFTWARE

DIVISIÓN DE TECNOLOGÍAS DE LA INFORMACIÓN
INGENIERÍA EN GESTIÓN Y DESARROLLO DE SOFTWARE

PRODUCTO 2

NOMBRE DEL ALUMNO(A):

ALEJANDRO RAMÍREZ ROMERO

GUADALUPE JAÉN FLORES

DIANA LAURA VELÁZQUEZ GARCÍA

ALAN MANUEL MARTÍNEZ VALDEZ

NOMBRE DE/LA PROFESOR (A):

CRUZ SALAS SOLÍS

16 DE OCTUBRE DE 2023

Introducción	2
Planteamiento del problema	3
Objetivo	5
Casos de pruebas a realizar al software	6
Herramienta para la automatización de pruebas	9
Proceso de pruebas	10
Planificación de Pruebas	10
Pruebas Unitarias	10
Pruebas de Integración	10
Pruebas de Funcionalidad	10
Pruebas de Usabilidad	11
Pruebas de Seguridad	11
Pruebas de rendimiento	11
Pruebas de compatibilidad	11
Pruebas de carga	11
Pruebas de Recuperación y Respaldo	12
Pruebas de Aprobación de Usuarios (UAT)	12
Generación de Informes	12
Resolución de Problemas	12
Lanzamiento y Monitoreo Continuo	12
Proceso de integración continua (Pipeline)	14
Control de versiones (Git)	14
Herramientas de Automatización de Pruebas	14
Plataforma de CI	15
Ejecución de Pruebas	16
Análisis de resultados	16
Notificaciones	16
Despliegue Automatizado	16
Registro y Almacenamiento de Artefactos	16
Retorno a la Etapa de Desarrollo	17
Monitoreo Continuo	17
Comunicación de resultado de pruebas	18
Generación de Informes de Pruebas	18
Informes de errores y problemas	18
Resultados de Pruebas Automatizadas	18
Resultados de Pruebas Manuales	18
Clasificación de Problemas	19
Notificaciones Inmediatas	19
Informe de resumen	19
Reuniones de Retroalimentación	19
Seguimiento de Problemas	19
Comentarios continuos	20
Documentación de Mejoras y Correcciones	20
Configuración y uso de las herramientas de compilación	21

INTRODUCCIÓN

Las aplicaciones de reseña de lugares son herramientas digitales que permiten a los usuarios compartir sus opiniones, calificaciones y comentarios sobre una amplia variedad de establecimientos y ubicaciones, como restaurantes, hoteles, tiendas, parques, museos y más. Estas aplicaciones han ganado una gran popularidad en la era digital y móvil, y desempeñan un papel crucial en la toma de decisiones de muchas personas al buscar lugares para visitar, comer, comprar o recibir.

Algunas de las aplicaciones de reseña de lugares más conocidas incluyen Yelp, TripAdvisor, Google Maps, Foursquare y Zomato, entre otras. Estas aplicaciones han revolucionado la forma en que las personas exploran y eligen lugares para visitar, al proporcionar acceso a una amplia base de datos de opiniones y experiencias de otros usuarios, lo que les ayuda a tomar decisiones más informadas y satisfactorias en su vida cotidiana y en sus viajes.

A continuación, profundizaremos el tema con base a nuestro proyecto acerca de una aplicación de reseñas de antros en la ciudad de Puebla.

PLANTEAMIENTO DEL PROBLEMA

En la actualidad, la ciudad de Puebla, México, es conocida por su vibrante vida nocturna y su amplia oferta de antros y bares para los residentes y turistas. Sin embargo, a pesar de la abundancia de opciones disponibles, los consumidores se enfrentan a desafíos significativos al intentar elegir un antro que se adapte a sus gustos y preferencias individuales. Esto se debe en gran parte a la falta de una plataforma de reseñas de antros en Puebla que proporcione información precisa y confiable sobre la calidad y la experiencia en estos establecimientos.

Implementaremos la aplicación en base a los siguientes puntos:

Falta de Información Confiable: Actualmente, no existe una aplicación de reseñas de antros en Puebla que ofrezca una amplia base de datos de opiniones y calificaciones sobre estos establecimientos. Esto deja a los consumidores en la oscuridad al tratar de determinar la calidad de un antro antes de visitarlo.

Experiencias Variadas: Los antros en Puebla varían ampliamente en términos de música, ambiente, precios y calidad de servicio. Los usuarios enfrentan dificultades para encontrar antros que se ajusten a sus gustos y necesidades específicas debido a la falta de información detallada.

Riesgos de Decisiones Erróneas: La falta de reseñas confiables puede llevar a los consumidores a tomar decisiones equivocadas al elegir un antro, lo que puede resultar en experiencias insatisfactorias y desperdicio de dinero.

Falta de Interacción y Comunidad: La comunidad de amantes de la vida nocturna en Puebla carece de una plataforma interactiva donde puedan compartir experiencias, consejos y recomendaciones sobre los antros locales.

Potencial para la Mejora: Los antros podrían beneficiarse de una retroalimentación constructiva proporcionada por los clientes a través de una plataforma de reseñas, lo que les permitiría realizar mejoras y ajustes en función de las opiniones de los usuarios.

OBJETIVO

Desarrollar una aplicación que permita a los usuarios revisar y calificar antros, compartir sus experiencias y ayudar a otros a tomar decisiones informadas sobre dónde pasar su tiempo en la vida nocturna de Puebla. Además, se espera que esta aplicación promueva una comunidad activa de usuarios que puedan interactuar, compartir consejos y contribuir al enriquecimiento de la vida nocturna en la ciudad.

CASOS DE PRUEBAS A REALIZAR AL SOFTWARE

Para garantizar el funcionamiento adecuado y la calidad de la aplicación de reseñas de antros en Puebla, es esencial realizar una serie de casos de prueba que abarquen diferentes aspectos y funcionalidades de la aplicación. A continuación, se describen algunos casos de prueba que podrían llevarse a cabo:

1. Registro de Usuario:

Caso de prueba 1.1: Verificar que un nuevo usuario pueda registrarse correctamente en la aplicación utilizando un correo electrónico válido y una contraseña segura.

Caso de prueba 1.2: Comprobar que el sistema valida la fortaleza de la contraseña durante el registro.

Caso de prueba 1.3: Asegúrese de que no se permita el registro con un correo electrónico ya registrado previamente.

2. Inicio de Sesión:

Caso de prueba 2.1: Verificar que un usuario registrado pueda iniciar sesión con éxito utilizando su correo electrónico y contraseña.

Caso de prueba 2.2: Confirmar que el sistema muestre un mensaje de error adecuado en caso de credenciales incorrectas.

3. Búsqueda de Antros:

Caso de prueba 3.1: Probar la funcionalidad de búsqueda para asegurarse de que los usuarios puedan encontrar antros por nombre, ubicación o categoría.

Caso de prueba 3.2: Verifique que los resultados de búsqueda sean precisos y relevantes.

4. Reseñas y Calificaciones:

Caso de prueba 4.1: Asegurarse de que los usuarios puedan agregar reseñas y calificaciones a los antros que han visitado.

Caso de prueba 4.2: Comprobar que las reseñas incluyan campos para la descripción, calificación numérica y fotos opcionales.

Caso de prueba 4.3: Verificar que los usuarios puedan editar y eliminar sus propias reseñas si es necesario.

5. Perfil de Usuario:

Caso de prueba 5.1: Confirmar que los usuarios pueden acceder a su perfil, donde puedan ver y editar su información personal.

Caso de prueba 5.2: Verificar que los usuarios puedan ver un registro de sus reseñas anteriores en su perfil.

6. Comentarios y Respuestas:

Caso de prueba 6.1: Probar la capacidad de los usuarios para comentar en las reseñas de otros usuarios.

Caso de prueba 6.2: Verificar que los dueños de antros puedan responder a las reseñas de los usuarios.

Caso de prueba 6.3: Asegúrese de que los comentarios se muestren en el orden correcto y se actualicen correctamente.

7. Integración de Mapas:

Caso de prueba 7.1: Verifique que los antros se muestren en un mapa interactivo con marcadores precisos.

Caso de prueba 7.2: Comprobar que los usuarios puedan obtener direcciones y direcciones de navegación a los antros desde la aplicación.

8. Compatibilidad Móvil:

Caso de prueba 8.1: Asegúrese de que la aplicación sea compatible con dispositivos móviles y que la interfaz sea responsiva en diferentes tamaños de pantalla.

HERRAMIENTA PARA LA AUTOMATIZACIÓN DE PRUEBAS

Una herramienta popular para la automatización de pruebas en aplicaciones para Android e iOS es Appium. Appium es una herramienta de código abierto que permite a los desarrolladores y probadores automatizar pruebas en aplicaciones móviles nativas, híbridas y basadas en la web en dispositivos Android e iOS. Algunas de las características destacadas de Appium incluyen:

- Soporte multiplataforma: Appium es compatible con dispositivos Android e iOS, lo que facilita la automatización de pruebas en ambas plataformas con un único conjunto de pruebas.
- Lenguajes de programación: Puedes escribir tus pruebas en varios lenguajes de programación populares, como Java, Python, Ruby, C#, etc.
- Integración con marcos de prueba: Appium se integra con marcos de prueba populares como TestNG, JUnit y otros, lo que facilita la gestión de pruebas y resultados.
- Acceso completo a las API nativas: Appium permite interactuar con las API nativas de Android e iOS, lo que te permite probar una amplia gama de funcionalidades.
- Emuladores y dispositivos reales: Puedes ejecutar pruebas en emuladores o dispositivos físicos, lo que brinda flexibilidad en cuanto a la configuración de tu entorno de prueba.
- Compatibilidad con aplicaciones híbridas y nativas: Appium es adecuado para probar tanto aplicaciones nativas como híbridas.
- Amplia comunidad y soporte: Appium tiene una comunidad activa de usuarios y ofrece soporte continuo y actualizaciones.

PROCESO DE PRUEBAS

El proceso de pruebas para una aplicación de reseñas de antros en la ciudad de Puebla es crítico para garantizar que la aplicación sea confiable, segura y cumpla con las expectativas de los usuarios. A continuación, describe un proceso de pruebas generales que puedes seguir:

Planificación de Pruebas

Identifica los objetivos de las pruebas y los escenarios clave que deben evaluarse.

Defina los casos de prueba, incluyendo pruebas de funcionalidad, usabilidad, seguridad y rendimiento.

Diseña un plan de pruebas que establezca el alcance, los cronogramas y los recursos necesarios.

Pruebas Unitarias

Comienza con pruebas unitarias para verificar que las funciones y componentes individuales de la aplicación funcionan correctamente.

Verifique que los módulos de la aplicación, como el sistema de autenticación, la gestión de reseñas y la integración de mapas, se comportan según lo previsto.

Pruebas de Integración

Realiza pruebas de integración para asegurarte de que los diferentes módulos de la aplicación funcionen juntos sin conflictos.

Compruebe la comunicación adecuada entre los componentes y que los datos se transmitan de manera correcta.

Pruebas de Funcionalidad

Ejecuta pruebas de funcionalidad para validar que la aplicación cumple con los requisitos especificados.

Verifica que los usuarios puedan registrarse, iniciar sesión, buscar antros, agregar reseñas, etc., sin problemas.

Pruebas de Usabilidad

Realice pruebas de usabilidad para evaluar la facilidad de uso de la aplicación.

Reúne a usuarios reales y solicita sus comentarios sobre la interfaz de usuario, la navegación y la experiencia en general.

Pruebas de Seguridad

Evalúa la seguridad de la aplicación mediante pruebas de seguridad, como la inyección de SQL, el cross-site scripting (XSS), y la gestión segura de contraseñas.

Verifica la protección de datos de usuario y la prevención de acceso no autorizado.

Pruebas de rendimiento

Lleva a cabo pruebas de rendimiento para evaluar la capacidad de la aplicación bajo cargas de trabajo simuladas.

Analiza el rendimiento de la aplicación bajo condiciones de tráfico pesado y evalúa los tiempos de respuesta.

Pruebas de compatibilidad

Asegúrate de que la aplicación sea compatible con diferentes dispositivos móviles (iOS y Android) y navegadores web.

Pruebe en una variedad de dispositivos y tamaños de pantalla.

Pruebas de carga

Realice pruebas de carga para verificar cómo responde la aplicación bajo cargas máximas simuladas, como un gran número de usuarios accediendo simultáneamente.

Pruebas de Recuperación y Respaldo

Evalúa la capacidad de la aplicación para recuperarse de posibles fallos, como caídas del servidor, y respaldar los datos de los usuarios de manera segura.

Pruebas de Aprobación de Usuarios (UAT)

Lleva a cabo pruebas de aprobación de usuarios con un grupo de usuarios representativos de Puebla para obtener comentarios finales y validación de la aplicación.

Generación de Informes

Documenta los resultados de las pruebas, incluyendo problemas identificados y su gravedad.

Proporciona informes detallados que incluyen capturas de pantalla y registros de pruebas.

Resolución de Problemas

Corrige los problemas identificados durante las pruebas y realiza pruebas de regresión para asegurarte de que las correcciones no afecten otras áreas de la aplicación.

Lanzamiento y Monitoreo Continuo

Después de que todas las pruebas hayan sido exitosas y los problemas resueltos, lanza la aplicación.

Implemente herramientas de monitoreo para supervisar la aplicación en producción y realice ajustes según sea necesario.

PROCESO DE INTEGRACIÓN CONTINUA (PIPELINE)

El proceso de Integración Continua (CI) es esencial en el desarrollo de una aplicación de reseñas de antros en la ciudad de Puebla. La CI implica la integración y prueba automática regular de cambios en el código, lo que permite detectar problemas rápidamente y garantizar que la aplicación se mantenga en un estado funcional en todo momento. Aquí te describe el proceso de Integración Continua específico para esta aplicación:

Control de versiones (Git)

El desarrollo de la aplicación se lleva a cabo en un sistema de control de versiones como Git. Cada vez que se realiza una modificación en el código, se crea una nueva rama.

El control de versiones con Git es esencial para el desarrollo de una aplicación de reseñas de antros en la ciudad de Puebla. Git te permite rastrear y administrar las diferentes versiones de tu código fuente, lo que facilita la colaboración en equipo, el seguimiento de cambios y la reversión a versiones anteriores si es necesario.

Herramientas de Automatización de Pruebas

Se utilizan herramientas de automatización de pruebas, como Appium o Selenium, para escribir casos de prueba automatizados que verifican la funcionalidad de la aplicación en diferentes escenarios.

Appium

Appium es una herramienta de automatización de pruebas de código abierto que permite automatizar aplicaciones móviles en iOS y Android. Puedes utilizarlo para automatizar las pruebas de la aplicación en dispositivos móviles y emuladores.

Plataforma de CI

Se configura una plataforma de CI, como Jenkins, Travis CI, CircleCI o GitLab CI/CD, para automatizar el proceso de integración y prueba.

Desencadenantes de Integración:

Se establecen desencadenantes (triggers) para la ejecución automática de las pruebas de CI. Estos desencadenantes pueden ser cambios en el repositorio de código, como confirmaciones en la rama principal (master) o en las ramas de características.

GitLab CI/CD

GitLab incluye su propia funcionalidad de CI/CD que se integra directamente con repositorios de GitLab. Es una opción conveniente si ya estás utilizando GitLab para la gestión de tu código fuente.

GitHub Actions

GitHub Actions es una solución de CI/CD integrada en GitHub que te permite automatizar el flujo de trabajo de CI directamente desde tu repositorio de GitHub.

Ejecución de Pruebas

Cuando se detecta un cambio en el repositorio de código, la plataforma de CI inicia automáticamente la ejecución de pruebas automatizadas. Esto puede incluir pruebas de unidad, pruebas de integración, pruebas de interfaz de usuario, pruebas de rendimiento, y más, según sea necesario.

Análisis de resultados

Después de que las pruebas se completen, la plataforma de CI analiza los resultados. Si todas las pruebas son exitosas, el proceso continúa. Si se detectan errores, se genera un informe detallado con información sobre los problemas encontrados.

Notificaciones

Se configuran notificaciones para informar a los miembros del equipo de desarrollo sobre los resultados de las pruebas. Esto puede incluir notificaciones por correo electrónico, mensajes en Slack o integraciones con otras herramientas de comunicación.

Despliegue Automatizado

Si todas las pruebas son exitosas, la aplicación se considera lista para quedar desplegada. Se puede configurar un proceso de implementación automática en la plataforma de CI para implementar la nueva versión en un entorno de producción o de pruebas.

Registro y Almacenamiento de Artefactos

Se almacenan artefactos generados durante el proceso de CI, como aplicaciones compiladas y archivos de registro. Estos artefactos pueden ser útiles para fines de auditoría y seguimiento.

Retorno a la Etapa de Desarrollo

Si se encuentran problemas durante la ejecución de pruebas, el equipo de desarrollo es notificado y puede corregir los errores. Luego, los cambios se vuelven a enviar al sistema de control de versiones, lo que desencadena un nuevo ciclo de CI.

Monitoreo Continuo

Después del despliegue, la aplicación se somete a monitoreo continuo para detectar cualquier problema en tiempo real. Los problemas que surgen en la producción pueden llevar a nuevas modificaciones en el código y desencadenar ciclos adicionales de CI.

COMUNICACIÓN DE RESULTADO DE PRUEBAS

La comunicación efectiva de los resultados de las pruebas es esencial para garantizar que los equipos de desarrollo y calidad estén informados sobre el estado de la aplicación y puedan tomar medidas para abordar los problemas identificados. Aquí se describe el proceso de comunicación de resultados de las pruebas para la aplicación de reseñas de antros en la ciudad de Puebla:

Generación de Informes de Pruebas

Después de que se ejecuten las pruebas, se generarán informes detallados que documentarán los resultados. Estos informes deben incluir detalles sobre las pruebas realizadas, los casos de prueba, las métricas de rendimiento y cualquier problema identificado.

Informes de errores y problemas

En el informe, se deben destacar claramente los errores y problemas identificados durante las pruebas. Cada problema debe incluir una descripción detallada, capturas de pantalla, pasos para reproducir el problema y su gravedad.

Resultados de Pruebas Automatizadas

Los resultados de las pruebas automatizadas, incluyendo pruebas de unidad, pruebas de integración, pruebas de interfaz de usuario y otras, se deben documentar y compartir con el equipo.

Resultados de Pruebas Manuales

Si se realizan pruebas manuales, se deben proporcionar informes detallados que incluyan los comentarios y observaciones de los probadores. Esto puede incluir comentarios sobre la usabilidad, la experiencia del usuario y otros aspectos críticos.

Clasificación de Problemas

Los problemas identificados durante las pruebas deben clasificarse según su gravedad y prioridad. Por lo general, se utilizan categorías como "Crítico", "Mayor", "Menor" y "Mejora" para clasificar los problemas.

Notificaciones Inmediatas

Los problemas críticos o bloqueantes deben notificarse de inmediato al equipo de desarrollo y otros miembros clave del proyecto para que puedan tomar medidas de corrección rápidamente.

Informe de resumen

Además de los informes detallados de problemas, se debe crear un informe de resumen que incluya una visión general de los resultados de las pruebas. Esto puede incluir estadísticas clave, como la cantidad de pruebas exitosas y fallidas, el rendimiento general y las métricas de calidad.

Reuniones de Retroalimentación

Organice reuniones de retroalimentación periódicas con el equipo de desarrollo y calidad para discutir los resultados de las pruebas. Durante estas reuniones, se pueden abordar preguntas, aclaraciones y planes de acción para abordar los problemas identificados.

Seguimiento de Problemas

Utiliza un sistema de seguimiento de problemas, como Jira, Trello o una herramienta similar, para registrar y dar seguimiento a los problemas identificados. Esto permite un seguimiento ordenado y garantiza que los problemas se aborden de manera adecuada.

Comentarios continuos

Fomenta una cultura de retroalimentación continua en la que los miembros del equipo estén dispuestos a comunicar y discutir los resultados de las pruebas de manera abierta. Esto puede incluir la colaboración para solucionar problemas y mejorar la calidad de la aplicación.

Documentación de Mejoras y Correcciones

A medida que se aborde los problemas identificados, documente las mejoras y correcciones realizadas en la aplicación. Esto asegura que el equipo tiene un historial completo de las acciones tomadas.

CONFIGURACIÓN Y USO DE LAS HERRAMIENTAS DE COMPILACIÓN

Se utilizará las siguientes herramientas de compilación para hacer la compilación del proyecto elaborado

Visual Studio Code

Visual Studio Code (VS Code) es un editor de código fuente gratuito y de código abierto desarrollado por Microsoft. Aunque se llama "Visual Studio", es más ligero y se centra principalmente en la edición de código, en lugar de ser un entorno de desarrollo completo como Visual Studio.

Dart

Dart es un lenguaje de programación desarrollado por Google que se utiliza principalmente para construir aplicaciones móviles, web y de escritorio. Dart es conocido por su sintaxis limpia y fácil de aprender, y se utiliza en el desarrollo de aplicaciones en Flutter, un popular framework de código abierto también creado por Google.

Se utiliza para escribir aplicaciones de alta calidad y rendimiento, ya que es un lenguaje orientado a objetos que admite programación orientada a aspectos. Dart se puede utilizar para desarrollar tanto el lado del cliente como el servidor de una aplicación, lo que lo hace versátil.

En cuanto a su uso, se escribe código en Dart utilizando un editor de texto o un entorno de desarrollo integrado (IDE) como Visual Studio Code con el complemento de Dart y Flutter instalado. Una vez que se ha escrito el código, se puede compilar a JavaScript para su ejecución en navegadores web o se puede utilizar directamente con el framework Flutter para construir aplicaciones móviles multiplataforma. La versatilidad y la facilidad de aprendizaje hacen que Dart sea una opción popular para los desarrolladores que buscan crear aplicaciones modernas y eficientes.

Flutter

Flutter es un framework de código abierto desarrollado por Google, diseñado para la creación de aplicaciones nativas para móviles, web y escritorio desde una sola base de código. Se destaca por su capacidad para construir interfaces de usuario atractivas y de alto rendimiento.

Utiliza un lenguaje de programación llamado Dart y ofrece un conjunto de herramientas que permiten a los desarrolladores escribir código una vez y desplegar la aplicación en múltiples plataformas, lo que ahorra tiempo y esfuerzo.

Para usar Flutter, se necesitan conocimientos básicos de programación en Dart. Se utiliza el entorno de desarrollo integrado (IDE) de preferencia, como Android Studio, Visual Studio Code o IntelliJ IDEA con el complemento de Flutter instalado. Los desarrolladores pueden crear interfaces de usuario mediante widgets, que son los bloques de construcción de la interfaz en Flutter. Estos widgets son personalizables y se combinan para formar la apariencia y funcionalidad de la aplicación

Chrome

Google Chrome es un navegador web desarrollado por Google, diseñado para permitir a los usuarios acceder y navegar por internet de manera rápida, segura y eficiente. Es un software que te permite visualizar páginas web, realizar búsquedas, acceder a sitios, ver videos, entre otras actividades en línea.

Xampp

XAMPP es un entorno de desarrollo que facilita la creación de sitios web dinámicos. Su nombre es un acrónimo que representa los componentes principales: X (cualquier sistema operativo), Apache (el servidor web), MySQL (la base de datos), PHP (lenguaje de programación) y Perl (otro lenguaje de programación).

Se utiliza principalmente para crear y probar sitios web de manera local antes de ponerlos en un entorno de producción. Al integrar un servidor web, una base de datos y lenguajes de programación, permite a los desarrolladores trabajar en aplicaciones web sin necesidad de conexión a internet, lo que facilita el desarrollo, pruebas y depuración de sitios web o aplicaciones.

Para usar XAMPP, debes descargar e instalar el paquete que se adapte a tu sistema operativo. Una vez instalado, puedes iniciar el panel de control y activar los servicios de Apache y MySQL. Luego, podrás colocar tus archivos de sitio web en el directorio de XAMPP y acceder a tu proyecto a través de un navegador web utilizando la URL local, por ejemplo, <http://localhost/tusitio>.

Mysql

MySQL es un sistema de gestión de bases de datos relacionales ampliamente utilizado. Se utiliza para almacenar, organizar y administrar datos. Es especialmente útil en aplicaciones web, como sitios web, aplicaciones empresariales y sistemas de gestión, ya que permite el almacenamiento y recuperación eficiente de información.