UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
SOFTWARE AVANZADO
CATEDRÁTICO: ING. JOSE MANUEL RUIZ
AUXILIAR: MANUEL ALEJANDRO DE MATA MAYEN

PROYECTO LABORATORIO - “FULL TRIP”

1. **Introducción**
2. **Objetivos.**
   General
   Específicos.
3. **Full Trip**
   3.1 Usuarios
   3.2 Registro e inicio de sesión
   3.3 Hoteles.
   3.4 Renta de autos.
   3.5 Vuelos.
   3.6 Reseñas.
4. **Requisitos tecnicos**
5. Consideraciones
6. Entregas
   FASE 1 - Documentación
   FASE 2
   FASE 3

Objetivos:

General:

- Abordar el problema dado desde la perspectiva de un arquitecto de software y así brindar una solución.
- Conocer y entender el concepto de pruebas unitarias.
- Comprender la importancia del trabajo en equipo.

Específicos:

- Proponer soluciones óptimas a un problema dado por el cliente.
- El estudiante podrá listar las diferentes pruebas que se pueden realizar en un proyecto de software.
- utilizar herramientas para realizar pruebas unitarias.

3. **Full trip**

“Full trip**”** es el sistema que se desarrollará para gestionar el viaje, estadía y transporte de todo turista alrededor del mundo; con el fin de garantizar una completa y agradable experiencia en época de post pandemia. La finalidad es centralizar los datos y que el cliente haga la menor cantidad de validaciones al momento de planificar su viaje.

3.1 Usuarios

1. **Turista:** Toda persona que desea hacer un viaje a cualquier parte del mundo.
2. **Hoteles**: Entienda que desea poner en el mercado del turismo las instalaciones de hoteles para que se puedan hacer reservaciones en sus instalaciones para la estadística de turistas.
3. **Renta de Autos**: Entidad que se dedica al alquiler de autos.
4. **Admin Vuelos**: Entidad que se dedica a la creación de vuelos disponibles, horarios etc.

Tanto los hoteles como las empresas de renta de autos, vuelos tienen el rol de “servicios
tercerizados” .

3.2 Registro e inicio de sesión

La página web debe de tener una página de inicio y en ella se debe de poder iniciar sesión o generar nuevos registros de usuarios para poder acceder al sistema y tener las vistas correspondientes a su tipo de usuario. Desde la página web se pueden registrar turistas y servicios tercerizados al sistema.

3.3 Hoteles

Cada hotel debe de crear un calendario con las reservaciones que va a poner a disposición de los turistas. Los usuarios tipo turista podrán hacer reservaciones y para generar una experiencia agradable, se debe de contar con un buscador donde se puedan generar consultas en
cascada de hoteles por:

- País
- Ciudad
- Cantidad de personas
- Rango de precios y/o rango de fechas.

3.4 Renta de Autos

El sistema debe de poder permitir ingresar carros para rentar de forma manual o por carga masiva desde el usuario de tipo renta autos.
El turista debe de tener poder hacer reserva de los autos que desea rentar o rentarlos en el momento.

3.5 Vuelos

Deben de poderse crear vuelos de forma manual. Existen dos tipos de viajes: sólo ida o ida y vuelta.

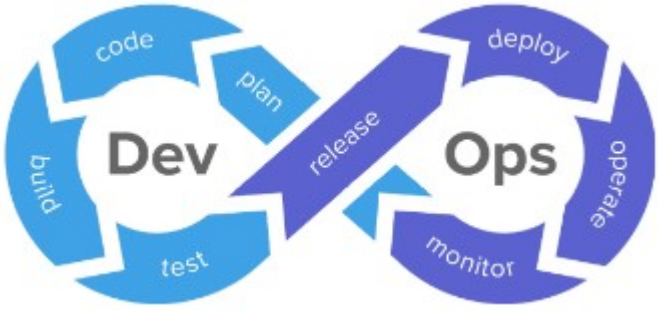
3.6 Reseñas

Todo servicio (vuelos, reservaciones de hoteles y autos) debe de poder tener la opción de poder dar una calificación al finalizar un servicio y que el turista pueda dejar un comentario respecto al servicio que utilizó. Estas reseñas se deben de poder visualizar dentro de la pantalla de inicio del usuario.

4. Requisitos técnicos.
    1. “Full trip” debe de contar con todos los servicios requeridos para su funcionamiento: backend, frontend, bases de datos (relacionales y/o no relacionales).
    2. Git: Uso de un repositorio para el control de versiones. Debe de hacer uso de un flujo de trabajo de Git. Debe de agregar al auxiliar al repositorio y darle los permisos de propietario. (gitlab: matalejandro18).
    3. Semantic Versioning: Correcto uso de ramas y tags para el control de versiones en el repositorio.
    4. Responsive Design (página web). El diseño de la aplicación debe de ser capaz de adaptarse a cualquier dispositivo.
    5. Documentación: La documentación debe de guardarse en el repositorio, en una rama con el nombre de “documentación”. Para la entrega de cada fase la documentación debe de estar actualizada.

       La documentación debe incluir:
        - Descripción de la metodología de desarrollo a utilizar y su justificación.
        - Descripción del modelo de branching a utilizar.
        - Toma de requerimientos
            - Funcionales
            - No funcionales
        - Para este punto tenga en cuenta que debe de aplicar la ingeniería de requerimientos y no solo listar los requerimientos ya que es parte fundamental para el desarrollo del proyecto. Debe de dejar claro que métricas utilizará para medir el cumplimiento de los requerimientos
        - Historias de usuario.
        - Descripción de las tecnologías a utilizar
        - Diagramas de actividades.
        - Arquitectura de sistemas.
        - Diagrama de datos o almacenamiento: ejemplo Modelo Entidad Relación.
        - Descripción de la seguridad de la aplicación. Toma en cuenta el proceso de inicio de sesión y que garantice que la persona que inicia sesión es realmente quien dice ser.
        - Mockups de las principales vistas para la página web y la aplicación móvil.
        - Documentación de las Pipelines para los servicios (para la fase final).

6. Pruebas unitarias: Generación de pruebas unitarias durante el desarrollo, estas pruebas deben de ser ejecutadas automáticamente en el ciclo DevOps. Al encontrarse un error debe de notificarse y detener el pipeline o stage.
7. Pruebas Funcionales: Generación de pruebas funcionales durante el desarrollo.
8. DevOps: Debe de implementarse todo el ciclo DevOps al finalizar el proyecto.



9. Pipelines: Los pipelines se ejecutarán automáticamente. Si se detectan cambios en el repositorio se debe ejecutar el stage correspondiente. Haga uso correcto de los Pull Request o Merge Request.
10. Issues: Uso de herramientas para el manejo de tareas, tickets o historias de usuarios, según la metodología que haya decidido utilizar. Debe de agregar al auxiliar con los permisos necesarios en la herramienta para dar seguimiento a su desarrollo.

Entrega 3

- Es requisito mínimo contar con la fase 2 para poder tener derecho a calificación de la fase 3. Esta fase implica ambiente DevOps
- Realizar pruebas Funcionales, deben de pasar todas las pruebas exitosas, realizar 2 pruebas funcionales por cada funcionalidad utilizada en el proyecto, de no pasar una prueba se penalizará con el 10%.
- Implementación de ci/cd con gitlab. en donde deberán ejecutar la aplicación en la Nube y pruebas unitarias.
- Para tener derecho a calificación toda la implementación debe de ser en un entorno en la nube deployando su aplicación utilizando herramienta de gitlab CI/CD.
- Monitoreo de código y aplicación utilizando Sonarqube.
- La entrega completa de la fase 3, Tag de entrega: v3.0.0.
- Fecha de Entrega 03/07/2022 a las 23:59 P.M.