**EJERCICIO INGENIERIA DE SOFTWARE II**

1. Definir el Escenario: Imagina que estás trabajando en un equipo de desarrollo de

software que está creando una aplicación móvil para la gestión de tareas. Define los

elementos de configuración relevantes, como el código fuente, la interfaz de usuario,

la documentación y los recursos gráficos.

2. Identificación de Configuración: Enumera y documenta todos los elementos de

configuración que componen la aplicación. Cada elemento debe tener una

descripción breve y una lista de sus relaciones y dependencias.

Identificación de Configuración:

1. Código: La base de la aplicación de software. Archivos en Python-Django para la lógica de la aplicación.
2. Interfaz del usuario: Diseños específicos para la pantalla de inicio, búsqueda y resultado de vuelos y alojamientos. Imágenes diferentes para cada sección detallada. Iconos en cada botón y elementos visuales en diferentes campos, como tarjetas de resultados.
3. Diseño gráfico: Imágenes en formato PNG para ilustrar cada destino, tanto en el inicio como en el resultado. Iconos de cada aerolínea característica también en formato PNG.
4. Pruebas: Pruebas unitarias que evalúan las unidades individuales del código como pruebas de integración para evaluar los diferentes componentes de la aplicación que funcionan en conjunto. Probar también si cumple con los requisitos y el rendimiento de la aplicación frente a diferentes condiciones.

3. Propuesta de Cambio: Plantea un escenario en el que se identifique la necesidad de

Un cambio en la aplicación, cómo agregar una nueva funcionalidad o corregir un

Error. Describe en detalle el cambio propuesto.

Agregar la opción para alquilar autos. Esta opción re direcciona automáticamente a la página de la empresa que se encargar de alquiler de autos.

4. Evaluación de Cambios: Como equipo, discutan el cambio propuesto y evalúen su

Impacto en la configuración existente. ¿Cómo afectará a otros elementos? ¿Es

Necesario realizar modificaciones en otros componentes?

Existe un cambio, ya que debemos modificar la pantalla de inicio y agregar la opción de “Alquiler de Autos”. Ya que es una nueva funcionalidad que se agregará. Agregar la logica para que al hacer click en la opción, re direccione a la correspondiente página.

5. Aprobación del Cambio: Simula un proceso de revisión y aprobación del cambio.

Designa a un "Controlador de Cambios" que evaluará y tomará una decisión sobre la

Aprobación del cambio.

El "Controlador de Cambios" evalúa el valor de la nueva función alquiler de autos y cómo beneficia a los usuarios.

Este cambio beneficia a los usuarios ya que les permite a los mismos tener la posibilidad de alquiler autos para su comodidad incluida en la página. También mantiene una lógica similar a los demás componentes del sistema donde mediante un ingreso de fechas se le muestra un listado porque facilita el desarrollo.

Aprobado.

6. Implementación del Cambio: Si el cambio es aprobado, realiza las modificaciones

Necesarias en los elementos de configuración afectados. Actualiza el código fuente,

La documentación y otros recursos según sea necesario.

Se realizan las modificaciones necesarias en el código fuente para la lógica de alquiler de autos. En la interfaz de usuario para agregar botones, se debe agregar en la página principal en el navbar la opción “Alquiler de autos” y también generar una página la cual muestra el resultado de los mismos.

7. Creación de una Nueva Versión: Una vez que se ha implementado el cambio, crea

Una nueva versión de la aplicación que incluya las modificaciones. Asigna un

Número de versión y documenta los cambios realizados en un registro de cambios.

La nueva versión se etiqueta como 1.2.0. Se documentan los cambios en un registro de cambios (Changelog.md). No es necesario que sea una versión 2.0.0 ya que no es un cambio masivo sino que es una nueva funcionalidad.

8. Control de versiones: Utiliza herramientas de control de versiones (como Git) para

Gestionar las diferentes versiones de la aplicación. Crea ramas, fusiona cambios y

Etiqueta versiones para simular el proceso de control de versiones.

Se utiliza Github. Se crea una rama de desarrollo, llamada Gianluca, para implementar la función de Alquiler de autos. Los cambios se fusionan en la rama principal y se etiqueta la versión 1.2.0 (Ya realizado en el repositorio ingenieria-software-2).

9. Auditoría de Configuración: Realiza una auditoría de la configuración para verificar

Que la nueva versión cumple con las políticas y líneas base establecidas. Identifica

Posibles desviaciones y asegúrate de que los cambios se hayan implementado

Correctamente.

Se verifica que la nueva versión cumple con las políticas. Se asegura que los cambios se implementaron correctamente en todos los elementos de configuración. Una posible desviación puede ser que el nuevo botón genero una sensación de “congestión visual”, mucho contenido, los usuarios pueden sentirse confundidos al encontrar múltiples botones en el inicio. Una acción propuesta para mejorarlo es explorar las opciones de rediseño de la interfaz de navegación o bien reorganizar los botones.

10. Presentación y Retroalimentación: Finaliza el ejercicio presentando la nueva versión

De la aplicación al equipo y discutiendo el proceso seguido. Solicita retroalimentación

Sobre la simulación y cómo se aplicaron los conceptos de Gestión de Configuración.

Se presenta la versión 1.2.0 al equipo. Se discute cómo se aplicaron los conceptos de Gestión de Configuración y se recopila retroalimentación de usuarios finales para futuras mejoras en el proceso. Se desarrolló inicialmente en la rama Gianluca y al estar testeado y ya funcionando se realizó un merge a la rama main para que esté disponible para los usuarios finales.

**Crear una carpeta que se llame:**

**ASSETS,** buscar fotos en google y ponerlas en la carpeta (Imágenes)

**INTERFAZ,** Carpeta con imágenes

**CODE,** código en python y escribirlo (“Hello World”)

**CHANGELOG M.D**, Aca se explican los cambios