

Manual de Mantenimiento

Juan Carlos Gómez Hernández¹

Universidad Linda Vista, Ex-Finca Santa Cruz No. 1, 29750, México correo
1@ulv.edu.mx

Este documento está dirigido al personal de mantenimiento y proporciona detalles sobre cómo realizar tareas de mantenimiento y actualizaciones del Simulador de Equilibrio. A continuación se presentan las secciones clave del manual.

1. Procedimientos de Mantenimiento

1.1. Revisión Periódica del Software.

- Realizar revisiones mensuales para asegurarse de que el simulador funcione correctamente.
- Verificar que todas las imágenes y recursos gráficos estén presentes y accesibles en las rutas especificadas.

1.2. Actualización de Bibliotecas

- Mantener actualizadas las bibliotecas de Python necesarias para el funcionamiento del simulador. Utilizar los siguientes comandos en la terminal:

```
pip install --upgrade pygame matplotlib canvas math
```

1.3. Monitoreo de Errores

- Registrar cualquier error o fallo que ocurra durante el uso del simulador.
- Revisar los registros de errores para identificar patrones o problemas recurrentes.

1.4. Pruebas de Funcionalidad

- Después de realizar actualizaciones, ejecutar pruebas para verificar que todas las funciones del simulador operen como se espera.
- Incluir pruebas específicas para cada funcionalidad principal, como la simulación, visualización de gráficos y ajustes de parámetros.

2. Plan de Actualizaciones

2.1. Frecuencia de Actualizaciones

- Realizar actualizaciones mayores cada seis meses, o según sea necesario si se identifican errores críticos.
- Implementar actualizaciones menores cuando se realicen mejoras o correcciones.

2.2. Registro de Cambios

- Mantener un registro detallado de las actualizaciones realizadas, incluyendo fechas, descripciones de cambios y responsables.
- Documentar cualquier nueva funcionalidad añadida o mejoras en la interfaz.

2.3. Pruebas Post-Actualización

- Después de cada actualización, llevar a cabo pruebas para asegurar que el simulador funcione correctamente y que no haya introducido nuevos errores.

3. Manejo de Errores

3.1. Identificación de Errores

- Monitorear el rendimiento del simulador y registrar cualquier error que ocurra durante su uso.
- Utilizar mensajes de error claros que ayuden a identificar la causa del problema.

3.2. Solución de Problemas Comunes

- Consultar la sección "Solución de Problemas Comunes" en el manual del usuario para abordar problemas frecuentes.
- Si un problema persiste, consultar la documentación técnica o buscar asistencia en foros especializados.

3.3. Restauración a Versiones Anteriores

- Mantener copias de seguridad del software antes de realizar actualizaciones importantes.
- En caso de un error crítico tras una actualización, restaurar la versión anterior utilizando las copias de seguridad.
- Usar Git como controlador de versiones.

4. Copias de Seguridad y Recuperación

4.1. Frecuencia de Copias de Seguridad

- Realizar copias de seguridad completas del simulador y sus recursos al menos una vez al mes.
- Asegurarse de incluir todos los archivos relevantes, incluidos gráficos, configuraciones y scripts.

4.2. Método de Copia

- Utilizar GitHub como herramientas automatizada para controlar las copias de seguridad regulares.
- Verificar periódicamente que las copias sean accesibles y estén completas.
- Usa el proceso abitual para subir tus cambios remotos a la nube de git (add, commit, pull, push)

5. Consideraciones Especiales Basadas en el Código

5.1. Manejo de Recursos Gráficos

- Asegurarse de que todas las imágenes (como Giro.png, Peso.png, etc.) estén disponibles en las rutas especificadas dentro del código. Si se cambian las ubicaciones, actualizar el código correspondiente.

5.2. Configuraciones Iniciales

- Verificar que las configuraciones iniciales (como dimensiones y colores) sean adecuadas y se mantengan consistentes con el diseño original del simulador.

5.3. Funciones Matemáticas

- Revisar periódicamente las funciones matemáticas (como calculateTensions y Solution) para garantizar que los cálculos sean precisos y eficientes.

5.4. Gestión del Ciclo Principal

- Asegurarse de que el ciclo principal (while running:) funcione correctamente sin bloqueos ni errores inesperados durante la ejecución.
- Monitorear eventos como clics del mouse y movimientos para garantizar que se manejen adecuadamente sin causar fallos en el programa.

5.5. Gestión Visual y Gráfica

- Comprobar que todas las funciones gráficas (como drawScene y crearGrafico) generen visualizaciones correctas sin errores visuales ni problemas en la representación gráfica.