Universidad de Guadalajara.

Centro Universitario de Ciencias Exactas e Ingenierías.

Ejercicio 3.3

Restaurar el estado de ejecución

Computación tolerante a fallas NRC 179961

Nombre del alumno: Loredo Padilla Orlando Javier

Código: 217560328

Departamento: Departamento de ciencias computacionales

Nombre del maestro: López Franco Michel Emanuel

Carrera: Ingeniería en computación

Sección: D06

Ciclo: 23B

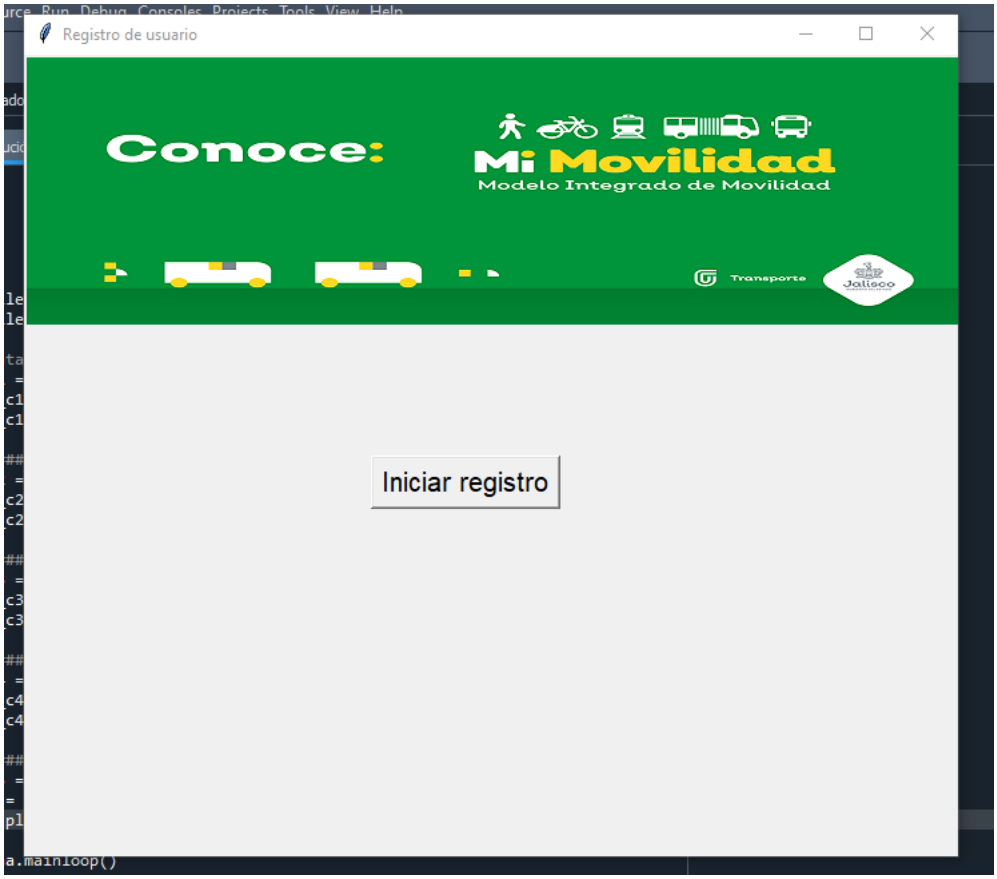
Fecha: 4/9/2023.

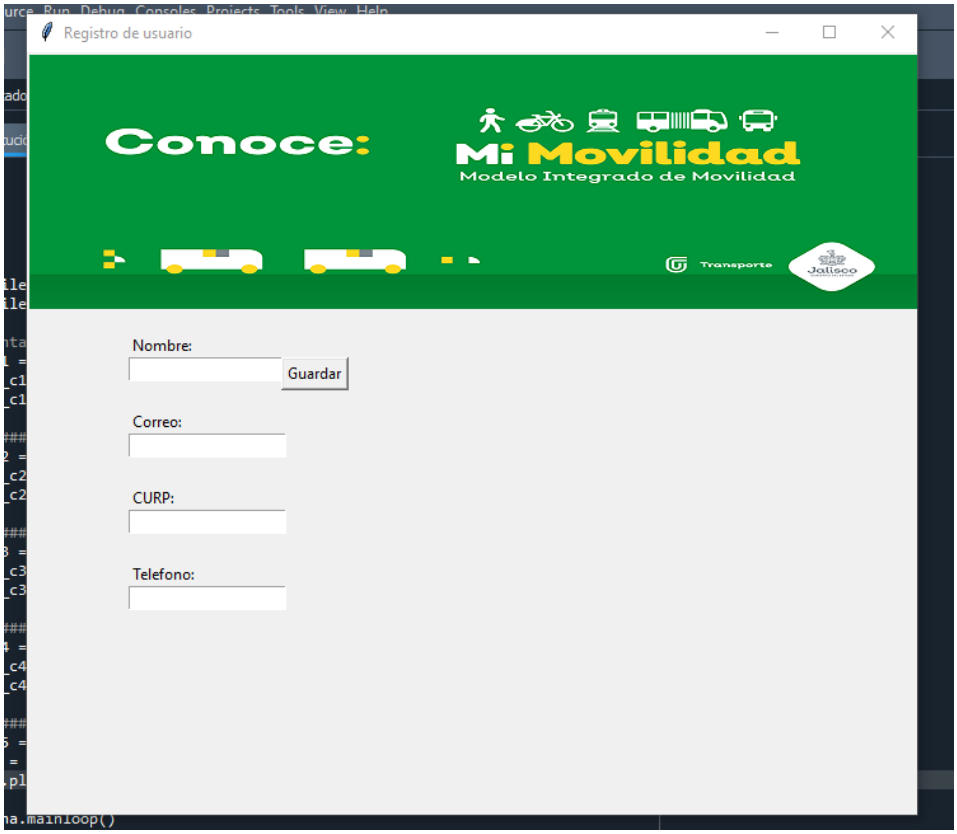
Introducción

Para esta práctica tendremos que crear un sistema para un programa en el cual si ocurre un error este al volver ser ejecutado se inicie en punto de guardado para evitar perder el progreso.

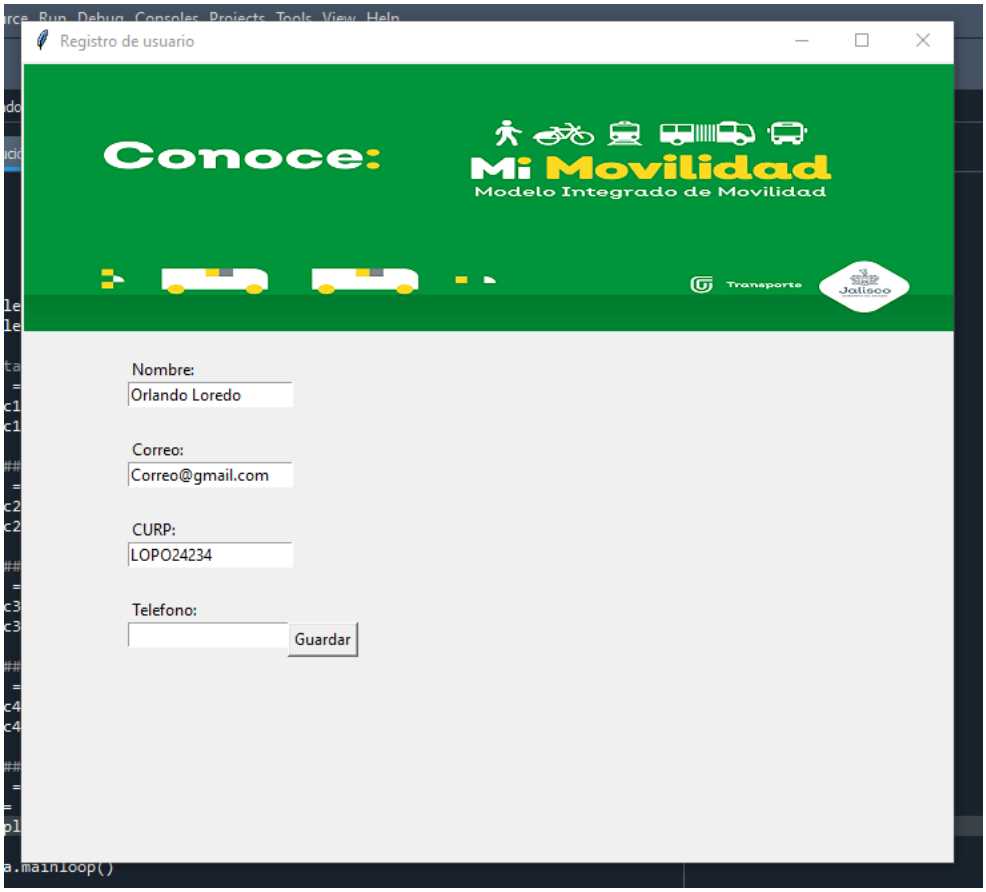
Programa

Funciona mediante una interfaz gráfica y simulará una página de registro. Al iniciar se nos presentará un botón para iniciar a ingresar nuestros datos, este lo usaremos para primero revisar si se encuentran datos guardados o no. En caso de no encontrar nada en el respaldo los campos estarán vacíos.

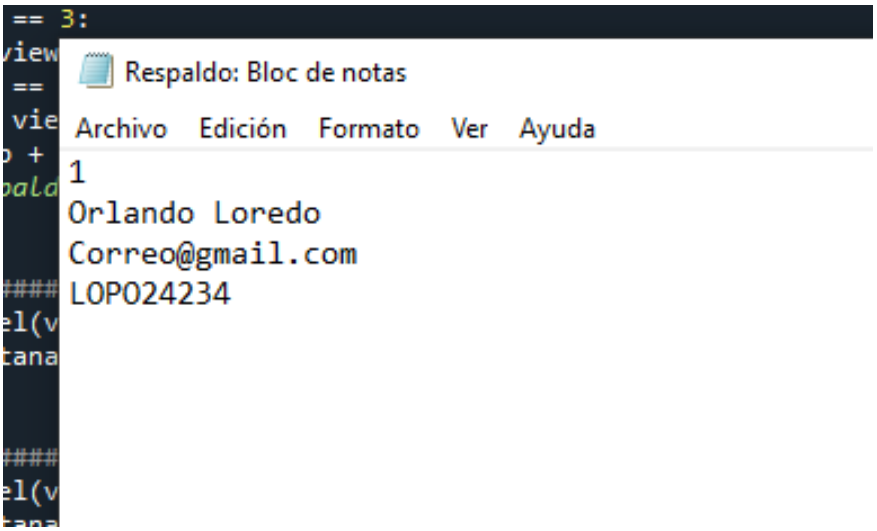




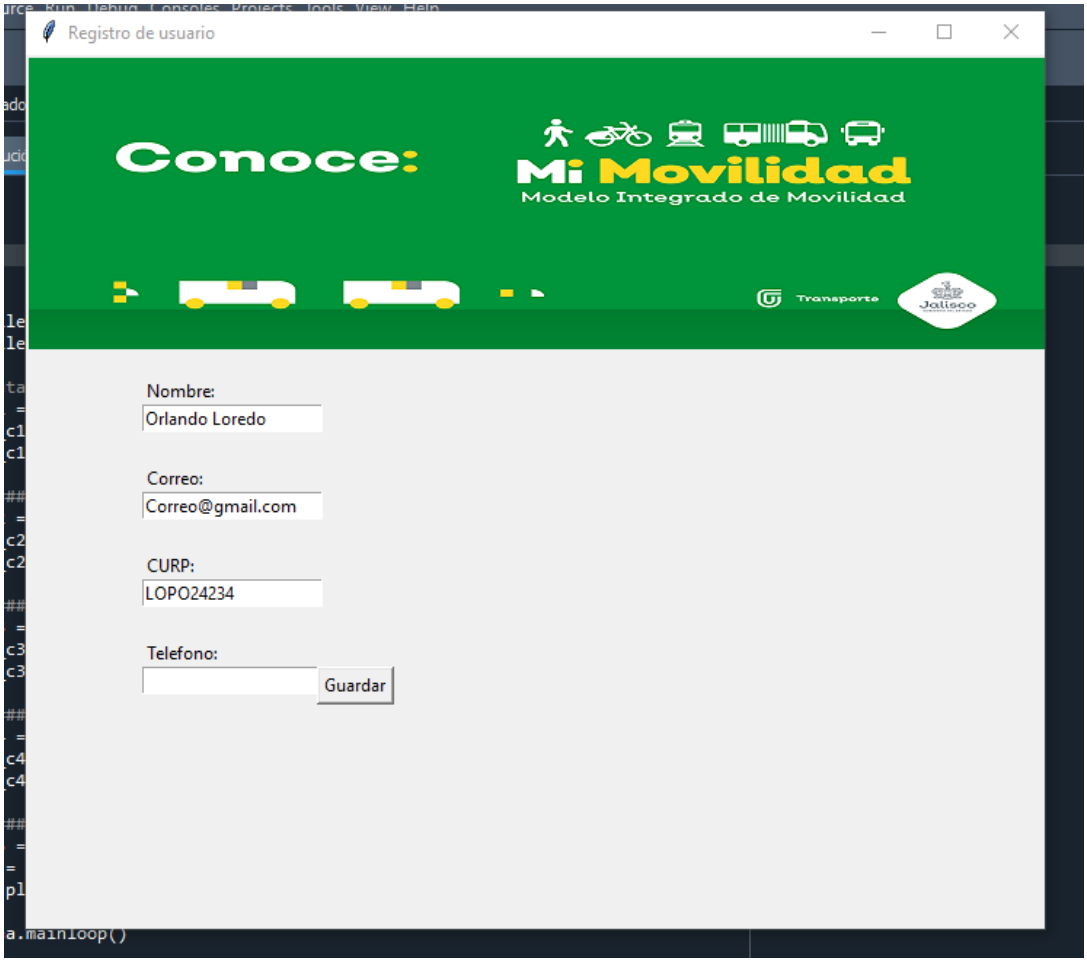
Si están vacíos iniciaremos desde el primer campo, para guardar la información presionaremos el botón para pasar al siguiente campo, cada vez que presionemos el botón estaremos respaldando la información.



La información se guardará en un .txt.



De esta manera si cerramos el programa, o se cierra por accidente que es para lo que está diseñado, al volverlo a ejecutar verificará la información que se tenía hasta ese momento y la cargará para avanzar desde el punto donde nos quedamos.



Conclusión

El programa fue un poco complicado de hacer, esto debido al sistema de restauración de ejecución ya que se tuvo que planear como funcionaría y como lo haría, si se respaldaba cada cierto tiempo esto debería de tomar en cuanta cada uno de los en como se estaban respondiendo, y al momento de respaldar si no se seguía el orden correcto la información se recuperaría mal. Para resolver esto decidí ir respaldando cada acción confirmada, de esta manera se guardarían los pasos ya hechos y al momento de ejecutar el programa se revisarían los que ya habían sido ejecutados para volver a cargarlos.

El trabajo es muy simple y lo consideraría como poco eficiente por eso me gustaría a futuro poder realizar una actualización o poder mejorar este sistema para futuros trabajos donde con una mejor planeación el sistema fuera mucho más optimo.

Referencias

- Herrera G. (2005). Clasificación y Análisis Post-Proceso de Defectos en Proyectos de Desarrollo de Software. Recuperado el 1/9/2023 en: https://cimat.repositorioinstitucional.mx/jspui/bitstream/1008/104/2/TE%20264.pdf
- Software Quality Exp (2023). What is Orthogonal Defect Classification (ODC)? en Medium. Recuperado el 1/9/2023 en: https://medium.com/@SWQuality3/what-is-orthogonal-defect-classification-odc-by-vivek-vasudeva-f2e49917f478
- S/A (S/F). Métodos y técnicas de prevención de defectos en Myservername.com. Recuperado el 1/9/2023 en: https://spa.myservername.com/defect-severity-priority-testing-with-examples
- Ventura M. (S/F). Costo de la Calidad en el Desarrollo de Software. Recuperado el 1/9/2023 en: http://www.paginaspersonales.unam.mx/files/69/Publica_20110622215614.pdf