TEC - Instituto Tecnológico de Costa Rica – Escuela de Ingeniería en Computación

Carrera: Ingeniería en Computación

Curso: Taller de Programación

Estudiantes: Esteban Azofeifa y Camilo Allón

Profesor: William Mata Rodríguez

Programa III (30%): PROYECTO: REVISIÓN TÉCNICA DE VEHÍCULOS (ReTeVe)

Fecha de entrega: 19 de junio 2023, 7:00 am

IC 1803 TALLER DE PROGRAMACIÓN

Escuela de Ingeniería en Computación, Tecnológico de Costa Rica

Contenido

Enunciado del proyecto:

Temas investigados

Software de control de versiones

¿De qué trata?

Este es un software encarga que funciona como un archivo de juegos. Ayuda a guardar el proceso de un programa de una manera más segura y menos riesgosa. Mientras que, en una computadora, se está peligrando siempre a que el archivo se corrompa, un software de este estilo no es así. También, en caso de que ocurra un error y el programa ya no funcione, es fácil poder acceder a una versión más antigua del trabajo y continuar de ahí. Además, gracias a estos softwares, se puede trabajar en equipo con otros programadores en el mismo archivo, pero cada uno con su versión. Es decir que cada miembro de un equipo puede tener diferentes versiones de un solo programa, las cuales se pueden unir en uno solo.

¿Cómo se usa?

Para guardar un trabajo en cualquier software de control de versiones, es necesario usar un repositorio. Este es simplemente un lugar donde se guarda el programa y el resto de archivos y módulos con los que funciona, módulos en este caso refiriéndose a trabajos creados por el usuario para usarse dentro de otro programa. No es necesario acceder desde el software para programar, aunque eso se puede hacer. Lo que se hace en vez de eso es acceder de forma remota al repertorio creado desde donde se trabaje. Todos los softwares de almacenamiento pueden funcionar de diferentes maneras, sin embargo, todos se pueden basar en operaciones donde se hace un cambio remoto, otro donde se mete ese cambio al repertorio remoto y, por último, otro comando para actualizar el repertorio remoto al repertorio en el software. Por ejemplo, para Git, estos serían los comandos: add, commit y push. Existen muchos otros comandos, peor esos, más init, el cual sirve para iniciar el repertorio remoto, son los más importantes.

En resumen

¿Qué es?

Un software de control de versiones se encarga de mantener a salvo diferentes versiones de uno o varios programas y así poder accesar a estas en cualquier momento. Se puede imaginar como un archivo de guardado de juego y cargar partida.

Importancia

Ayuda al trabajo en equipo, a unir esos trabajos en uno solo, ayuda a comunicación entre los integrantes de un grupo, esto gracias a funcionalidades de comentarios a compañeros por parte del líder. Y por supuesto, ayuda a siempre tener una copia del programa a la mano, sana y salva.

Algunos softwares:

1- Git: uno de los más comunes y más sencillas, a pesar que para personas nuevas para programación sea necesario pasar un rato aprendiendolos comandos para usarlo.
2- SVN
3- Mercurial
4- CVS
5- Monotone

Características:

- Puede trabajar con ramas o branches, estas se pueden unir si se les pide. Las ramas son como líneas temporales diferentes, en algunas se hizo algo diferente a lo que se hizo en otras. Esto es especialmente útil cuando varias personas intentan algo diferente para la misma opción, se pueden combinar las ideas o se puede elegir la opción más eficaz.
- Está lleno de comunidades de personas que suben sus programas para que otras usen como base para más proyectos.
- Se usa para controlar versiones de un programa y poder volver a ellas.
- Ayuda al trabajo en equipo en el mismo programa y ayuda con la comunicación.
- Cada software de control de versiones usa su propia lista de comando. Las acciones principales que realizan estos son: iniciar el repertorio, acceder a un repertorio

remoto, añadir cambios dentro de ese repertorio remoto y hacerlos y, por último, poner los cambios dentro del repertorio en el software.

Algoritmos para implementar ABB:

¿Qué es eso?

Un ABB o árbol binario es una estructura de datos usada para organizar información más eficientemente que en una lista, tupla o diccionario. Cada dato de esta estructura se le conoce como un nodo, el primero de todos, se conoce como raíz. Cada uno de los nodos puede tener dos nodos más es tos se les llamará sus hijos, el hijo izquierdo siempre será menor que su raíz y el hijo derecho será mayor. Esto se puede notar más fácilmente cuando se debe guardar una mayor cantidad de datos. En el caso de Python, las estructuras ABB no existen a diferencia de las listas o tuplas que si tienen hasta un código dentro del lenguaje. Por lo tanto, hay que trabajarlo por medio de clases y usar programación orientada a objetos.

¿Cómo se usa?

Para recorrer un árbol binario es necesaria la recursión de pila, esta es la única manera de moverse dentro de esta estructura. Lo esencial para lograrlo es siempre estar comparando un nodo o un dato dentro del nodo que sea sepa que se puede comparar con los demás, para moverse entre hijos izquierdo y derechos según sea el caso. Lo bueno de los nodos es que pueden guardar casi que cualquier otra estructura diferente, ya sean listas, tuplas, strings, diccionarios, etc. Así se pueden agregar, buscar y cambiar datos dentro del ABB y sus nodos.

Nuevos componentes de GUI:

Textbox

¿Qué es?

Es similar a un entry box mucho más grande para guardar mayores cantidades de texto. Es posible ajustar el texto en diferentes espacios, darle fuente y tamaño y vincularlo a acciones cómo el que cuando se presione un botón su texto cambie.

¿Cómo se usa?

Muy similar a un entry, se puede posicionar con place, pack o grid, se pueden cambiar fuente de letra y su tamaño, se puede obtener lo que está dentro gracias a textvariable, se puede justificar un texto hacia la izquierda derecha o que se ponga centrado, etc.

Checkbox

¿Qué es eso?

Un widget usado para marcar varias opciones al mismo tiempo, a diferencia del radiobutton el cual solo marca una a la vez. Se usa para definir si una o varias condiciones o circunstancias se cumplen a la vez.

¿Cómo se usa?

Similar a un radiobutton, se puede poner texto, a eso s ele puede poner fuente y tamaño, se pueden posicionar con pack, grid o place. Se les puede asignar diferentes valores si están seleccionados o no.

Scrollbar

¿Qué es eso?

Un widget usado para subir y bañar en una ventana. Ya que las pantallas solo presentan un espacio limitado para ver información, no toda va a caber, por eso es que los scrollbars son necesarios, ayudan a ver información que de lo contrario no sería visible. Se puede usar vertical u horizontalmente.

¿Cómo se usa?

Se debe poner en un frame para servir, si se intenta poner directamente en la ventana, no es posible. Eso no es suficiente, el scrollbar ayuda a mover otro tipo de widgets como listboxes o canvas, por lo tantos en necesario ligarlas a esos widgets para hacer que el scrollbar funcione. Se puede poner vertical u horizontalmente. Solo se puede usar pack u grid para colocarlo, al estar en un frame no se puede usar place.

Listbox

¿Qué es?

Un widget capaz de ostrar una lista de opciones ya desplegada sin necesidad presionar ningún botón. Simplemente a la hora de darle click la opción se selecciona.

¿Cómo se usa?

Si se tiene varios listboxes es necesario que ponerles en su código una variable para que no se deseleccionen al momento de seleccionar el otro. Se puede usar un scrollbar para ver sus opciones. Se puede posicionar con grid, pack o place.

Combobox

¿Qué es?

Un widget que es una mezcla entre un cascade y un entry box. Permite escribir en él y despliega una lista de opciones para seleccionar sui se presiona una flecha en su parte derecha. Ya trae un scrollbar implementado en caso que se pase del límite de longitud de la cascade.

¿Cómo se usa?

Se puede posicionar con pack, grid o place. Se puede poner de diferentes colores, se puede usar para que se elija de un grupo de opciones específicas, le puede dar longitud, etc.

Diseño de la solución

Entre las estructuras de datos más importantes está el árbol binario, se usa para guardar la información en todas las citas, sus nodos y hojas. Ayuda a buscar información de una forma más rápida que con una lista u diccionario. Esta organizado por el número de fecha, las citas se posicionan en el árbol basándose en el número de fecha. Si la fecha va después que la raíz, se sitúa a la derecha, de lo contrario a la izquierda. Esta estructura es la base de toda la estructura y que la información se preserve.

El tablero también es una estructura de datos que fue muy importante, es una lista en la que se guardan todas las configuraciones de los botones. El tablero de revisión está compuesto

por botones, los cuales con cada comando cambian su texto dependiendo de cuál placa se esté moviendo.

También se usa un diccionario para guardar la información de cada una de las fallas, posiciones y estados de las placas que estén en proceso de revisión. Está organizada por las posiciones en las que se encuentra cada placa en el tablero.

Archivos

El archivo configuración_riteve.dat guarda la información de la configuración en una lista la cual en todos los programas se extrae para ajusta todo.

El archivo.dat llamado Lista_Fallas se usa para guardar las fallas que se registren en el programa y usarse en la parte del tablero. Contiene un diccionario de fallas con sus códigos, descripciones y si son fallas graves o leves.

El archivo.dat números_citas se usa para guardar el contenido del árbol y el número de cita, de esta forma si se cierra el programa se puede volver a abrir con los mismos datos e información de las citas pasadas. Guarda una lista con dos elementos una lista vacía, donde se guardan los contenidos del árbol y el número de cita por el que se va.

Por último, se tiene el archivo.dat registro_de_citas este al final quedó fuera de uso, pero originalmente era para guardar el árbol conforma fuera alterándose. Esto lo haría guardando el árbol como tal. Sin embargo, se teminó usando el .dat anterior.

Conclusiones

Problemas encontrados:

Hacer que, si se cierra el tablero por accidente, la información se quede ahí y el cliente no tenga que volver a meter las placas.

Solución: Crear una copia del tablero con el que guarde la información y se hagan los cambios.

Hacer que cuando se elija la hora de inicio del horario de la estación que no se pueda elegir una antes.

Solución: Para esto se usaron combo boxes, al llenar la primera la segunda despliega solo las horas posteriores a la anterior para seleccionar.

Hacer que el comando F saque todo y envíe un correo.

Para lo del correo se hizo una función que adjunta documentos, para la emisión de resultados y se creó otra que crea los pdfs necesarios. Además se hizo que eliminara el texto del botón en el que se encontrara.

Crear un árbol binario funcional.

Solución: Uso de la recursión de pila por medio de la comparación del número de cita para encontrar citas específicas y comparación de la fecha para organizarlas de una manera que tenga sentido.

Aprendizajes obtenidos

Mejores estrategias para tkinter:

Aprendimos a usar frames de una manera más eficiente y sencilla. El uso de frames más pequeños ayuda a situar widgets donde uno quiere mucho más fácil que con un simple pack. Además, usar side =x y anchor = Y en pack ayuda a orientarse.

Uso de árboles binarios:

Se aprendió a usar los árboles binarios de manera eficiente y crearlos usando recursión depila. Además, buscar cosas dentro de estas estructuras se facilita si se compara un dato con otro ya preexistente dentro. Ejemplo el número de cita es único para cada cita, por lo que, si se compara, es posible llegar a lo que se está buscando.

Uso de Github: Se aprendieron los comandos para a usarlo y la utilidad que se les puede dar a la hora de trabajar en grupo. Esto gracias a los repertorios remotos y los comandos usados para implementarlos en el repertorio de Github.

Uso de datetime: Se aprendió a configurar fechas, las actuales y otras futuras o pasadas. Además, se aprendió a ponerles formato.

Estadísticas de tiempos

Actividad realizada	**Horas**
Análisis del problema	10 horas
Diseño de algoritmos	10 horas
Investigación software de versiones	3 horas
Investigación sobre ABB	4 horas
Investigación sobre elementos adicionales de GUI	3 horas
Investigación de uso de diferentes módulos	3 horas
Programación	35 horas
Documentación interna	4 horas
Pruebas	8 horas
Elaboración del manual de usuario	2 hora
Elaboración de documentación del proyecto	4 hora
TOTAL	83 horas

Lista de revisión del proyecto:

LISTA DE REVISIÓN DEL PROYECTO Concepto	**Avance 100/%/0**
Programación de citas implementando ABB con recursión	100%
Asignación manual de citas	100%
Asignación automática de citas	100%
Cancelar citas	100%
Ingreso de vehículos a la estación	100%
Despliegue del tablero de revisión	100%
Ejecución de los comandos del tablero	100%
Registro de fallas por cita de revisión	100%
Resultado de la revisión en PDF	100%
Certificado de tránsito en PDF	100%
Envío de correos electrónicos (cita, resultado de la revisión)	100%
Lista de fallas (CRUD)	100%
Configuración del sistema	100%
Ayuda: manual de usuario en pantalla	100%
TOTAL	**100%**

Bibliografía

Morcuende, S. (21 de octubre del 2022). 5 sistemas de control de versiones imprescindibles en los proyectos de desarrollo. *Between.* https://impulsate.between.tech/5-sistemas-control-versiones