

Instituto Tecnológico de Costa Rica

Escuela de Computación, Ingeniería en Computación



Curso IC-1803

Taller de Programación

Grupo 03

Proyecto Programado #2

"ReviTec"

Elaborado por:

Katerine Guzmán Flores

2019390523

Profesor del curso:

William Mata Rodriguez

Semestre II

2019

Contenido

Enunciado del proyecto	4
Objetivos del proyecto	4
Definición del proyecto: Revisión técnica de vehículos (ReviTec)	5
Funcionalidades del Programa	6
A) Programar citas	6
B) Cancelar citas	8
C) Ingreso de vehículos a la estación	9
D) Tablero de revisión	9
E) Resultado de la revisión.	12
F) Lista de fallas	12
G) Ayuda	12
H) Acerca de	12
I) Salir	13
Temas Investigados	14
Módulo Datetime	14
Reportlab	14
Encode():	14
Decode():	14
Withdraw	14
Módulo smtpplib	15
Módulo os	15
Mainloop	15
Listbox	15
Diseño y explicación de la solución	15
Estructuras de Datos	15
Archivos	16
Conclusiones del Trabajo	17
Problemas Encontrados y Soluciones	17
Aprendizajes Obtenidos	17
Estadística de tiempos	18

Enunciado del proyecto

Objetivos del proyecto

- Reforzar los conocimientos acerca de la metodología de desarrollo de programas de mayor escala:
 - Entender el problema.
 - Diseñar la solución: dividir el problema en problemas más pequeños.
 - Codificar la solución.
 - Probar y evaluar el programa.
- Aplicar y reforzar aspectos del lenguaje Python 3
 - Uso de estructuras condicionales y de repetición de procesos.
 - Desarrollo de funciones.
 - Utilización de archivos.
 - Utilización de estructuras de datos nativas de Python tales como las secuencias y diccionarios.
 - Utilización de estructuras de datos creadas por el programador como las colas y matrices.
- Usar nuevas características de interfaces gráficas de usuario (GUI).
- Aplicar buenas prácticas de programación: documentación interna y externa del programa, reutilización de código, nombres significativos, eficiencia del programa, evaluar alternativas, uso de técnica divide y vencerás (dividir el problema en partes, desarrollar cada una de esas partes), etc.
- Validación de los datos de entrada: todos los datos de entrada se deben validar según restricciones que se indican en cada uno de ellos.
- Fomentar en el estudiante la investigación: temas no tratados en el curso pero necesarios para hacer el proyecto. Dichos temas deben ser explicados detalladamente en la documentación del proyecto.

Definición del proyecto: Revisión técnica de vehículos (ReviTec)

Se desarrollará un programa para administrar el proceso de revisión que se hace en una estación especializada. La revisión consiste en examinar cada vehículo anualmente para emitir un certificado que indica si el vehículo es apto para circular. De no ser apto para circular el certificado no se emite y se dan 15 días para hacer una reinspección. Hay un máximo de tres reinspecciones, sino pasa en alguna de ellas, el vehículo debe ser sacado permanentemente de circulación. El programa tendrá funciones para el registro de las citas de revisión de cada vehículo, un tablero que muestra en qué parte de la revisión está el vehículo una vez que ingrese a una línea de trabajo y ver los resultados de la revisión.

En esta estación hay seis líneas de trabajo: una línea de trabajo es una cola (primero que entra, primero que sale) donde están los vehículos que se están revisando.

En cada línea de trabajo hay cinco puestos de revisión:

- **Puesto 1:** revisiones generales (luces, llantas, vidrios, etc.).
- **Puesto 2:** banco de amortiguadores.
- **Puesto 3:** sistema de frenado.
- **Puesto 4:** detector de holguras.
- **Puesto 5:** emisión de contaminantes.

En cada puesto el vehículo puede salir con estas condiciones:

- Sin fallas.
- Con fallas leves.
- Con fallas graves.

Si al final de la línea de trabajo el vehículo presenta al menos una falla grave, debe volver a reinspección en otra cita cuando la falla sea arreglada.

Antes del puesto uno en cada línea de trabajo están los vehículos en una “cola de espera” para ingresar a la revisión. Cuando entran a la revisión (a partir del puesto uno) pasan de la “cola de espera” a la “cola de revisión”. Después del puesto cinco se da el resultado de la revisión. En cada puesto de revisión de cada línea de trabajo puede haber sólo un vehículo.

El programa tendrá un menú principal desde el cual se accederán las funciones del programa las veces que sea necesario. Cada opción debe ofrecer la posibilidad de regresar al menú anterior preservando la información registrada. Usted tiene completa libertad para diseñar la interfaz gráfica siempre y cuando cumpla con los requerimientos de información (funcionalidades del programa) que se indican seguidamente.

Funcionalidades del Programa

A) Programar citas

La estación trabaja de las 6 am hasta las 10 pm y las citas se programan cada 20 minutos en cada línea de trabajo. Es decir, a las 6 am se puede programar una cita, a las 6:20, 6:40, 7:00, etc., por cada línea. Ya que hay 6 líneas de trabajo se pueden programar un máximo de 6 citas a la misma hora. Con esta opción se van a registrar las citas con los siguientes datos:

- Número de cita (entero > 0) que es único, no puede existir números de cita duplicados.
- Tipo de cita (código de 1 dígito): 0-primera vez, 1-primera reinspección, 2-segunda reinspección, 3-tercera reinspección. Este control lo va a llevar el usuario.

- Número de placa (string de 6 caracteres que puede tener dos formatos: el primer formato consiste de números entre 1 y 999999, el segundo formato consiste exactamente de 3 letras mayúsculas seguidas por 3 dígitos, ejemplo: AXB015).
- Marca del vehículo (string de 15 caracteres).
- Modelo (string de 15 caracteres).
- Propietario (string de 40 caracteres).
- Teléfono.
- Correo electrónico (string).
- Dirección física (string de 40 caracteres).
- Fecha y hora de la cita: a solicitud del usuario puede ser manual o automática. Manual es cuando el usuario puede pedir una fecha y hora específica para su cita, el programa debe validar que pueda ser aceptada. Automática es cuando el programa sugiere la siguiente cita disponible considerando la fecha y hora actual del sistema:
 - Fecha de la cita (un solo número de 8 dígitos –ddmmaaaa- representando una fecha válida).
 - Hora de la cita (un solo número de 4 dígitos –hhmm- representando una hora válida en el formato de 24 horas). Ejemplo: 1520 (3 y 20 de la tarde).
- Estado de la revisión: “PENDIENTE” cuando se saca la cita. Otros valores se le asignan cuando se cancela la cita y cuando termina la revisión, ver más adelante cuando termina el último puesto de revisión.

Cuando se da una cita se envía un correo electrónico al propietario con esos datos. Los datos de las citas se deben mantener en un archivo para futuras referencias. Puede usar un diccionario para llevar la información de citas.

Validaciones en esta opción:

- Cada dato debe ser del tipo, tamaño y características especificadas, de lo contrario el programa enviará un mensaje de error para ese dato. El error se mantendrá en pantalla hasta que el usuario le indique al programa que continúe para que pueda dar el dato correctamente.
- Para los strings se especifica un tamaño máximo. El mínimo debe ser al menos 1 carácter, excepto que se diga lo contrario.
- Solo pueden darse citas según lo descrito anteriormente: 6 citas máximo a una misma hora, cada 20 minutos (a las 00, 20 y 40 minutos).
- No pueden haber varias citas en un mismo día para un mismo número de placa, excepto que sea una reinspección.
- Las citas son de la fecha y hora del día actual en adelante, es decir, si hoy es viernes 4 de octubre y son las 10 de la mañana, la cita debe ser posterior a ello.

B) Cancelar citas

Esta opción va a servir para que un usuario cancele su cita. Debe dar el número de la cita y el número de placa. Actualiza en los datos de la cita el estado de la revisión a "CITA CANCELADA".

Validaciones en esta opción:

- Cada dato debe ser del tipo, tamaño y características especificadas, de lo contrario el programa enviará un mensaje de error para ese dato. El error se mantendrá en pantalla hasta que el usuario le indique al programa que continúe para que pueda dar el dato correctamente.
- El número de cita debe corresponder con el número de placa.
- Se pueden cancelar solo citas existentes de lo contrario se genera un error.
- Si la placa está en un puesto de revisión no se puede cancelar la cita.

- Antes de cancelar una cita debe pedir una confirmación de que efectivamente quiere hacer la cancelación.

C) Ingreso de vehículos a la estación

Esta opción se usará cuando el usuario se presente a su cita en la estación.

Requiere los siguientes datos:

- Número de cita.
- Placa.

Validaciones en esta opción:

- Cada dato debe ser del tipo, tamaño y características especificadas, de lo contrario el programa enviará un mensaje de error para ese dato. El error se mantendrá en pantalla hasta que el usuario le indique al programa que continúe para que pueda dar el dato correctamente.
- Verificar que la cita sea para el día que se está trabajando (fecha del sistema) y corresponda con la placa.
- Verificar que el vehículo no esté en alguna cola.
- Cobro por la revisión o inspección (valor entre 12000 y 50000)

Si todo está correcto el sistema lo envía automáticamente a la “cola de espera” de la línea de trabajo con menos vehículos. Si todas están iguales lo envía a cualquiera.

D) Tablero de revisión

Debe desplegar los vehículos (placas) que se encuentran en los diferentes puestos de cada línea de trabajo. La primera vez que se entra a esta opción al iniciar el día el tablero va a estar vacío, luego refleja la información actualizada en tiempo real.

Fecha: dd/mm/aaaa. (día tomado del sistema)

Línea	Puesto 1	Puesto 2	Puesto 3	Puesto 4	Puesto 5
1	ABC001	QWE121	JHF101	OWA201	PLN501
2	BAC501	PLN521	OJB504	JBN562	
3					
4					
5					
6					

COMANDO: _____

El tablero tendrá una línea de comandos que servirá para ir moviendo los vehículos en cada puesto de cada línea de trabajo. Cada comando es un string que puede tener cualquiera de los siguientes formatos:

COMANDO SIGNIFICADO

LA: El vehículo en la “cola de espera” de la línea número L debe avanzar al puesto número 1. La “cola de espera” y la “cola de revisión” de esta línea se actualizan. Ejemplos del comando: 1A, 2A, etc.

TXXXXXX El vehículo con la placa XXXXXX (hay que validar que el vehículo esté en algún puesto) debe avanzar al siguiente puesto y también todos los vehículos en puestos anteriores avanzan un puesto. Se actualizan la “cola de espera” y la “cola de revisión” de la línea. Ejemplos: TPLN501, TJBN562, TXYZ001, T5200, etc.

UXXXXXX El vehículo con la placa XXXXXX (hay que validar que el vehículo esté en algún puesto) debe avanzar al siguiente puesto siempre y cuando esté libre. Avanza únicamente este vehículo. No altera la “cola de espera” y los otros puestos permanecen sin cambios. Con este cambio solo se actualiza la “cola de revisión”.

CXXXXXX Cancelación. El vehículo con la placa XXXXXX no va a continuar en la revisión. Se actualiza la “cola de revisión”. Actualiza en los datos de la cita el estado de la revisión a “REVISIÓN CANCELADA” lo que significa que posteriormente debe sacar otra cita para una nueva revisión.

FXXXXXXYYYY Al vehículo con la placa XXXXXX hay que agregarle la falla con el código YYYY. En caso de que la falla sea grave, la placa se pondrá en rojo en el tablero a partir de ese momento. Las fallas presentadas para el vehículo se deben mantener registradas en un archivo para referencias posteriores.

M Regresar al menú anterior. La información que se tenga en el tablero debe ser preservada para continuar procesándola cuando se regrese a esta opción.

Validaciones en esta opción: Cada dato debe ser validado para que el flujo en el proceso pueda darse, de lo contrario el programa enviará un mensaje de error para ese dato. No podemos pasar a un puesto de trabajo una placa que no esté en la parte respectiva del proceso, ejemplos: para pasar de la “cola de espera” a la “cola de revisión” el vehículo tiene que estar en el primer lugar de esa “cola de espera”, para avanzar a un puesto el mismo debe estar libre. El error se mantendrá en pantalla hasta que el usuario le indique al programa que continúe para que pueda dar el dato correctamente

La información en el tablero de control se mantiene actualizada: cada movimiento que se haga en cualquier puesto debe ser reflejado en tiempo real (en el momento).

Cuando el vehículo sale del último puesto, el programa debe actualizar automáticamente el estado de la revisión para esta placa y cita: - Estado “APROBADA”

si no presenta fallas o todas son leves. - Estado “PARA REINSPECCIÓN” si al menos una falla es grave. - Estado “SACAR DE CIRCULACIÓN” si la cita es de tercera reinspección y la revisión presentó al menos una falla grave. - Estado “CANCELACIÓN” cuando se usó el comando de cancelación.

E) Resultado de la revisión.

Esta opción servirá para emitir el certificado de la revisión, ya sea aprobado o no. Se debe pedir el número de cita y placa para el cual se requiere el resultado. Todos los datos de la cita deben ir a un archivo tipo PDF para que pueda ser impresos.

Adicionalmente se debe dar la lista de fallas encontradas: código, descripción y tipo de falla. El resultado también se envía al correo electrónico.

F) Lista de fallas

Diccionario cuyos elementos tienen la siguiente estructura - Llave: número de falla (entero entre 1 y 9999) . - Valor: tupla con dos valores o descripción de la falla (string 60 caracteres). o tipo de falla (L: falla leve, G: falla grave). Esta parte del programa debe permitir que el usuario pueda actualizar el diccionario mediante operaciones de agregar, borrar y cambiar elementos. Mantener estos datos en un archivo ya que se van a usar cada vez que se usa el programa.

G) Ayuda

Esta opción desplegará el manual del usuario.

H) Acerca de

Esta opción la usaremos para desplegar información “Acerca del programa” donde pondremos al menos los datos del nombre del programa, la versión, la fecha de creación y el autor.

I) Salir

Esta opción se usa para salir del programa.

Temas Investigados

Módulo Datetime

Este módulo propiamente de Python proporciona clases para la manipulación de fechas y horas simples y complejas del sistema de la computadora. Utiliza el siguiente formato:

Formato

```
>>>ahora = datetime.datetime.now()
>>>ahora.hour
>>>ahora.minute
>>>ahora.year
>>>ahora.day
>>>ahora.month
```

Reportlab

Se utiliza para crear documentos PDF desde Python. Se trata de una librería muy extensa y con muchas funcionalidades, desde pequeños textos y figuras geométricas a grandes gráficos e ilustraciones, todo ello puede ser incluido dentro de un PDF.

Encode():

Sirve para codificar los datos en bits.

Decode():

Decodifica la información, que fue codificada anteriormente con la función encode, al texto original.

Withdraw

Esconde la ventana de la pantalla.

Módulo smtplib

El módulo smtplib define un objeto de sesión de cliente SMTP que se puede usar para enviar correo a cualquier máquina de Internet con un daemon de escucha SMTP o ESMTP.

Módulo os

Es un módulo que provee funciones para interactuar con el sistema operativo. Viene con el standard de módulos de python. Este módulo provee una forma portable de usar el operating system dependiendo de la funcionalidad.

Mainloop

Es un loop infinito usado para correr la aplicación. Espera para que ocurra un evento y procesa el evento hasta que la ventana se cierra.

Listbox

Las listbox en python se utilizan para desplegar una lista de elementos donde el usuario puede seleccionar estos.

Diseño y explicación de la solución

Solucion

Para la realización de este proyecto primero se realizó la interfaz porque de esta manera podía visualizar como quedaría mi programa final. Posteriormente se realizaron todas las validaciones necesarias que se solicita por cada ventana. Se dibujó una matriz de 5x6 donde se muestra el tablero de la revisión. Además por cada cita se

guarda esta información en archivos y se escriben y leen cada vez que se ocupe. Al final de realizar todas las validaciones y que el programa esté funcionando correctamente se crea el pdf y se envía al correo que este caso se puso el correo del estudiante para que se pueda visualizar esta función.

Archivos

Para el desarrollo de este programa se hace un uso intensivo de la funcionalidad de los archivos, ya que este permite que se escriba la información de las citas, se lea y se cierre. Dentro de este programa la información de cada cita se guarda en un diccionario junto con otra información relevante de la persona, asimismo como información del vehículo. Además en la solución de este proyecto se utiliza un archivo también para guardar la información de las fallas. En la ventana de fallas, la diferencia es que el usuario puede agregar, modificar y eliminar la información que contiene el código y el tipo de falla. Esta información también se escribe con diccionarios y se trabaja similar como en la ventana de programar citas.

Conclusiones del Trabajo

Problemas Encontrados y Soluciones

Al principio se planeaba guardar la información en listas por cada cita para ir guardando la información de las citas. Sin embargo, se dificultó acceder a esta información para todas las citas creadas. Posteriormente, se cambió el algoritmo y se decidió guardar la información en llaves donde cada llave son los nombres de las variables como número de cita, número de placa, teléfono, correo electrónico, y los valores son los entry que se obtienen del usuario.

Por otra parte, al momento de crear el PDF, se analizaron muchas formas de crearlo. La primera opción fue utilizar una plantilla e ir escribiendo sobre la misma. La segunda opción fue crear el PDF desde cero e ir integrando la información desde cero conforme a las coordenadas.

Aprendizajes Obtenidos

Con la realización de este trabajo se entendió de una manera más amplia cómo se puede guardar la información más fácilmente en diccionarios y posteriormente en archivos, lo cual permite un mejor acceso a la información para utilizarla posteriormente.

Además se aprendió cómo enviar correos desde python con diferentes módulos y adjuntar archivos como PDF. La creación del PDF fue de gran importancia en este proyecto, así como enviar correos ya que en muchos programas se utilizan estos medios de información.

Por otra parte, se hizo un uso intensivo de for para recorrer ya sean los diccionarios o la matriz. Se comprendió que hay diferentes formas de hacer una matriz y trabajarla.

Estadística de tiempos

Actividad realizada	Horas
Análisis del problema	5
Diseño de algoritmos	12
Investigación de otras funciones en tkinter y librerías	8
Programación	42
Documentación interna	2
Pruebas	5
Elaboración manual de usuario	2
Elaboración documentación del proyecto	5
Total	81

Lista de conceptos a evaluar y análisis de resultados

Concepto	Puntos	Puntos obtenidos	Avance 100/%/0	Análisis de resultados
Programar citas y registrarlas en archivo	10		100	
Validación de programar citas	5		100	
Asignación manual de citas	1		100	
Asignación automática de citas	2		100	
Cancelar citas	1		100	
Validación de cancelar cita	1		100	
Ingreso de vehículos a la estación	5		100	
Validar ingreso de vehículos a la estación	5		100	
Despliegue del tablero de revisión	15		100	
Ejecución de los comandos del tablero	15		100	
Registro de fallas por vehículo	5		100	
Actualizar archivo de citas con resultados de la revisión	5		100	
Resultado de la revisión en PDF	10		100	
Ayuda: manual de usuario en pantalla	5		100	

Envío de correos electrónicos (cita, resultado de la revisión)	10		100	
Lista de fallas: actualización y archivo	5		100	
TOTAL				
Funcionalidades adicionales				