

***“Continúa estudiando, el cansancio es temporal,
la satisfacción es para siempre.”***

Anónimo



DEFINICIÓN DEL PROYECTO

PARQUEO

Desarrollar un programa para registrar las operaciones que se hacen en un estacionamiento de vehículos: las entradas y las salidas de los mismos, así como el cobro respectivo mediante una simulación de un cajero automático.

El programa usará archivos para almacenar la información.

Tendrá una **interfaz gráfica de usuario (GUI: Graphical User Interface)**. Puede hacer cambios en la interfaz gráfica previamente acordados con el profesor.

Para manejar cada espacio del parqueo se usará el diccionario **parqueo** con esta estructura:

Llave: número_espacio, entero entre 1 y la cantidad total de espacios del parqueo registrada en la configuración.

Valor: lista con datos del vehículo en este espacio.

[placa, fecha_hora_entrada, fecha_hora_pago, valor_pagado]

Cuando un vehículo ingresa se busca en el diccionario **parqueo** el primer espacio disponible para asignarlo (del 1 hacia adelante) y registrar los datos de placa y fecha_hora_entrada.

Con la entrada de un vehículo el dato fecha_hora_pago es un string vacío y valor_pagado es cero.

Cuando un vehículo paga se actualiza su respectivo elemento en el diccionario **parqueo** con los datos reales de fecha_hora_pago y valor_pagado.

Cuando el vehículo sale físicamente se actualiza el diccionario **parqueo**: el elemento se borra para indicar que el espacio está libre, y los datos de este elemento eliminado se agregan a la lista **detalle_de_uso** cuyos elementos son listas con la siguiente estructura:

[placa, fecha_hora_entrada, fecha_hora_pago, valor_pagado, fecha_hora_salida, numero_espacio]

Note que la lista **detalle_de_uso** contiene un elemento para cada pago hecho por los vehículos que salieron físicamente

Formato del string fecha_hora (hh: hora de 0 a 23, mm: minutos de 0 a 59):
hh:mm-dd/mm/aaaa

OBJETIVOS DEL PROYECTO

- Continuar aplicando el ciclo completo de la metodología general de desarrollo de programas:
 - Entender el problema
 - Diseñar algoritmo
 - Codificar algoritmo
 - Probar y evaluar programa
- Aplicar y reforzar aspectos del lenguaje Python 3.
 - Uso de diversos componentes del lenguaje.
 - Desarrollo de funciones.
 - Manejo de la técnica de iteración para repetición de procesos.
 - Utilizar estructuras de datos nativas de Python tales como las secuencias (listas, tuplas, strings), diccionarios.
- Aplicar buenas prácticas de programación básicas: documentación interna y externa del programa, reutilización de software, nombres significativos.
- Validar los datos de entrada: todos los datos de entrada se deben validar según restricciones que se indican en cada uno de ellos. Luego de cualquier mensaje o aviso, el programa debe esperar a que el usuario de <Intro> para continuar, se busca que el usuario pueda ver esos mensajes.
- Usar archivos de datos.

- **Usar algún software de control de versiones de software, por ejemplo Git o algún otro que usted decida.**
- **Fomentar habilidades investigativas (lectura-escritura, procesos cognitivos, trabajo colaborativo, socialización del conocimiento, obtención de información, análisis, motivación, conciencia del autoaprendizaje).** En el desarrollo de proyectos es común que aparezcan diversos aspectos afines sobre los cuales conocemos poco o nada. Estas situaciones las han podido vivir en trabajos anteriores y las seguirán viviendo, son inherentes a los quehaceres. Y para resolver nos queda investigar acerca de esos aspectos. Los mismos deben ser explicados detalladamente en la documentación del proyecto. Para este proyecto al menos debe investigar acerca de:
 - Software de control de versiones:
 - ¿Qué es?
 - Importancia en ingeniería de software.
 - Explicar las funciones utilizadas del software de control de versiones usado en el proyecto.
 - Envío de correos electrónicos.
 - Nuevas características de tkinter exploradas en este proyecto.
 - Cualquier otro aspecto necesario para ofrecer su solución.

REQUERIMIENTOS DE INFORMACIÓN

Iniciar con un menú principal implementado con botones para estas funcionalidades:

Configuración
Cargar cajero
Saldo del cajero
Ingresos de dinero
Entrada de vehículo
Cajero del parqueo
Salida de vehículo
Ayuda

Usted puede agregar otras funcionalidades que vayan a mejorar el producto.

Se puede salir del programa usando la opción de cerrar “X” en la GUI.

Estando en cualquier ventana del programa se puede volver al menú principal usando la opción de cerrar “X” en la GUI.

Seguidamente se explica el detalle de las funcionalidades.

A) CONFIGURACIÓN

PARQUEO - CONFIGURACIÓN

Cantidad de espacios en el parqueo (entero ≥ 1)	<input type="text"/>
Precio por hora (flotante con máximo 2 decimales ≥ 0)	<input type="text"/>
Pago mínimo (flotante con máximo 2 decimales ≥ 0)	<input type="text"/>
Correo electrónico del supervisor	<input type="text"/>
Minutos máximos para salir después del pago (entero ≥ 0)	<input type="text"/>
Tipos de moneda (máximo 3 tipos, enteros ≥ 0):	
Moneda 1, la de menor denominación (ejemplo 50)	<input type="text"/>
Moneda 2, denominación siguiente a la anterior (ejemplo 100)	<input type="text"/>
Moneda 3, denominación siguiente a la anterior (ejemplo 500)	<input type="text"/>
Tipos de billetes (máximo 5 tipos, enteros ≥ 0):	
Billete 1, el de menor denominación (ejemplo 1000)	<input type="text"/>
Billete 2, denominación siguiente a la anterior (ejemplo 2000)	<input type="text"/>
Billete 3, denominación siguiente a la anterior (ejemplo 5000)	<input type="text"/>
Billete 4, denominación siguiente a la anterior (ejemplo 10000)	<input type="text"/>
Billete 5, denominación siguiente a la anterior (ejemplo 20000)	<input type="text"/>
<input type="button" value="Ok"/>	<input type="button" value="Cancelar"/>

El correo electrónico es un string con un correo existente, debe verificar que este correo realmente exista ya que se usará para que el programa pueda enviarle mensajes al supervisor administrativo.

Botón **Ok**: los valores dados en esta ventana se deben mantener durante la ejecución del programa, sustituyen valores previos que tengan las variables de configuración.

Botón **Cancelar**: se cancelan, es decir, no se hacen cambios en los valores de las variables de configuración, ellas siguen manteniendo los valores que tenían antes de ingresar a esta ventana.

Luego regresa al menú principal.

Validaciones:

- Las denominaciones deben seguir las reglas indicadas: por ejemplo, si la moneda 1 es 50, la siguiente moneda debe ser > 50 o 0.
- Cuando un tipo de moneda sea 0 las siguientes también serán 0.
- Cuando un tipo de billete sea 0 los siguientes también serán 0.

- Las denominaciones se pueden cambiar solamente cuando el cajero este vacío, es decir, saldos en cero para todas las denominaciones.
- Usted debe asegurarse que los datos registrados en la configuración permitan que los cambios (vuelos) que se deben dar al usuario se puedan hacer con esa configuración. Por ejemplo, no podría aceptar un pago mínimo de 75 pesos con una moneda de 100 sino tiene monedas de 25 en las denominaciones.
- Pueden hacerse cambios a la configuración solamente cuando el parqueo está vacío.

B) CARGAR CAJERO

PARQUEO – CARGAR CAJERO						
DENOMINACIÓN	SALDO ANTES DE LA CARGA		C A R G A		SALDO	
	CANTIDAD	TOTAL	CANTIDAD	TOTAL	CANTIDAD	TOTAL
Monedas de 50	0	0	<input type="text" value="1.000"/>	50.000	1.000	50.000
Monedas de 100	0	0	<input type="text" value="1.000"/>	100.000	1.000	100.000
Monedas de 500	0	0	<input type="text" value="100"/>	50.000	100	50.000
TOTAL DE MONEDAS	0	0	2.100	200.000	2.100	200.000
Billetes de 1000	0	0	<input type="text" value="500"/>	500.000	500	500.000
Billetes de 2000	0	0	<input type="text" value="100"/>	200.000	100	200.000
Billetes de 5000	0	0	<input type="text" value="100"/>	500.000	100	500.000
Billetes de 10000	0	0	<input type="text" value="50"/>	500.000	50	500.000
Billetes de 20000	0	0	<input type="text" value="0"/>	0	0	0
TOTAL DE BILLETES	0	0	760	1.700.000	760	1.700.000
TOTAL DEL CAJERO						1.900.000

Cuando el programa inicia el cajero empieza con cero en todos los saldos de cantidades de las diferentes denominaciones. Los saldos se van modificando con los pagos de los clientes y sus respectivos cambios, además de las cargas manuales que se hacen en esta función. Las cargas se pueden hacer en cualquier momento y cuantas veces se necesiten

La sección SALDO ANTES DE LA CARGA debe reflejar los valores que tienen las variables al momento de ingresar a esta función.

Este proceso actualiza los montos de dinero en el cajero para las diferentes denominaciones. En la sección CARGA se va registrando la CANTIDAD que se está cargando en el cajero por cada denominación. Conforme se da la cantidad se va actualizando la columna TOTAL junto con las columnas CANTIDAD y TOTAL de la sección SALDO.

Botón **Ok**: establece las cantidades de la sección SALDO como las cantidades actuales de cada denominación en el cajero, por tanto debe actualizar las variables que usa para tales efectos.

Botón **Cancelar**: las cantidades de las denominaciones no son modificadas con los valores registrados en la ventana, en su lugar se mantienen con los valores que tenían antes de ingresar a esta función.

Botón **Vaciar cajero**: todas las cantidades y totales se ponen en cero, esta ventana debe reflejar este hecho, es decir todas las columnas deben verse en cero. Luego esta opción debe desaparecer de la lista de opciones y vuelven a quedar disponibles las otras para que el usuario pueda escogerlas nuevamente.

Luego regresa al menú principal.

Validaciones:

- Cantidad (entero ≥ 0).

C) SALDO DEL CAJERO

PARQUEO – SALDO DEL CAJERO						
DENOMINACIÓN	ENTRADAS		SALIDAS		SALDO	
	CANTIDAD	TOTAL	CANTIDAD	TOTAL	CANTIDAD	TOTAL
Monedas de 50	1.000	50.000	0	0	1.000	50.000
Monedas de 100	1.009	100.900	4	400	1.005	100.500
Monedas de 500	100	50.000	0	0	100	50.000
TOTAL DE MONEDAS	2.109	200.900	4	400	2.105	200.500
Billetes de 1000	501	501.000	2	2.000	499	499.000
Billetes de 2000	100	200.000	0	0	100	200.000
Billetes de 5000	101	505.000	0	0	101	505.000
Billetes de 10000	50	500.000	0	0	50	500.000
Billetes de 20000	10	200.000	0	0	10	200.000
TOTAL DE BILLETES	762	1.906.000	2	2.000	760	1.904.000

Ok

Esta función se usa para consultar el dinero que tiene el cajero. La sección ENTRADAS contiene las cantidades que han entrado al cajero por los conceptos de cargas y pagos. La sección SALIDAS contiene las cantidades que han salido por concepto de cambios (vuelos). La sección SALDO contiene ENTRADAS – SALIDAS.

Botón **Ok**: regresa al menú principal.

D) INGRESOS DE DINERO

PARQUEO – INGRESOS DE DINERO	
Del día	<input type="text" value="dd/mm/aaaa"/>
Al día	<input type="text" value="dd/mm/aaaa"/>
TOTAL DE INGRESOS EN EFECTIVO	xxx.xxx.xxx
TOTAL DE INGRESOS POR TARJETA DE CRÉDITO	xxx.xxx.xxx
TOTAL DE INGRESOS	xxx.xxx.xxx
ESTIMADO DE INGRESOS POR RECIBIR	xxx.xxx.xxx

Ok

Con esta función se calculan los ingresos reales y estimados durante un período de tiempo. El usuario registra el período y el programa comienza con los cálculos.

Los ingresos de dinero (en efectivo y tarjeta) se calculan tomando todos los pagos que se han realizado en el período según el diccionario **parqueo** y la lista **detalle_de_uso**.

Los vehículos que no han salido físicamente pero que ya han realizado el pago están en el diccionario **parqueo**.

Los pagos de los vehículos que han salido físicamente están en la lista **detalle_de_uso**.

Para calcular el estimado de ingresos hay que considerar los vehículos que no han pagado: estos se encuentran en el diccionario **parqueo** y no tienen datos de pago. Los cálculos se hacen tomando el tiempo transcurrido desde su entrada hasta la fecha y hora que tenga en el momento el sistema.

Botón **Ok**: regresa al menú principal.

E) ENTRADA DE VEHÍCULO

PARQUEO – ENTRADA DE VEHÍCULO

Espacios disponibles xxxxxxxxxxx

SU PLACA

Campo asignado XXXXXXXXXXXXX

Hora de entrada hh:mm dd/mm/aaaa

Precio por hora xxxxxxxxxxx

Puede entrar al parqueo si hay espacios disponibles. La hora y fecha de entrada se toma del sistema.

Botón **Ok**: reserva espacio. Los datos se registran en el diccionario **parqueo**.

Botón **Cancelar**: no reserva espacio.

Luego regresa a pedir otra placa.

Validaciones:

- La placa no debe estar en el diccionario **parqueo**, esto significaría que el vehículo ya está dentro del mismo.
- Calcular los espacios disponibles. Sino hay espacios desplegar el mensaje

NO HAY ESPACIO (color rojo y letras que sobresalgan).

F) CAJERO DEL PARQUEO

CAJERO DEL PARQUEO

XXXXXX POR HORA

Paso 1: SU PLACA XXXXXXXXXX

HORA DE ENTRADA HH:MM dd/mm/aaaa

HORA DE SALIDA HH:MM dd/mm/aaaa

TIEMPO COBRADO XXh YYm zzzd

Paso 2: SU PAGO EN: MONEDAS

50

100

500

1000

2000

5000

10000

20000

BILLETES

TARJETA DE CRÉDITO

A PAGAR

XXXXXXX

Pagado

XXXXXX

Cambio

XXXXXX

Las monedas y billetes son botones

Paso 3: SU CAMBIO EN:

MONEDAS	BILLETES
XX DE 50	XX DE 1000
XX DE 100	XX DE 2000
XX DE 500	XX DE 5000
	XX DE 10000

Anular el pago

Esta operación simula “un cajero automático de parqueo”.
 Debe desplegar una ventana con los datos indicados aquí.
 Se pide la placa para obtener datos de entrada y salida para cobro.
 El cobro debe calcularse según la configuración y se despliega en la casilla A PAGAR.
 El pago se puede hacer con monedas, billetes y tarjeta de crédito.
 Para simular el pago en lugar de dar dinero físicamente se presionan los botones respectivos de MONEDAS, BILLETES y TARJETA DE CRÉDITO del paso 2. Por ejemplo, si va a pagar 350 pesos con 3 monedas de 100 y una de 50 debe presionar 3 veces el botón de la moneda de 100 y 1 vez el botón de la moneda de 50. Cada vez que se presiona un botón de dinero se acumula el monto correspondiente en la casilla **Pagado**.
 El usuario puede presionar estos botones mientras lo **Pagado** sea menor que **A PAGAR**. En el momento que la casilla **Pagado** sea igual o mayor a la casilla **A PAGAR** ya no debe aceptar más pagos. En su lugar debe calcular y desplegar el cambio (vuelto) junto con su desglose de moneda (paso 3). También actualiza datos del pago en diccionario **parqueo**.
 En caso de pagar con una tarjeta de crédito debe posicionar el cursor en esa casilla y dar un número natural de 10 dígitos exactos. De la tarjeta se toma la diferencia que exista entre **A**

PAGAR y Pagado. No hay validación de aceptación de la tarjeta de crédito, vamos a asumir que siempre se hace.

Al finalizar el pago se debe crear un recibo que incluya los datos del paso 1 (formato PDF).

Botón **Anular el pago**: se puede presionar mientras no se haya hecho el pago total. Sirve para anular el pago devolviendo el dinero que ha pagado el usuario.

Después de registrar o anular el pago el programa vuelve al paso 1 para otro pago.

Validaciones:

- El cálculo del cambio con sus respectivas denominaciones (desglose de moneda) debe considerar todos los casos posibles. Siempre debe dar la mínima cantidad de denominaciones según los saldos que maneja el cajero. Considere diferentes casos, entre ellos:

- o Caso 1: No se puede dar el cambio. Ejemplo: A pagar 1.650, pago con 2.000, no hay monedas de 50 (saldo de estas monedas es 0): el cambio de 350 requiere 1 moneda de 50 pero no se puede porque no hay. En estos casos el pago no se acepta y debe devolver el pago que ha hecho el cliente exactamente con las mismas denominaciones. Se envía un correo electrónico al supervisor con un mensaje sobre la situación presentada para que se presente a cargar el cajero.

- o Caso 2: Dar el cambio pero no representa la mínima cantidad de denominaciones. Puede pasar cuando el saldo de algunas denominaciones no es suficiente. Por ejemplo: A pagar 1.700 con un billete de 2.000, en caso de haber solo 1 moneda de 100 los otros 200 de vuelto se pueden completar con 4 monedas de 50 si las hubiera.

- Luego de realizar un pago el programa debe enviar un correo al supervisor con las denominaciones que tengan una cantidad menor a 5.

- Considere que hay un pago mínimo.

G) SALIDA DE VEHÍCULO

PARQUEO – SALIDA DE VEHÍCULO

SU PLACA

El espacio ocupado por el vehículo se libera hasta que dicho vehículo haga la salida física. Esto es debido a que puede hacer el pago pero permanecer en el parqueo. Por eso mientras el vehículo no haga este proceso, se asume que sigue permaneciendo en el parqueo, ocupando el mismo espacio.

Botón **Ok**: actualiza diccionario **parqueo** (elimina elemento) y lista **detalle_de_uso** (se agrega elemento) según se mencionó anteriormente. Además, si no puede hacer la salida porque excedió el tiempo para ello, automáticamente al diccionario **parqueo** le agrega un elemento donde la fecha y hora de entrada se toman de la fecha y hora de pago del elemento eliminado, reflejando que es una nueva entrada del vehículo para que pueda hacer el pago en el momento que considere oportuno. Luego regresa al dato de placa para otra salida.

Validaciones:

- La placa debe estar en el diccionario **parqueo** y haber pagado.

- Tiempo transcurrido entre el pago y la salida física: si es mayor a lo permitido según la configuración, no se permite la salida del vehículo, en cuyo caso debe dar la siguiente información:

No puede salir porque excedió el tiempo permitido para ello.

Tiempo máximo para salir luego del pago xxxxxxxxxxxx

Tiempo que usted ha tardado xxxxxxxxxxxx

Debe regresar al cajero a pagar la diferencia.

H) AYUDA

Esta opción la usaremos para que el usuario pueda ver el Manual de Usuario directamente en la computadora (despliega el pdf respectivo).

ACERCA DE

Puede poner esta opción 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.

SALIR

Puede poner esta opción para salir del programa. También se puede salir con la opción de cerrar “X” en la GUI.

USO DE ARCHIVOS

Cada vez que inicie la ejecución del programa debe leer los datos que están en los archivos para asignarlos a las estructuras en memoria. La primera vez que se corre el programa los archivos están vacíos, luego se van actualizando con las operaciones que haga el usuario. Cada vez que termina la ejecución del programa debe grabar los datos en los archivos.

ARCHIVO **configuración.dat**

Contiene los datos de configuración.

Archivo tipo string por líneas: cada dato de configuración se debe poner en una línea.

ARCHIVO **parqueo.dat**

Contiene el diccionario parqueo.

Archivo binario usado con el módulo pickle.

ARCHIVO **detalle_de_uso.dat**

Contiene la lista detalle_de_uso.

Archivo binario usado con el módulo pickle.

ARCHIVO **cajero.dat**

Debe contener los datos que usted necesita para manejar el cajero. Usted define esos datos y el tipo de archivo que va a utilizar.

DOCUMENTACIÓN DEL PROYECTO

REQUISITOS PARA REVISAR EL PROYECTO

- a- El programa debe tener documentación interna.
- b- Durante todo el desarrollo del proyecto debe usar un software de control de versiones.
- c- El programa debe usar una interfaz tipo GUI.
- d- La nota de la documentación del proyecto sirve para aceptar o rechazar el proyecto: se revisan los proyectos que cumplan con la documentación en un 90% o más.

Enviar vía tecDigital, sección EVALUACIONES / PROGRAMAS, una carpeta comprimida (solo tipo zip) de nombre **programa3_su_nombre** que contenga las siguientes partes:

Parte 1: Documentación del proyecto (**nombre: documentación_parqueo.PDF**).

- Portada. (1P)
- Contenido. (2P)
- Enunciado del proyecto. (2P)
- Temas investigados (material no estudiado en el curso). (35P)
 - Por cada uno de estos temas debe poner el marco teórico: de qué trata, cómo se usa.
- Conclusiones del trabajo: (15P)
 - Problemas encontrados y soluciones a los mismos.
 - Aprendizajes obtenidos.
- Estadística de tiempos: un cuadro que muestre el detalle de las actividades que realizó y las horas invertidas en cada una de ellas. La estadística permite medir el esfuerzo dedicado al trabajo en términos de actividades y tiempos, lo cual puede ser una base para calcular el esfuerzo requerido en futuros trabajos (5P). Ejemplos de actividades:

Actividad Realizada	Horas
Análisis del proyecto	
- Leer y entender el proyecto	
- Etc.	
Diseño de algoritmos	
Investigación de ...	
Programación	
Documentación interna	
Pruebas	
Elaboración del manual de usuario	
Elaboración de documentación del proyecto	
Etc.	
TOTAL	

- Lista de revisión del proyecto (PONGA ESTA LISTA EN PÁGINA NUEVA). (20P)
- Por cada concepto de la lista de revisión usted debe completar el % de avance, puntos obtenidos según ese avance y el análisis de resultados de su proyecto.

- 100: Totalmente desarrollado. No hace falta análisis excepto que requiera hacer alguna observación.
- Un % específico, por ejemplo 80 significaría un desarrollo parcial del 80%. En el análisis indicar tres partes: ¿qué hace?, ¿qué falta?, ¿por qué no se completó?
- 0: No desarrollado. En el análisis indicar ¿por qué no se desarrolló?.

- Partes que desarrolló adicionales a los requerimientos.

- Manual de usuario (**nombre: manual_de_usuario_parqueo.PDF** (20P))

Es un documento de comunicación técnica utilizado para guiar a las personas que usan el software. Explica paso a paso cómo usar cada una de las funcionalidades del programa. Apóyese en imágenes, capturas de pantallas, menús, diagramas y los aspectos que considere van a servir como una guía útil para que el usuario pueda usar el programa. Puede tomar como referencia algún manual de usuario de alguna aplicación de software o esta misma especificación del proyecto con los ajustes correspondientes.

Parte 2: Programa fuente (**nombre: parqueo.py**) y todos los objetos necesarios para ejecutar el programa.

LISTA DE REVISIÓN DEL PROYECTO

Concepto	Puntos originales	Avance 100/%/0	Puntos obtenidos	Análisis de resultados
Menú principal	2			
Configuración	10			
Cargar cajero	8			
Saldo del cajero	4			
Ingresos:				
Reales	8			
Estimados	8			
Entrada de vehículo	5			
Cajero del parqueo paso 1	5			
Cajero del parqueo paso 2:				
Registrar pago	5			
Devolver el pago	3			
Enviar correos	5			
Recibo del pago	4			
Cajero del parqueo paso 3	5			
Salida de vehículo:				
Eliminar elemento de parqueo	3			
Agregar elemento a detalle_de_uso	3			
Entrada automática por exceder tiempo de salida	3			
Validación de datos	8			
Ayuda (manual de usuario desplegado en el programa)	5			
Archivos:				
Leer archivos	3			
Grabar archivos	3			
TOTAL	100			
Funciones desarrolladas adicionalmente				

La calificación de las validaciones de los datos de entrada está como un concepto aparte.
 La calificación de las validaciones de procesos está dentro del concepto respectivo.

Última línea