Fecha de Entrega: 12 de abril, 11pm

OBJETIVOS DEL PROYECTO

Este es el primer proyecto de programación del curso cuyos objetivos son:

- Empezar a aplicar el ciclo completo de la metodología de desarrollo de programas a situaciones de mayor alcance:
  - 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.
  - Uso de estructuras condicionales y de repetición de procesos.
  - Manejo de la técnica de iteración para repetición de procesos.
- 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
- Inducir al estudiante a la investigación: aquellos temas no tratados en el curso pero que los necesita para hacer el proyecto. Dichos temas deben ser explicados detalladamente en la documentación del proyecto. Entre los tópicos a investigar para este proyecto específico están:
  - Importancia de la documentación de software.
  - Programación por eventos.
  - Uso de interfaz gráfica de usuario (GUI: **G**raphical **U**ser **I**nterface) en el desarrollo de software.
  - Desarrollo de GUI con tkinter: biblioteca incluida en Python. tkinter se le considera el estándar de facto para la programación de GUI con Python. En esta parte hay que describir los elementos específicos de tkinter usados para este proyecto.
  - Herramientas de depuración (debugger) en programación.
  - Explicación de las funciones del "debugger" del IDE que está usando.
  - Así como cualquier otro aspecto necesario para ofrecer su solución.

DEFINICIÓN DEL PROYECTO: PARQUEO

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

El programa tendrá una GUI que inicia con un menú principal desde el cual se accederá su funcionalidad, es decir, las diferentes operaciones que el programa hace. Usted puede agregar otras funcionalidades que vayan a mejorar el producto. También puede hacer cambios a la interfaz gráfica pero deben seguir cumpliendo los requerimientos del programa que se indican seguidamente.

REQUERIMIENTOS DEL PROGRAMA

Lo primero que desplegará la interfaz gráfica será una barra de menú con estas opciones:

- Configuración
- Dinero del cajero (submenú)
  - Cargar cajero
  - Saldo del cajero
  - Ingresos de dinero
- Entrada de vehículo
- Cajero del parqueo
- Salida de vehículo
- Ayuda
- Acerca de

1) CONFIGURACIÓN

Cantidad de espacios en el parqueo (entero >=1) [ ]

Precio por hora (flotante >=0) [ ]

Pago mínimo (entero >=0) [ ]
En caso de no existir este mínimo ponga 0, de esta manera se cobra el tiempo de acuerdo al redondeo de cobro indicado a continuación.

Redondear el cobro a los siguientes minutos (entero entre 0 y 60) [ ]

Minutos máximos para salir después del pago (entero >=0) [ ]

Tipos de moneda (máximo 3 tipos, enteros >= 0)
- Moneda 1, la de menor denominación (ejemplo 50) [ ]
- Moneda 2, denominación siguiente a la anterior (ejemplo 100) [ ]
- Moneda 3, denominación siguiente a la anterior (ejemplo 500) [ ]

Tipos de billetes (máximo 5 tipos, enteros >= 0)
- Billete 1, el de menor denominación (ejemplo 1000) [ ]
- Billete 2, denominación siguiente a la anterior (ejemplo 2000) [ ]
- Billete 3, denominación siguiente a la anterior (ejemplo 5000) [ ]
- Billete 4, denominación siguiente a la anterior (ejemplo 10000) [ ]
- Billete 5, denominación siguiente a la anterior (ejemplo 20000) [ ]

Ok Cancelar

Cada uno de los valores de la configuración se va a guardar en una variable (por ejemplo: precio_por_hora, tipo_moneda1, etc.). Al inicio del programa estas variables estarán en cero, luego con esta opción se pueden ir modificando.

Para la cantidad de espacios del parqueo se usará la lista **parqueo**, la cual inicialmente tendrá todos sus elementos como listas vacías. Por ejemplo si la cantidad de espacios es 10 la lista tendrá [[], [], [], [], [], [], [], [], [], []]. El espacio 1 está en el elemento 0, el espacio N está en elemento N – 1.

Cuando un vehículo ingresa se busca en la lista **parqueo** el primer espacio disponible para asignarlo y registrar los datos de placa y fecha_hora_entrada.

Estructura de la lista **parqueo**:

[[placa, fecha_hora_entrada, fecha_hora_pago, valor_pagado], …]

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

hh:mm-dd/mm/aaaa

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 la lista **parqueo** con los datos reales de fecha_hora_pago y valor_pagado.

Cuando un vehículo sale físicamente se actualiza la lista **parqueo:** se elimina ese elemento y se cambia por una lista vacía para indicar que el espacio está libre, el elemento eliminado se agrega a la lista **detalle_de_uso** 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.

Botón **Ok**: sustituye los valores de las variables correspondientes con los valores que se han dado en esta 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 de cualquiera de estos botones 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.
- Cuando un tipo de moneda sea 0 las siguientes también serán 0.

- Cuanto 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 (vueltos) 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.
- La cantidad de espacios se puede modificar solamente cuando el parqueo está vacío.
-

2) CARGAR CAJERO

	SALDO ANTES DE LA CARGA		C A R G A		SALDO	
DENOMINACIÓN	CANTIDAD	TOTAL	CANTIDAD	TOTAL	CANTIDAD	TOTAL
Monedas de 50	0	0	1.000	50.000	1.000	50.000
Monedas de 100	0	0	1.000	100.000	1.000	100.000
Monedas de 500	0	0	100	50.000	100	50.000
TOTAL DE MONEDAS	0	0	2.100	200.000	2.100	200.000
Billetes de 1000	0	0	500	500.000	500	500.000
Billetes de 2000	0	0	100	200.000	100	200.000
Billetes de 5000	0	0	100	500.000	100	500.000
Billetes de 10000	0	0	50	500.000	50	500.000
Billetes de 20000	0	0	0	0	0	0
TOTAL DE BILLETES	0	0	760	1.700.000	760	1.700.000
TOTAL DEL CAJERO						1.900.000

Ok Cancelar

Actualiza montos de dinero de las diferentes denominaciones para dar vueltos. Las columnas de la sección SALDO se actualizan conforme se van registrando datos en la columna CANTIDAD de la sección CARGA.

Inicialmente el cajero empieza con cero en todos los saldos de cantidades de las diferentes denominaciones. Los saldos se van modificando con las cargas, los pagos y los cambios. Las cargas se pueden hacer en cualquier momento y cuantas veces se necesiten.

Botón **Ok**: adiciona las cantidades de la sección CARGA a las variables de las denominaciones.

Botón **Cancelar**: los valores de las variables de las denominaciones no son modificados con los valores en esta ventana, se mantienen con los valores que tenían antes de ingresar a esta ventana.

Luego de cualquiera de estos botones regresa al menú principal.

Validaciones:

- Cantidad (entero >=0).

3) SALDO DEL CAJERO

	ENTRADAS		SALIDAS		SALDO	
DENOMINACIÓN	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

Vaciar cajero ☐

Ok Cancelar

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 (vueltos).

La sección SALDO contiene ENTRADAS – SALIDAS.

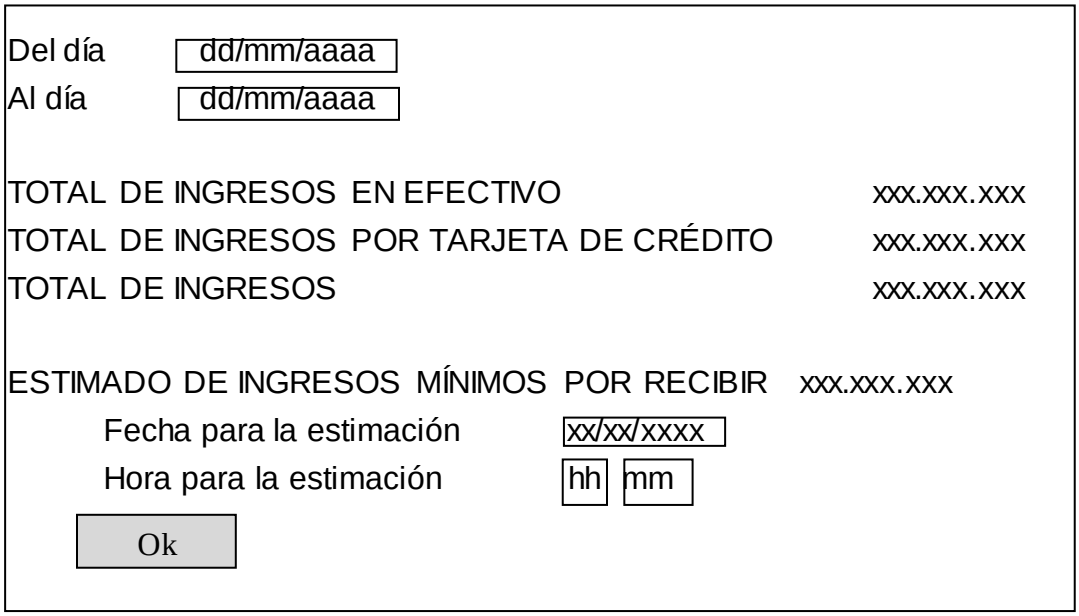
Opción **Vaciar cajero**: esta opción inicialmente está desactivada cuando se ingresa a esta ventana, es decir, no se va a vaciar el cajero, por tanto los saldos de las

diferentes denominaciones se mantienen. El usuario puede activar la opción (con un check) lo cual significa que los saldos de las denominaciones se pondrán en cero.

Botón **Ok**: aplica la opción **Vaciar cajero** y regresa al menú principal.

Botón **Cancelar:** omite cualquier acción en esta ventana (en este caso la opción de **Vaciar cajero**) y regresa al menú principal.

4) INGRESOS DE DINERO



Los ingresos de dinero se calculan tomando todos los pagos que se han realizado en el intervalo de días según las listas **parqueo** y **detalle_de_uso**.

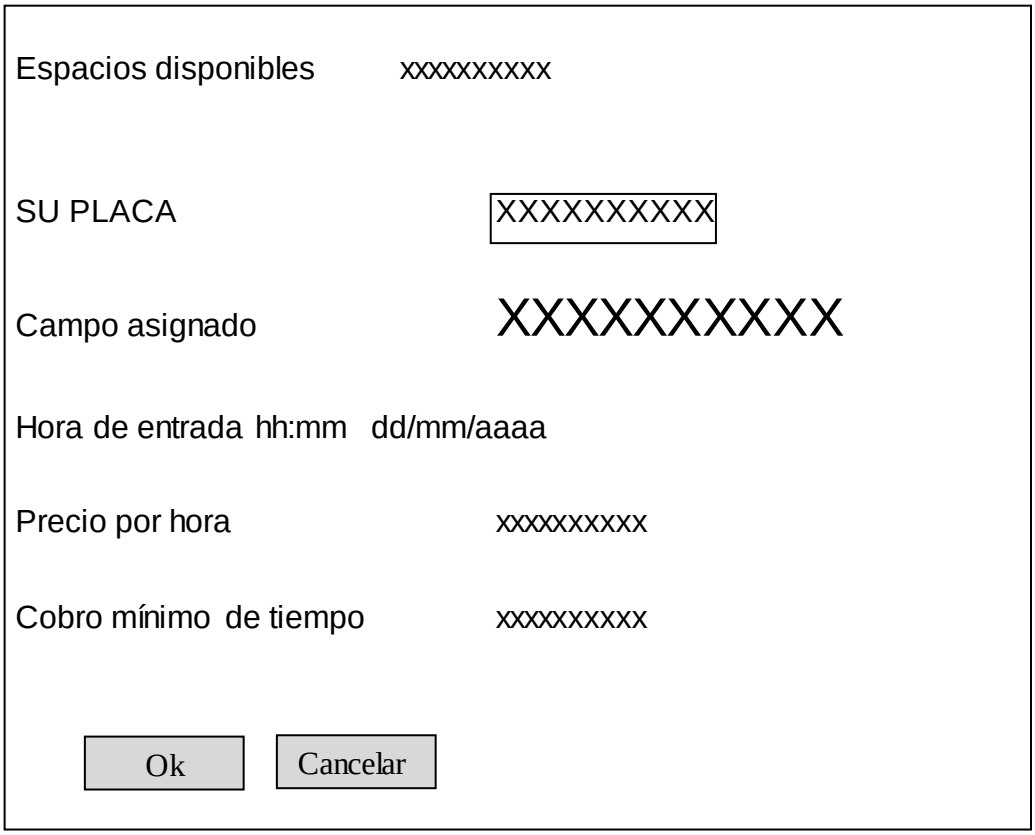
Los pagos de los vehículos que no han salido físicamente están en la lista **parqueo**.

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

El estimado de ingresos hay que calcularlo considerando todas la entradas de vehículos que no han sido pagadas , se encuentran en la lista **parqueo** y no tienen datos de pago. Los cálculos se hacen tomando el tiempo transcurrido desde su entrada hasta fecha y hora que el usuario registre para la estimación, sino da estos datos se toman del sistema.

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

5) ENTRADA DE VEHÍCULO



Puede entrar al parqueo si hay espacio disponible (elementos con una tupla vacía) en cuyo caso se registran los datos en la lista **parqueo**.

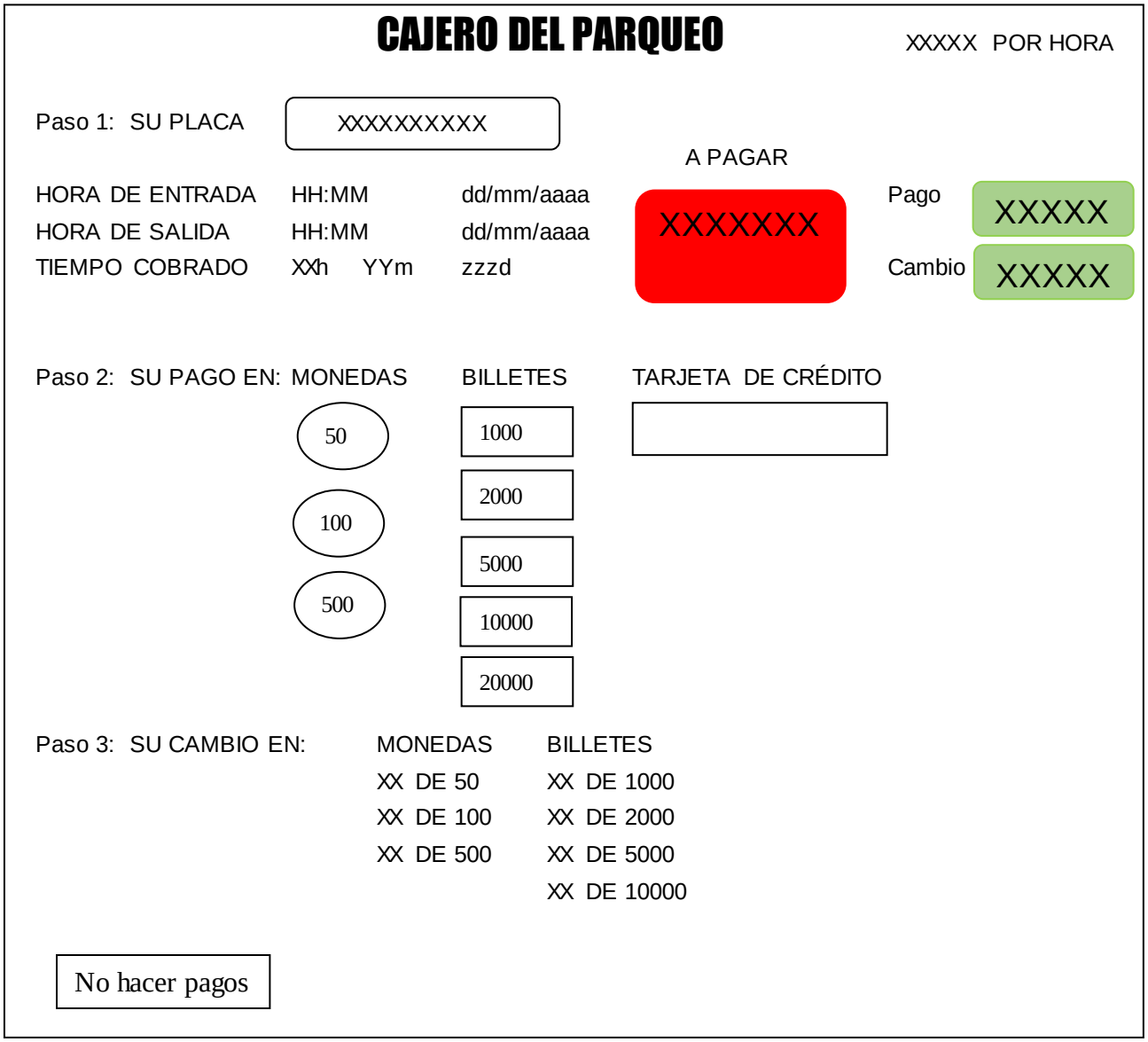
Botón Ok: reserva espacio y regresa al menú.
Botón Cancelar: no reserva espacio y regresa al menú

Validaciones:

- Calcular los espacios disponibles. Sino hay espacios desplegar el mensaje **NO HAY ESPACIO** (color rojo y letras que sobresalgan).

- La placa no debe estar en la lista **parqueo**, esto significaría que el vehículo ya está dentro del mismo.

6) CAJERO DEL PARQUEO



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.

Si va a pagar con una moneda (por ejemplo 50), presiona ese botón y acumula ese monto en la casilla **Pago**. Cada vez que se presiona un botón de moneda o billete se acumula esa denominación en la casilla **Pago**.

El usuario puede presionar estos botones mientras el **Pago** sea menor que **A PAGAR**. En el momento que la casilla de **Pago** sea igual o mayor a la casilla **A PAGAR** ya no debe aceptar más monedas o billetes. En su lugar debe calcular y desplegar el cambio (vuelto) junto con su desglose de moneda.

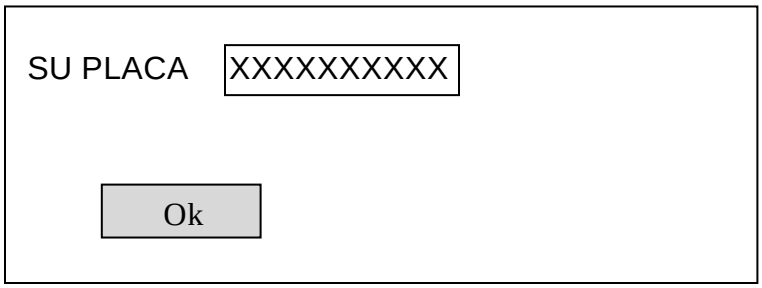
En caso de pagar con una tarjeta de crédito debe posicionar el cursor en esa casilla y dar un número natural de 4 dígitos exactos, tomando de la tarjeta la diferencia que exista entre **A PAGAR** y **Pago**. No hay validación de aceptación del crédito, vamos a asumir que siempre se hace. Una vez hecho el pago con tarjeta este no se devuelve.

Botón **No hacer pagos:** termina esta operación y regresa al menú principal. Si algún pago está procesándose en el momento, entonces devolver las denominaciones correspondientes.

Validaciones:

- El cálculo del cambio con sus respectivas denominaciones (desglose de moneda) debe considerar todos los casos posibles. En caso de ser posible debe dar la mínima cantidad de denominaciones, sin embargo considere diferentes casos, entre ellos:
    - Caso: 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 retornar el mismo pago que ha hecho el cliente. Se envía mensaje al cliente con la situación presentada y debe esperar a que carguen el cajero.
    - Caso: Dar el cambio pero no representa la mínima cantidad de denominaciones. Puede pasar cuando el saldo de algunas es 0. Por ejemplo: A pagar por 1.650, pago con 2.000, no hay monedas de 100: el cambio de 350 sería con 7 monedas de 50.
- El dato de pago mínimo en la configuración tiene prioridad sobre el redondeo de minutos. Por ejemplo un usuario que usó solo 20 minutos el parqueo, si el cobro debe redondearse a 0.5 horas con un precio por hora de 600, el cálculo daría 300, pero si el pago mínimo indica 400 entonces se cobra los 400.

7) SALIDA DE VEHÍCULO



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 las listas **parqueo** y **detalle_de_uso** según se mencionó. Además sino puede hacer la salida porque excedió el tiempo para ello, automáticamente la lista **parqueo** se actualiza en el mismo elemento con los datos actuales de fecha y hora reflejando que es una nueva entrada de vehículo para que pueda hacer el pago en el momento que considere oportuno. Luego regresa al menú principal.

Validaciones:

- La placa debe estar en la lista **parqueo**.
- 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 xxxxxxxxxx
Tiempo que usted ha tardado xxxxxxxxxx

Debe regresar al cajero a pagar la diferencia.

8) AYUDA

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

9) 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.

10) SALIR

Esta opción se usa para salir del programa (también se puede salir con el botón de cerrar “X” en la interfaz gráfica).

REQUISITOS PARA REVISAR EL PROYECTO

a) **Presentar la documentación del proyecto indicada en la sección siguiente. La nota de la documentación del proyecto sirve para aceptar o rechazar el proyecto: se revisan los proyectos que cumplan con este requisito en un 90% o más.**

b) **El programa debe estar documentado internamente.**

c) **El programa debe usar una GUI.**

DOCUMENTACIÓN DEL PROYECTO

Enviar vía tecDigital, sección EVALUACIONES / PROGRAMAS, una carpeta comprimida (solo de tipo .zip) de nombre **programa1** que contenga las siguientes partes:

- Parte 1: Documentación del proyecto
  (**nombre: documentación_parqueo.PDF**).
  - Portada. (1 p)
  - Contenido. (2 p)
  - Enunciado del proyecto. (2 p)
  - Temas investigados (material no estudiado en el curso). (35 p)
    - Por cada uno de estos temas debe poner el marco teórico: de qué trata, cómo se implementa.
    - Para este trabajo debe investigar al menos:
      - Importancia de la documentación de software.
      - Programación por eventos.
      - El uso de interfaces gráficas de usuario (GUI) en el desarrollo de software.
      - Desarrollo de GUI con tkinter: biblioteca incluida en Python. tkinter se le considera el estándar de facto para la programación de GUI con Python. En esta parte hay que describir los elementos específicos de tkinter usados para este proyecto.
      - Herramientas de “debugger” en programación.
      - Funciones del “debugger” del IDE de Python que está usando.
      - Así como cualquier otro aspecto necesario para ofrecer su solución.

- Conclusiones del trabajo: (15 p)
  - 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. (5 p)

  Ejemplos de actividades:

Actividad Realizada	**Horas**
Análisis de requerimientos	
Diseño de algoritmos	
Investigación de ...	
Programación	
Documentación interna	
Pruebas	
Elaboración del manual de usuario	
Elaboración de documentación del proyecto	
Etc.	
TOTAL	

- Rúbrica de evaluación y análisis de resultados (PONGA LA HOJA DE LA RÚBRICA EN PÁGINA NUEVA). (15 p)
  - Tome la rúbrica de evaluación y por cada concepto calificado Usted debe indicar el % de 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 el motivo.
  - Partes que desarrolló adicionales a los requerimientos.

Concepto	**Puntos**	**% de avance 100 / x / 0**	**Puntos obtenidos**
Menú con barra de opciones:	4		
Configuración	15		
Cargar cajero	5		
Saldo del cajero	5		
Ingresos	10		
Entrada de vehículo	5		
Cajero del parqueo	28		
Algoritmo de desglose de moneda en el vuelto con sus diferentes casos	12		
Salida de vehículo	10		
Acerca de / Salir	1		
Ayuda (manual de usuario desplegado en el programa)	5		
TOTAL	**100**		
Funciones desarrolladas adicionales			

- Manual de usuario (**nombre: manual_de_usuario_parqueo.PDF**). (25 p)

  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.

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

IMPORTANTE: CONOCIMIENTO DE LA SOLUCIÓN PRESENTADA. En la revisión del trabajo el estudiante debe demostrar un completo dominio de la solución que implementó, tanto desde el punto de vista técnico (uso de Python) como de la funcionalidad del programa.

Última línea

"...mira con optimismo el estudio que estás haciendo.

Estás aquí porque te estás formando para la vida.

Estás entrenando tu cerebro y tu inteligencia para ser una persona de bien que aporte muchas cosas a una sociedad actual carente de muchos valores.

Pon energía y entusiasmo que el estudio puede ser pesado, pero encontrarás muchos beneficios con tus logros alcanzados.

DIOS te siga bendiciendo..."