



Clase	Atributos	Metodos
Smartparking(Main)	ninguno	main():void
ConexionArduino	+valores: String[3] -est: Estacionamiento -ph: PanamaHitek_Arduino +events: SerialEventListener	-split(lectura: String): String[] -recorrerNumeros(valores: String[]): void -pattern2(group: String): void -pattern3(group: String): void
Estacionamiento	+lotes : ArrayList<Lote> +cambio: ArrayList<Boolean>	-llenadoTotalNombre() : void +copiarEstado(): void +actualizarEstado(posicion: int, valorConexion: int) : void -addLoteNombre(): void +botonActualizar(camb: ArrayList<Boolean>, lables: ArrayList<JPanel>): void

Lote	-uso: String -estado: boolean -comparar: boolean -md: ManejoDatos -nombre: String	+getEstado(): boolean +setEstado(entrada: int): void
GuiPrincipal	-mostrar: JButton -imagen: JLabel -version: JLabel -contenido: Container +est: Estacionamiento	-configuracionVentana(): void -inicioComponentes(): void -imageURL(miLabel: JLabel): void -accionBoton(): void +actionPerformed(e :(ActionEvent)) : void
GuiEstacionamiento	-volver: JButton -text: JLabel -actualizar: JButton -panel_1: JPanel -panel_2: JPanel -panel_3: JPanel -panel_4: JPanel -panel_5: JPanel -panel_6: JPanel -panel_7: JPanel -panel_8: JPanel -panel_9: JPanel -panel_10: JPanel -panel_11: JPanel -panel_12: JPanel -panel_13: JPanel -panel_14: JPanel -panel_15: JPanel -panel_16: JPanel -panel_17: JPanel -panel_18: JPanel -panel_19: JPanel -panel_20: JPanel -panel_21: JPanel -disponible: JPanel -ocupado: JPanel	-configuracionVentana(): void -inicioComponentes(): void -configPaneles(): void -accionBoton(): void +actionPerformed(e :(ActionEvent)) : void

	-contenido: Container -est: Estacionamiento -lblOcupado: JLabel -lblDisponible: JLabel -espacio: ArrayList<JPanel>	
CalcularEstadia	-diaEstadia -horaEstadia -minutoEstadia -segundoEstadia	-ajustarMes(mes: String): void -ajustarDia(mes: String): void -ajustarHora(mes: String): void -ajustarMinuto(mes: String): void -ajustarSegundo(mes: String): void -extraerMes(charRaw: char[]): String -extraerValor(horaChar : char[], posicion : int) : int +calcular(dataRAW : String) : String
ManejoDatos		+aniadirArchivo(lista : String, Path : String) : void

La clase “SmartParking” es el main del programa, y como tal, su única función es inicializar algunas variables y servir como punto de inicio para el programa.

La clase “ConexionArduino” se encarga de establecer la conexin entre el programa principal (escrito en Java) y el programa controlador del dispositivo Arduino.

La clase “Estacionamiento” sirve como un “almacén”, guardando la información relevante más reciente. Su metodo “addLoteNombre” agrega un nuevo “Lote” para ser registrado por la clase “Estacionamiento”.

El método “llenadoTotalNombre” itera el método “addLoteNombre” hasta alcanzar la capacidad máxima de la clase “Estacionamiento”.

El método “actualizarEstado”, tal como indica su nombre, actualiza el estado actual de todos los “Lotes” del “Estacionamiento”. Este método es ejecutado a intervalos de tiempo regulares.

El método “botonActualizar” actualiza la información del “Estacionamiento” de manera manual.

El método “copiarEstado” obtiene el estado actual de los “Lotes” del “Estacionamiento” para ser usado por otros métodos.

La clase "Lote" representa una parte de un "Estacionamiento" y su función es indicar su estado actual. Dado lo anterior, sus únicos métodos son un "getter" y un "setter" para su propio estado (disponible u ocupado).

La clase "GuiPrincipal" es la encargada de crear y mostrar la ventana principal para interacción entre el usuario final y el programa.

El método "configuracionVentana" configura la ventana, "inicioComponentes" inicia las componentes de la ventana, "imageURL" descarga la imagen central que utiliza la ventana. El método "accionBoton" despliega al usuario información de contacto del desarrollador para poder contactar a este.

El método "actionPerformed" cierra la ventana actual y crea una nueva ventana a partir de la clase "GuiEstacionamiento".

La clase "GuiEstacionamiento" es la encargada de crear y mostrar una ventana donde se despliega la información más reciente del "Estacionamiento".

Al igual que la clase anterior, "GuiEstacionamiento" posee sus propios métodos "configuracionVentana" e "inicioComponentes", los cuales cumplen un rol similar que en la clase anterior.

El método "configPaneles" configura los "JPanel's" que utiliza "GuiEstacionamiento".

El método "accionBoton" solicita una actualización del estado actual del "Estacionamiento" y modifica el despliegue en la ventana para que refleje el estado actual.

El método "actionPerformed" cierra la ventana actual y retorna al usuario a la "ventana principal".

La clase "CalcularEstadia" es un conjunto de atributos y métodos necesarios para estimar la cantidad de tiempo que un "Lote" se encontró "ocupado".

El método "extraerMes" obtiene el mes en el cual el "Lote" volvió a estar "disponible".

El método "extraerValor" extrae de un "String" el valor numérico del dato solicitado (hora, día, minuto o segundo).

El método "calcular" se encarga de estimar el "tiempo de estadia" de un vehículo en un "Lote" determinado. Para ello solicita la información obtenida por "extraerMes" y "extraerValor" y luego asegura valores coherentes con los métodos "ajustarMes", "ajustarDia", "ajustarHora", "ajustarMinuto" y "ajustarSegundo".

Los métodos "ajustarMes", "ajustarDia", "ajustarHora", "ajustarMinuto" y "ajustarSegundo" aplican una corrección a los valores estimados por "calcular", ya que dicho método opera sin las consideraciones necesarias a la hora de trabajar con múltiples unidades de tiempo.

La clase "ManejoDatos" se encarga de almacenar información relevante en un archivo de texto. Para ello utiliza su único método "aniadirArchivo".