**UNIVERSIDAD EAN**

**FACULTAD DE ESTUDIOS EN AMBIENTES VIRTUALES**

**DESARROLLO DE SOFTWARE**

**GUIA 2**

**AUTOR**

**JHOAN ALEXANDER CASTRO NAVARRETE**

**JULIAN DAVID ACOSTA CLAVIJO**

**JUAN FERNANDO SAENZ SANCHEZ**

**DOCENTE**

**COBO CAMPO LUIS ARMANDO**

**BOGOTÁ, D.C., AGOSTO DE 2021**

Métodos:

* buscarRangoImpuesto()

Se crearon objetos de tipo “RangoImpuesto” en el cual cada uno guarda los rangos y porcentajes. Luego se utiliza un bucle y se utiliza el método “contieneA()” del objeto rangos para determinar si el valor esta dentro de ese rango. Si el valor esta dentro de un cierto rango entonces se devuelve el rango que le corresponde a ese valor.

* calcularPago()

Se creo una variable “costoTotal” para guardar el impuesto base. Si el usuario selecciona algún descuento como “ProntoPago”, “Servicio Publico” o “Traslado de cuenta” entonces mediante unas condicionales se evalúa si alguno de estos descuentos está activo. En el caso de que algún descuento este activo entonces ese descuento será aplicado a “costoTotal”. Por otro lado, si ningún descuento es activado entonces solo se devolverá el impuesto sin descuento.

* buscarVehiculoMasCaro()

Se hizo creo una variable llamada “precioAlto” donde se guarda el precio del vehículo más caro. Luego se hizo uso de un bucle donde compara los precios de cada vehículo y mientras esto sucede “precioAlto” guarda el vehículo con mayor valor basado en las comparaciones. Finalmente devuelve el vehículo más caro.

* buscarVehiculoPorLinea()

La solución fue utilizar un bucle para comparar la variable “línea” con las líneas de los vehículos si hay una coincidencia entonces se devuelve el vehículo de dicha línea. Por otro lado, si no se encuentra la línea ingresada por el usuario entonces devolverá un valor nulo.

* buscarVehiculoPorMarca()

La solución fue utilizar un bucle para comparar la variable “marca” con las marcas de los vehículos si hay una coincidencia entonces se devuelve el vehículo de dicha línea. Por otro lado, si no se encuentra la línea ingresada por el usuario entonces devolverá un valor nulo.

* buscarVehiculoMasAntiguo()

Se soluciono utilizando una variable “año” en el que se guarda el año del vehículo. Luego se utiliza un bucle que recorre todos los vehículos comparándose unos con otros mientras la variable “año” guarda el vehículo con mayor antigüedad hasta que el bucle se detiene y devuelve el vehículo con mayor antigüedad.

* promedioPreciosVehiculos()

Se utilizo la variable “promedio” y se utilizó un bucle para sumar los precios de los vehículos en la variable “promedio” y dividir el resultado de acuerdo con la longitud del array “vehículos”. Finalmente se devuelve el promedio.

Enlace a repositorio en Github: <https://github.com/JULIAN-ACOSTA-C/G2_Act2_DSW>