

Trabajo Practico Nro. 4 Alejandro Nahuel

Bongioanni 2°C

Se realizo una aplicación para gestionar el ingreso y egreso de vehículos a un Estacionamiento. Este permitirá realizar ingresos de manera manual y a su vez simulará un ingreso automático al sistema.

The screenshot shows a web application interface for a parking system. The title bar at the top reads "Estacionamiento 24 Hs" with a close button on the right. The interface is divided into a left sidebar and a main content area. The sidebar has a blue background and contains a location pin icon, a toggle switch labeled "Habilitar Ingreso" which is currently turned on, and three input fields for hourly rates: "Importe de la Hora", "Auto:", "Camioneta:", and "Moto:". Below these fields is a blue padlock icon. At the bottom of the sidebar, a digital clock displays "16:42:46". The main content area has a light gray background and a large empty box labeled "Listado de Vehiculos:". At the bottom of the main area, a status bar shows "Capacidad: 256/Ocupados: 0" next to a progress bar.

Al iniciar el sistema, este no hará nada hasta que se habilite el ingreso de vehículos, para poder iniciar será necesario establecer los precios de la hora para cada tipo de vehículo, de lo contrario no se podrá habilitar el ingreso.

Cada vehículo que ingrese generara un Archivo a modo de ticket en una carpeta denominada factura en el escritorio de la PC, cuando se retire, adicionara a este archivo la hora de salida y el cargo por el estacionamiento. También guardar en la tabla “RegistroEstacionamiento” los datos del vehículo que se retira.

La solución cuenta con los siguientes proyectos:

- 1. Clases vehículos:** Esta se compone de 4 clases 1 de tipo abstracta y 3 selladas
 - a. Vehículo:** Es la clase base de la cual heredaran los diferentes tipos de vehículos, en ella se encuentra:
 - i. Atributos: Patente, Hora de Ingreso, Hora de Egreso.
 - ii. Métodos: **Validar Patente** de tipo privado que chequeará el ingreso de dicho dato, **Cargo de estacionamiento**, que en la clase base solo retornará la diferencia de horas, en las clases derivadas deberá adicionarse el importe de la hora según el tipo de vehículo. **Mostrar Datos** es protegido y muestra los datos de cada vehículo, estos se harán públicos desde la sobre escritura del método **ToString()**.
 - iii. Sobre cargas: Dos vehículos son iguales si tienen la misma patente
 - b. Automóvil, Moto Y camioneta:** Estas clases heredaran de vehículo y tienen un atributo de propio de clase (valorHora) que asigna un valor diferente a la hora de Estacionamiento a cada tipo de vehículo, este se podrá cambiar desde la Propiedad **ValorHora**. También tendrá:
 - i. Métodos: Validar Hora: comprobara que el valor de la hora ingresado desde la propiedad sea numérico y no sea negativo. **Cargo De estacionamiento** multiplica la cantidad de horas obtenidas desde la base por el valor hora de cada vehículo.
- 2. Clases Estacionamiento:** Estacionamiento, tiene un nombre, un listado de vehículos y una capacidad finita la cual podrá ser ocupada. Solo se puede realizar una única instancia de la clase Estacionamiento. Los vehículos se agregarán y eliminarán del estacionamiento a través de los operadores aritméticos (+/-). Estos operadores según sea la acción devolverán un mensaje a ser impreso en el ingreso o en la salida.
- 3. Clases BBDD:** VehiculosDAO, es la clase encargada de leer y guardar registros en la base de datos. Esta tiene un delegado y un evento actualizar, estos son partícipes de simular el ingreso de vehículos de forma automática.
- 4. Clases Archivos:** Tickets, debe de implementar los métodos Guardar y Leer de tipo texto y XML. Guardar Ticket de Texto, genera un nuevo documento en la carpeta Factura en el escritorio de la PC con el nombre de Patente y la hora de ingreso de cada Vehículo. Al retirarse un vehículo, modificara esta información adicionándole el cargo de la estadía.

5. Clases Excepciones: Posee las diferentes excepciones que el sistema pudiera emitir y controlar.

La siguiente Query posee las sentencias necesarias para generar la Base de datos y adicionar los registros para el funcionamiento del proyecto. Adicionalmente se adjunta una carpeta con estos datos.



Nombre	Fecha de modificacion	tipo	tamaño
ClasesArchivos	21/11/2020 15:32	Carpeta de archivos	
ClasesBBDD	21/11/2020 15:21	Carpeta de archivos	
ClasesEstacionamiento	21/11/2020 17:07	Carpeta de archivos	
ClasesExcepciones	21/11/2020 13:04	Carpeta de archivos	
ClasesVehiculos	21/11/2020 14:41	Carpeta de archivos	
ImagenesTP4	21/11/2020 13:04	Carpeta de archivos	
MItestUnitario	21/11/2020 13:04	Carpeta de archivos	
packages	21/11/2020 13:04	Carpeta de archivos	
Test	21/11/2020 17:13	Carpeta de archivos	
TestResults	21/11/2020 13:04	Carpeta de archivos	
TP_N4	21/11/2020 16:31	Carpeta de archivos	
Bongioanni.Alejandro.Div.2C.TP4.sln	21/11/2020 15:20	Visual Studio Solu...	5 KB
Base de Datos	21/11/2020 16:28	Carpeta de archivos	

En dicha carpeta también se encuentra un BK de la base de datos.