



## INSTRUMENTO DE EVALUACIÓN 02 JAVA

INFORMACIÓN GENERAL				
CODIGO PARA IDENTIFICACIÓN DEL INSTRUMENTO: ADSO-03-IE03				
FECHA DE APLICACIÓN: 02/10/2023				
PROGRAMA DE FORMACIÓN: TECNOLOGO EN ANALISIS Y DESARROLLO DE SOFTWARE.				
ID:				
CENTRO: Atención Sector Agropecuario				
RESULTADO DE APRENDIZAJE:				
<ul style="list-style-type: none"><li>Codificar el software de acuerdo con el diseño establecido.</li></ul>				
CRITERIOS DE EVALUACIÓN:				
<ul style="list-style-type: none"><li>Aplicar estándares de codificación.</li></ul>				
EVIDENCIA:	Desempeño	Producto	X	Conocimiento
NOMBRE DEL APRENDIZ:				
NOMBRE DEL INSTRUCTOR- TUTOR: OSCAR ANDRES LOAIZA PABON				

### ACTIVIDAD

En grupos de 4 aprendices, deben realizar dos aplicaciones:

- Aplicación de escritorio en JAVA, con conectividad a internet para consumo de API.
- API en PHP usando lenguaje nativo o algún framework.

---

#### REQUERIMIENTOS DE LA API EN PHP:

**Endpoint para crear parqueaderos:** Debe existir una ruta que permita a los administradores crear nuevos parqueaderos con la siguiente información:

- Nombre del parqueadero
- Dirección del parqueadero

**Endpoint para crear vendedores:** La API debe tener una ruta para que los administradores puedan agregar nuevos vendedores a los parqueaderos. Los datos requeridos son:

- Identificación del vendedor
- Nombre del vendedor
- Correo electrónico
- Contraseña

**Endpoint para asignar vendedores a parqueadero:** La API debe tener una ruta para que los administradores puedan asignar vendedores a los parqueaderos, un parqueadero puede tener asociado muchos vendedores y un vendedor solo puede estar asociado a un parqueadero. Los datos requeridos son:

- Id del parqueadero
- Id del vendedor

**Endpoints para RUD (Read, Update and Delete) de vendedores y parqueaderos:** Se deben crear las rutas para completar el CRUD de vendedores y parqueaderos. Se debe tener en cuenta que a los vendedores se le pueden deshabilitar el acceso a la aplicación por medio de un estado.

---

## REQUERIMIENTOS DE LA APLICACIÓN EN JAVA:

**Login:** sección de inicio de sesión para los usuarios Administrador y Vendedor. Se debe validar que los usuarios ingresen sus datos de acceso de forma correcta y que a su vez los usuarios se encuentren habilitados para el acceso. En caso de no permitirse el acceso se debe visualizar el respectivo mensaje de error.

**Usuario Administrador:** Para el administrador, la aplicación debe proporcionar las siguientes funcionalidades:

- Crear nuevos parqueaderos.
- Crear nuevos vendedores.
- Editar datos de vendedores.
- Editar datos de parqueaderos.
- Visualizar la lista de parqueaderos y vendedores.
- Visualizar vendedores sin asignación.
- Asignar vendedores a los parqueaderos existentes.

**Interfaz de Vendedor:** Para los vendedores, la aplicación debe permitir lo siguiente:

- Visualizar la información del parqueadero asignado.
- Definir la tarifa del servicio de parqueo para los diferentes tipos de vehículos (Camiones, Camionetas, Motos y Carros).
- Registrar nuevos vehículos en el sistema.
- Registrar la entrada de vehículos en el parqueadero.
- Al registrar la entrada del vehículo en el parqueadero, el sistema debe permitir imprimir el ticket con la información del vehículo para entregar al cliente, ese mismo ticket puede ser usado para facilitar el proceso de salida del vehículo.
- Registrar la salida de vehículos en el parqueadero, este proceso debe realizar el cálculo del costo total del servicio dada la cantidad de horas.
- Al registrar la salida o revisar el historial, la aplicación debe permitir al usuario realizar impresión del ticket o factura del servicio de parqueo.
- Ver el estado actual del parqueadero (Vehículos actuales).
- Buscar un vehículo por placa y mostrar el estado en pantalla, si el vehículo se encuentra ingresado mostrar el tiempo total hasta el momento de la consulta. Adicional mostrar el historial de registros.

---

## ENTREGA:

Para la entrega de los productos de software desarrollados, deben preparar un repositorio en GitHub. Cada aprendiz debe crear una rama con la nomenclatura **dev0X**, donde X es el número del desarrollador. El repositorio debe contener en total 5 ramas: 1 rama master y 4 rama devs.

Deben asociarme como colaborador, para realizar la revisión del trabajo realizado.

Email: [oaolaiza9@misena.edu.co](mailto:oaolaiza9@misena.edu.co)

**Fecha de entrega:** viernes 13 de octubre.