**EVALUACIÓN 3**

**ASIGNATURA: PGY2121 – DESARROLLO DE SOFTWARE**

|  |  |  |
| --- | --- | --- |
| **PUNTAJE MÁXIMO:** | **45 puntos** | **NOTA** |
| **PUNTAJE OBTENIDO** |  |
| **FECHA** |  |
| **DURACIÓN** | **120 minutos** |

**INDICACIONES GENERALES**

La nota 4,0 se obtiene logrando un 60% del puntaje total.

Modalidad: Resolución de un caso práctico. Integrantes: **1** o **2** personas.

**LEA ATENTAMENTE EL DOCUEMENTO ANTES DE INICIAR EL DESARROLLO DE LA SOLUCIÓN.**

**ENUNCIADO.**

Dado el buen funcionamiento del proyecto del Taller Mecánico, el dueño ha decidido agregar nuevas funcionalidades a la aplicación para mejorar las actuales características.

Los nuevos requerimientos que debe cumplir la aplicación son las siguientes:

* Utilizar lenguaje Java.
* Ambiente escritorio.
* Arquitectura **MVC** visto en clases.
* Motor de Base de datos **MySql**.
* Nombre de la base de datos: **TALLERMECANICO**.
* Tablas: VEHICULO, PERSONA, MARCA, CARGO, COMUNA Y ESPECIALIDAD.
* Agregar 5 filas a cada tabla, para verificar el funcionamiento de la aplicación.
* Clases abstractas: Vehículo y Persona.
* Subclases: Motocicleta, Especialidad, Empleado y Cliente.
* Clases para aplicar **colaboración**: Cargo y Comuna.
* Crear el **CRUD** para cada una de las tablas, usando todas las clases mencionadas.
* Empleados tienen sueldo y opcionalmente especialidad.
* Utilizar ***MENU*** para acceder a las distintas opciones.

Para cada formulario, validar que la información que el usuario ingresa cumpla con las siguientes condiciones:

1. Rut: solo números.
2. Digito: solo 1 carácter y que estén en los rangos del 1 al 9, más el cero y k.
3. Cantidad de caracteres de los textos, según lo señalado por cada tabla.
4. Sueldo solo para empleados y debe estar en el rango de 400.000 a 2.500.000.
5. Las FK requeridas en las tablas, deben ser seleccionadas, por el usuario, desde una lista (jComboBox).
6. Usar radios para diferenciar el ingreso de cliente o empleado.
7. Usar checkbox para el campo habilitado.
8. Otras que considere apropiadas para el caso planteado.
9. Todos los datos son obligatorios según el objeto/clase creado.
10. Para una correcta funcionalidad de la aplicación debe:
    * Agregar los eventos necesarios.
    * Aplicar mejoras que permitan facilitar el uso de la app, por parte del usuario.

**Usar SOLO los códigos y reglas vistas en clases. Plagios serán evaluados con nota mínima.**

Script a utilizar:

CREATE TABLE Marca

(

idMarca int primary key auto\_increment,

nombre varchar(50) not null,

UNIQUE(nombre)

);

CREATE TABLE Cargo

(

idCargo int primary key auto\_increment,

nombre varchar(50) not null,

UNIQUE(nombre)

);

CREATE TABLE Especialidad

(

idEspecialidad int primary key auto\_increment,

nombre varchar(50) not null,

habilitado int not null,

UNIQUE(nombre)

);

CREATE TABLE Comuna

(

idComuna int primary key auto\_increment,

nombre varchar(50) not null,

UNIQUE(nombre)

);

CREATE TABLE Persona

(

idPersona int primary key auto\_increment,

idCargo int null,

idEspecialidad int null,

idComuna int not null,

rut int not null,

digito varchar(1) not null,

nombre varchar(50) not null,

apellido varchar(50) not null,

sueldo int not null,

esCliente int not null,

habilitado int not null,

UNIQUE(rut),

FOREIGN KEY(idCargo) REFERENCES Cargo(idCargo),

FOREIGN KEY(idEspecialidad) REFERENCES Especialidad(idEspecialidad),

FOREIGN KEY(idComuna) REFERENCES Comuna(idComuna)

);

CREATE TABLE Vehiculo

(

idVehiculo int primary key auto\_increment,

idPersona int not null,

idMarca int not null,

patente varchar(6) not null,

modelo varchar(100) not null,

litrosMaletero int null,

cantidadPuerta int null,

largoManillar int null,

estilo varchar(100) null,

UNIQUE(patente),

FOREIGN KEY(idPersona) REFERENCES Persona(idPersona),

FOREIGN KEY(idMarca) REFERENCES Marca(idMarca)

);