

## Trabajo Práctico N° 4

En la cuarta entrega del TP realizaremos la programación de todos los scripts de los objetos necesarios para poder dar funcionalidad al sistema.

**Objetivo:** Comprender y profundizar el funcionamiento de la Programación Orientada a Objetos utilizando el lenguaje de programación PHP.

**Introducción:** El departamento de servicio técnico de una empresa requiere de un sistema para mantener organizada la distribución de su equipamiento, así como también el registro de direcciones de IP. La empresa cuenta con diferentes oficinas las cuales se distribuyen con equipos de computación, éstos equipos se encuentran inventariados con una nomenclatura formada por las cuatro primeras letras del nombre de la oficina más un número consecutivo e independiente a cada oficina. Cada equipo además del código de inventario están asociado a un empleado.

**Observación:** Dentro de la clase 8 se encuentra el la base de datos y los scripts necesarios para el funcionamiento de la aplicación bajo el nombre de **Trabajo Práctico N° 4 (Anexo)**.

### Puntos a realizar en la entrega:

1. Crear una clase la cual administre la información de los empleados de la empresa.
2. Crear una clase la cual administre la información de los equipos de la empresa.
3. Crear las clases para administrar la información de las oficinas y los equipos que la conforman.

### Base de datos

#### Tabla empleado:

- idempleado int(11) NOT NULL AUTO\_INCREMENT PRIMARY KEY.
- apellido varchar(60) NOT NULL.
- nombre varchar(60) NOT NULL.
- legajo int(11) NOT NULL.

#### Tabla equipo:

- idequipo int(11) NOT NULL AUTO\_INCREMENT PRIMARY KEY.
- idempleado int(11) NOT NULL (Relación a la tabla empleado).
- descripcion varchar(60) NOT NULL.
- ip varchar(15) NOT NULL.
- mac varchar(17) NOT NULL.

Tabla oficina:

- idoficina int(11) NOT NULL AUTO\_INCREMENT PRIMARY KEY.
- nombre varchar(100) NOT NULL.

Tabla oficinaequipo:

- idoficinaequipo int(11) NOT NULL AUTO\_INCREMENT PRIMARY KEY.
- idoficina int(11) NOT NULL (Relación a la tabla oficina).
- idequipo int(11) NOT NULL (Relación a la tabla equipo).
- fecha timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP.

## Desarrollo

### 1. Crear una clase la cual administre la información de los empleados de la empresa.

Crear un script para la clase Empleado dentro de la carpeta /sge/empleado/includes/. La clase debe contener los atributos privados para el id de empleado, apellido, nombre y legajo, y los métodos getter y setter para cada uno de los atributos. Además deberá contar con los siguientes métodos los cuales utilizan el objeto BaseDeDatos ubicado en /sge/includes/php/:

- **insertar:** Este método debe permitir insertar un nuevo registro en la tabla empleado utilizando los atributos del objeto para completar la información, también debe obtener el id generado para este registro y asignarlo al atributo id de empleado.
- **modifica:** Este método debe actualizar el registro cuya clave sea igual al id de empleado.
- **eliminar:** Este método debe eliminar el registro cuya clave sea igual al id de empleado.
- **buscar:** Este método recibe como parámetro el id del empleado el cual se desea cargar la información desde la base de datos y asignarlos a los atributos del objeto.
- **getAll:** Este método deberá devolver un array de objetos del tipo Empleado con la información de cada registro.

### 2. Crear una clase la cual administre la información de los equipos de la empresa.

Crear un script para la clase Equipo dentro de la carpeta /sge/equipo/includes/. La clase debe contener los atributos privados para el id de equipo, descripción, dirección de IP, dirección de MAC y uno el cual debe contener un objeto del tipo Empleado, y los métodos getter y setter para cada uno de los atributos, el método setter para empleado debe recibir un objeto del tipo Empleado como parámetro. Además deberá contar con los siguientes métodos los cuales utilizan el objeto BaseDeDatos ubicado en /sge/includes/php/:

- **insertar:** Este método debe permitir insertar un nuevo registro en la tabla equipo utilizando los atributos del objeto para completar la información, también debe obtener el id generado para este registro y asignarlo al atributo id de equipo.
- **modifica:** Este método debe actualizar el registro cuya clave sea igual al id de equipo.
- **eliminar:** Este método debe eliminar el registro cuya clave sea igual al id de equipo.
- **buscar:** Este método recibe como parámetro el id del equipo el cual se desea cargar la información desde la base de datos y asignarlos a los atributos del objeto.
- **getAll:** Este método deberá devolver un array de objetos del tipo Equipo con la información de cada registro.

### 3. Crear las clases para administrar la información de las oficinas y los equipos que la conforman.

Crear un script para la clase OficinaEquipo dentro de la carpeta /sge/oficina/includes/. La clase debe contener los atributos privados para el id de la relación entre oficina y equipo, el id de la oficina y uno el cual debe contener un objeto del tipo Equipo, y los métodos getter y setter para cada uno de los atributos, el método setter para equipo debe recibir un objeto del tipo Equipo como parámetro. Además deberá contar con los siguientes métodos los cuales utilizan el objeto

BaseDeDatos ubicado en /sge/includes/php/:

- **insertar:** Este método debe permitir insertar un nuevo registro en la tabla oficinaequipo utilizando los atributos para completar la información, también debe obtener el id generado para este registro y asignarlo al atributo id de la relación entre oficina y equipo.
- **eliminar:** Este método debe eliminar el registro cuya clave sea igual al id de la relación entre oficina y equipo.
- **buscar:** Este método recibe como parámetro el id de la relación entre oficina y equipo el cual se desea cargar la información desde la base de datos y asignarlos a los atributos del objeto.
- **buscarPorIdOficina:** Este método recibe como parámetro el id de la oficina y debe buscar todos los registros de la relación. Deberá devolver un array de objetos del tipo OficinaEquipo con la información de cada registro obtenido.
- **getAll:** Este método deberá devolver un array de objetos del tipo OficinaEquipo con la información de cada registro.

Crear un script para la clase Oficina dentro de la carpeta /sge/oficina/includes/. La clase debe contener los atributos privados para el id de la oficina, nombre y uno el cual debe contener un array de objeto del tipo OficinaEquipo, los métodos getter y setter para id de oficina y nombre, y solo el método getter para el atributo que contiene el array de objetos del tipo OficinaEquipo. Además deberá contar con los siguientes métodos los cuales algunos utilizan el objeto BaseDeDatos ubicado en /sge/includes/php/:

- **insertar:** Este método debe permitir insertar un nuevo registro en la tabla oficina utilizando el atributo nombre para completar la información, también debe obtener el id generado para este registro y asignarlo al atributo id de oficina.
- **modifica:** Este método debe actualizar el registro cuya clave sea igual al id de oficina.
- **eliminar:** Este método debe eliminar el registro cuya clave sea igual al id de oficina.
- **buscar:** Este método recibe como parámetro el id de equipo oficina de la cual se desea cargar la información desde la base de datos y asignarlos a los atributos del objeto.
- **getAll:** Este método deberá devolver un array de objetos del tipo Oficina con la información de cada registro.
- **getCantidadEquipos:** Este método deberá devolver la cantidad de equipos que tiene la oficina.
- **agregarEquipo:** Este método debe recibir un objeto del tipo OficinaEquipo como parámetro y agregarlo al atributo que contiene el array de objetos del tipo OficinaEquipo, también debe verificar que el equipo no se repita en el array. Una forma de verificar que no se repita de manera fácil, es utilizar el id del objeto que se recibe como índice del array.
- **quitarEquipo:** Este método de recibir un objeto del tipo OficinaEquipo y quitarlo del atributo que contiene el array con los con los objetos del mismo tipo.



## Entrega

La cuarta entrega deberá contener el proyecto con todos los scripts modificados, y deberá ser cargado a la plataforma como un único archivo comprimido en formato \*.rar o \*.tar.gz.