

**PROGRAMACION ORIENTADA A OBJETOS I**

**SEMANA 7**

Jorge Peñaloza Guerra

15 de mayo de 2023

ingeniería en informática

# **DESARROLLO**

Caso

En la empresa Qstar.CA, el departamento de TI, necesita expandir su red de notebook para un total de 30 empleados, para ello se está proyectando establecer un listado de equipamiento de hardware. Considerando que deben llevar el control de la asignación de los equipos a cada empleado. En vista de que el listado se lleva de forma manual, la empresa toma la decisión de contratarte como técnico en informática para la elaboración de una aplicación que permita llevar el control del listado de los empleados y de sus equipos.

Por eso te solicitan que se debe registrar todos los datos del notebook que son: serial, modelo, tipo de procesador, marca y tipo de sistema operativo. En vista de que se quiere llevar un control de la asignación del equipo, se debe registrar la fecha y su ubicación dentro de la empresa, también se deben registrar los datos del empleado que son: código del empleado, sus nombres, apellidos, cargo y correo

**Según lo planteado elabora lo siguiente:**

1. Construye un programa en java que contenga la clase principal y sus componentes (nombre, constructores, atributos y métodos), que deben estar presente el sistema de control de equipos.

En la figura 1 se observa el diagrama UML definitivo de la aplicación y en código de programación del 1 al 5 se encuentran todos los códigos de programación necesarios para el funcionamiento de la aplicación

Figura 1

Diagrama UML Sistema de control de equipos

Captura de pantalla de un celular con letras

Descripción generada automáticamente

Fuente: Elaboración propia

Código de programación 1

Clase: SistemaControlDeEquipos

*/\*\*  
 \* <h1>Sistema de Control de Equipos</h1>  
 \*  
 \* <p>Este programa administra los equipos, y sus asignaciones a los empleados  
 \* de la empresa Qstar.CA. Permite registrar y actualizar la información de los  
 \* equipos y empleados, así como asignar un equipo a un empleado.</p>  
 \*  
 \* <p><b>Nota:</b> Este programa es un ejemplo y puede ser mejorado y adaptado  
 \* según las necesidades específicas de la empresa.</p>  
 \*  
 \* @author Jorge Peñaloza  
 \* @version 1.0  
 \* @since 2023-05-13  
 \* @lastModified 2023-05-15  
 \* @return vacio  
 \*/*import java.util.Scanner;  
public class SistemaControlDeEquipos {  
 private Asignacion[] asignaciones; // Declaración del array de asignaciones  
 private int numAsignaciones; // Contador del número de asignaciones registradas  
  
 public SistemaControlDeEquipos() {  
 this.asignaciones = new Asignacion[100]; // Tamaño inicial del array de asignaciones  
 this.numAsignaciones = 0; // Inicializar el contador en 0  
 }  
 public static void main(String[] args) {  
 // Crear una instancia de SistemaControlDeEquipos  
 SistemaControlDeEquipos sistema = new SistemaControlDeEquipos();  
 // Menú principal  
 Scanner scanner = new Scanner(System.*in*);  
 int opcion;  
 do {  
 System.*out*.println("----- MENÚ PRINCIPAL -----");  
 System.*out*.println("1. Registrar nueva asignación");  
 System.*out*.println("2. Buscar asignación por número de serie del equipo");  
 System.*out*.println("3. Buscar asignación por código del empleado");  
 System.*out*.println("4. Actualizar estado de una asignación");  
 System.*out*.println("5. Generar reporte de asignaciones");  
 System.*out*.println("0. Salir");  
 System.*out*.print("Ingrese su opción: ");  
 opcion = scanner.nextInt();  
  
 switch (opcion) {  
 case 1:  
 sistema.registrarNuevaAsignacion();  
 break;  
 case 2:  
 System.*out*.print("Ingrese el número de serie del equipo: ");  
 String serialEquipo = scanner.next();  
 Asignacion asignacionPorSerial = sistema.buscarAsignacionPorNotebook(serialEquipo);  
 if (asignacionPorSerial != null) {  
 System.*out*.println("Asignación encontrada:");  
 System.*out*.println("Empleado: " + asignacionPorSerial.getEmpleado().getNombres() + " " + asignacionPorSerial.getEmpleado().getApellidos());  
 System.*out*.println("Equipo: " + asignacionPorSerial.getEquipo().getModelo() + " (" + asignacionPorSerial.getEquipo().getSerial() + ")");  
 System.*out*.println("Fecha de asignación: " + asignacionPorSerial.getFechaAsignacion());  
 System.*out*.println("Ubicación: " + asignacionPorSerial.getUbicacion());  
 } else {  
 System.*out*.println("No se encontró ninguna asignación con el número de serie " + serialEquipo);  
 }  
  
 System.*out*.println();  
 break;  
 case 3:  
 System.*out*.print("Ingrese el código del empleado: ");  
 String codigoEmpleado = scanner.next();  
 Asignacion asignacionPorEmpleado = sistema.buscarAsignacionPorEmpleado(codigoEmpleado);  
 if (asignacionPorEmpleado != null) {  
 System.*out*.println("Asignación encontrada:");  
 System.*out*.println("Empleado: " + asignacionPorEmpleado.getEmpleado().getNombres() + " " + asignacionPorEmpleado.getEmpleado().getApellidos());  
 System.*out*.println("Equipo: " + asignacionPorEmpleado.getEquipo().getModelo() + " (" + asignacionPorEmpleado.getEquipo().getSerial() + ")");  
 System.*out*.println("Fecha de asignación: " + asignacionPorEmpleado.getFechaAsignacion());  
 System.*out*.println("Ubicación: " + asignacionPorEmpleado.getUbicacion());  
 } else {  
 System.*out*.println("No se encontró ninguna asignación con el código de empleado " + codigoEmpleado);  
 }  
 break;  
 case 4:  
 // Pedir los datos de la asignación y el nuevo estado  
 // Actualizar el estado de la asignación llamando al método actualizarEstadoAsignacion()  
 break;  
 case 5:  
 sistema.generarReporteAsignaciones();  
 break;  
 case 0:  
 System.*out*.println("Saliendo del programa...");  
 break;  
 default:  
 System.*out*.println("Opción inválida. Intente nuevamente.");  
 break;  
 }  
 System.*out*.println();  
 } while (opcion != 0);  
  
 scanner.close();  
 }  
 */\*\*  
 \* Registra una nueva asignación de un equipo a un empleado.  
 \*/* public void registrarNuevaAsignacion() {  
 // Pedir los datos necesarios para la nueva asignación  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Ingrese los datos de la nueva asignación:");  
  
 // Pedir los datos del empleado  
 System.*out*.print("Código del empleado: ");  
 String codigoEmpleado = scanner.nextLine();  
 System.*out*.print("Nombres del empleado: ");  
 String nombresEmpleado = scanner.nextLine();  
 System.*out*.print("Apellidos del empleado: ");  
 String apellidosEmpleado = scanner.nextLine();  
 System.*out*.print("Cargo del empleado: ");  
 String cargoEmpleado = scanner.nextLine();  
 System.*out*.print("Correo del empleado: ");  
 String correoEmpleado = scanner.nextLine();  
  
 // Crear un objeto Empleado con los datos ingresados  
 Empleado empleado = new Empleado(codigoEmpleado, nombresEmpleado, apellidosEmpleado, cargoEmpleado, correoEmpleado);  
  
 // Pedir los datos del equipo  
 System.*out*.print("Número de serie del equipo: ");  
 String serialEquipo = scanner.nextLine();  
 System.*out*.print("Modelo del equipo: ");  
 String modeloEquipo = scanner.nextLine();  
 System.*out*.print("Tipo de procesador del equipo: ");  
 String tipoProcesadorEquipo = scanner.nextLine();  
 System.*out*.print("Marca del equipo: ");  
 String marcaEquipo = scanner.nextLine();  
 System.*out*.print("Tipo de sistema operativo del equipo: ");  
 String tipoSistemaOperativoEquipo = scanner.nextLine();  
  
 // Crear un objeto Equipo con los datos ingresados  
 Equipo equipo = new Equipo(serialEquipo, modeloEquipo, tipoProcesadorEquipo, marcaEquipo, tipoSistemaOperativoEquipo);  
  
 // Pedir los datos de la asignación  
 System.*out*.print("Fecha de asignación: ");  
 String fechaAsignacion = scanner.nextLine();  
 System.*out*.print("Ubicación del equipo: ");  
 String ubicacionEquipo = scanner.nextLine();  
  
 // Crear una nueva asignación con los objetos Empleado y Equipo, y los demás datos ingresados  
 Asignacion asignacion = new Asignacion(empleado, equipo, fechaAsignacion, ubicacionEquipo);  
  
  
 // Agregar la asignación al array de asignaciones  
 asignaciones[numAsignaciones] = asignacion;  
 numAsignaciones++;  
  
 System.*out*.println("Nueva asignación registrada correctamente.");  
 System.*out*.println();  
  
 //scanner.close();  
 }  
  
 */\*\*  
 \* Busca una asignación por el número de serie del equipo.  
 \*  
 \* @param serial Número de serie del equipo.  
 \* @return Asignación encontrada o null si no se encontró ninguna asignación.  
 \*/* public Asignacion buscarAsignacionPorNotebook(String serial) {  
 // Recorrer el array de asignaciones para buscar la asignación por el número de serie del equipo  
 for (int i = 0; i < numAsignaciones; i++) {  
 if (asignaciones[i].getEquipo().getSerial().equals(serial)) {  
 return asignaciones[i]; // Retornar la asignación encontrada  
 }  
 }  
 return null; // No se encontró ninguna asignación con el número de serie dado  
 }  
  
 */\*\*  
 \* Busca una asignación por el código del empleado.  
 \*  
 \* @param codigoEmpleado Código del empleado.  
 \* @return Asignación encontrada o null si no se encontró ninguna asignación.  
 \*/* public Asignacion buscarAsignacionPorEmpleado(String codigoEmpleado) {  
 // Recorrer el array de asignaciones para buscar la asignación por el código del empleado  
 for (int i = 0; i < numAsignaciones; i++) {  
 if (asignaciones[i].getEmpleado().getCodigoEmpleado().equals(codigoEmpleado)) {  
 return asignaciones[i]; // Retornar la asignación encontrada  
 }  
 }  
 return null; // No se encontró ninguna asignación con el código de empleado dado  
 }  
  
 */\*\*  
 \* Actualiza el estado de una asignación.  
 \*  
 \* @param asignacion Asignación a actualizar.  
 \* @param nuevoEstado Nuevo estado de la asignación.  
 \*/*

public void actualizarEstadoAsignacion(Asignacion asignacion, String nuevoEstado) {  
 // Lógica para actualizar el estado de una asignación  
 }  
  
 */\*\*  
 \* Genera un reporte de todas las asignaciones realizadas.  
 \*/* public void generarReporteAsignaciones() {  
 System.*out*.println("Reporte de Asignaciones:");  
 System.*out*.println("-----------------------");  
  
 if (numAsignaciones == 0) {  
 System.*out*.println("No hay asignaciones registradas.");  
 } else {  
 for (int i = 0; i < numAsignaciones; i++) {  
 Asignacion asignacion = asignaciones[i];  
 System.*out*.println("Asignación " + (i + 1) + ":");  
 System.*out*.println("Empleado: " + asignacion.getEmpleado().getNombres() + " " + asignacion.getEmpleado().getApellidos());  
 System.*out*.println("Equipo: " + asignacion.getEquipo().getModelo() + " (" + asignacion.getEquipo().getSerial() + ")");  
 System.*out*.println("Fecha de asignación: " + asignacion.getFechaAsignacion());  
 System.*out*.println("Ubicación: " + asignacion.getUbicacion());  
 System.*out*.println();  
 }  
 }  
 }  
}

Fuente: Elaboración propia

Código de programación 2

Clase: Persona

*/\*\*  
 \* La clase Persona representa a una persona en general.  
 \* Contiene los atributos comunes a todas las personas, como nombres y apellidos.  
 \*  
 \* @author Jorge Peñaloza  
 \* @version 1.0  
 \* @since 13/05/2023  
 \*/*public class Persona {  
 private String nombres;  
 private String apellidos;  
  
 */\*\*  
 \* Constructor de la clase Persona.  
 \*  
 \* @param nombres Los nombres de la persona.  
 \* @param apellidos Los apellidos de la persona.  
 \*/* public Persona(String nombres, String apellidos) {  
 this.nombres = nombres;  
 this.apellidos = apellidos;  
 }  
  
 */\*\*  
 \* Obtiene los nombres de la persona.  
 \*  
 \* @return Los nombres de la persona.  
 \*/* public String getNombres() {  
 return nombres;  
 }  
  
 */\*\*  
 \* Establece los nombres de la persona.  
 \*  
 \* @param nombres Los nombres de la persona.  
 \*/* public void setNombres(String nombres) {  
 this.nombres = nombres;  
 }  
  
 */\*\*  
 \* Obtiene los apellidos de la persona.  
 \*  
 \* @return Los apellidos de la persona.  
 \*/* public String getApellidos() {  
 return apellidos;  
 }  
  
 */\*\*  
 \* Establece los apellidos de la persona.  
 \*  
 \* @param apellidos Los apellidos de la persona.  
 \*/* public void setApellidos(String apellidos) {  
 this.apellidos = apellidos;  
 }  
}

Fuente: Elaboración propia

Código de programación 3

Clase: Empleado

*/\*\*  
\* La clase Empleado representa a un empleado en la empresa.  
\* Hereda los atributos y métodos de la clase Persona y agrega atributos específicos para los empleados, como código de empleado, cargo y correo.  
\*  
\* @author Jorge Peñaloza  
\* @version 1.0  
\* @since 13/05/2023  
\*/*public class Empleado extends Persona {  
 private String codigoEmpleado;  
 private String cargo;  
 private String correo;  
  
 */\*\*  
 \* Constructor de la clase Empleado.  
 \*  
 \* @param nombres Los nombres del empleado.  
 \* @param apellidos Los apellidos del empleado.  
 \* @param codigoEmpleado El código del empleado.  
 \* @param cargo El cargo del empleado.  
 \* @param correo El correo del empleado.  
 \*/* public Empleado(String nombres, String apellidos, String codigoEmpleado, String cargo, String correo) {  
 super(nombres, apellidos);  
 this.codigoEmpleado = codigoEmpleado;  
 this.cargo = cargo;  
 this.correo = correo;  
 }  
  
 */\*\*  
 \* Obtiene el código del empleado.  
 \*  
 \* @return El código del empleado.  
 \*/* public String getCodigoEmpleado() {  
 return codigoEmpleado;  
 }  
  
 */\*\*  
 \* Establece el código del empleado.  
 \*  
 \* @param codigoEmpleado El código del empleado.  
 \*/* public void setCodigoEmpleado(String codigoEmpleado) {  
 this.codigoEmpleado = codigoEmpleado;  
 }  
  
 */\*\*  
 \* Obtiene el cargo del empleado.  
 \*  
 \* @return El cargo del empleado.  
 \*/* public String getCargo() {  
 return cargo;  
 }  
  
 */\*\*  
 \* Establece el cargo del empleado.  
 \*  
 \* @param cargo El cargo del empleado.  
 \*/* public void setCargo(String cargo) {  
 this.cargo = cargo;  
 }  
  
 */\*\*  
 \* Obtiene el correo del empleado.  
 \*  
 \* @return El correo del empleado.  
 \*/* public String getCorreo() {  
 return correo;  
 }  
  
 */\*\*  
 \* Establece el correo del empleado.  
 \*  
 \* @param correo El correo del empleado.  
 \*/* public void setCorreo(String correo) {  
 this.correo = correo;  
 }  
}

Fuente: Elaboración propia

Código de programación 4

Clase: Equipo

*/\*\*  
 \* La clase Equipo representa un notebook en la empresa Qstar.CA.  
 \* Cada equipo tiene un número de serie, modelo, tipo de procesador, marca, tipo de sistema operativo, fecha de asignación y ubicación.  
 \*  
 \* @author Jorge Peñaloza  
 \* @version 1.0  
 \* @since 13/05/2023  
 \*/*public class Equipo {  
  
 // Atributos de la clase Equipo  
 private String serial;  
 private String modelo;  
 private String tipoProcesador;  
 private String marca;  
 private String tipoSistemaOperativo;  
 private String fechaAsignacion;  
 private String ubicacion;  
  
 */\*\*  
 \* Constructor de la clase Equipo.  
 \*  
 \* @param serial Número de serie del equipo.  
 \* @param modelo Modelo del equipo.  
 \* @param tipoProcesador Tipo de procesador del equipo.  
 \* @param marca Marca del equipo.  
 \* @param tipoSistemaOperativo Tipo de sistema operativo del equipo.  
 \*/* //Equipo equipo1 = new Equipo("12345", "Modelo 2023", "Intel Core i7", "Dell", "Windows 10");  
 public Equipo(String serial, String modelo, String tipoProcesador, String marca, String tipoSistemaOperativo) {  
 this.serial = serial;  
 this.modelo = modelo;  
 this.tipoProcesador = tipoProcesador;  
 this.marca = marca;  
 this.tipoSistemaOperativo = tipoSistemaOperativo;  
  
 }  
  
 // Métodos get y set para cada atributo  
  
 public String getSerial() {  
 return serial;  
 }  
  
 public void setSerial(String serial) {  
 this.serial = serial;  
 }  
  
 public String getModelo() {  
 return modelo;  
 }  
  
 public void setModelo(String modelo) {  
 this.modelo = modelo;  
 }  
  
 public String getTipoProcesador() {  
 return tipoProcesador;  
 }  
  
 public void setTipoProcesador(String tipoProcesador) {  
 this.tipoProcesador = tipoProcesador;  
 }  
  
 public String getMarca() {  
 return marca;  
 }  
  
 public void setMarca(String marca) {  
 this.marca = marca;  
 }  
  
 public String getTipoSistemaOperativo() {  
 return tipoSistemaOperativo;  
 }  
  
 public void setTipoSistemaOperativo(String tipoSistemaOperativo) {  
 this.tipoSistemaOperativo = tipoSistemaOperativo;  
 }  
}

Fuente: Elaboración propia

Código de programación 5

Clase: Asignacion

*/\*\*  
 \* La clase Asignacion representa una asignación de un Equipo a un Empleado en la empresa Qstar.CA.  
 \* Cada asignación tiene un empleado, un equipo, una fecha de asignación y una ubicación.  
 \*  
 \* @author Jorge Peñaloza  
 \* @version 1.0  
 \* @since 13/05/2023  
 \*/*

public class Asignacion {  
  
 */\*\*  
 \* El empleado al que se le asigna el equipo.  
 \*/* private Empleado empleado;  
  
 */\*\*  
 \* El equipo que se asigna al empleado.  
 \*/* private Equipo equipo;  
  
 */\*\*  
 \* Fecha en la que se realiza la asignación del equipo.  
 \*/* private String fechaAsignacion;  
  
 */\*\*  
 \* Ubicación del equipo dentro de la empresa.  
 \*/* private String ubicacion;  
  
 */\*\*  
 \* Constructor de la clase Asignacion.  
 \*  
 \* @param empleado El empleado al que se le asigna el equipo.  
 \* @param equipo El equipo que se asigna al empleado.  
 \* @param fechaAsignacion Fecha de asignación del equipo.  
 \* @param ubicacion Ubicación del equipo dentro de la empresa.  
 \*/*

public Asignacion(Empleado empleado, Equipo equipo, String fechaAsignacion, String ubicacion) {  
 this.empleado = empleado;  
 this.equipo = equipo;  
 this.fechaAsignacion = fechaAsignacion;  
 this.ubicacion = ubicacion;  
 }  
  
 */\*\*  
 \* Retorna el empleado al que se le asignó el equipo.  
 \*  
 \* @return Empleado al que se le asignó el equipo.  
 \*/* public Empleado getEmpleado() {  
 return empleado;  
 }  
  
 */\*\*  
 \* Establece el empleado al que se le asigna el equipo.  
 \*  
 \* @param empleado Empleado al que se le asigna el equipo.  
 \*/* public void setEmpleado(Empleado empleado) {  
 this.empleado = empleado;  
 }  
  
 */\*\*  
 \* Retorna el equipo que se asignó al empleado.  
 \*  
 \* @return Equipo que se asignó al empleado.  
 \*/* public Equipo getEquipo() {  
 return equipo;  
 }  
  
 */\*\*  
 \* Establece el equipo que se asigna al empleado.  
 \*  
 \* @param equipo Equipo que se asigna al empleado.  
 \*/* public void setEquipo(Equipo equipo) {  
 this.equipo = equipo;  
 }  
  
 */\*\*  
 \* Retorna la fecha en que se realizó la asignación.  
 \*  
 \* @return Fecha en que se realizó la asignación.  
 \*/* public String getFechaAsignacion() {  
 return fechaAsignacion;  
 }  
  
 */\*\*  
 \* Establece la fecha en que se realiza la asignación.  
 \*  
 \* @param fechaAsignacion Fecha en que se realiza la asignación.  
 \*/* public void setFechaAsignacion(String fechaAsignacion) {  
 this.fechaAsignacion = fechaAsignacion;  
 }  
  
 */\*\*  
 \* Retorna la ubicación del equipo dentro de la empresa.  
 \*  
 \* @return Ubicación del equipo dentro de la empresa.  
 \*/* public String getUbicacion() {  
 return ubicacion;  
 }  
  
 */\*\*  
 \* Establece la ubicación del equipo dentro de la empresa.  
 \*  
 \* @param ubicacion Ubicación del equipo dentro de la empresa.  
 \*/* public void setUbicacion(String ubicacion) {  
 this.ubicacion = ubicacion;  
 }  
  
}

Fuente: Elaboración propia

2. Utiliza accesadores y mutadores de clase, para definir atributos dentro de una clase en el sistema de control de equipos.

En las figuras 2 y 3 se pueden ver ejemplo de como se utilizaron, pero en las clases Equipo, Persona, empleado y asignación está completamente implementado.

Figura 2

Ejemplo de uso de accesador

Texto

Descripción generada automáticamente

Fuente: Elaboración propia

Figura 3

Ejemplo de uso de mutador

Texto

Descripción generada automáticamente

Fuente: Elaboración propia

3. Emplea el método de herencia en java para la reutilización del código y desarrollar la aplicación con el fin de darle una solución al caso planteado

En la figura 1 se puede ver el concepto de herencia en la clase Empleado heredando métodos y atributos de la clase Persona con diagrama UML, y en la figura 4 se puede apreciar la implementación en JAVA

Figura 4

Ejemplo de uso de herencia



Fuente: Elaboración propia

**REFERENCIAS BIBLIOGRÁFICAS**

IACC. (2022). Programación Orientada a Objetos con Java. En *Programación orientada a objetos I. Semana 7.*