

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Computación

Lenguajes de Programación

Tarea Programada #4

Kevin Walsh Muñoz

Jonathan Rojas Vargas

Sede San Carlos

15/11/2015



Síntesis del problema

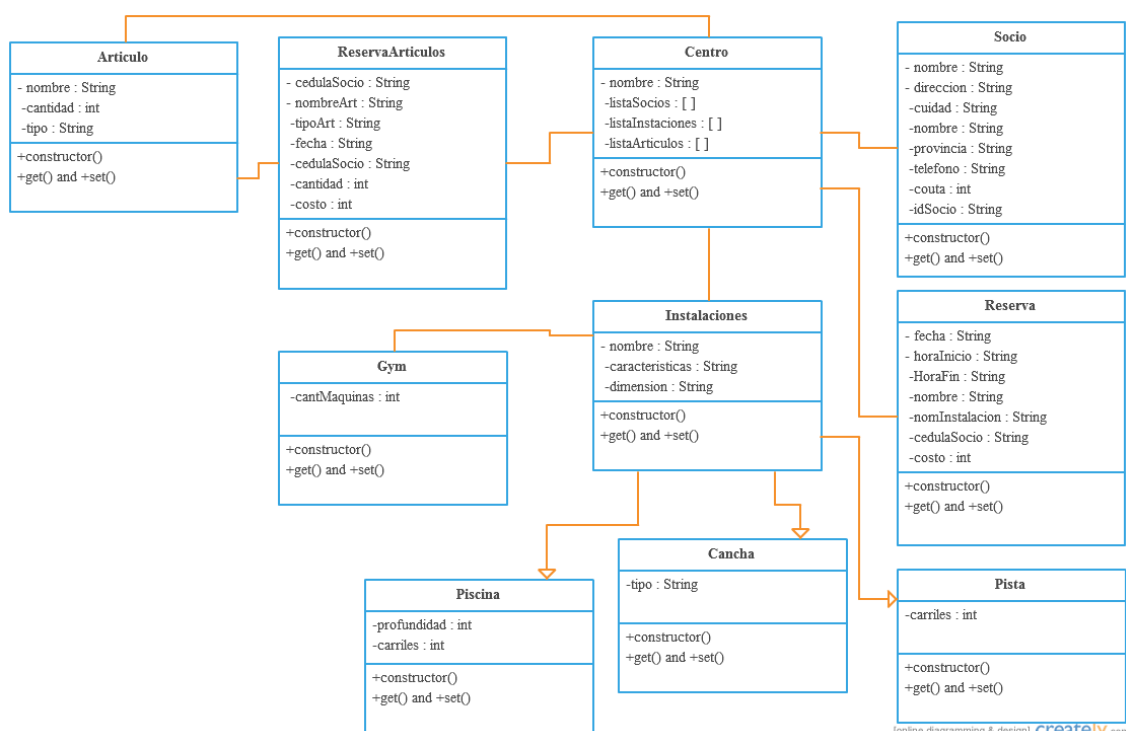
Es necesario implementar un sistema que permita administrar un centro deportivo, el cual posee instalaciones (piscinas, canchas, pistas y gimnasio). Este centro deportivo es utilizado por socios. Los cuales podrán realizar reservaciones de las instalaciones y además artículos deportivos que utilizaran en dicha instalación. Estos artículos contarán con sus propias características.

Cuando se realice una reservación será necesario almacenar varios datos (nombre de la instalación, fecha, hora inicio, hora fin, artículos que reservó). Toda la información debe almacenarse en una base de datos en postgres

Además de los requisitos mencionados anteriormente el código del programa debe dividirse siguiendo el patrón Modelo Vista Controlador (MVC). También el sistema debe permitir cargar de forma dinámica diferentes plugins, utilizando interfaces de java. Y por último se deben diseñar tres diferentes plugins que se deben cargar dinámicamente.

Diseño – Estructuras y Soluciones Finales

Para crear los diferentes objetos del sistema se utilizó como base el siguiente diagrama de clases:



Descripción de las Clases

La clase principal es la de tipo **Centro**. Al iniciar el programa se crea un objeto de tipo Centro en cual se almacenan los diferentes objetos en diferentes listas (listaSocios, listaArticulos, listaInstalaciones) las cuales contienen diferentes objetos que serán explicados a continuación.

listaSocios: Esta lista almacena a los distintos socios, los cuales son instanciados de la clase **Socios** la cual contiene la información necesaria de un socio.

listaArticulos: Esta lista almacena a los distintos artículos, los cuales son instanciados de la clase **Artículos** la cual contiene la información de un artículo.

listaInstalaciones: Esta lista almacena a las distintos instalaciones, los cuales son instanciados de la clase **Instalaciones** la cual contiene la información de una instalación.

Existen cuatro clases con sus propios atributos (**Cancha, Pista, Gym, Piscina**) Estas clases heredan de la clase **Instalaciones** la cual contiene los atributos nombre, características y dimensiones.

Y por último en la clase **ReservaArticulo** la cual representa los datos de reservar un artículo esta implementa de **Articulo** y **Centro**.

Lenguaje y Base de Datos

Para realizar el sistema se utilizó Java, basándose en el principio Modelo Vista Controlador. Se almacenaron los datos del sistema en una base de dato mediante una conexión con postgres. A continuación se muestra un ejemplo de una conexión.

```
String cadena = "jdbc:postgresql://localhost:5432/baseLenguajes";
String user = "postgres";
String pass = "12345";
try {
    Class.forName("org.postgresql.Driver");
    Connection conex = DriverManager.getConnection(cadena,user,pass);
    java.sql.Statement st = conex.createStatement();
    String sql = "insert into socio(nombre, cedula, telefono, provincia, ciudad, direccion, cuota) "
        + "values ('"+n+"', '"+ced+"', '"+tel+"', '"+pro+"', '"+ciud+"', '"+dir+"',"+ cuota+"");
    ResultSet result = st.executeQuery(sql);

    result.close();
    st.close(); conex.close();
} catch(Exception exc) {
    System.out.println("Errorx:"+e-xc.getMessage());
}
```

Plugins

Se diseñaron tres plugins los cuales cargar de forma dinámica utilizando una interface de Java. Estos fueron:

Cargar Reservaciones: Este plugin crea un HTML con todas las reservaciones que se han realizado en el centro. Desplegándolas en una tabla y mostrando la instalación que se reservó, hora de inicio y fin de la reservación y fecha.

Cargar Reservaciones Socio: Este plugin crea un archivo “.TXT” con los datos de las reservaciones de un cliente y los distintos artículos que pidió para dicha reservación.

Crear Invitación: Este plugin crea una invitación con sus datos (fecha, nombre de instalación, etc.)

Comentarios Finales

Poner en práctica este paradigma es muy provechoso ya que permite abstraer de una forma muy acertada a la realidad a los distintos objetos.

Bibliografía

javaya. (11 de 15 de 2015). Obtenido de <http://www.javaya.com.ar/detalleconcepto.php?codigo=87>

oracle. (15 de 11 de 2015). Obtenido de <https://docs.oracle.com/javase/7/docs/api/javax/swing/JFileChooser.html>

Oracle. (15 de 11 de 2015). *Java*. Obtenido de <http://www.oracle.com/technetwork/java/painting-140037.html>

w3schools. (15 de 11 de 2015). Obtenido de http://www.w3schools.com/html/html_tables.asp