Memoria

Contenido

[Estudio del problema 2](#_Toc413235877)

[Análisis de requisitos 2](#_Toc413235878)

[Solución propuesta 2](#_Toc413235879)

[Control de versiones 2](#_Toc413235880)

[Diagrama E/R 3](#_Toc413235881)

[Modelo relacional 4](#_Toc413235882)

[Diagrama de casos de uso 5](#_Toc413235883)

[Diagrama de clases 6](#_Toc413235884)

[Pojos 6](#_Toc413235885)

[HibernateUtil 7](#_Toc413235886)

[DAO 8](#_Toc413235887)

[Listado de peticiones al webservice 9](#_Toc413235888)

# Estudio del problema

Se trata de crear un webservice que permita llevar la gestión básica de un gimnasio. Los elementos a gestionar son: alumnos, monitores y tablas junto con los ejercicios de los que constan dichas tablas.

# Análisis de requisitos

El webservice debe permitir la gestión de alumnos, monitores y tablas. Deberá ofrecer como principales funcionalidades:

* Alumnos:
  + Dar de alta nuevos alumnos
  + Listar los alumnos registrados
  + Borrar un alumno registrado
* Ejercicios
  + Listar los ejercicios disponibles
* Monitores
  + Dar de alta nuevos monitores
  + Listar los monitores registrados
  + Borrar un monitor registrado
* Tablas
  + Crear tablas nuevas incluyendo el listado de ejercicios asociados a dicha tabla.
  + Listar las tablas existentes ofreciendo la posibilidad de visualizar la tabla junto con la lista de ejercicios asociados a dicha tabla
  + Borrar una tabla creada previamente

Los monitores podrán adoptar 2 roles:

1. **Administradores**: pueden realizar todas las operaciones anteriormente descritas.
2. **Monitores** (no administradores): pueden realizar todas las operaciones anteriormente descritas excepto las correspondientes a dar de alta monitores y listar monitores.

# Solución propuesta

La solución propuesta, se basa en la creación de un webservice basado en RestFull. Se utilizará como sistema de almacenamiento, MySQL. La información devuelta por el webservice, deberá presentarse indistintamente tanto en formato XML como en formato JSON.

# Control de versiones

Se puede hacer seguimiento del proyecto a través de Subversion:

<https://github.com/orzoweitop/rexia>

# Diagrama E/R

# Modelo relacional



# Diagrama de casos de uso

# Diagrama de clases

## Pojos

Servicios



## HibernateUtil



## DAO



# Listado de peticiones al webservice

NOTA: las operaciones marcadas con (\*) requieren de privilegios.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | TIPO | PATH | MÉTODO | DESCRIPCIÓN | CONSUME | PRODUCE |
| OK | GET | **/alumnos** | getListaAlumnos() | Devuelve un listado con los alumnos registrados en la BBDD |  | application/xml; application/json |
| OK | GET | **/alumnos/{id}** | getAlumno(int id) | Devuelve el alumno especificado |  | application/xml; application/json |
| OK | POST | **/alumnos** | annadirAlumno(Alumno) | Registra un nuevo alumno en la BBDD | application/xml | application/xml; application/json |
| OK | PUT | **/alumnos** | modificarAlumno(Alumno) | Modifica/actualiza la información de un alumno. | application/xml | application/xml; application/json |
| OK | DELETE | **/alumnos/{id}** | borrarAlumno(int id) | Borra el alumno de la BBDD correspondiente al id pasado por parámetro |  | text/plain |
| OK | GET | **/alumnos/count** | contarAlumnos() | Devuelve el total de alumnos registrados en la BBDD |  | text/plain |
| OK | GET | **/ejercicios** | getListaEjercicios() | Devuelve un listado con los ejercicios registrados en la BBDD |  | application/xml; application/json |
| OK | GET | **/ejercicios/{id}** | getEjercicio(int id) | Devuelve el ejercicio especificado |  | application/xml; application/json |
| OK | GET | **/ejercicios/count** | contarEjercicios() | Devuelve el total de ejercicios registrados en la BBDD |  | text/plain |
| OK | GET\* | **/monitores** | getListaMonitores() | Devuelve un listado con los monitores registrados en la BBDD |  | application/xml; application/json |
| OK | GET\* | **/monitores/{id}** | getMonitor(int id) | Devuelve el monitor especificado |  | application/xml; application/json |
| OK | POST | **/monitores** | annadirMonitor(Monitor) | Registra un nuevo monitor en la BBDD | application/xml | application/xml; application/json |
| OK | PUT | **/monitores** | modificarMonitor(Monitor) | Modifica/actualiza la información de un monitor | Application/xml | application/xml; application/json |
| OK | DELETE | **/monitores/{id}** | borrarMonitor(int id) | Borra el monitor de la BBDD correspondiente al id pasado por parámetro |  | text/plain |
| OK | GET | **/monitores/count** | contarMonitores() | Devuelve el total de monitores registrados en la BBDD |  | text/plain |
| OK | GET\* | **/monitores/{id}/alumnos** | getListaAlumnos(Monitor) | Devuelve la lista de alumnos asociada a un monitor |  | application/xm; application/json |
| OK | GET | **/tablas** | getListaTablas() | Devuelve un listado con las tablas registradas en la BBDD |  | application/xm; application/json |
| OK | GET | **/tablas/{id}** | getTabla(int id) | Devuelve la tabla especificada |  | application/xml; application/json |
| OK | POST | **/tablas** | annadirTabla(Tabla) | Guarda una tabla en la BBDD | application/xml | application/xml; application/json |
| OK | POST | **/tablas/líneas** | annadirLinea(Linea) | Guarda una línea asociada a una tabla en la BBDD | application/xml | application/xml; application/json |
| OK | PUT | **/tablas** | modificarTabla(Tabla) | Modifica/actualiza la información de una tabla | application/xml | application/xml; application/json |
| OK | DELETE | **/tablas/{id}** | borrarTabla(int id) | Borra la tabla de la BBDD correspondiente al id pasado por parámetro |  | text/plain |
| OK | GET | **/tablas/count** | contarTablas() | Devuelve el total de tablas registradas en la BBDD |  | text/plain |
| OK | GET | **/tablas/alumno/{id}** | getListaTablasAlumnos(Alumno) | Devuelve un listado con las tablas creadas para dicho alumno |  | application/xm; application/json |
| OK | GET | **/tablas/{id}/líneas** | getListaEjercicios(Tabla) | Devuelve la lista de ejercicios asociada a la tabla especificada |  | application/xml; application/json |

NOTA: la url base para el proyecto será: <http://localhost:8080/Proyecto/webservice/>