

Crea un nuevo pojo llamado **Alumno** con los siguientes campos (recuerda agregar su constructor default, sobrecargado y sus getters y setters)

```
private int idAlumno;
private String nombre;
private String apellidoPaterno;
private String apellidoMaterno;
private String matricula;
private String correo;
private String fechaNacimiento;
private int idCarrera;
private String carrera;
private int idFacultad;
private String facultad;
private byte[] foto;
private String fotoBase64;
```

Crea un nuevo mapper **AlumnoMapper.xml** (recuerda agregarlo al **mybatis-config.xml**) con los siguiente:

```
<!DOCTYPE mapper
PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="alumno">

    <select id="obtener-todos" resultType="pojo.Alumno" >
        SELECT idAlumno, nombre, apellidoPaterno, apellidoMaterno, matricula,
        correo, fechaNacimiento, a.idCarrera, c.carrera, c.idFacultad, f.facultad
        FROM alumno a
        INNER JOIN carrera c ON c.idCarrera = a.idCarrera
        INNER JOIN facultad f ON f.idFacultad = c.idFacultad
    </select>

    <select id="obtener-por-carrera" resultType="pojo.Alumno" parameterType="int" >
        SELECT idAlumno, nombre, apellidoPaterno, apellidoMaterno, matricula, correo, fechaNacimiento
        FROM alumno
        WHERE idCarrera = #{idCarrera}
    </select>

</mapper>
```

Agrega una nueva clase en dominio llamada **AlumnoImp** y agrega lo siguiente:

```
public static List<Alumno> obtenerAlumnos() {
    List<Alumno> alumnos = null;
    SqlSession conexionBD = MyBatisUtil.getSession();
    if (conexionBD != null) {
        try {
            alumnos = conexionBD.selectList("alumno.obtener-todos");
            conexionBD.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    return alumnos;
}

public static List<Alumno> obtenerAlumnosPorCarrera(int idCarrera) {
    List<Alumno> alumnos = null;
    SqlSession conexionBD = MyBatisUtil.getSession();
    if (conexionBD != null) {
        try {
            alumnos = conexionBD.selectList("alumno.obtener-por-carrera", idCarrera);
            conexionBD.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    return alumnos;
}
```

Posteriormente agrega los siguientes servicios (AlumnoWS) para obtener todos los alumnos y los alumnos pertenecientes a una carrera.

```
@Path("obtener-todos")
@GET
@Produces(MediaType.APPLICATION_JSON)
public List<Alumno> obtenerAlumnos() {
    return AlumnoImp.obtenerAlumnos();
}

@Path("obtener-todos-carrera/{idCarrera}")
@GET
@Produces(MediaType.APPLICATION_JSON)
public List<Alumno> obtenerAlumnosCarrera(@PathParam("idCarrera") Integer idCarrera) {
    if (idCarrera != null && idCarrera > 0) {
        return AlumnoImp.obtenerAlumnosPorCarrera(idCarrera);
    }
    throw new BadRequestException();
}
```

**PRUEBA** tus dos nuevos servicios en Postman para garantizar que funcionen correctamente.

Ahora agrega dos nuevos pojos uno llamado **Facultad** con los siguientes atributos (recuerda agregar su constructor default, sobrecargado y sus getters y setters)

```
private int idFacultad;  
private String facultad;
```

y otro llamado **Carrera** (recuerda agregar su constructor default, sobrecargado y sus getters y setters)

```
private int idCarrera;  
private String carrera;  
private int idFacultad;
```

Ahora en el mapper **CatalogoMapper.xml** agrega lo siguiente:

```
<select id="obtener-facultades" resultType="pojo.Facultad" >  
    SELECT idFacultad, facultad FROM facultad  
</select>  
  
<select id="obtener-carreras-facultad" resultType="pojo.Carrera" parameterType="int" >  
    SELECT idCarrera, carrera FROM carrera WHERE idFacultad = #{idFacultad}  
</select>
```

Ahora en la clase **CatalogoImp** y agrega lo siguiente:

```

public static List<Facultad> obtenerFacultades() {
    List<Facultad> facultades = null;
    SqlSession conexionBD = MyBatisUtil.getSession();
    if (conexionBD != null) {
        try {
            facultades = conexionBD.selectList("catalogo.obtener-facultades");
            conexionBD.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    return facultades;
}

public static List<Carrera> obtenerCarrerasFacultad(int idFacultad) {
    List<Carrera> carreras = null;
    SqlSession conexionBD = MyBatisUtil.getSession();
    if (conexionBD != null) {
        try {
            carreras = conexionBD.selectList("catalogo.obtener-carreras-facultad", idFacultad);
            conexionBD.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    return carreras;
}

```

Posteriormente agrega los siguientes servicios (CatalogoWS) para obtener todas las facultades y las carreras asociadas por facultad.

```

@Path("obtener-facultades")
@GET
@Produces(MediaType.APPLICATION_JSON)
public List<Facultad> obtenerFacultades() {
    return CatalogoImp.obtenerFacultades();
}

@Path("obtener-carreras-facultad/{idFacultad}")
@GET
@Produces(MediaType.APPLICATION_JSON)
public List<Carrera> obtenerCarrerasFacultad(@PathParam("idFacultad") Integer idFacultad) {
    if (idFacultad != null && idFacultad > 0) {
        return CatalogoImp.obtenerCarrerasFacultad(idFacultad);
    }
    throw new BadRequestException();
}

```

**PRUEBA** tus dos nuevos servicios en Postman para garantizar que funcionen correctamente.