

Creacion del compilador y Runner

c.cmd

```
rm -rf \opt\apache-tomcat-9.0.52\webapps\ROOT\  
rm -rf \opt\apache-tomcat-9.0.52\webapps\ROOT.war  
mvn clean install -DskipTests
```

r.cmd

```
mv target\ROOT \opt\apache-tomcat-9.0.52\webapps\  
cd \opt\apache-tomcat-9.0.52\bin\  
java -cp tomcat-juli.jar;bootstrap.jar  
org.apache.catalina.startup.Bootstrap  
cd C:\cursos\nombregroupo\clase[zzz]\
```

Primer Servlet

```
public class AlumnosController extends HttpServlet {  
    public void doGet(HttpServletRequest request,  
        HttpServletResponse response)  
        throws IOException, ServletException {  
        response.setContentType("text/html");  
        PrintWriter out = response.getWriter();  
        out.println(System.currentTimeMillis());  
        response.setStatus(200);  
    }  
}
```

Tareas previas antes de realizar el Alumnos Controller

Mostrar el faker

Hacer un controller que se llame inventar alumno, para facilitar los post

Mostrar la configuracion de los mappings

Alumnos Controller

```
public class AlumnosController extends HttpServlet {  
    public void doGet(HttpServletRequest request,  
        HttpServletResponse response)  
        throws IOException, ServletException {  
        response.setContentType("application/json");  
        PrintWriter out = response.getWriter();  
  
        var conector = new ConectorJPA();  
        var em = conector.getEntityManager();  
        var qr = em.createQuery("SELECT a FROM  
Alumno a", Alumno.class);  
  
        var a = qr.getResultList();  
        var strAlumno = new Gson().toJson(a);  
  
        out.println(strAlumno);  
        response.setStatus(200);  
    }  
}
```

```
public void doPost(HttpServletRequest request,
    HttpServletResponse response)
    throws IOException, ServletException {

    response.setContentType("application/json");

    BufferedReader br =
        new BufferedReader(new
InputStreamReader(request.getInputStream()));
    var strJson = new StringBuilder();
    for (String linea = br.readLine(); linea != null; linea
= br.readLine()) {
        strJson.append(linea);
    }

    var alumno = new Gson().fromJson(strJson.toString(),
Alumno.class);

    System.out.println(alumno.id);
    System.out.println(alumno.nombre);
    System.out.println(alumno.apellido);

    var conector = new ConectorJPA();
    var em = conector.getEntityManager();
    var tx = em.getTransaction();
    tx.begin();
    em.persist(alumno);
    tx.commit();

    response.setStatus(201);
```

```
}  
    public void doPut(HttpServletRequest request,  
        HttpServletResponse response)  
        throws IOException, ServletException {  
  
        response.setContentType("text/html");  
        PrintWriter out = response.getWriter();  
        out.println("PUT");  
        response.setStatus(200);  
    }  
    public void doDelete(HttpServletRequest request,  
        HttpServletResponse response)  
        throws IOException, ServletException {  
  
        response.setContentType("text/html");  
        PrintWriter out = response.getWriter();  
        out.println("DELETE");  
        response.setStatus(200);  
    }  
}
```