

Los EJB Async forman parte de **Java EE 6** y permiten realizar peticiones de forma asíncrona, algo que a veces necesario . Vamos a crear un EJB que tenga un método que funciona de forma asíncrona:

```
package com.arquitecturajava.ejb;

import java.util.concurrent.Future;

import javax.ejb.AsyncResult;
import javax.ejb.Asynchronous;
import javax.ejb.LocalBean;
import javax.ejb.Stateless;

/**
 * Session Bean implementation class EJBA sincrono
 */
@Stateless
@LocalBean
public class EJBA sincrono implements EJBA sincronoLocal {

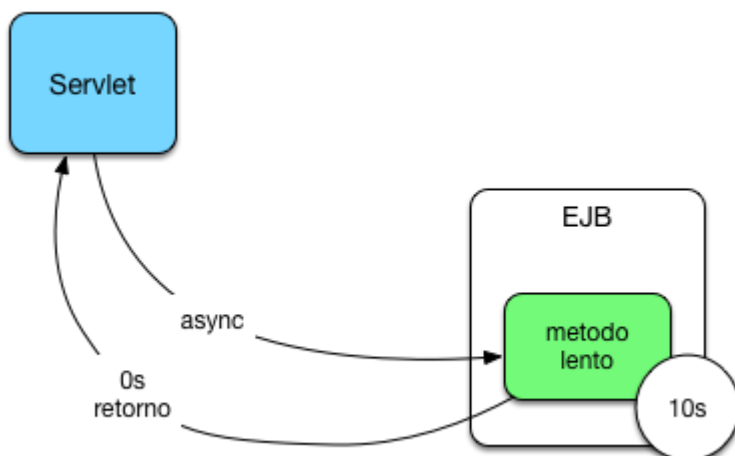
    @Override
    @Asynchronous
    public void&nbsp; metodoLento() {

        try {
            Thread.sleep(10000);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```

```
}  
  
System.out.println("que lento va esto");  
  
}  
  
}
```

Invocando un EJB Async

Acabamos de crear nuestro primer EJB Asíncrono utilizando la anotación `@Asynchronous` . En este caso tenemos un método que tarda 10 segundos en ejecutarse. Vamos a invocar este EJB desde un Servlet , en principio el Servlet debería tardar 10 segundos en ejecutarse ya que el EJB duerme 10 segundos . Sin embargo al tratarse de un EJB asíncrono el Servlet no se verá bloqueado y devolverá un mensaje de forma inmediata.



Vamos a ver como se construye el código del Servlet y como invoca al EJB:

```
package com.arquitecturajava.servlet;

import java.io.IOException;
import java.io.PrintWriter;

import javax.ejb.EJB;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.arquitecturajava.ejb.EJBAsincronoLocal;
@WebServlet("/ServletEJBAsync")
public class ServletEJBAsync extends HttpServlet {
    private static final long serialVersionUID = 1L;

    @EJB
    EJBAsincronoLocal ejb;

    public ServletEJBAsync() {
        super();
        // TODO Auto-generated constructor stub
    }

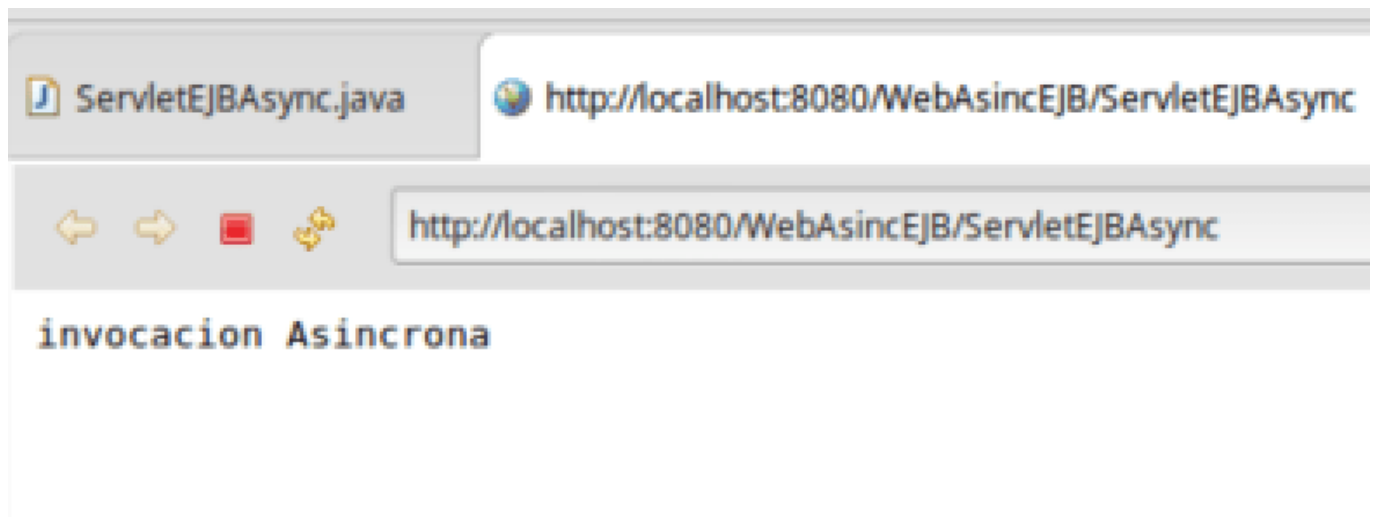
    protected void doGet(HttpServletRequest request, HttpServletResponse
    response) throws ServletException, IOException {

        ejb.metodoLento();

        PrintWriter pw= response.getWriter();
        pw.println("invocacion Asincrona");
    }
}
```

```
pw.close();  
  
}  
}
```

El Servlet nos devolverá el resultado de forma inmediata :



Mientras por el otro lado el mensaje saldrá en en la consola del servidor de aplicaciones al cabo de 10 segundos.

```
12 INFO [org.jboss.as] (controller boot thread) JBoss  
51 INFO [stdout] (EJB default - 1) que lento va esto
```

Los EJBs asíncronos pueden ser útiles a la hora de realizar tareas cuyo tiempo de ejecución se dispare o en situaciones en las que tenemos que realizar varias tareas de forma asíncrona y luego enlazarlas.

Otros artículos relacionados:

[Introducción EJB](#)

[EJBs Remotos](#)