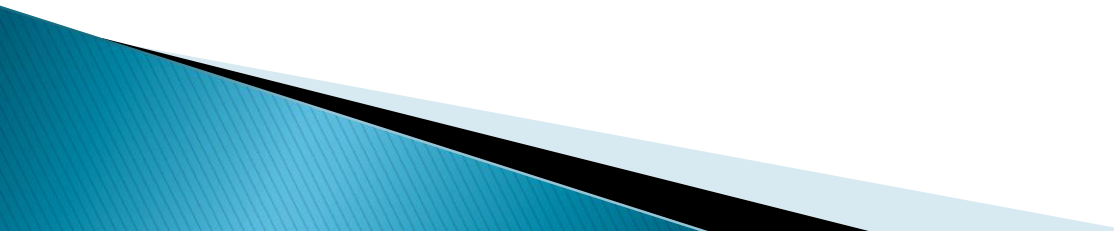


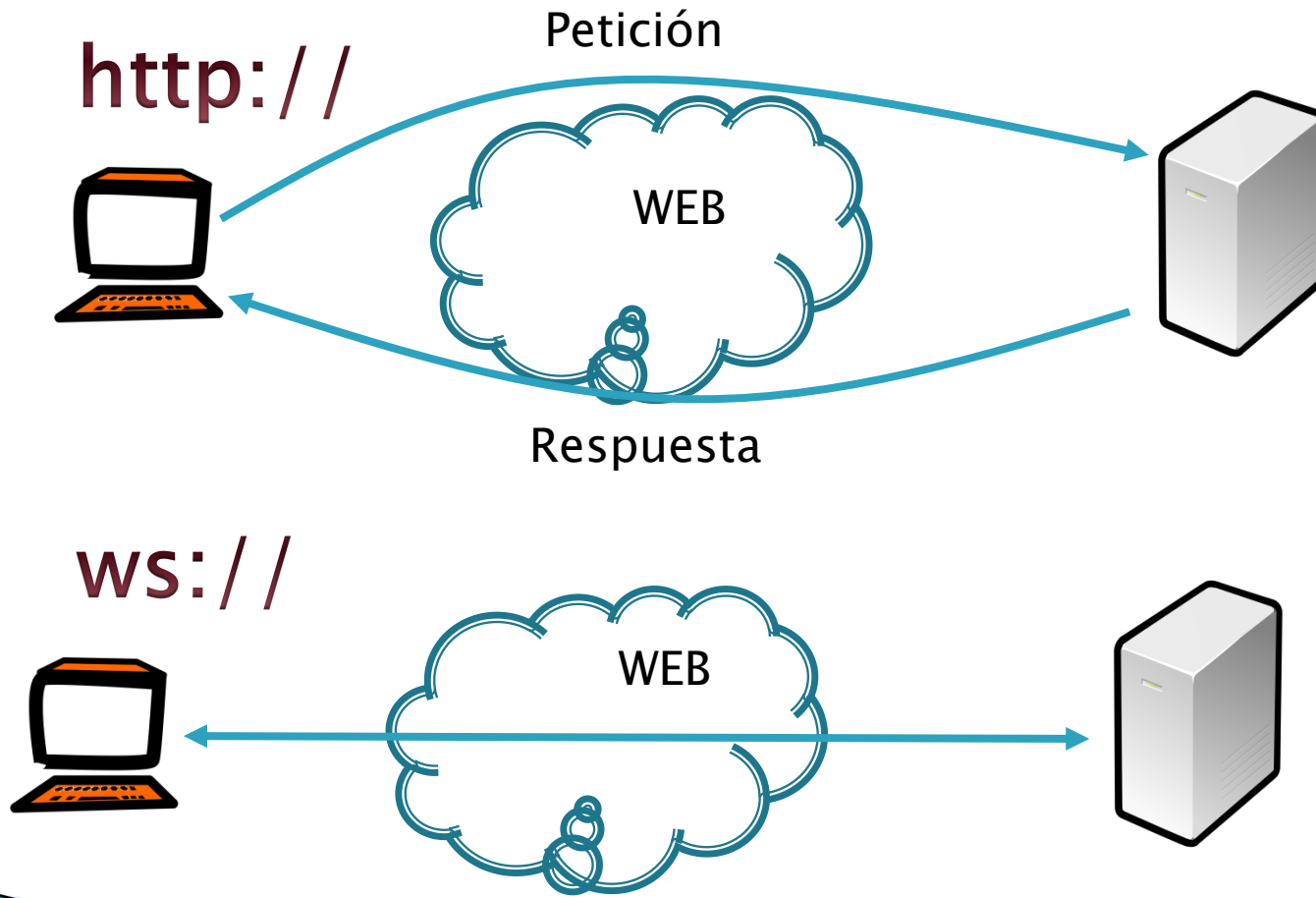
# Introducción a los websockets



# ¿Qué es WebSocket?

- Tecnología/protocolo que permite una comunicación bidireccional en un contexto cliente/servidor
  - Proporciona un canal sobre TCP, en el que tanto cliente como servidor pueden enviar datos
  - Implementación en servidor con WebSocket API de Java EE 7
  - Implementación en cliente con objeto WebSocket HTML5
- 

# WebSocket vs HTTP



# API websocket

javax.validation.constraints  
javax.validation.constraintvalidation  
javax.validation.executable  
javax.validation.groups  
javax.validation.metadata  
javax.validation.spi  
javax.websocket  
**javax.websocket.server**  
javax.ws.rs  
javax.ws.rs.client  
javax.ws.rs.container  
javax.ws.rs.core  
javax.ws.rs.ext

javax.websocket.server

## Interfaces

*HandshakeRequest*  
*ServerApplicationConfig*  
*ServerContainer*  
*ServerEndpointConfig*

## Classes

*ServerEndpointConfig.Builder*  
*ServerEndpointConfig.Configurator*

## Annotation Types

*PathParam*  
*ServerEndpoint*

OVERVIEW **PACKAGE** CLASS USE TREE DEPRECATED INDEX HELP

PREV PACKAGE NEXT PACKAGE FRAMES NO FRAMES

## Package javax.websocket

This package contains all the WebSocket APIs common to both the client and server side.

See: Description

### Interface Summary

Interface	Description
<b>ClientEndpointConfig</b>	The ClientEndpointConfig is a special kind of endpoint configuration object that configures client endpoints.
<b>CloseReason.CloseCode</b>	A marker interface for the close codes.
<b>Decoder</b>	The Decoder interface holds member interfaces that define how a developer can provide developer defined custom objects.
<b>Decoder.Binary&lt;T&gt;</b>	This interface defines how a custom object (of type T) is decoded from a web socket message.
<b>Decoder.BinaryStream&lt;T&gt;</b>	This interface defines how a custom object is decoded from a web socket message in stream mode.
<b>Decoder.Text&lt;T&gt;</b>	This interface defines how a custom object is decoded from a web socket message in text mode.
<b>Decoder.TextStream&lt;T&gt;</b>	This interface defines how a custom object of type T is decoded from a web socket message in stream mode.
<b>Encoder</b>	The Encoder interface defines how developers can provide a way to convert their custom objects into a web socket message.
<b>Encoder.Binary&lt;T&gt;</b>	This interface defines how to provide a way to convert a custom object into a binary web socket message.

# Implementación en servidor

```
@ServerEndpoint("/dir")
public class EjemploWebSocket {
    //métodos del ciclo de vida de la conexión
    @OnMessage
    public String onMessage(String mensaje) {
        //tareas a realizar cuando se
        //recibe un mensaje
    }

    @OnOpen
    public void open(){
        //tareas a realizar cuando se establece
        //una conexión
    }

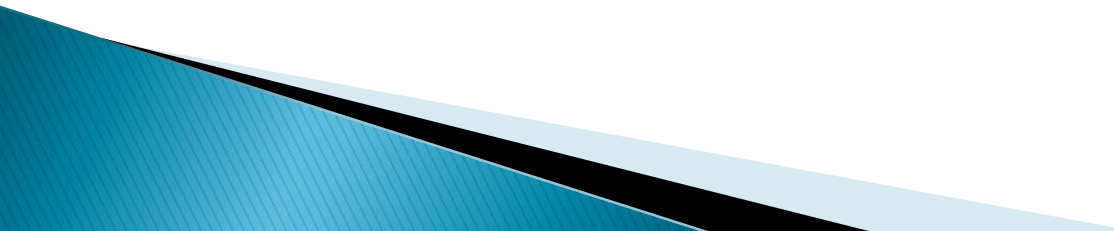
    @OnClose
    public void cierre(){
        //tareas a realizar cuando se cierra
        //una conexión
    }
}
```

# Implementación en cliente

```
var uri="ws://servidor:puerto/app/dir";
var miWebSocket= new WebSocket(uri);
miWebSocket.onopen=function(evento) {
    //tareas a realizar al abrir la
    //conexión
}

miWebSocket.onmessage=function(evento) {
    //tareas a realizar al recibir
    //un mensaje del servidor
}

miWebSocket.onclose=function(evento) {
    //tareas a realizar al cerrar
    //la conexión
}
```



# Utilización

- Juegos on-line
  - Aplicaciones de chat
  - Actualización de datos en tiempo real
- 