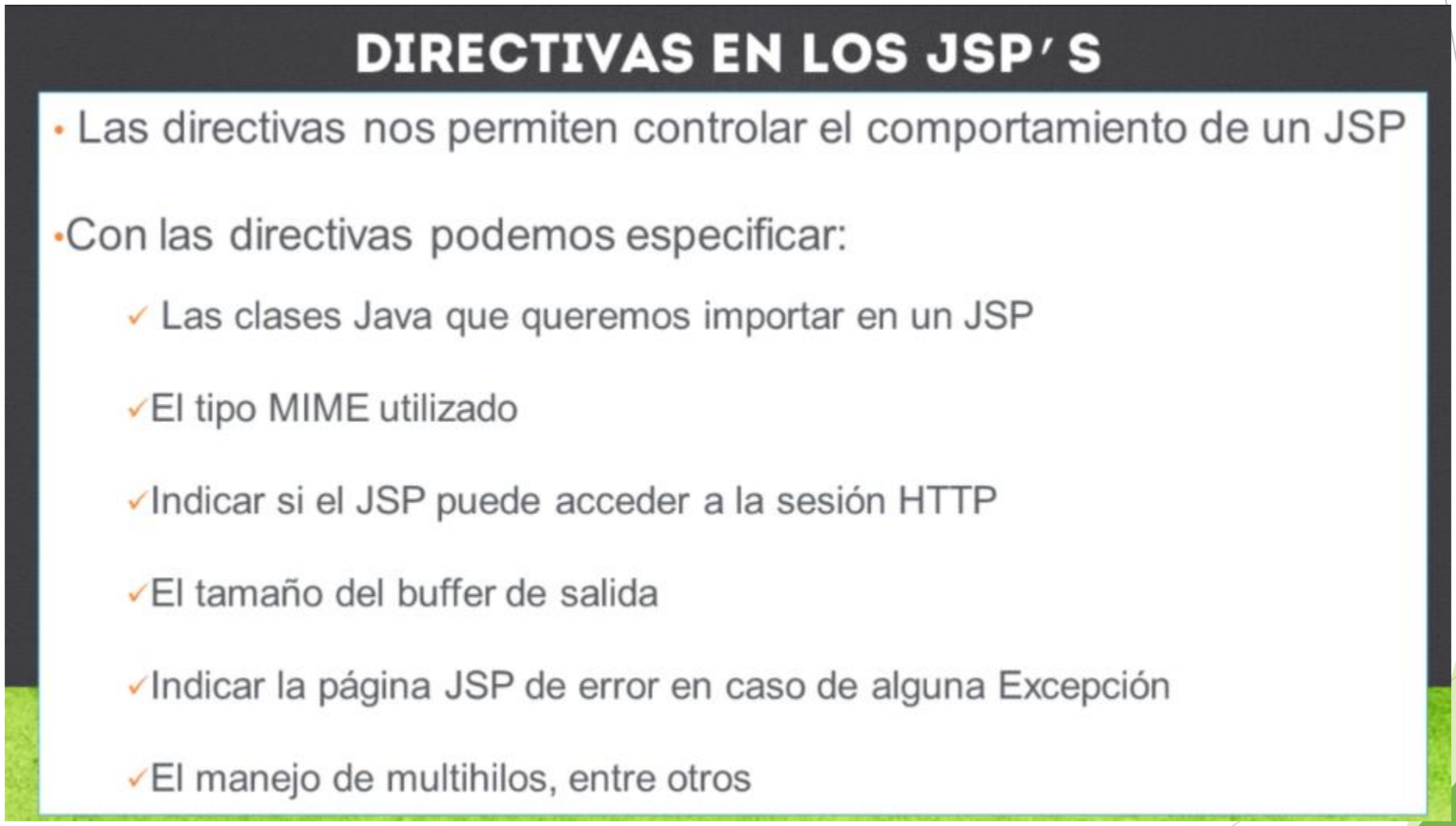
DIRECTIVAS


DIRECTIVAS EN LOS JSP´S
• Las directivas nos permiten controlar el comportamiento de un JSP
•Con las directivas podemos especificar:
✓ Las clases Java que queremos importar en un JSP
✓El tipo MIME utilizado
✓Indicar si el JSP puede acceder a la sesión HTTP
✓El tamaño del buffer de salida
✓Indicar la página JSP de error en caso de alguna Excepción
✓El manejo de multihilos, entre otros

Vamos a revisar a continuación el tema de las directivas en los JSP's.

Las directivas nos permiten controlar el comportamiento de un JSP, por ejemplo las clases que vamos a utilizar dentro de un JSP y hacer el import de clases Java, especificar el tipo MIME con el que vamos a responder a nuestro cliente.

También con estas directivas podemos especificar si un JSP va a participar en el concepto de sesiones, o podemos especificar el tamaño del buffer de salida, especificar el JSP de error en caso de que tengamos alguna excepción dentro de nuestro JSP, o indicar si nuestro JSP va a manejar el concepto de multihilos, entre varias características más.

Veamos a continuación a más detalle varias de las directivas que podemos utilizar en nuestros JSP's.

ATRIBUTOS EN LAS DIRECTIVAS DE UN JSP

- Atributo **import**:

  <%@ page import="paquete.Clase1 , paquete.ClaseN" %>

- Atributo **contentType**:

  <%@ page contentType="MIME-Type" %>

- Atributo **session**:

  <%@ page session="true" %>

- Atributo **isELIgnored**:

  <%@ page isELIgnored="false" %>

En esta lección vamos a revisar la directiva de page. Algunos de los atributos que podemos indicar en la directiva page son:

Import: El atributo de import dentro de la directiva page nos va a permitir especificar cuáles son las clases que vamos a importar dentro de nuestro JSP e indirectamente a nuestro servlet generado a partir de la solicitud a nuestro JSP. Aunque el import lo podemos declarar en cualquier parte de nuestro JSP lo recomendable es ponerlo al principio como una buena práctica. Para especificar las clases que queremos importar en nuestro JSP podemos utilizar la sintaxis **<%@ page import="paquete.Clase1, paquete.ClaseN" %>.** Podemos detectar que es una directiva debido a que estamos utilizando ahora el símbolo de @ y posteriormente el nombre page. Posteriormente utilizamos el atributo import e indicar el nombre completamente calificado de nuestras clases, es decir, incluimos el paquete y el nombre de nuestra clase y si queremos especificar más de una clase a utilizar vamos a separarlas por coma.

Posteriormente tenemos el atributo **contentType <%@page contentType="MIME-Type" %> .** Este atributo nos permite especificar el tipo de respuesta a nuestro cliente web.

Session: <% @page session="true" %> También vamos a poder indicar por medio de esta directiva page si el JSP va poder acceder al objeto sesión que se haya creado anteriormente por ejemplo desde un JSP o desde otro servlet. Por default un JSP está configurado para que podamos acceder al objeto session, si queremos indicar lo contrario tendremos que especificar false en esta directiva y si especificamos false en esta directiva lo que sucede es que dentro de nuestras clases implícitas de un JSP ya no vamos a tener acceso a nuestra variable.

Atributo isELIgnored: <%@ page isELIgnored="false" %> También tenemos otro atributo llamado isELIgnored esto lo que significa es que si queremos deshabilitar el manejo de expression language que posteriormente vamos a revisar, tendríamos que indicarlo por medio de esta directiva page indicando en el atributo isELIgnored el valor de true, por default un JSP a partir de la versión 2.4 puede utilizar de manera automática el concepto de Expression Language pero anterior a la versión 2.3 de los servlets los JSPs por default no pueden acceder al lenguaje conocido como Expression Language.

ATRIBUTOS EN LAS DIRECTIVAS DE UN JSP CONT

- Atributo **buffer**:

```
<%@ page buffer="tamañoEnKb" %>
```

- Atributo **errorPage**:

```
<%@ page errorPage="url relativo al JSP de error" %>
```

- Atributo **isErrorPage**:

```
<%@ page isErrorPage="true" %>
```

- Atributo **isThreadSafe**:

```
<%@ page isThreadSafe="true" %>
```

- Atributo **extends:**

```
<%@ page extends="paquete.NombreClase" %>
```

Atributo **buffer: <%@ page buffer=”tamañoEnKb” %>** dentro de esta misma directiva page tenemos acceso al atributo buffer. Con este atributo buffer podemos especificar el tamaño en Kb que puede contener nuestro buffer del JSP y si llegamos al tamaño especificado en este JSP entonces se hace un **flush o vaciado** de manera automática de todo el flujo que hayamos agregado a nuestro printWriter o a nuestro output servlet dependiendo el caso.

También tenemos los atributos **errorPage** con la siguiente sintaxis:

<%@ page errorPage=”url_relativo_al_JSP_de_error” %> con el atributo erroPage vamos a especificar cual va a ser el JSP que va a manejar las excepciones en caso de que el JSP provoque un error, y para configurar el JSP que va a manejar los errores tenemos que indicar la directiva **isErrorPage** con la siguiente sintaxis:

<%@ page isErrorPage=”true” %> indicando que ese JSP si es el JSP que va a manejar los errores, por default los JSPs tienen el valor de esta directiva igual a false debido a que los JSPs no van a manejar las excepciones de manera automática sino hay que configurarlos para que puedan acceder a la pila de errores.

También podemos observar el atributo **isThreadSafe** con sintaxis **<%@ page isThreadSafe="true" %>** Por default el contenedor de servlets, considera a un JSP como seguro para ser accedido por múltiples hilos, es decir, el valor por default es true. Si se desea que el contenido del JSP no sea accedido por distintos hilos entonces se debe especificar este valor como false.

Por ultimo vamos a mencionar el **atributo extends <%@ page extends="paquete.NombreClase" %>** nos va a permitir heredar de una clase según especifiquemos en ese atributo y por lo tanto nuestro JSP va a heredar las características de la clase que especifiquemos normalmente ese atributo únicamente se utiliza si es que queremos agregar comportamiento de terceros aunque este concepto rara vez lo configuramos y únicamente se van a utilizar si es que algún software en particular alguna solución que compremos nos solicite hacer un extends de clases ya creadas por este vendedor.

Vamos a poner a continuación en práctica alguno de estos atributos utilizando el concepto de directivas de un JSP.