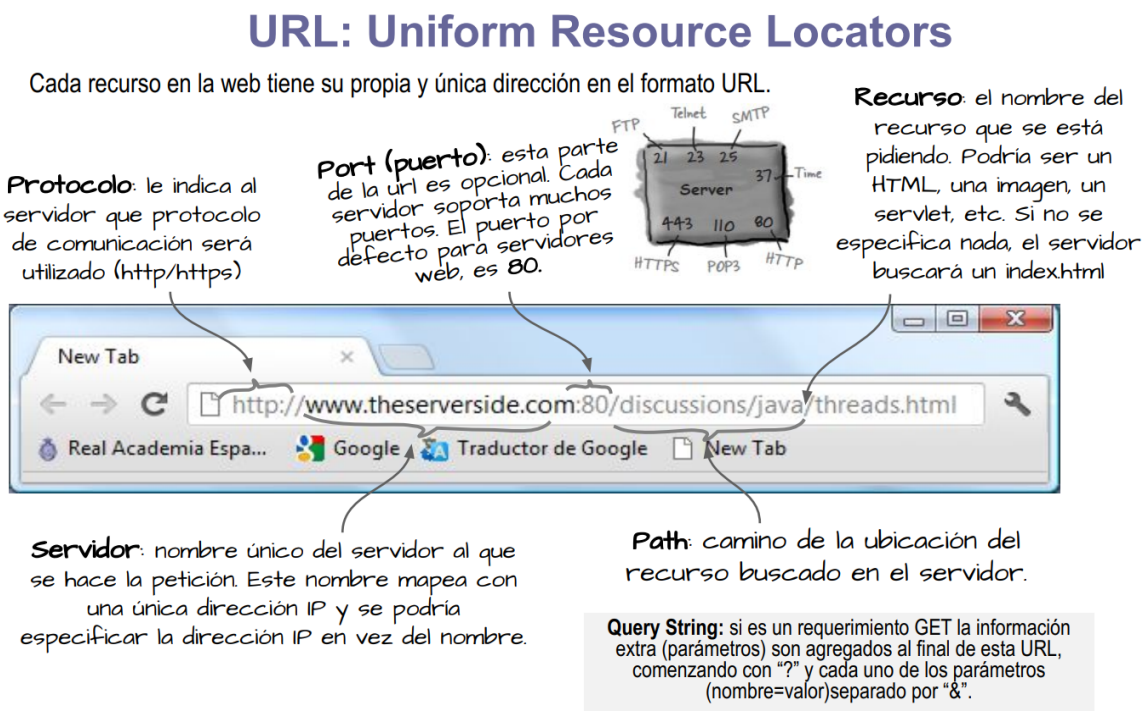
Práctica 1
Conceptos básicos.
Para este punto usará el navegador Chrome o Mozilla Firefox y analizará los intercambios de mensajes HTTP entre el navegador y los servidores Web.
Usando la tecla F12 se habilitará la herramienta para el desarrollador.

Responda las siguientes preguntas:
a. ¿Qué es una URL y cómo se compone?
Una URL (del inglés Uniform Resource Locator) es una cadena de texto que identifica la ubicación de un recurso en la web. En otras palabras, es la dirección que se utiliza para acceder a un sitio web en Internet. Las URL se emplean para enlazar distintos recursos, como páginas web, imágenes, archivos de audio o video, y otros contenidos digitales.

Esta dirección se compone de:

'https://' que sería el protocolo de acceso para las páginas de internet. Otro ejemplo es el 'ftp://' que es el protocolo para descarga de ficheros.
'www' que es la dirección del recurso.
'.significados' sería el nombre del dominio y '.com' es el tipo de dominio. Estos pueden ser genéricos como .net, .org, .mobi o territoriales como .mx, .ar, .cl. Hoy en día el tipo de dominio se está diversificando pudiente contener frases completas como .google o .maps.



b. ¿Qué componentes de la URL son opcionales?
Los parámetros constituyen una parte opcional de la URL y se utilizan para transmitir información adicional al servidor. Los parámetros se indican después del signo de interrogación (?) y sirven para especificar valores que se deben pasar al servidor. Por ejemplo, en la URL «http://ejemplo.com/buscar?consulta=ejemplo», «consulta=ejemplo» es un parámetro que se utiliza para indicar al servidor que se debe realizar una búsqueda de la palabra «ejemplo».

c. ¿Cómo se especifica en el encabezado HTTP el recurso al que se quiere acceder?
En una solicitud HTTP, el recurso al que se quiere acceder se especifica a través de la línea de solicitud y el encabezado Host.

Línea de Solicitud (Request Line):
La línea de solicitud se encuentra en la primera línea de la solicitud HTTP y tiene la siguiente estructura:

Método URI Protocolo/versión

- Método: Es el verbo HTTP que indica la acción que se desea realizar en el recurso. Ejemplos comunes son GET, POST, PUT, DELETE, etc.
- URI (Uniform Resource Identifier): Es una cadena que identifica de manera única el recurso al que se desea acceder. Puede ser una URL (Uniform Resource Locator) o un URI genérico.
- Protocolo/versión: Es la versión del protocolo HTTP que se está utilizando. Por ejemplo, HTTP/1.1.

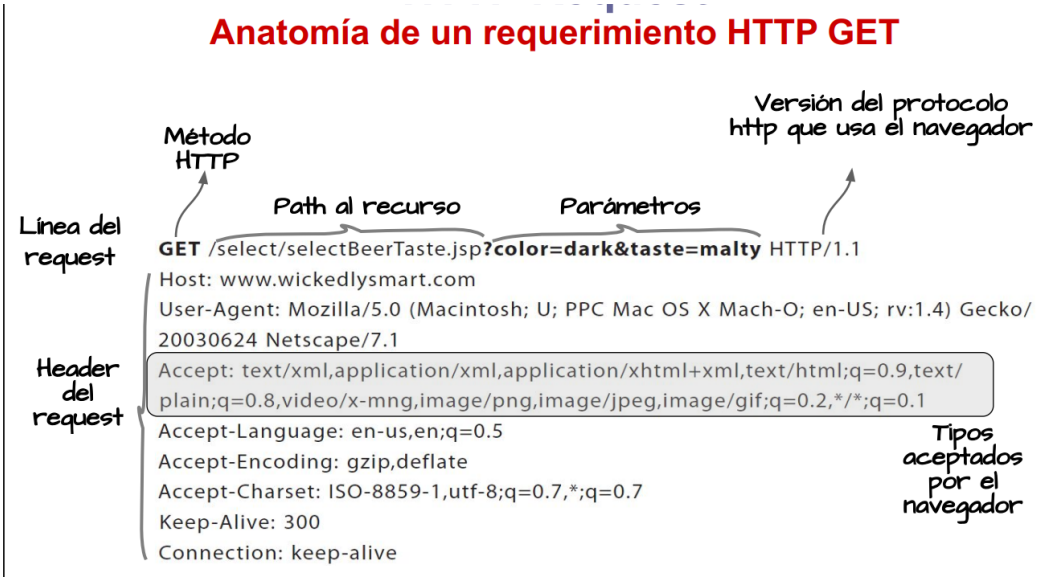
Ejemplo de una línea de solicitud GET: GET /ruta-del-recurso HTTP/1.1
Encabezado "Host":
El encabezado Host se utiliza para indicar el nombre de dominio del servidor al que se está enviando la solicitud. Es esencial cuando se accede a sitios web alojados en servidores compartidos, donde un mismo servidor físico puede alojar varios sitios web con diferentes nombres de dominio.

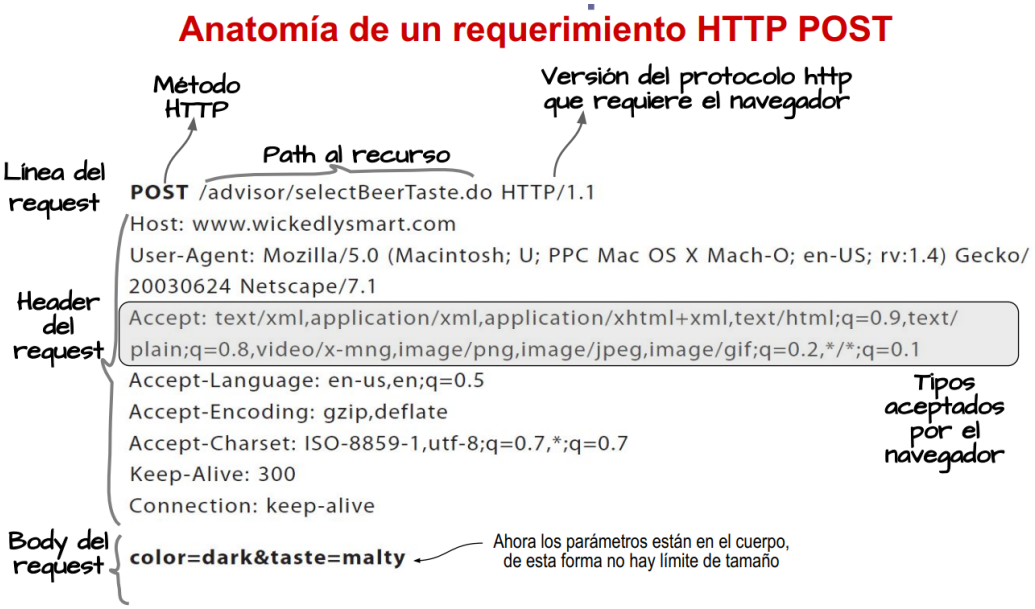
Ejemplo de un encabezado "Host": Host: www.ejemplo.com

d. ¿Cómo se diferencian los mensajes de requerimiento HTTP de los mensajes de respuesta?
Se diferencian por su estructura y contenido. Cada una cumple un propósito específico en la comunicación entre el cliente y el servidor.

Mensajes de Requerimiento (Request):
Los mensajes de requerimiento son enviados por el cliente al servidor para solicitar un recurso o realizar una acción específica en el servidor. Estos mensajes contienen información sobre la operación que se desea llevar a cabo y el recurso involucrado. La estructura básica de un mensaje de requerimiento es:

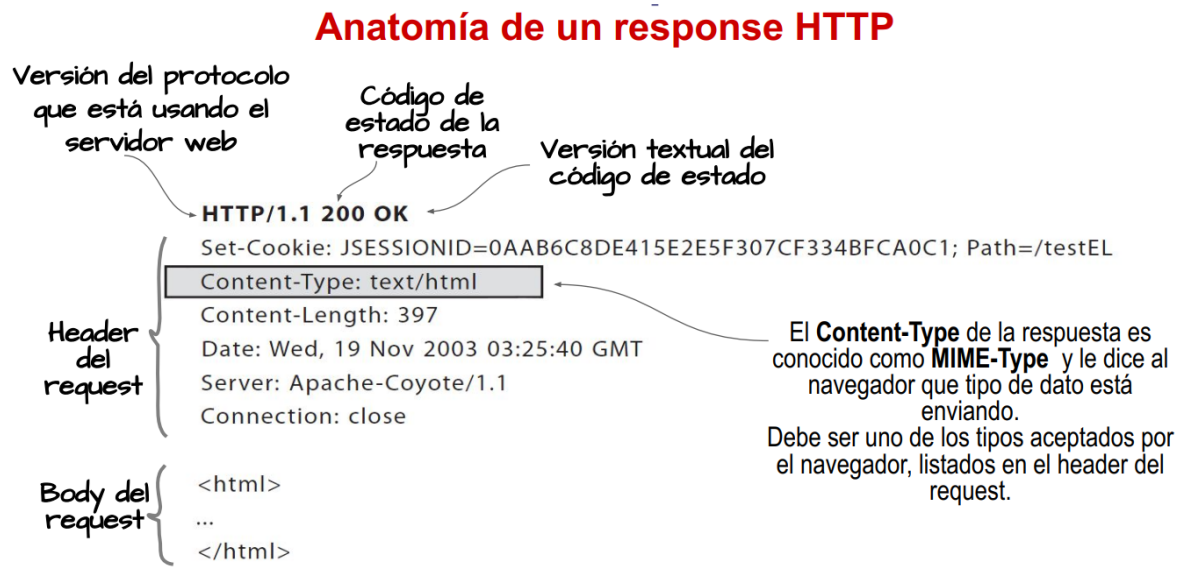
- Línea de Solicitud (Request Line): Esta línea contiene el método HTTP, el URI (Uniform Resource Identifier) del recurso y la versión del protocolo.
- Encabezados (Headers): Los encabezados proporcionan información adicional sobre la solicitud, como información de autenticación, tipos de contenido aceptados, información sobre el cliente, etc.
- Cuerpo (Body): Algunas solicitudes pueden contener un cuerpo que contiene datos enviados al servidor, como en el caso de las solicitudes POST.



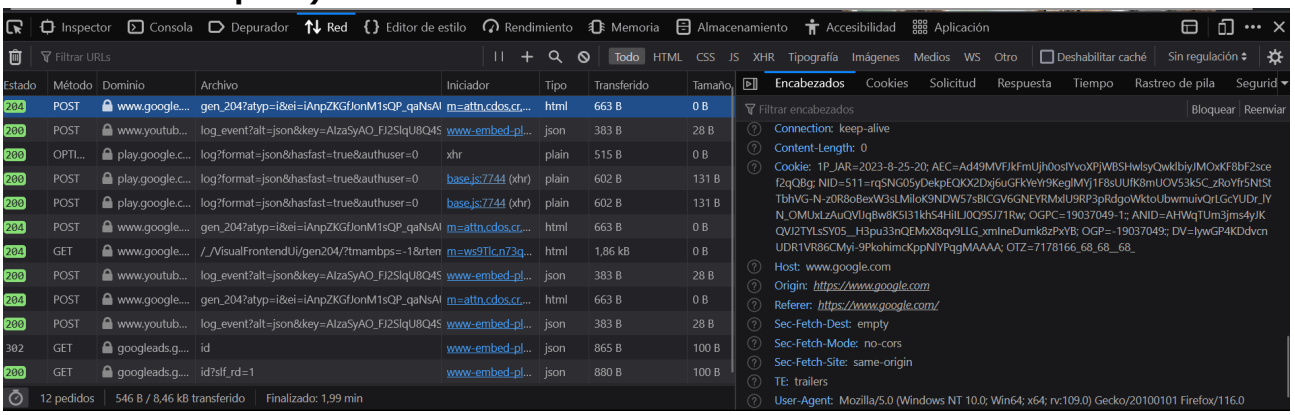


Mensajes de Respuesta (Response):
Los mensajes de respuesta son enviados por el servidor al cliente en respuesta a una solicitud previa. Estos mensajes contienen la información solicitada o los resultados de la acción realizada en el servidor. La estructura básica de un mensaje de respuesta es:

- Línea de Estado (Status Line): Esta línea contiene la versión del protocolo, el código de estado y una frase descriptiva.
- Encabezados (Headers): Al igual que en las solicitudes, los encabezados proporcionan información adicional sobre la respuesta, como el tipo de contenido, la longitud del contenido, la fecha de la respuesta, etc.
- Cuerpo (Body): El cuerpo de la respuesta contiene los datos que se envían de vuelta al cliente. Puede ser HTML, JSON, imágenes, archivos, etc., dependiendo del recurso solicitado y el resultado.



e. ¿Cómo sabe el servidor qué navegador está utilizando el visitante? (Revise la información provista por la herramienta para el desarrollador (F12) respecto de los Headers del Request)



Abajo a la derecha el encabezado User-Agent indica el navegador.
El User-Agent request header (en-US) es una cadena característica que le permite a los servidores y servicios de red identificar la aplicación, sistema operativo, compañía, y/o la versión del user agent (en-US) que hace la petición.

f. ¿Cómo se envían los parámetros en los mensajes con método POST?
En el protocolo HTTP, los parámetros se pueden enviar en los mensajes utilizando el método POST a través del cuerpo (body) de la solicitud, lo que es especialmente útil para enviar datos más grandes o sensibles, como formularios. Con el método GET los parámetros se pasan a través de la URL y tiene limitaciones –cada servidor soporta una cantidad limitada de caracteres.

g. ¿Qué son las Cookies? ¿Cómo se envían las Cookies a través de HTTP?
Una cookie HTTP, cookie web o cookie de navegador es una pequeña pieza de datos que un servidor envía al navegador web del usuario. El navegador guarda estos datos y los envía de regreso junto con la nueva petición al mismo servidor. Las cookies se usan generalmente para decirle al servidor que dos peticiones tienen su origen en el mismo navegador web lo que permite, por ejemplo, mantener la sesión de un usuario abierta. Las cookies permiten recordar la información de estado en vista a que el protocolo HTTP es un protocolo sin estado.

Las cookies se utilizan principalmente con tres propósitos:

- Gestión de Sesiones → Inicios de sesión, carritos de compras, puntajes de juegos o cualquier otra cosa que el servidor deba recordar
- Personalización → Preferencias de usuario, temas y otras configuraciones
- Rastreo → Guardar y analizar el comportamiento del usuario

Al recibir una solicitud HTTP, un servidor puede enviar un encabezado Set-Cookie con la respuesta. La cookie generalmente es almacenada por el navegador, y luego se envía con solicitudes hechas al mismo servidor dentro de un encabezado HTTP Cookie.

h. ¿Cuál es la diferencia entre HTTP y HTTPS? ¿Cuáles son los puertos por default (por defecto) de los mismos?
Los mensajes HTTP son de texto plano, lo que significa que las personas no autorizadas pueden acceder a ellos y leerlos fácilmente a través de Internet. Por el contrario, HTTPS transmite todos los datos de forma cifrada.
Para habilitar HTTPS, los sitios web necesitan un certificado SSL/TLS emitido por una autoridad de certificación confiable. Este certificado verifica la identidad del sitio web y garantiza la autenticidad de la conexión segura.
El puerto por defecto de HTTP es el 80 y de HTTPS es el 443.

2. Códigos de error HTTP. En función de la información que se encuentra disponible en la
siguiente URL, https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html, responda:
a. ¿Cómo se categorizan los tipos de respuestas HTTP según su código de estado?
Los tipos de respuestas HTTP se categorizan en:

1xx: Código informativo. Este código de estado indica una respuesta provisional, que consta únicamente de la línea de estado y encabezados opcionales, y termina con una línea vacía. No hay encabezados obligatorios para esta clase de código de estado. Dado que HTTP/1.0 no definió ningún código de estado 1xx, los servidores NO DEBEN enviar una respuesta 1xx a un cliente HTTP/1.0 excepto en condiciones experimentales.

2xx: OK. Esta clase de código de estado indica que la solicitud del cliente fue recibida, comprendida y aceptada exitosamente.

3xx: Redirecciòn. Indica que el agente de usuario debe tomar medidas adicionales para cumplir con la solicitud. La acción requerida PUEDE ser realizada por el agente de usuario sin interacción con el usuario si y sólo si el método utilizado en la segunda solicitud es GET o HEAD. Un cliente DEBE detectar bucles de redireccionamiento infinitos, ya que dichos bucles generan tráfico de red para cada redireccionamiento.

4xx: Error de parte del cliente. Está destinada a casos en los que el cliente parece haber cometido un error. Excepto cuando responde a una solicitud HEAD, el servidor DEBE incluir una entidad que contenga una explicación de la situación del error y si se trata de una condición temporal o permanente. Estos códigos de estado son aplicables a cualquier método de solicitud. Los agentes de usuario DEBEN mostrar cualquier entidad incluida al usuario.

5xx: Error de parte del servidor. Indican casos en los que el servidor sabe que ha cometido un error o es incapaz de realizar la solicitud.

b. Complete la siguiente tabla indicando el significado de los códigos de estado

Código	**Descripción**
200 OK	La solicitud ha tenido éxito. El contenido que es devuelto en la respuesta depende del método que se usó en la solicitud.
301 Movido permanente mente	Al recurso solicitado se le ha asignado un nuevo URI permanente y cualquier referencia futura a este recurso DEBE utilizar uno de los URI devueltos.
302 Encontrada	El recurso solicitado reside temporalmente en un URI diferente. Dado que la redirección puede modificarse en ocasiones, el cliente DEBE continuar usando el URI de solicitud para solicitudes futuras.
304 No modificada	Si el cliente ha realizado una solicitud GET condicional y se permite el acceso, pero el documento no ha sido modificado, el servidor DEBE responder con este código de estado.
403 prohibida	El servidor entendió la solicitud, pero se niega a cumplirla.
404 No encontrada	El servidor no ha encontrado nada que coincida con el URI de solicitud. No se da ninguna indicación de si la condición es temporal o permanente.

500 Error Interno del Servidor	El servidor encontró una condición inesperada que le impidió cumplir con la solicitud.

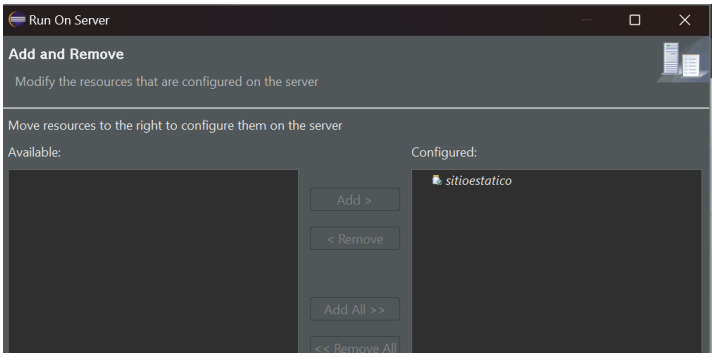
c. En el browser, escriba la URL: http://www.eldia.com, ¿cuáles de estos códigos aparecen?
Apareció el código 200.

d. Verifique como se redirecciona http://www.guarani-informatica.unlp.edu.ar a una conexión segura (use la pestaña Network de la herramienta para el desarrollador que brinda el browser, en el caso de usar Chrome, verifique que se encuentra chequeada la opción "Preserve Log"). ¿En qué campo de la respuesta se indica a donde redireccionar?
Hay varios códigos 200, pero en el momento del login y logout hubo códigos de redirección 302, y en el encabezado "location" indica hacia donde redireccionar.

3- HTML 4. (Referencia para más información:
https://developer.mozilla.org/en-US/Learn/HTML)
a. Usando eclipse cree un proyecto web estático llamado "sitioestatico".
b. Dentro de la carpeta "public" y usando el menú contextual, agregue una página HTML llamada paginav4.html (haciendo click en Next, le permitirá elegir la versión de HTML, elija versión html 4.01 Transitional)
c. En la página paginav4.html escriba entre los tags HTML <body> </body> el siguiente contenido

```
<div id="header">
   <h1>Mi sitio Web</h1>
</div>
<div id="menu">
   <ul>
      <li>Música</li>
      <li>Deporte</li>
      <li>El tiempo</li>
   </ul>
</div>
<div id="content">
   <h2>Sección de Música</h2>
   <div id="post">
      <h2>Artículo 1</h2>
      <p>Lorem ipsum….</p>
      <p>Vivamus sollicitudin molestie…. </p>
   </div>
   <div id="post">
      <h2>Artículo 2</h2>
      <p>Lorem ipsum….</p>
      <p>Phasellus consequat pretium….</p>
   </div>
</div>
<div id="footer">
   <p>&amp;#169; 2015 Mi sitio Web. Todos los derechos reservados.</p>
</div>
```

d. Para visualizar su página, vamos a publicarla en un servidor básico provisto por Eclipse.
i. Seleccione su página. Usando el menú contextual ejecute Run As... Run on Server
ii. Analice el contenido en esta ventana
iii. Seleccione el servidor Basic - HTTP Preview
iv. Haga click en Next . ¿Qué información se muestra en cada panel?



v. Haga click en Finish

e. Vamos a agregarle los estilos para visualizar un poco mejor la página. Agregue dentro de la sección <head></head> el siguiente contenido **codigo css**

4. Adaptación a HTML5. (Referencia para más información: https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5)

En este punto vamos a convertir la página HTML4 creada en el punto 3 a una página HTML5. Tenga en cuenta el significado de los siguientes tags en HTML5:

<HEADER>	encabezado de la página web
<FOOTER>	pié de página de la página web
<NAV>	Define un conjunto de links (enlaces) para navegar
<SECTION>	Se define un bloque de elementos relacionados.Normalmente contiene un heading (título)
<ARTICLE>	Se define un bloque de elementos relacionados, pero independiente, de modo que puede estar disponible “stand alone”, por ej. para ser usado en un RSS
<DIV>	Se define como un bloque de elementos hijos

Realice las siguientes modificaciones:

a. Copie y pegue su página paginav4.html sobre la carpeta WebContent y renombre este nuevo archivo como paginav5.html.

b. Modifique el encabezado: <!DOCTYPE html>
c. Modifique el meta de codificación: <meta charset="utf-8" />

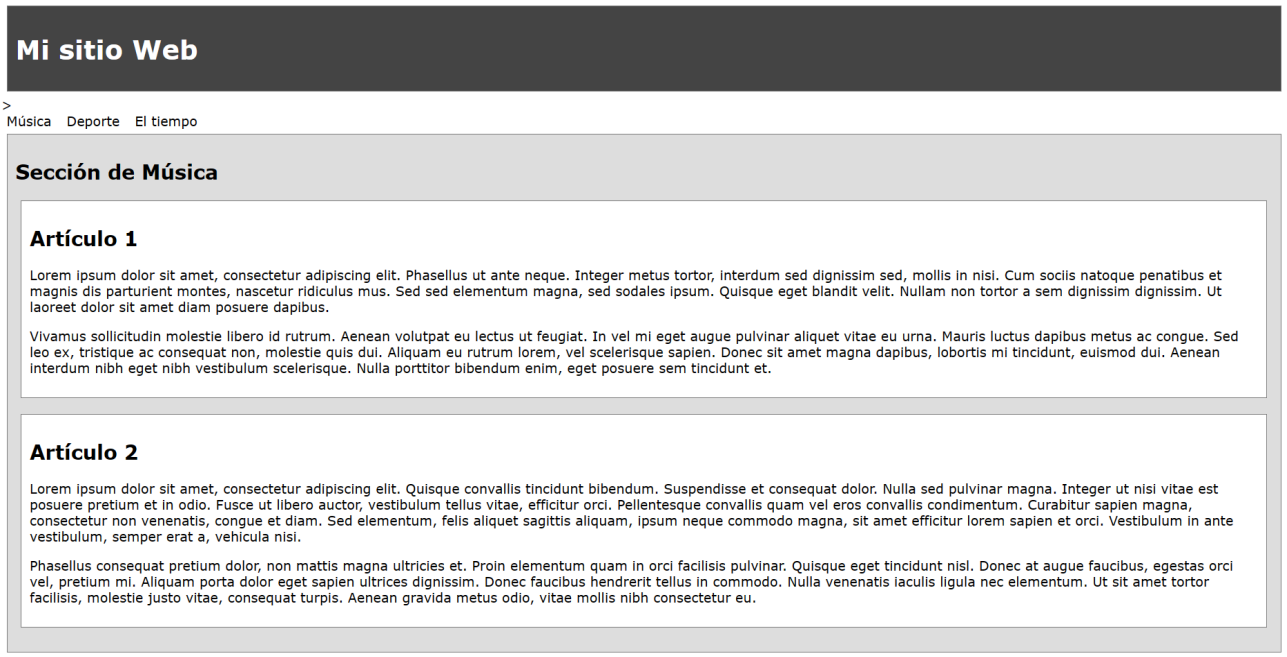
d. Adapte el encabezado (id=”header”) y pié de página (id=”footer”) usando las etiquetas <HEADER> y <FOOTER>

```
<header>
  <h1>Mi Sitio Web</h1>
</header>
.
.
.
<footer>
  <p>&#169; 2015 Mi sitio Web. Todos los derechos reservados.</p>
</footer>
```

e. Adapte el menú (id=”menu”) a HTML5 con la etiqueta <NAV>

```
<nav>
  <ul>
    <li>Música</li>
    <li>Deporte</li>
    <li>El tiempo</li>
  </ul>
</nav>
```

f. Adapte la sección principal (id=”content”) a HTML5 con la etiqueta <SECTION>



5. Estilos. (Referencia para más información: http://www.w3schools.com/css/css_intro.asp)
Los estilos pueden estar embebidos dentro de la página HTML ó asociados en un archivo externo. Esta última opción es la que vamos a usar en la práctica.
a. En el proyecto y dentro la carpeta WebContent cree una carpeta llamada estilos
b. Dentro de la carpeta estilos cree un archivo llamado estilo.css
c. Copie en el archivo estilo.css el contenido del tag <style>
d. Verifique que efectivamente ahora la página html perdió los estilos (puede necesitar borrar la caché del navegador).
e. Agregue a la página HTML la referencia a la hoja de estilos usando el tag:
<link rel="stylesheet" href="estilos/estilo.css" />

f. Ejecute su página en el servidor provisto por eclipse como lo hizo en el punto 3 y verifique que se visualiza correctamente el contenido.

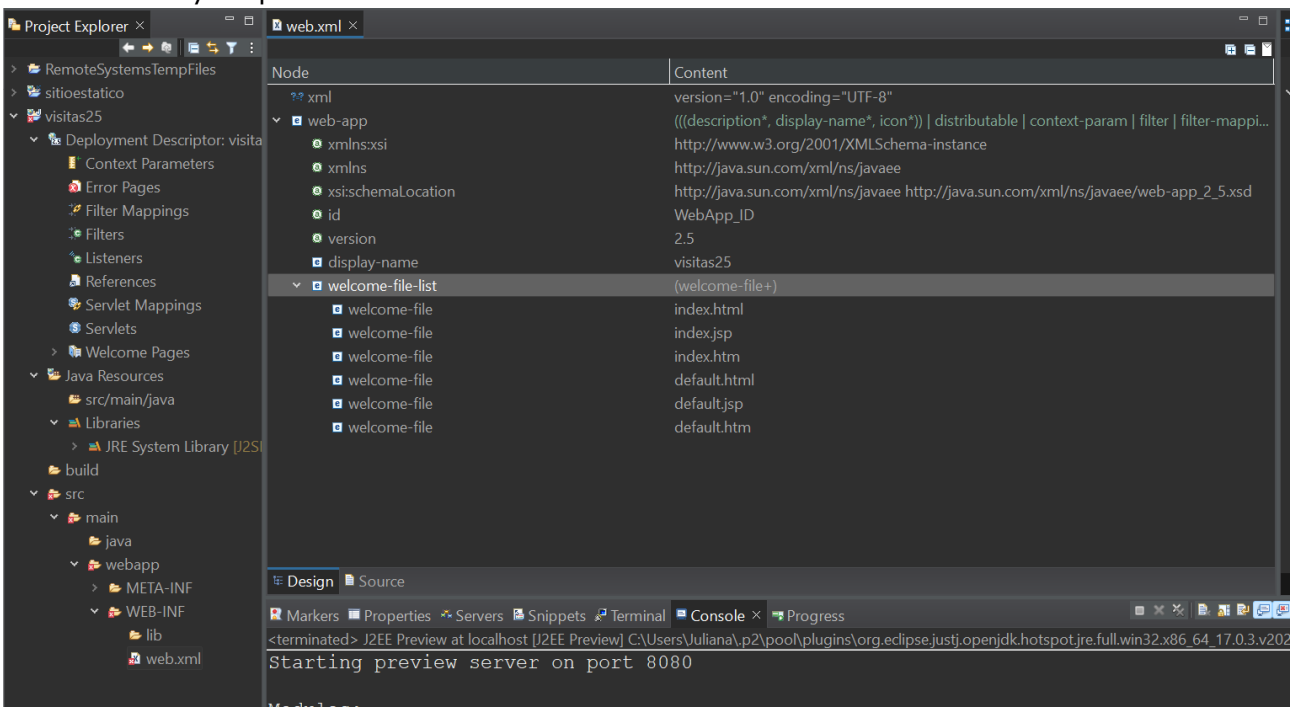
g. Modifique a su gusto la hoja de estilos (colores, tipografía, etc).

6. Web dinámica y Servlets.

Cree un proyecto web dinámico llamado visitas25, indicando en el combo "Dynamic Web module versión" la versión 2.5. Asigne como nombre de la aplicación web (context root) el nombre premio.

a. Analice la estructura de directorios que muestra la vista Project Navigator.

b. Despliegue la carpeta WEB-INF que está dentro de la carpeta webapp Analice la estructura de directorios y en particular el archivo web.xml.



Si está utilizando Tomcat 10, deberá actualizar el tag web-app con la especificación jakarta:
<web-app xmlns="https://jakarta.ee/xml/ns/jakartaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee
https://jakarta.ee/xml/ns/jakartaee/web-app_2_5.xsd" version="2.5">

c. Escriba una página HTML llamada index.html, que contenga un formulario con un campo de entrada de texto donde el usuario escribirá su nombre y un botón de tipo submit para enviar los datos del formulario al servlet del inciso d.

d. Escriba un servlet llamado Premio que:

- Tome el nombre del usuario del requerimiento HTTP (Request) que recibe como parámetro
- Mantenga actualizada la cantidad de requerimientos al servlet (visitas).
- Luego el servlet debe generar una página HTML para mostrar un mensaje personalizado. El mensaje personalizado debe leerse de un parámetro de inicialización del servlet. El mensaje debería contener 2 caracteres especiales que indiquen donde se insertarán el nombre y la cantidad de visitas, como se muestra a continuación:

  ¡Felicitaciones @! eres el visitante número # de nuestro sitio y has sido seleccionado para el gran premio TTPS - Cursada APROBADA

  Luego, si Charly completa el formulario y es el visitante 347, el resultado del servlet sería:

  ¡Felicitaciones Charly! eres el visitante número 347 de nuestro sitio y has sido seleccionado para el gran premio TTPS - Cursada APROBADA

  Nota: use el método replace de la clase String para el reemplazo de los caracteres especiales
- Analice nuevamente el contenido de la carpeta WEB-INF\classes y el archivo web.xml