|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TOMCAT** | | | | | | | **11/01/2023** |
|  |
|  |
|  |  | | | | | | |
| Vamos a la pag.web oficial de Tomcat y vamos a zona descarga:  <https://tomcat.apache.org/download-80.cgi>  Elegiremos para este caso la versión “[64-bit Windows zip](https://dlcdn.apache.org/tomcat/tomcat-8/v8.5.84/bin/apache-tomcat-8.5.84-windows-x64.zip)”. En esta tabla podemos ver la equivalencia según JEE necesitemos (v.3.1 que equivale a la v.7) con el TomCat que lo soporta: “https://tomcat.apache.org/whichversion.html”  Descargamos el \*.zip y lo descomprimimos en la carpeta donde tenemos nuestro proyecto de eclipse, en mi caso “D:\00Curso\CursosHarzeta\CURSO-Hazerta-1-2023”. Ahora tendremos una carpeta del servidor tomcat y al lado la de eclipse y la de workspace.  Nos vamos a **ECLIPSE**.  Donde lo teníamos abierto, con nuestro proyecto abierto. En las pestañas inferiores de eclipse, debajo de donde se abre la class, veremos una pestaña que pone Server. Ahí un texto que nos dice que agreguemos un servidor. Damos clic ahí.  En la ventana que sale tenemos que buscar “Apache”, tras desplegar este menú, buscar Tomcat 8.5 Server.  En configuración deberemos indicar o que nos aparezca “localhost” y “Tomcat v.8.5 Server at localhost”  (next)  Name: Apache Tomcat v8.5  Directory: buscamos ruta en donde se descomprimió el tomcat que nos descargamos y descomprimimos antes.  En mi caso “D:\00Curso\CursosHarzeta\CURSO-Hazerta-1-2023\apache-tomcat-8.5.84”  (finish)  Ya nos debe a ver agregado el tomcat al apartado de servidor de Eclipse, a parte que nos debe de a ver creado más carpetas.  Ahora abajo en donde antes estaba vacio, nos aparecerá el tomcat que hemos agregado, botón derecho encima de “server” y le damos a “star” o a la derecha al botón verde. La primera vez, Windows puede ser que nos consulte si queremos darle permiso para conectarse. Le damos siempre que sí.  Si miramos en “console”, nos aparecerá mucho texto en rojo, eso no quiere decir que sea error, sino que nos lo pone así. Hay que mirar la última línea que nos debe de poner que arranco correctamente y el tiempo que tardó en ms (milisegundos). | | | | | | | |
| **ARREGLO FALLO o ERROR que nos da en import y en métodos** | | | | | | | |
| Nos situamos encima de “primer-proyecto” y le damos a botón derecho/propiedades/**Project Facets.**  Nos aparece una ventana como la que se ve en la imagen. Nos aparece el “dynamic Web Module” (versión JEE 3.1), “Java” (v.17) y “JavaScript” (v.1.0). Lo dejamos todo lo que nos pone señalado. Y pulsamos a la derecha en “Runtimes”.  Ahí señalamos el servidor que agregamos y que ahora nos aparece. “Apache Tomcat v 8.5”.  Damos a “Apply and Close”. | |  | | | | | |
|  |  | | | | | | |
| Listo, ya se deben de a ver quitado los errores que nos aparecía antes y debe aparecer en el explorer de nuestra izq. Una carpeta “Server”.  En JEE a parte de la especificaciones, nace falta la implementación que lo pone el servidor. Se complica cuando tengamos que añadir cosas específicas. Todo normalmente no lo encontraremos vacio y lo tenemos que ir rellenando. | | | | | | | |
|  | | | | |  | | |
| Un **Servlet** se crea por herencia y de clase abstracta. Establecida la herencia, puedes sobrescribir lo que desees. Nunca instanciamos y ni llamamos a los métodos de una clase, lo hará el servidor por nosotros. Sobrescribimos cualquier método que haya llamado el servidor.  Java no permite herencia múltiple, sino en cascada.  Nos colocamos en el código de “Primer\_Servlet.java” y en cualquier parte del código le damos a botón derecho: Source/Override.Implements Methods,… Elegimos los métodos init y destroy. Y Tras dale a ok, nos aparecerá estas líneas de código en el archivo que tenemos abierto. | | | | |
| Si escribimos un poco de código y le pulsamos “**Contrl + BarraEspaciadora**” 🡪 te sale automaticamente | | | | | | | |
| **@WebServlet (description=”Primer Servlet de prueba …** 🡪 son **anotaciones** que aportan información de la clase, se denominan metadatos, puede ser que te lo exijan en el proyecto. En este caso indican que están definiendo el componente Servlet con una serie de parámetros.  Sintaxis:  @nombre (parámetro = “….”, parámetro=”….”, etc)  Es una forma de definir componentes, en el servidor también se le pude hacer en forma de ficheros. En la definición en el proyecto te pueden indicar como hacer los comentarios, si te lo exigen. | | | | | | | |
| **Programáticamente** 🡪 está en la v.3.1 o v.4.0 y es otra forma de arrancar la aplicaciones y definir componentes; se crea una clase estableciendo una herencia a partir de la clase “web-aplication”, a partir de ahí se instancia todos los componentes y el tomcat sabe que tiene que llamar a las distintas clases para llamar a los componentes. (no la piden en la certificación)  La petición solo entrará en la clase si tiene la url que viene definida en el **urlPatterns**, sino es la misma, no entrará en el componente y no hará lo que indiquemos en la clase.  El nombre del Servlet no debe porque figurar en la url y por tanto no tiene por qué ser iguales.  **CONFIGURACIÓN SERVIDOR**  Doble clic en  y nos saldrá la siguiente pantalla: | | | | | | | |
| |  | | --- | |  |  |  |  | | --- | --- | | Importante que debemos tener señalado la opcion “Use Tomcat installation (takes control of Tomcat installation)”. Los puertos dejamos los que están a menos que estemos seguro de que van a estar ocupados.  Y otra que podemos cambiar es el “Timeouts” que podemos ponerlo a 120 el Star y a 120 ó 125 el Stop. Pero estas dos es opcional. Lo importane es tener señalado el “Use Tomcat installation….”.  Si no te aparece habilitado para marcarlo, mira a ver que servidor este parado y que no tengas añadido ningun proyecto a él. |  | |  | | | | | | | | |
| **AÑADIR PROYECTO AL SERVIDOR**  Tras esto, botón derecho encima de Server “Tomcat v8.5…” (como antes) pero ahora le damos a “**Add and Remove**” y nos sale la pantalla que vemos a la izquierda.  Ahí nos debe aparecer nuestro proyecto en el cuadro de la izquierda, señalamos y damos “Add” y se nos pasará al cuadro de la derecha. A continuación a (Finish).  Y ya debe aparecernos como aquí abajo. Añadido al servidor y listo para ser probado. | | | | | |  | |
| **CONFIGURACIÓN DEL EXPLORER** | | | | | | | |
| Vamos aconfigurar la salida de nuestro proyecto, de cara para probarlo, para que nos salga en el explorer predeterminado que nos trae Eclipse, y no en otro que tengamos instalado en nuestro computador. Para proyecto pequeños y con poco estilo o diseño puede venir muy bien, si es ya más complejo mejor utilizar uno que tengamos en el ordenador como Mozilla, Edge, Chrome,… (el que deseemos).  En eclipse vamos a la pestaña superior del menu “Window” y le damos a “Preference”, ahí vamos a “General/Web Browser”.  Debemos elegir la pestaña “Use internal web browser” y abajo que se quede señalado "Default system web browser”.  Damos a Apply and close. | | |  | | | | |
| **PROBANDO EL CÓDIGO**  Ya lo tenemos listo para probar lo que llevamos hecho de código.  Debemos arrancar servidor dandole boton derecho “star” o en el boton que nos aparece abajo a la derecha en verde, una vez que hayamos señalado nuestro servidor “Tomcat v8.5 …”  Ahora le damos arriba en los accesos rapido de menu al circulo con forma de globo terrestre “Open Web Browser”.    Se nos abre una ventana del explorer donde nos deja poner una url …  Sintaxis: [Protocolo://dominio:puerto/proyecto/](http://localhos:puerto/root/UrlPatter(del)componenteProyecto    Le damos al intro y nos debe salir lo que hemos escrito como respuesta en el “Primer\_Servlet.java”. En consola debe aparecer el texto que le pusimos en el constructor y en el metodo init. Y arriba nos aparecerá el texto que pusimos en el metodo doGet.  **CICLO DE VIDA**, este será su recorrido:  Servidor recibe petición sobre ese componente, servidor crea componente y llama al constructor, a continuación llama al metodo init; y despues llama al metodo doget. El metodo post no se ejecuta (no pasa por ahí) porque no lo tiene indicado como tal.  Metodo Get 🡪 aparece la información que se pase en la barra de dirección del explorer.  Metodo Post 🡪 no aparece ninguna información en la barra direción del explorer.  Esto se repetira todas la veces, podemos darle al intro en url y debe de repetirno el recorrido de antes, hasta que paremos el servidor. Es cuando la aplicación muere.  Al arrancar de nuevo el servidor, todo se sincroniza y todo se queda a la escucha de la petición que le llege. Cuando llegue la petición es cuando el servidor crea el componente. Todo no tiene porque ser así, el ciclo de vida, puede ser que haya cosas que al arrancar el servidor ya las cree, porque así se le indique. | | | | | | | |
| **PETICION DE DOMINIO**  Retocamos el web.xml y le indicamos en la etiqueta <welcome-file> una dirección donde estará nuestra pag jsp (/jsp/inicio.jsp) que podrá ser un formulario o lo que queramos, en este caso remite a nuestra pag.web anterior creada de inicio.html y que esta mandará mostrar el mensaje que le pusimos. | | | |  | | | |
| En el url ponemos:  <http://localhost:8080/primer-proyecto/>  Esta vez se puso todo en el post, por tanto en la barra direcciones no aparece nada y nos muestra en “ver codigo fuente” con boton derecho, un codigo html plano, sin nada extraño. | | | | | | | |
|  | | | | | |  | |
|  |  | | | | | | |