



TÉCNICO SUPERIOR EN DESARROLLO DE
APLICACIONES WEB

Departamento de Informática

PROYECTO

Abatech

Manual Técnico

Autor/es: Alejandro Rodríguez Álvarez

Curso Académico: DAW 2ºB

Índice

- 1. Introducción**
- 2. Arquitectura de la aplicación**
 - 2.1. Frontend**
 - 2.1.1. Tecnologías usadas**
 - 2.1.2. Entorno de desarrollo**
 - 2.2. Backend**
 - 2.2.1. Tecnologías usadas**
 - 2.2.2. Entorno de desarrollo**
- 3. Documentación técnica**
 - 3.1. Análisis**
 - 3.2. Desarrollo**
 - 3.3. Pruebas realizadas**
- 4. Proceso de despliegue**
- 5. Propuesta de mejoras**
- 6. Bibliografía**

1. Introducción

El presente proyecto consiste en el desarrollo de una tienda en línea utilizando tecnologías modernas para garantizar un sistema robusto, escalable y seguro. La aplicación ha sido diseñada siguiendo el patrón de arquitectura MVC (Modelo-Vista-Controlador), lo que facilita la organización y mantenimiento del código. Además, se han implementado diversos patrones de diseño como DAO (Data Access Object) y Factory para la gestión eficiente de los datos en la base de datos.

Funcionalidades Desarrolladas

1. Gestión de Usuarios

- Registro de nuevos usuarios con validación de datos en el lado cliente y servidor.
- Inicio de sesión seguro mediante credenciales encriptadas con MD5.
- Recuperación y modificación de datos de perfil, exceptuando el correo electrónico y NIF.
- Cambio de contraseña con validación de contraseña actual.
- Carga y actualización de imagen de perfil (avatar).

2. Catálogo de Productos

- Visualización de productos con sus detalles, incluyendo nombre, descripción, precio e imagen.
- Clasificación de productos por categorías.
- Búsqueda de productos mediante filtros avanzados (nombre, precio, categoría, marca).

3. Carrito de Compras

- Agregar productos al carrito con la posibilidad de modificar la cantidad.
- Persistencia del carrito para usuarios anónimos por un período de dos días mediante cookies.

- Sincronización automática del carrito al iniciar sesión.
- Eliminación de productos del carrito o del carrito completo.
- Actualización en tiempo real de la cantidad de productos mediante AJAX.

4. Gestión de Pedidos

- Generación de pedidos a partir del carrito de compras.
- Cálculo del importe total del pedido, incluyendo el IVA.
- Historial de pedidos para usuarios registrados, con detalle de cada compra.
- Estados de pedidos: en carrito y finalizado.

5. Seguridad y Validaciones

- Uso de sesiones para gestionar la autenticación de usuarios.
- Validaciones en el lado cliente con JavaScript y en el lado servidor con Java EE.
- Protección contra inyecciones SQL y XSS.
- Encriptación de contraseñas para mejorar la seguridad.

Este proyecto ha sido desarrollado utilizando Java EE 7, Java 11, MySQL 11, Apache Tomcat 9, Bootstrap y Maven. Se han implementado pruebas automatizadas con JUnit y Selenium para garantizar el correcto funcionamiento del sistema.

A lo largo del desarrollo se ha buscado optimizar el rendimiento y asegurar una experiencia de usuario fluida e intuitiva. La organización del código en paquetes específicos y la utilización de tecnologías modernas han permitido construir una aplicación flexible y escalable.

2. Arquitectura de la aplicación

Tecnologías Utilizadas en el Desarrollo de la Aplicación

1. Base de Datos

1.1. Tecnología Usada

- **MySQL 11:** Se ha elegido MySQL debido a su eficiencia en la gestión de bases de datos relacionales, su amplia documentación y compatibilidad con Java EE.

1.2. Justificación

- MySQL es un sistema de gestión de bases de datos robusto, con una gran comunidad de soporte y optimización para aplicaciones web de comercio electrónico. Además, su capacidad para manejar grandes volúmenes de datos y su facilidad de integración con Java lo convierten en la elección ideal para este proyecto. Su escalabilidad permite adaptarse a futuras expansiones del sistema sin comprometer el rendimiento.

2. Frontend

2.1. Tecnologías Usadas

- **Bootstrap:** Utilizado para el diseño y estilización de la interfaz gráfica, proporcionando un diseño responsivo y adaptable a diferentes dispositivos.
- **JSP, JSTL y Expression Language (EL):** Se usaron para la generación dinámica de contenido y la lógica de presentación, permitiendo la reutilización de código y facilitando la separación entre la lógica y la interfaz de usuario.

2.2. Entorno de Desarrollo

- **IntelliJ IDEA:** Se utilizó debido a su integración con herramientas de desarrollo Java EE, su facilidad de uso y sus potentes herramientas de depuración.
- **Apache Tomcat 9:** Servidor de aplicaciones utilizado para desplegar la aplicación, proporcionando un entorno estable y seguro para la ejecución de los servlets y JSP.

3. Backend

3.1. Tecnologías Usadas

- **Java 11 con Java EE 7:** Se ha utilizado Java EE por su modularidad y soporte para aplicaciones empresariales escalables, permitiendo la implementación de patrones de diseño como DAO y MVC.
- **Maven:** Para la gestión de dependencias y construcción del proyecto, facilitando la administración de librerías y la automatización del proceso de compilación y empaquetado.
- **Jakarta Servlet API 4.0:** Para la gestión de peticiones HTTP, garantizando una interacción eficiente entre el cliente y el servidor.

3.2. Entorno de Desarrollo

- **IntelliJ IDEA:** IDE utilizado para el desarrollo de la aplicación, ofreciendo herramientas avanzadas para la gestión del código y depuración.
- **Apache Tomcat 9:** Entorno de ejecución para los servlets y JSP, asegurando un comportamiento estable y óptimo de la aplicación.

4. Herramientas para Pruebas

4.1. Tecnologías Usadas

- **JUnit 5:** Para la ejecución de pruebas unitarias en la lógica de negocio, garantizando la fiabilidad del código.
- **Mockito 3.9:** Para la simulación de dependencias en pruebas, permitiendo probar componentes de manera aislada.
- **Selenium WebDriver:** Para pruebas de interfaz de usuario y automatización, asegurando que la aplicación funcione correctamente en diferentes navegadores y dispositivos.

4.2. Justificación

- Estas herramientas permiten realizar pruebas automatizadas, asegurando la estabilidad y calidad del software. Las pruebas unitarias y de integración garantizan que los módulos funcionen correctamente antes de ser implementados en producción. Además, la automatización de pruebas de interfaz mejora la experiencia del usuario al reducir errores visuales y funcionales.

5. Estructura del Proyecto

El proyecto sigue el patrón **MVC (Modelo-Vista-Controlador)** y está organizado de la siguiente manera:

- **Modelo (Model):** Contiene las clases DTO y DAO encargadas de la gestión de datos, asegurando una correcta separación de responsabilidades.
- **Vista (View):** Basada en JSP, JSTL, Bootstrap y CSS para la presentación de la aplicación, asegurando un diseño atractivo y funcional.
- **Controlador (Controller):** Implementado con Servlets y Java EE para la lógica de negocio, manejando las peticiones del usuario y la interacción con la base de datos.

5.1. Estructura de Directorios

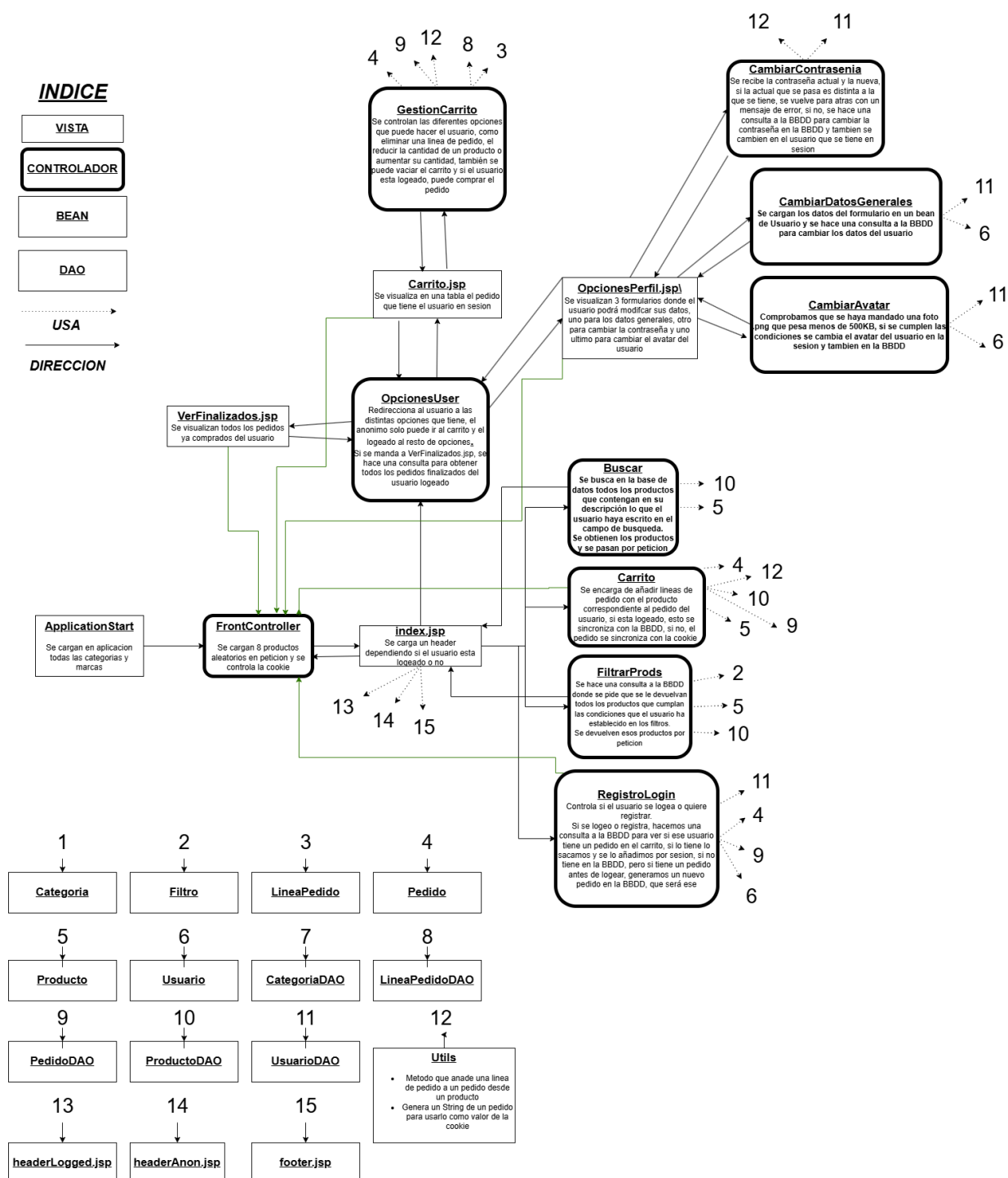
```
/Abatech
├── src/main/java/es.abatech
│   ├── beans
│   ├── controllers
│   ├── DAO
│   ├── DAOFactory
│   ├── events
│   ├── filters
│   └── models
├── src/main/webapp
│   ├── CSS
│   ├── IMG
│   ├── INC
│   ├── JS
│   ├── JSP
│   ├── META-INF
│   ├── WEB-INF
│   └── index.jsp
```

Esta organización garantiza la separación de responsabilidades y facilita el mantenimiento del código. La estructura modular permite una mayor escalabilidad y facilita la incorporación de nuevas funcionalidades sin afectar el rendimiento del sistema.

6. Diagramas del Proyecto

A continuación, se presentan los diagramas utilizados en el desarrollo del proyecto:

- **Diagrama de Flujo:** Representa la secuencia de interacciones en la aplicación, permitiendo visualizar el comportamiento del sistema y optimizar los procesos.



Proyecto “Desarrollo de Aplicaciones Web”

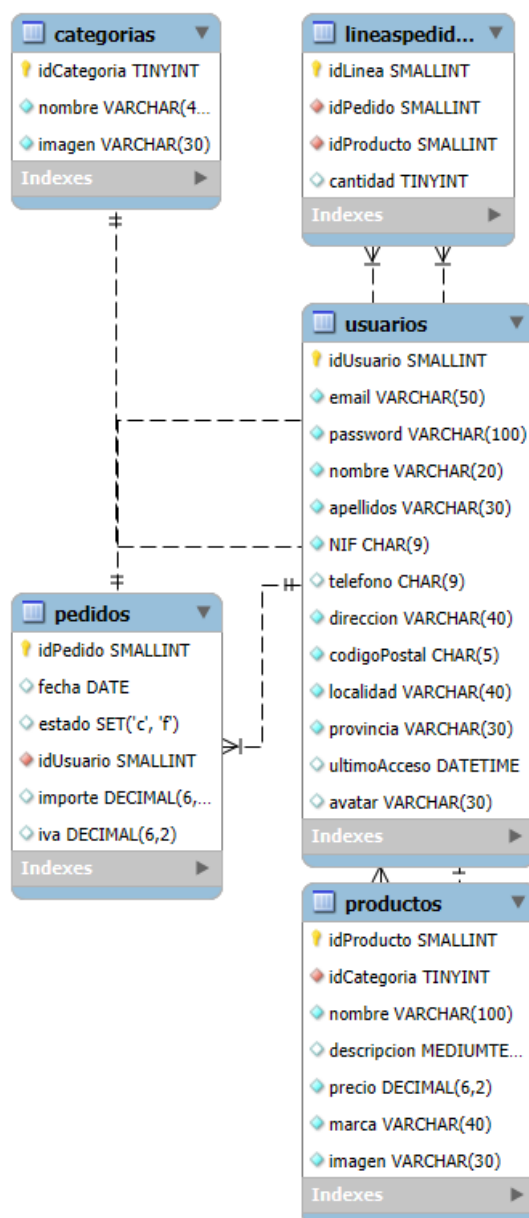
Título del Proyecto:



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo

- **Diagrama de Entidad-Relación:** Muestra la estructura de la base de datos y sus relaciones, asegurando la correcta organización y almacenamiento de la información.



3. Documentación técnica

El desarrollo de la aplicación ha requerido un proceso meticuloso de análisis, diseño, implementación y pruebas para garantizar su correcto funcionamiento, escalabilidad y mantenibilidad. En esta sección se recoge la documentación técnica generada a lo largo del proyecto, con especial énfasis en los diagramas utilizados, la arquitectura de la solución y las pruebas realizadas para validar el software.

3.1. Análisis

La fase de análisis es fundamental para la correcta planificación y estructuración de la aplicación. Durante esta etapa se han elaborado diversos diagramas que han permitido definir con precisión los requisitos del sistema, las interacciones entre los actores y los componentes que conforman la solución.

Diagramas Utilizados

1. Diagrama de Casos de Uso

- Representa las interacciones entre los distintos actores del sistema (usuarios, administradores, sistemas externos) y las funcionalidades que proporciona la aplicación.
- Permite identificar los requisitos funcionales y definir el alcance del proyecto.

2. Diagrama de Clases

- Muestra la estructura de las clases que componen la aplicación, definiendo sus atributos, métodos y relaciones.
- Es clave para diseñar una arquitectura modular, escalable y orientada a objetos.

3. Diagrama de Entidad-Relación (DER)

- Representa el modelo de datos del sistema, estableciendo las entidades, sus atributos y las relaciones entre ellas.
- Permite optimizar la base de datos, asegurando la integridad y evitando redundancias en la información.

4. Diagrama de Componentes

- Describe los diferentes módulos que conforman la aplicación y sus relaciones.
- Ayuda a definir la separación de responsabilidades y la interacción entre las capas del sistema.

5. Diagrama de Despliegue

- Representa la infraestructura donde se ejecutará la aplicación, incluyendo servidores, bases de datos, conexiones de red y otros elementos técnicos.
- Es útil para planificar el despliegue y la distribución de los servicios en el entorno de producción.

Documentación Adicional

Además de los diagramas mencionados, se ha generado documentación detallada que describe los requisitos funcionales y no funcionales, los flujos de trabajo de los usuarios y las especificaciones técnicas clave para el desarrollo del sistema.

3.2. Desarrollo

Una vez definido el análisis del sistema, se ha procedido a la fase de desarrollo, donde se han implementado las funcionalidades y se han creado diversos diagramas que detallan el comportamiento del software en ejecución.

Diagramas Utilizados

1. Diagrama de Secuencias

- Representa la interacción entre los diferentes objetos del sistema a lo largo del tiempo.
- Permite visualizar el flujo de mensajes entre los componentes del sistema en escenarios específicos.

2. Modelo Lógico del Sistema

- Detalla la organización de los módulos, su relación y cómo interactúan entre ellos.
- Facilita la estructuración del código y la división del trabajo en el equipo de desarrollo.

3. Diagrama de Estados

- Representa los distintos estados por los que puede pasar una entidad del sistema.
- Útil para modelar comportamientos dinámicos, como el ciclo de vida de una orden de compra.

4. Diagrama de Flujo de Datos

- Muestra cómo la información fluye a través del sistema, desde la entrada hasta la salida.
- Es clave para comprender el procesamiento de datos y mejorar el rendimiento de la aplicación.

Documentación Adicional

Además de los diagramas técnicos, se ha elaborado documentación sobre las tecnologías utilizadas, las decisiones arquitectónicas y las mejores prácticas aplicadas en el desarrollo del sistema.

3.3. Pruebas Realizadas

Para garantizar la estabilidad y correcto funcionamiento de la aplicación, se han llevado a cabo diversas pruebas en distintas etapas del desarrollo. Estas pruebas han permitido detectar y corregir errores, asegurando que el software cumple con los requisitos establecidos.

Tipos de Pruebas Realizadas

1. Pruebas Unitarias

- Se han evaluado métodos individuales de las clases más críticas del sistema.
- Se han utilizado datos de entrada controlados para verificar el correcto funcionamiento de cada componente de manera aislada.

2. Pruebas de Integración

- Se ha comprobado la comunicación entre los distintos módulos del sistema.
- Se ha verificado que la integración entre el backend, frontend y base de datos funcione correctamente.

3. Pruebas Funcionales

- Se ha validado que cada funcionalidad de la aplicación se comporte según lo esperado.
- Se han probado los principales flujos de usuario, asegurando una experiencia óptima.

4. Pruebas de Rendimiento

- Se ha medido la velocidad de respuesta del sistema bajo diferentes niveles de carga.
- Se ha optimizado el acceso a la base de datos y el procesamiento de datos en el backend.

5. Pruebas de Seguridad

- Se han evaluado posibles vulnerabilidades en la autenticación, gestión de sesiones y acceso a datos sensibles.
- Se han implementado medidas de seguridad para prevenir ataques como SQL Injection y Cross-Site Scripting (XSS).

Estructura de la Documentación de Pruebas

Para cada prueba realizada, se ha documentado la siguiente información:

- **Componente Evaluado:** Se especifica la clase, método o módulo probado.
- **Juego de Datos de Entrada:** Se describe la información utilizada en cada prueba.
- **Resultado Esperado y Resultado Obtenido:** Se comparan los valores esperados con los valores generados en la ejecución de la prueba.
- **Herramientas de Pruebas Usadas:**

Herramienta	Descripción
JUnit 5	Para pruebas unitarias y validación de la lógica de negocio.
Mockito 3.9	Para simular dependencias y probar la interacción entre módulos.
Selenium WebDriver	Para pruebas automatizadas en la interfaz de usuario.
Postman	Para pruebas de servicios REST y validación de API.

Conclusión

La documentación técnica generada ha sido clave para la correcta implementación del sistema. Los diagramas han facilitado la comprensión de la arquitectura y las interacciones del

Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo

software, mientras que las pruebas realizadas han asegurado su calidad y correcto funcionamiento.

Esta documentación servirá como referencia para futuras ampliaciones y mantenimiento del sistema, garantizando una base sólida para el crecimiento del proyecto.

4. Proceso de despliegue

1. Introducción

El despliegue de aplicaciones web requiere la elección de un servidor adecuado que garantice un correcto funcionamiento, estabilidad y seguridad. En este proyecto, hemos decidido utilizar **Apache Tomcat 9** como servidor de aplicaciones debido a su compatibilidad con **Java EE 7**, su eficiencia en el manejo de aplicaciones basadas en **servlets y JSP**, y su facilidad de configuración y administración. A continuación, se detallan las razones por las cuales Tomcat ha sido la mejor elección para este despliegue.

2. Características y Ventajas de Apache Tomcat

2.1. Compatibilidad con Java EE y JSP

Dado que nuestra aplicación está desarrollada en **Java EE 7**, utilizando tecnologías como **Servlets, JSP, JSTL y Expression Language (EL)**, necesitábamos un servidor de aplicaciones que brindara soporte nativo para estas tecnologías. **Apache Tomcat** es un servidor ampliamente utilizado para aplicaciones basadas en **Java EE** y proporciona un entorno de ejecución optimizado para estas tecnologías.

2.2. Ligereza y Eficiencia

A diferencia de otros servidores de aplicaciones más pesados como **WildFly o GlassFish**, **Tomcat es una solución ligera y eficiente** que permite desplegar aplicaciones sin un consumo excesivo de recursos. Esto es especialmente útil para aplicaciones que no requieren un servidor de aplicaciones empresarial completo, sino un contenedor web optimizado.

2.3. Facilidad de Configuración y Administración

Tomcat es conocido por su facilidad de instalación y configuración. La estructura modular de su configuración mediante archivos **server.xml y web.xml** permite personalizar la ejecución de la aplicación sin una curva de aprendizaje elevada. Además, su integración con **IntelliJ IDEA** facilita el desarrollo y prueba de la aplicación antes de su despliegue en producción.

2.4. Seguridad y Gestión de Usuarios

Tomcat ofrece mecanismos integrados para la gestión de autenticación y autorización mediante **Realms**, **JDBC Realms** y **JAAS**, lo que permite una administración segura de los usuarios. Esto es fundamental en aplicaciones de comercio electrónico, donde la protección de los datos de los usuarios es prioritaria.

2.5. Despliegue Flexible y Soporte para WAR Files

Nuestra aplicación está empaquetada en un archivo **WAR (Web Application Archive)**, un formato estándar para el despliegue de aplicaciones Java en servidores. Tomcat permite desplegar aplicaciones mediante la simple copia del archivo **WAR** en su directorio de aplicaciones, facilitando la actualización y mantenimiento del sistema.

3. Comparación con Otras Alternativas

Aunque existen otros servidores de aplicaciones como **GlassFish**, **JBoss (WildFly)** y **Payara**, Tomcat ha sido la mejor opción para este proyecto por las siguientes razones:

- **GlassFish y Payara** son servidores Java EE completos, pero su administración es más compleja y requieren más recursos que Tomcat.
- **JBoss (WildFly)** está orientado a aplicaciones empresariales más grandes y con una mayor carga de trabajo, lo que lo hace innecesario para este proyecto.
- **Tomcat** es más ligero, fácil de administrar y suficiente para ejecutar aplicaciones basadas en **servlets** y **JSP** sin necesidad de configuraciones avanzadas.

4. Integración con el Entorno de Desarrollo

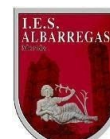
El uso de **IntelliJ IDEA** junto con **Apache Tomcat 9** permite un flujo de trabajo eficiente, con integración directa para la depuración y prueba de la aplicación antes de su despliegue final. Además, Tomcat se integra fácilmente con **Maven**, lo que facilita la gestión de dependencias y la construcción del proyecto.

5. Proceso de despliegue

Para desplegar nuestro proyecto en Tomcat, tendremos primero instalar tomcat en nuestro equipo, eso lo haremos desde la página de Tomcat.

Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo

La imagen muestra la interfaz web de Apache Tomcat. En la parte superior, hay un navegador con la URL `tomcat.apache.org/download-90.cgi`. El encabezado incluye el logo de Tomcat y el de la Fundación Apache. A la izquierda, un menú de navegación con categorías como 'Apache Tomcat', 'Download', 'Documentation' y 'Maven Plugin'. El contenido principal, titulado 'Tomcat 9 Software Downloads', contiene: un mensaje de bienvenida; un enlace 'Quick Navigation' a las versiones; una sección 'Release Integrity' que explica la verificación de firmas; una lista de 'Mirrors' con un selector de URL; y una sección '9.0.99' con un enlace a 'README'. En la parte inferior, se muestra la lista de distribuciones binarias, comenzando con 'Core'.

En este caso instalaremos la versión 9, una vez la tengamos descargado, tendremos un zip que extraeremos en nuestro equipo, una vez extraído, iremos a la ruta de la carpeta desde la consola de comandos.

```
C:\Windows\System32\cmd.e
Microsoft Windows [Versión 10.0.26100.3037]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\tomcat\apache-tomcat-9.0.94>
```

Una vez ahí, iremos a la ruta de `/bin` y ejecutaremos el archivo `catalina.bat`, junto a `start`, el comando será este.

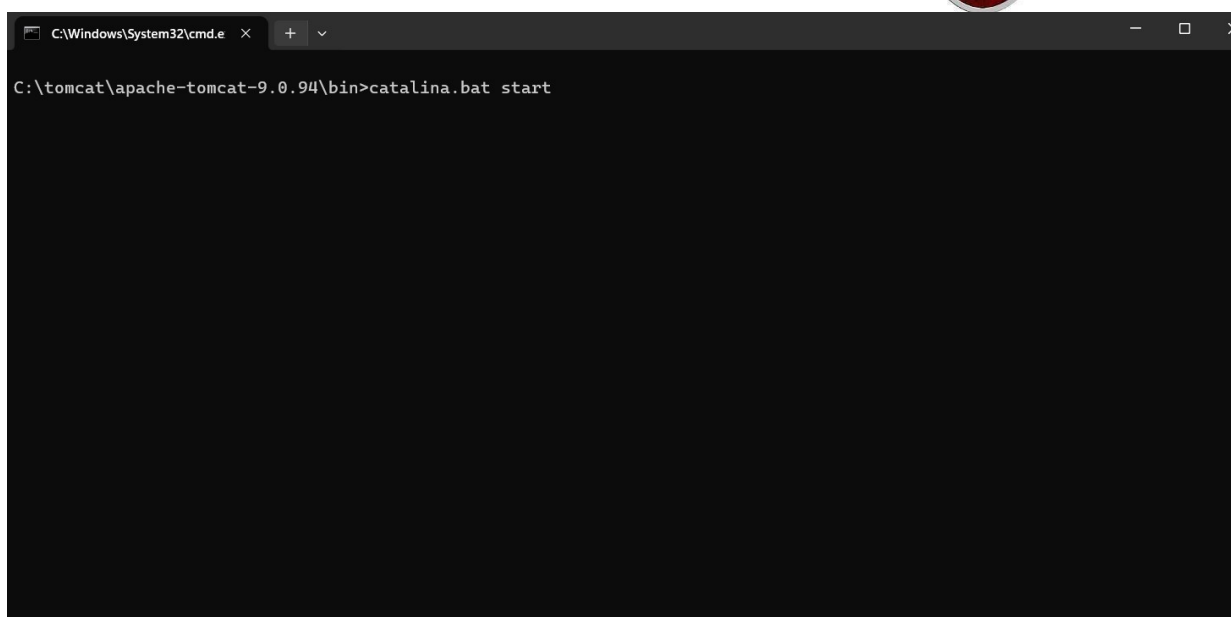
Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:

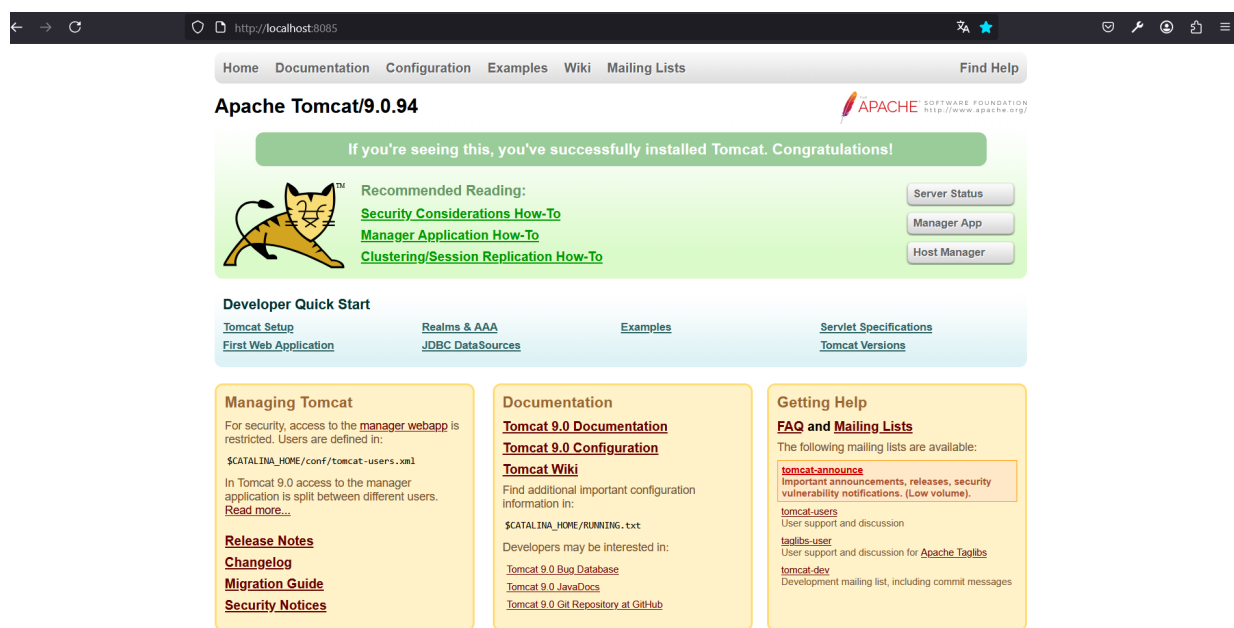


JUNTA DE EXTREMADURA

Consejería de Educación y Empleo



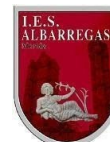
Una vez hecho esto, tomcat estará en funcionamiento. Para acceder a la zona de despliegue de tomcat, en nuestro navegador pondremos la IP del equipo donde está tomcat funcionando y accederemos a la pagina por defecto de tomcat.



Antes de hacer nada más, crearemos la base de datos y los usuarios necesarios para el uso de la aplicación, para eso usaremos el script de creación que está subido en Github, en la ruta [/BBDD/Abatech.sql](#).

Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:

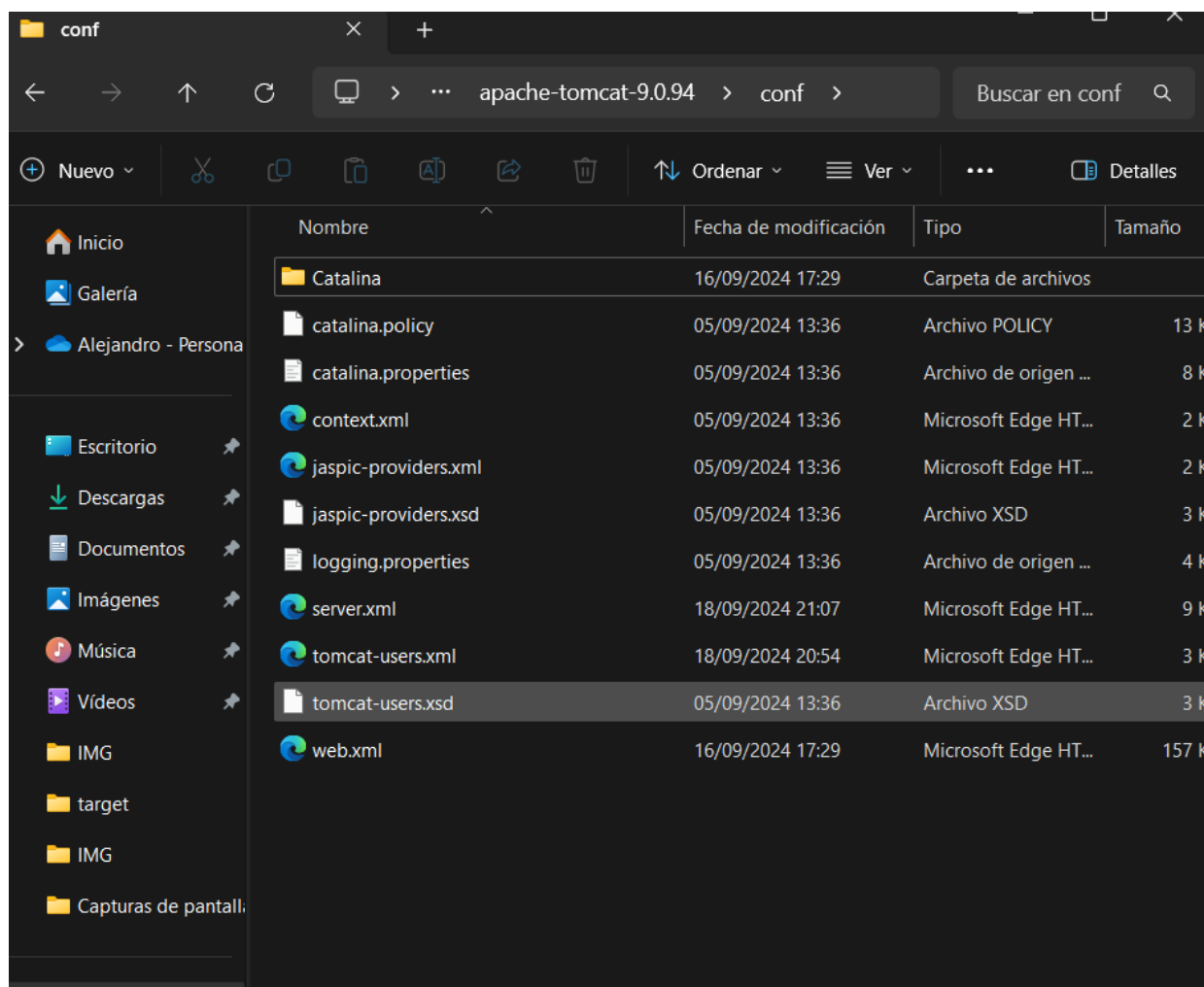


JUNTA DE EXTREMADURA

Consejería de Educación y Empleo

```
1 CREATE DATABASE IF NOT EXISTS 'abatech' /*!40100 DEFAULT CHARACTER SET utf8mb3 COLLATE utf8mb3_spanish_ci */ /*!80016 DEFAULT ENCRYPTION='N' */;  
2 USE 'abatech';  
3 -- MySQL dump 10.13 Distrib 8.0.34, for Win64 (x86_64)  
4 --  
5 -- Host: 127.0.0.1 Database: abatech  
6 -----  
7 -- Server version      8.0.35  
8  
9 CREATE USER IF NOT EXISTS 'abatech'@'%' IDENTIFIED BY 'abatech';  
10 GRANT ALL PRIVILEGES ON abatech.* TO 'abatech'@'%';  
11  
12 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@CHARACTER_SET_CLIENT */;  
13 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@CHARACTER_SET_RESULTS */;  
14 /*!40101 SET @OLD_COLLATION_CONNECTION=@COLLATION_CONNECTION */;  
15 /*!50503 SET NAMES utf8 */;  
16 /*!40103 SET @OLD_TIME_ZONE=@TIME_ZONE */;  
17 /*!40103 SET TIME_ZONE='+00:00' */;  
18 /*!40014 SET @OLD_UNIQUE_CHECKS=@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;  
19 /*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;  
20 /*!40101 SET @OLD_SQL_MODE=@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;  
21 /*!40111 SET @OLD_SQL_NOTES=@SQL_NOTES, SQL_NOTES=0 */;  
22
```

Una vez creada la BBDD, nos meteremos en los archivos de tomcat, en la ruta /conf, para modificar el archivo de tomcat-users.xml.



Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo

Entraremos en ese archivo con un editor de texto, y añadiremos esta línea `<user username="tomcat" password="s3cret" roles="manager-gui"/>`.

```
- manager-status - allows access to the status pages only

The users below are wrapped in a comment and are therefore ignored. If you
wish to configure one or more of these users for use with the manager web
application, do not forget to remove the <!-- ... --> that surrounds them. You
will also need to set the passwords to something appropriate.
<!--
<user username="admin" password="<must-be-changed>" roles="manager-gui"/>
<user username="robot" password="<must-be-changed>" roles="manager-script"/>
-->
<!--
The sample user and role entries below are intended for use with the
examples web application. They are wrapped in a comment and thus are ignored
when reading this file. If you wish to configure these users for use with the
examples web application, do not forget to remove the <!-- ... --> that surrounds
them. You will also need to set the passwords to something appropriate.
-->
<!--
<role rolename="tomcat"/>
<role rolename="role1"/>
<user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
<user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
<user username="role1" password="<must-be-changed>" roles="role1"/>
-->
<role rolename="manager-gui"/>

<user password="admin" roles="manager-script,admin" username="admin"/>
<user username="tomcat" password="s3cret" roles="manager-gui"/>
</tomcat-users>
```

Esto añade un usuario con el que podremos acceder a la zona de despliegue de tomcat, ahora pararemos el tomcat y lo volveremos a encender. Para pararlo, haremos lo mismo que para encenderlo, pero con stop en vez de con start.

```
C:\tomcat\apache-tomcat-9.0.94\bin>catalina.bat stop
Using CATALINA_BASE:   "C:\tomcat\apache-tomcat-9.0.94"
Using CATALINA_HOME:   "C:\tomcat\apache-tomcat-9.0.94"
Using CATALINA_TMPDIR: "C:\tomcat\apache-tomcat-9.0.94\temp"
Using JRE_HOME:        "C:\jdk\jdk-11.0.2"
Using CLASSPATH:       "C:\tomcat\apache-tomcat-9.0.94\bin\bootstrap.jar;C:\tomcat\apache-tomcat-9.0.94\bin\tomcat-juli.jar"
Using CATALINA_OPTS:   ""
NOTE: Picked up JDK_JAVA_OPTIONS:  --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED
--add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.util.concurrent=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED

C:\tomcat\apache-tomcat-9.0.94\bin>catalina.bat start
Using CATALINA_BASE:   "C:\tomcat\apache-tomcat-9.0.94"
Using CATALINA_HOME:   "C:\tomcat\apache-tomcat-9.0.94"
Using CATALINA_TMPDIR: "C:\tomcat\apache-tomcat-9.0.94\temp"
Using JRE_HOME:        "C:\jdk\jdk-11.0.2"
Using CLASSPATH:       "C:\tomcat\apache-tomcat-9.0.94\bin\bootstrap.jar;C:\tomcat\apache-tomcat-9.0.94\bin\tomcat-juli.jar"
Using CATALINA_OPTS:   ""

C:\tomcat\apache-tomcat-9.0.94\bin>
```

Una vez reiniciado, volveremos a la página principal del tomcat desplegado y haremos click en Manager App, una vez se nos pedirá un usuario y contraseña, estos datos son los del usuario que hemos creado anteriormente en los archivos de tomcat.

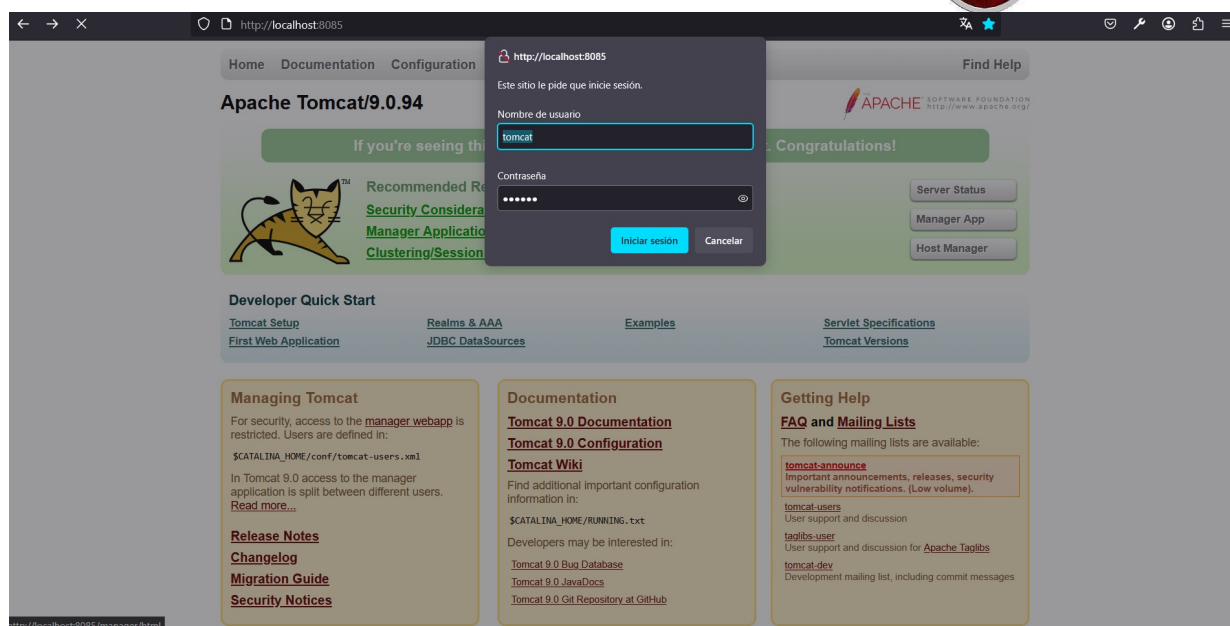
Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:





JUNTA DE EXTREMADURA

Consejería de Educación y Empleo



Una vez entremos en esa zona, estaremos en la zona de despliegue de tomcat.



Gestor de Aplicaciones Web de Tomcat

Mensaje: OK - Replegada aplicación en trayectoria de contexto [/RodriguezAAA1A1_war]

Gestor

[Listar Aplicaciones](#)[Ayuda HTML de Gestor](#)[Ayuda de Gestor](#)[Estado de Servidor](#)

Aplicaciones					
Ruta	Versión	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
/	Ninguno especificado	Welcome to Tomcat	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/examples	Ninguno especificado	Servlet and JSP Examples	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/host-manager	Ninguno especificado	Tomcat Host Manager Application	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/manager	Ninguno especificado	Tomcat Manager Application	true	1	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos

Desplegar
Desplegar directorio o archivo WAR localizado en servidor

Trayectoria de Contexto (opcional):

Version (for parallel deployment):

Ahora bajaremos en la página hasta la zona de archivo war a desplegar.

Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo

← → ↻ http://localhost:8085/manager/html/undeploy?path=/RodriguezAAA1A1_war&org.apache.catalina.filters.CSRF_NONCE=2EE2EB777B822039FCE359D

/manager	Ninguno especificado	Tomcat Manager Application	true	1	Expirar sesiones: sin trabajar 2:30 minutos
Arrancar Parar Recargar Replegar					
Expirar sesiones sin trabajar 2:30 minutos					

Desplegar

Desplegar directorio o archivo WAR localizado en servidor

Trayectoria de Contexto (opcional):

Version (for parallel deployment):

URL de archivo de Configuración XML:

URL de WAR o Directorio:

Desplegar

Archivo WAR a desplegar

Seleccione archivo WAR a cargar No se ha seleccionado ningún archivo.

Desplegar

Configuration

Re-read TLS configuration files

TLS host name (optional)

Re-read

Diagnósticos

Revisa a ver si una aplicación web ha causado fallos de memoria al parar, recargar o replegarse.

Este chequeo de diagnóstico disparará una colección completa de basura. Utilízalo con extremo cuidado en sistemas en producción.

TLS connector configuration diagnostics

<input type="button" value="Cifrados"/>	List the configured TLS virtual hosts and the ciphers for each.
<input type="button" value="Certificados"/>	Lista los virtual hosts configurados con TLS y la cadena de certificado para cada uno de ellos.
<input type="button" value="Trusted Certificates"/>	List the configured TLS virtual hosts and the trusted certificates for each.

Le haremos click en la zona de examinar para buscar en nuestro equipo el .war del proyecto que hemos realizado.

← → ↻ http://localhost:8085/manager/html/undeploy?path=/RodriguezAAA1A1_war&org.apache.catalina.filters.CSRF_NONCE=2EE2EB777B822039FCE359D

/manager	Ninguno especificado	Tomcat Manager Application	true	1	Expirar sesiones: sin trabajar 2:30 minutos
Arrancar Parar Recargar Replegar					
Expirar sesiones sin trabajar 2:30 minutos					

Desplegar

Desplegar directorio o archivo WAR localizado en servidor

Trayectoria de Contexto (opcional):

Version (for parallel deployment):

URL de archivo de Configuración XML:

URL de WAR o Directorio:

Desplegar

Archivo WAR a desplegar

Seleccione archivo WAR a cargar Abatech-1.0.war

Desplegar

Configuration

Re-read TLS configuration files

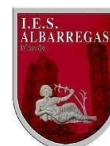
TLS host name (optional)

Re-read

Una vez elegido, le daremos a desplegar y si hemos hecho todo bien, nuestro proyecto ya estará desplegado.

Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo



Gestor de Aplicaciones Web de Tomcat

Mensaje:	OK				
Gestor					
Listar Aplicaciones		Ayuda HTML de Gestor	Ayuda de Gestor		Estado de Servidor
Aplicaciones					
Ruta	Versión	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
/	Ninguno especificado	Welcome to Tomcat	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/Abatech-1.0	Ninguno especificado		true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/examples	Ninguno especificado	Servlet and JSP Examples	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/host-manager	Ninguno especificado	Tomcat Host Manager Application	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/manager	Ninguno especificado	Tomcat Manager Application	true	1	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos

Para acceder a nuestro proyecto, podemos hacer click en el nombre de nuestro proyecto desde la zona de aplicaciones, o poner la IP del equipo donde tenemos desplegado el tomcat, junto al nombre del .war que hemos desplegado en este caso sería el nombre del .war es Abatech-1.0.

Filtros

Filtrar

Categorías

Marcas

Precio máximo

Asus A88X-PLUS

Marca: Asus

Motherboard A88X con FM2 ...

Precio: 82,00 €

Ver detalles

Creative Inspire T3150

Marca: Creative

Escucha música sin cables...

Precio: 52,00 €

Ver detalles

AMD A6-5400K 3.60Ghz

Marca: Amd

Gran procesador que deja ...

Precio: 45,00 €

Ver detalles

GARANTÍA

Garantía 6 meses

Marca: Nullware

NullWare te ofrece una ex...

Precio: 100,00 €

Ver detalles

5. Propuestas de mejoras.

1. Mejora en la Seguridad

- **Hashing y Salting de Contraseñas:** Actualmente se usa MD5 para almacenar contraseñas, lo que no es seguro. Se recomienda el uso de **BCrypt** o **Argon2** para un mayor nivel de seguridad.
- **Protección contra SQL Injection:** Validar y parametrizar todas las consultas SQL para evitar ataques de inyección.
- **Protección contra XSS y CSRF:** Implementar cabeceras de seguridad y utilizar tokens CSRF en formularios sensibles.
- **Control de Sesiones:** Implementar expiración de sesiones y regeneración de identificadores de sesión tras el login.
- **Autenticación en Dos Factores (2FA):** Agregar la posibilidad de autenticación en dos pasos para usuarios registrados.

2. Optimización de Base de Datos

- **Índices en Tablas Críticas:** Agregar índices en campos de búsqueda frecuente como Email en usuarios y Nombre en productos para mejorar el rendimiento.
- **Normalización de Datos:** Evaluar si la estructura de la base de datos requiere normalización adicional para evitar redundancias.
- **Optimización de Consultas:** Revisar y optimizar consultas SQL para mejorar tiempos de respuesta.
- **Backups Automatizados:** Implementar un sistema de copias de seguridad automáticas de la base de datos.

3. Mejoras en la Experiencia de Usuario (UX/UI)

- **Diseño Responsivo:** Asegurar que la interfaz sea completamente adaptable a dispositivos móviles.
- **Autocompletado y Sugerencias de Búsqueda:** Implementar un sistema de sugerencias en la barra de búsqueda para mejorar la navegación.
- **Historial de Pedidos Mejorado:** Permitir a los usuarios registrados filtrar y ordenar sus pedidos finalizados por fecha, importe o estado.
- **Feedback en Tiempo Real:** Mostrar mensajes de error y validaciones de formularios de manera inmediata sin necesidad de recargar la página.

4. Optimización del Rendimiento

- **Uso de Caché:** Implementar caché en el lado del servidor y cliente para mejorar la velocidad de carga de productos y carritos.
- **Optimización de Imágenes:** Comprimir imágenes y utilizar formatos modernos como WebP para mejorar tiempos de carga.
- **Carga Diferida (Lazy Loading):** Aplicar carga diferida en imágenes y productos para mejorar la velocidad de navegación.
- **Minificación de Recursos:** Reducir y combinar archivos CSS y JavaScript para mejorar el tiempo de carga.

5. Nuevas Funcionalidades

- **Wishlist o Lista de Deseos:** Permitir a los usuarios guardar productos para futuras compras.
- **Valoraciones y Reseñas de Productos:** Agregar un sistema de comentarios y puntuaciones para productos.
- **Notificaciones por Email:** Enviar notificaciones automáticas sobre cambios en el estado del pedido.
- **Modo Invitado para el Carrito:** Permitir que usuarios anónimos guarden su carrito sin necesidad de registro.
- **Implementación de Métodos de Pago Adicionales:** Integración con pasarelas de pago como PayPal, Stripe o Bizum.

6. Mejoras en la Gestión Administrativa

- **Panel de Administración Mejorado:** Crear un dashboard con métricas de ventas, pedidos y usuarios.
- **Gestión de Stock Automatizada:** Notificar al administrador cuando un producto esté por agotarse.
- **Historial de Cambios en Productos:** Guardar un registro de modificaciones en productos para auditoría.

7. Internacionalización y Accesibilidad

- **Multi-idioma:** Permitir la selección de idiomas para una mayor cobertura global.
- **Compatibilidad con Lector de Pantalla:** Mejorar la accesibilidad siguiendo estándares WCAG.

Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:



Consejería de Educación y Empleo

- **Modo Oscuro:** Implementar un modo oscuro para mejorar la experiencia del usuario.

Estas mejoras fortalecerán la aplicación, aumentando su seguridad, eficiencia y facilidad de uso, garantizando una mejor experiencia tanto para usuarios como administradores.

6. Bibliografía

Ambler, S. W. (2004). *The Object Primer: Agile Model-Driven Development with UML 2.0*. Cambridge University Press.

Bauer, C., & King, G. (2015). *Java Persistence with Hibernate*. Manning Publications.

Bloch, J. (2018). *Effective Java* (3rd ed.). Addison-Wesley Professional.

Fowler, M. (2002). *Patterns of Enterprise Application Architecture*. Addison-Wesley Professional.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.

Hall, M. (2006). *Core Servlets and JavaServer Pages (JSP), Volume 1: Core Technologies* (2nd ed.). Prentice Hall.

Holzner, S. (2003). *Inside XML*. New Riders.

Liang, Y. D. (2018). *Introduction to Java Programming and Data Structures, Comprehensive Version* (11th ed.). Pearson.

Sierra, K., & Bates, B. (2008). *Head First Java* (2nd ed.). O'Reilly Media.

Sun Microsystems. (2009). *JavaServer Pages (JSP) Specification*. Oracle Corporation.

Torres, R. (2017). *Desarrollo de Aplicaciones Web con Java EE y PrimeFaces*. Alfaomega Grupo Editor.

Ullman, L. (2019). *PHP and MySQL for Dynamic Web Sites* (5th ed.). Peachpit Press.

Oracle. (2023). *Java Platform, Enterprise Edition (Java EE) Documentation*. Oracle. Recuperado el 7 de febrero de 2025, de <https://www.oracle.com/java/technologies/java-ee-glance.html>

Apache Software Foundation. (2023). *Apache Tomcat 9.0 Documentation*. Recuperado el 7 de febrero de 2025, de <https://tomcat.apache.org/tomcat-9.0-doc/index.html>

MySQL. (2023). *MySQL 8.0 Reference Manual*. Oracle. Recuperado el 7 de febrero de

Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto:



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo

2025, de <https://dev.mysql.com/doc/refman/8.0/en/>

MDN Web Docs. (2023). *JavaScript and AJAX Overview*. Mozilla Foundation. Recuperado el 7 de febrero de 2025, de <https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX>