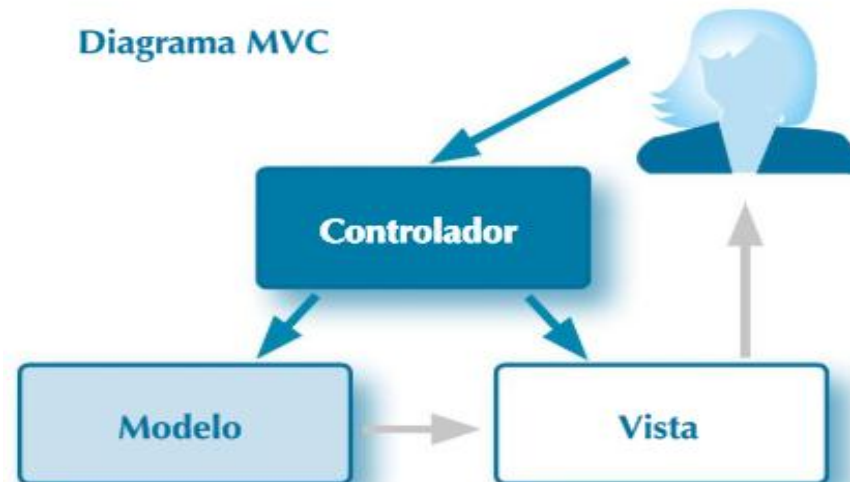


Arquitectura y configuración básica del servidor de aplicaciones



La arquitectura que se emplea en este tipo de aplicaciones es un patrón software que separa la lógica de negocio y los datos de la parte representativa que observa el usuario, los eventos y las comunicaciones entre los distintos componentes. A este modelo se le denomina *Modelo-Vista-Controlador (MVC)*, y se basa en la separación de módulos para su posterior mantenimiento y reutilización de código, además de la facilidad de detectar errores en caso de fallos. Es importante separar la representación de los datos (Vista) de la parte interna o modelo de datos, como, por ejemplo, la base de datos. Como se puede observar en la siguiente figura:



Los tres grandes componentes en los que se divide el modelo son los siguientes:

- a) *Modelo*: es el componente que se encarga de representar la información con la que la aplicación trabaja. Por lo que su función es la de realizar las consultas y modificaciones con base en los privilegios definidos previamente en el análisis de requisitos de la aplicación. La petición llega por parte del controlador y este componente ejecuta la acción y la presenta a la *Vista*.
- b) *Vista*: visualiza el modelo con un tipo de representación para interactuar con el usuario. Normalmente es la interfaz de la aplicación, ya que puede ser una aplicación web, aplicación de escritorio, o cualquier aplicación en cualquier entorno o sistema operativo.
- c) *Controlador*: es el módulo más importante, como su propio nombre indica, que controla a los otros dos componentes. Responde a peticiones realizadas por el usuario (pulsar un botón de la interfaz), y partir de ahí se comunica con el modelo para manipularlo haciendo cambios en la vista. Se puede concluir que es el mediador entre el *Modelo* y la *Vista*.

Un servidor de aplicaciones es un software global que permite dar servicio a las aplicaciones que se publican en el mismo, dando soporte para la seguridad, para las transacciones, administración de sesiones, registro de logs, etc.

Existe un abanico de servidores de aplicaciones que son usados en el tejido productivo empresarial. Se van a comentar los más usados:

- ✓ *WildFly (JBoss Application Server)*: es un software implementado por JBoss y está desarrollado en Java. Está disponible para todos los sistemas operativos del mercado y está basado en los proyectos punteros de software open source. Si se analiza la estructura interna, está basado en dos núcleos principales:
 - *JBoss Module*: controla la carga de los recursos de la clase contenedor.
 - *Modular Service Container*: administra los servicios usados por el contenedor, por ejemplo, instala y desinstala los diferentes servicios.
- ✓ *GlassFish*: es un software de código abierto desarrollado por Sun Microsystems para la plataforma Java EE. Se encuentra disponible la versión GlassFish 5.0.1 con Java EE 8. Este proyecto open source proporciona un proyecto estructurado de desarrollo que permite tener disponibles las nuevas funcionalidades de Java.
- ✓ *JOnAs*: es un software libre indicado para servidores de aplicaciones desarrollado por el consorcio ObjectWeb.
- ✓ *Apache-Tomcat*: es un servidor de aplicaciones open source implementando Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket technologies. Estos módulos son desarrollados por Java Community Process. Oracle certifica con la versión de Java correspondiente. La versión estable de Apache-Tomcat es la 9.0.37 con JDK 8.

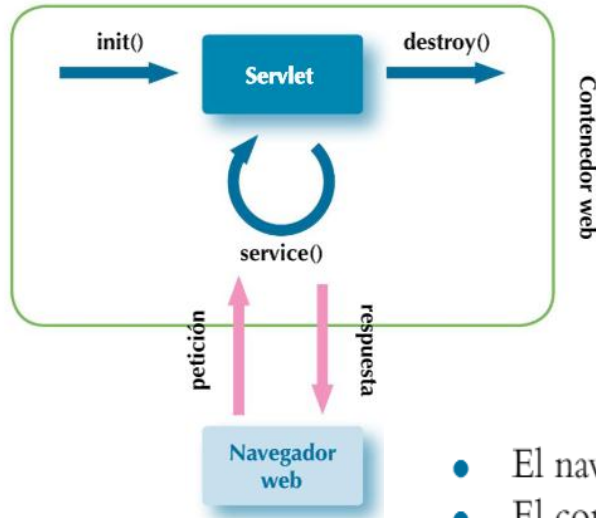
Servidor de aplicaciones Apache-Tomcat



El servidor de aplicaciones que se va a instalar y configurar es Apache-Tomcat, por ser uno de los más usados en el mundo empresarial, también por su facilidad de uso y por su aceptación en la comunidad de desarrolladores de software.

Apache-Tomcat es un contenedor de Servlets que se puede usar para compilar y ejecutar aplicaciones realizadas en Java. Tomcat puede funcionar de manera autónoma pero usualmente se puede utilizar con otros productos para permitir una mayor compatibilidad con las distintas tecnologías del mercado, y de esta forma incrementar su funcionalidad.

Funcionamiento Apache-Tomcat



- El navegador web del cliente solicita una página al servidor HTTP.
- El contenedor de servlets procesa la petición y la asigna al servlet apropiado.
- El servlet elegido es el encargado de generar el texto de la página web y entregarla al contenedor de servlets.
- El contenedor devuelve la página web al navegador del cliente.

Por otro lado, además de servlets, se pueden ejecutar en la capa web más componentes, como, por ejemplo, Java Server Pages (JSP). Es un software que permite generar páginas web dinámicas principalmente (también genera otro tipo de documentos, como xml, etc.). Estos archivos .jsp se compilan y se transforman en un servlet.