Hello, Ubaldo Acosta greets you. Welcome back to this JavaServer Faces Course.

In this first lesson we will review an introduction to JavaServer Faces (JSF), which is one of the most used technologies for creating Web applications with Java. We will also study the basic elements that will allow us to understand the JSF role today.

So if you're ready, we will too. Let's start immediately.

© All rights reserved Global Mentoring

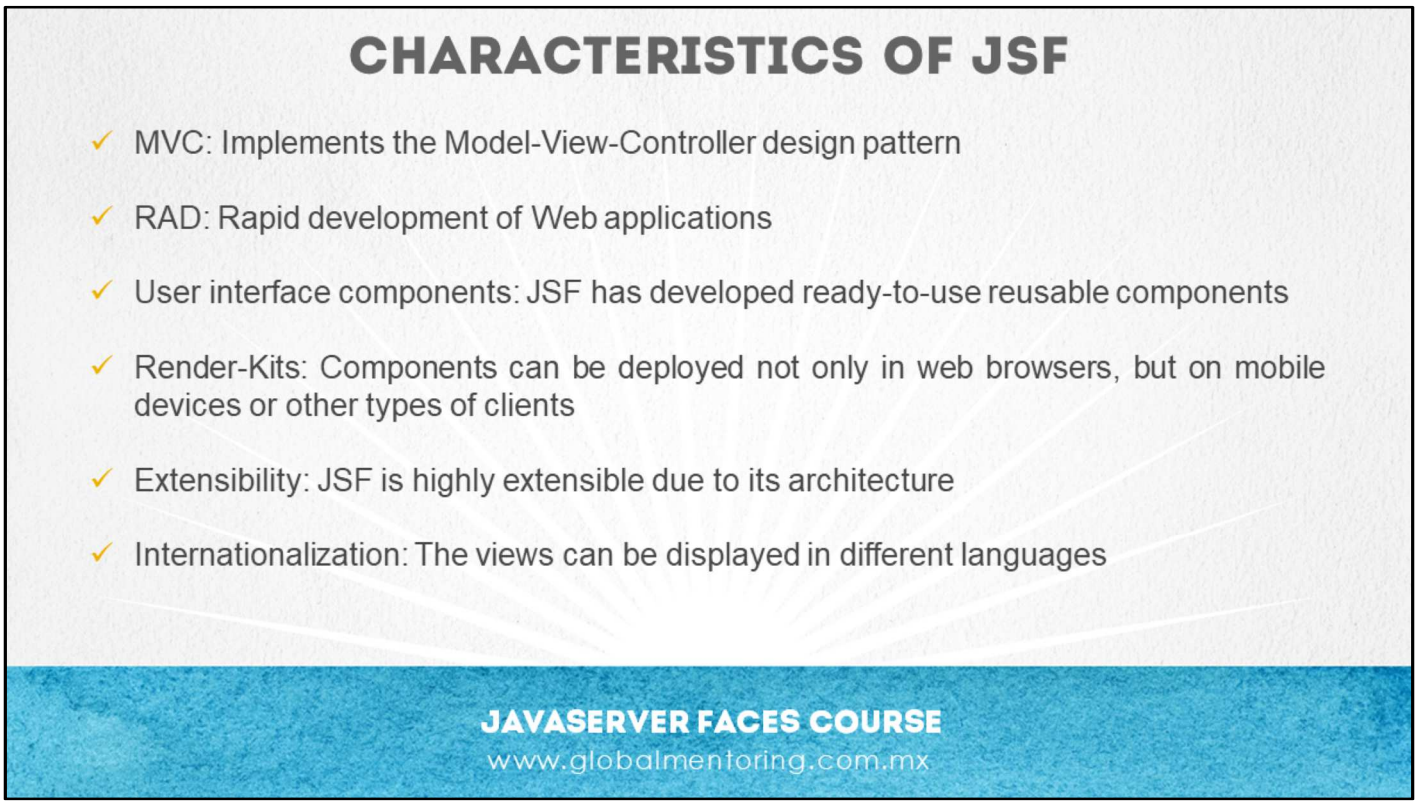
JSF was designed to simplify the construction of the user interface for Web applications. One of the key points was the support of the "tools" created for this type of Java applications.

JSF has the support of Java IDEs such as:

- Eclipse
- MyEclipse
- NetBeans
- IntelliJ IDEA
- BEA Workshop
- Oracle Jdeveloper

It also has a strong support of Java Application Servers with the aim of supporting the applications created in JSF.

© All rights reserved Global Mentoring



Some of the most important characteristics of JSF are:

MVC: The JSF framework implements the Model-View-Controller design pattern, providing an event-oriented approach.

RAD: Due to the number of ready-to-use components, JSF streamlines the development of Web applications for Java.

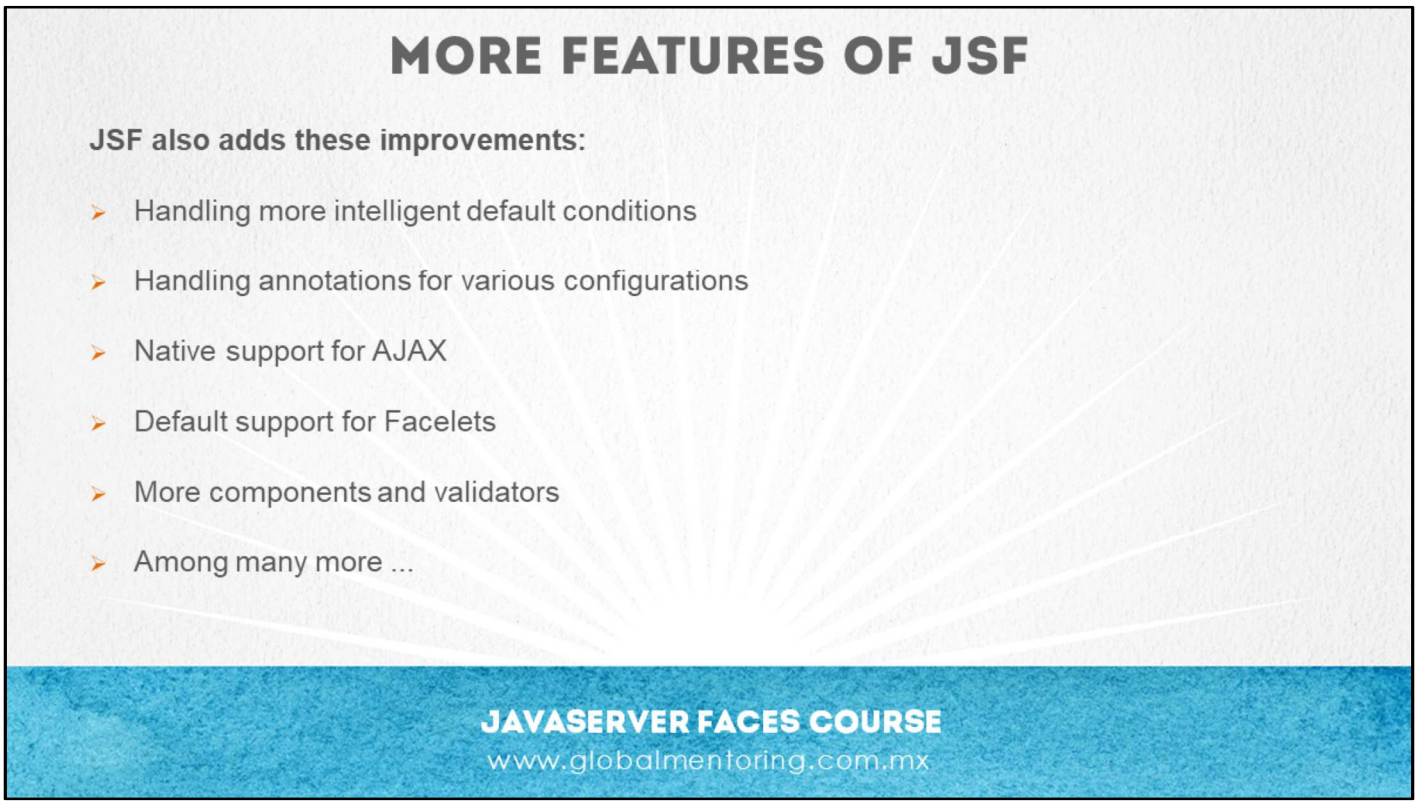
User interface components (GUI): JSF provides an API to create complex HTML components, including JavaScript and integrated CSS, creating reusable components.

Render-Kits: Components can be deployed not only in Web browsers, but in mobile devices or other types of clients.

Extensibility: JSF allows to create new components more easily, so there are several frameworks that extend the power of JSF and Ajax, such as richFaces, iceFaces, among others.

Internationalization: The views can be displayed in different languages.

© All rights reserved Global Mentoring



JSF adds more improvements to make this technology easier to use today.

Some of these improvements are:

- Handling more intelligent default conditions: This applies to simple navigation cases, which are no longer required to be added to the configuration file faces-config.xml

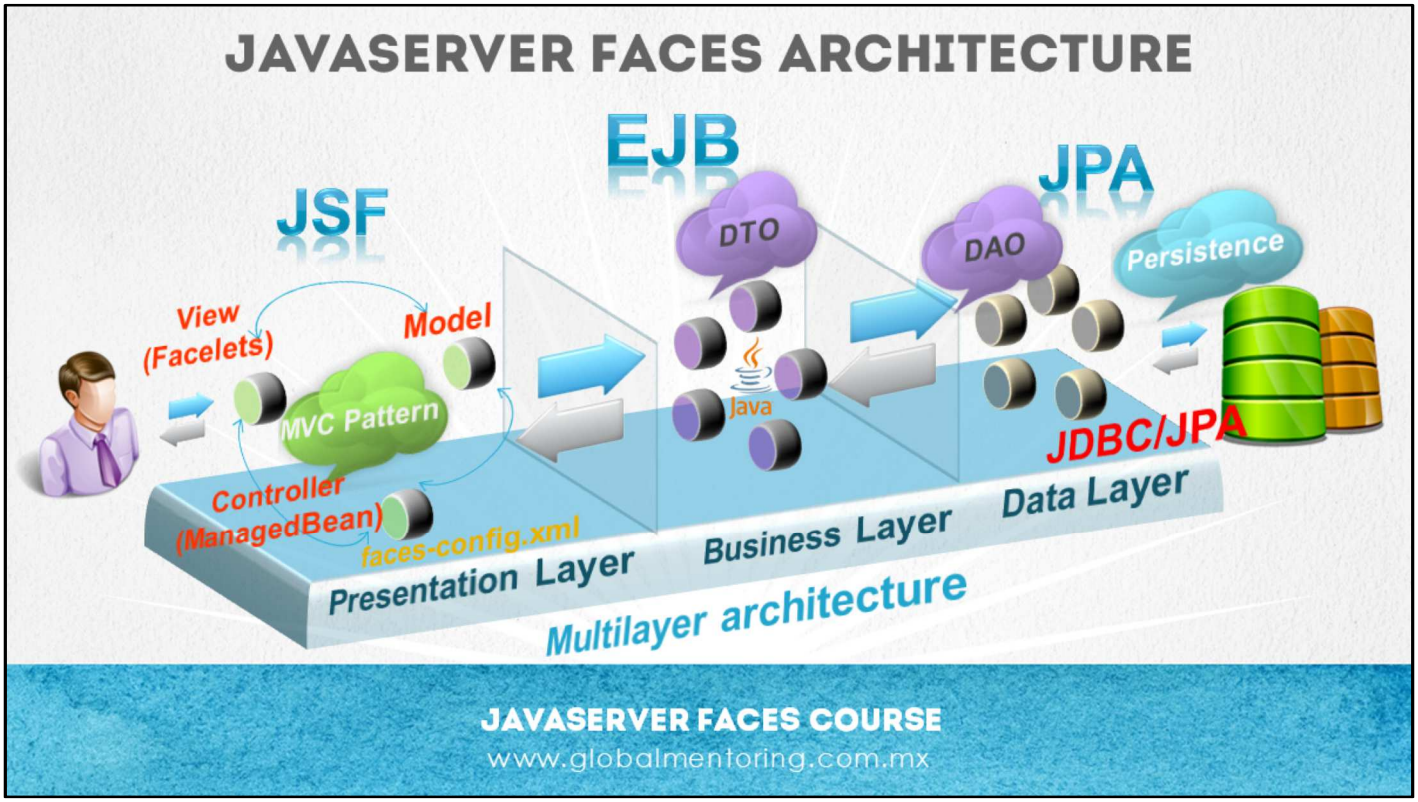
- Handling annotations for various configurations: This greatly simplifies adding a Managed Bean to our application, avoiding its declaration in the faces-config.xml file. Later we will see the concept of Managed Beans.

- Native support for AJAX: AJAX technology is already part of the life cycle of JSF.

- Default support for Facelets: Facelets technology takes into account the life cycle of JSF, unlike JSP's.

- More components and validators: New components have been created and added to the standard JSF library. Among many more features.

© All rights reserved Global Mentoring

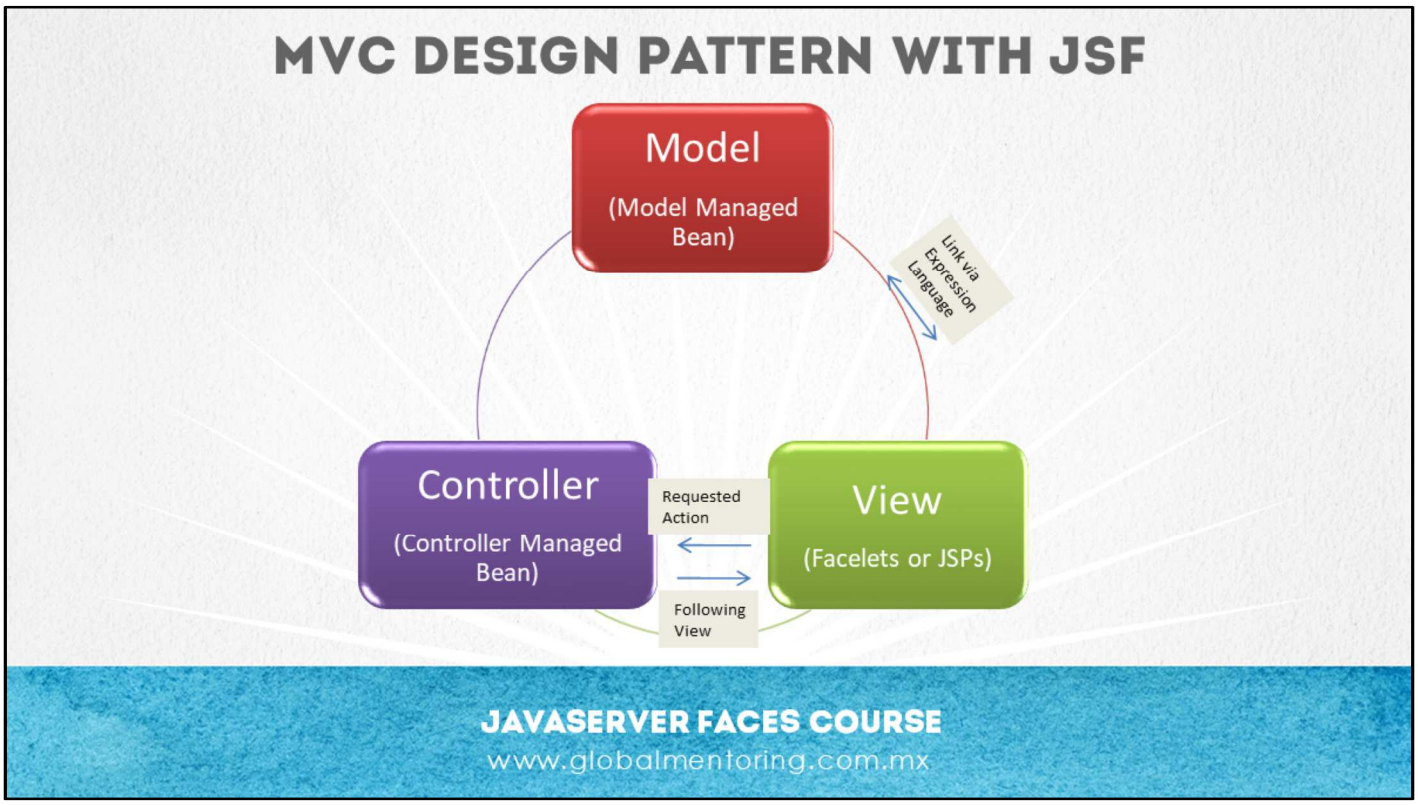


In the figure we can see the multi-layer architecture that we will be using throughout the course. We will study the bases to create this type of Java Web Business applications, applying design patterns such as MVC, DAO, DTO, among others, and above all with the best practices of Java Web systems development using JavaServer Faces technology.

The final Java Web project represents the Web layer of a Java business application, in which you will find the best practices so that you are ready when you get to your real-world projects, and in this way you will find many similarities that will make your work and the development of your projects much simpler.

The final project will guide you to create the Web layer for your own Java Web projects using JavaServer Faces. This layer can be integrated with different technologies, such as EJB and JPA, or Spring and Hibernate or JPA, depending on the selected architecture and technologies.

© All rights reserved Global Mentoring



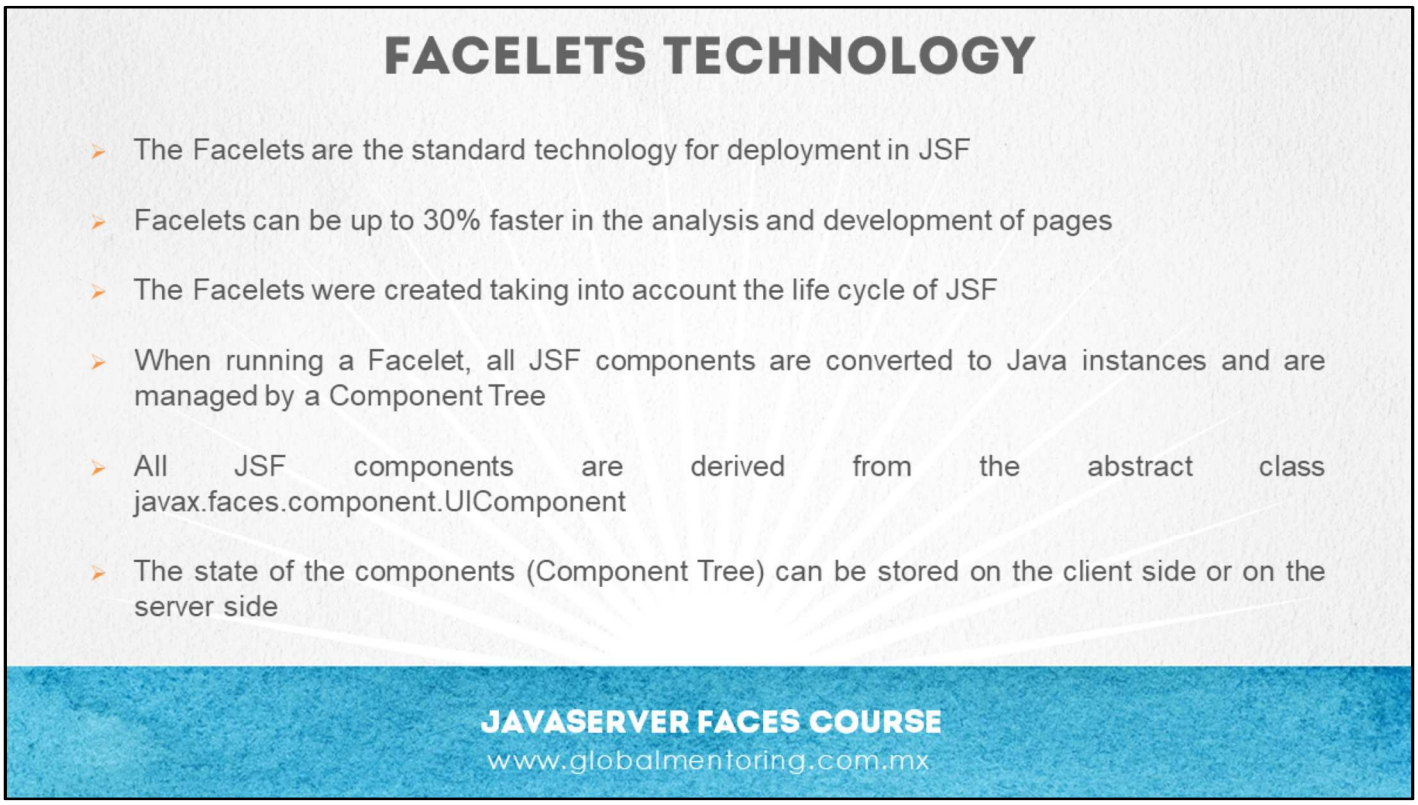
JSF is a web application framework that implements the MVC design pattern, with a clear separation of responsibilities:

- The Model: Contains data from the user interface and is responsible for storing the data of the Web application. It can be implemented with pure Java classes (POJO: Plain Old Java Object) or with Managed Bean of Model (they do not contain logic of the application nor manage the flow of it).

- The View: Defines the user interface with a hierarchy of components, using the standard library of JSF, Expression Language (EL), JSTL, among other technologies to facilitate the deployment of the Model information. The technology used by default in JSF are the Facelets. Using the JSF EL language, it is possible to link and use the Model objects and integrate it with the view for the display of information.

- The Controller: Manages the user's interactions and the navigation or flow of the application. It is implemented with Managed Beans. We will study this concept in detail later.

© All rights reserved Global Mentoring

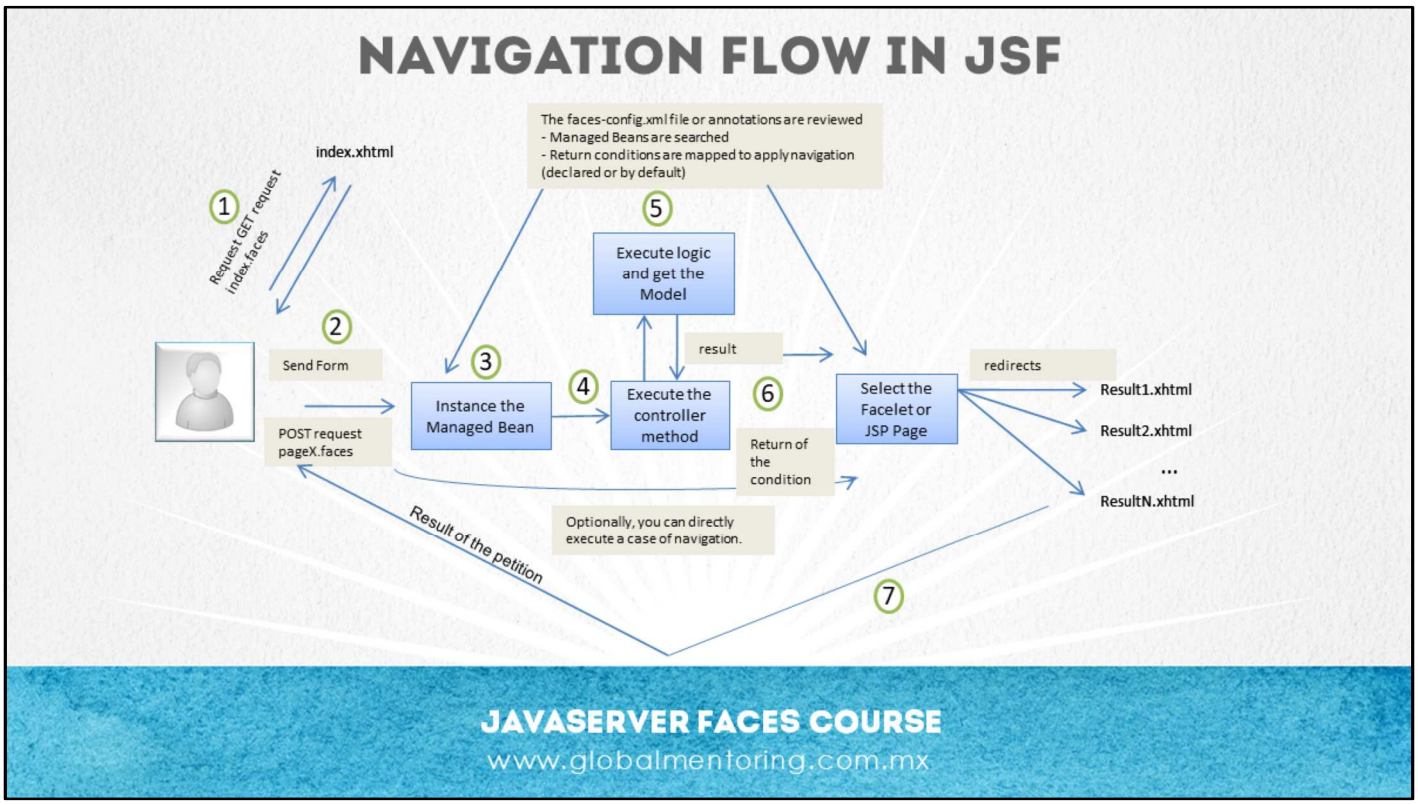


The deployment technology (Page Description Language - PDL) by default is known as Facelets, and is used in place of the JPS's (JavaServer Pages).

The Facelets have the advantage that they were created taking into account the life cycle of JSF, including the following characteristics:

- Facelets are XHTML documents, so the XML parser makes debugging easier than in JSPs.
- When running a Facelet, all JSF components are converted to Java instances and are managed by a Component Tree.
- When using Facelets it is easier to encapsulate code and thus create reusable components.
- Facelets allow you to create templates and thus more easily define new views from the defined templates.

© All rights reserved Global Mentoring



We will explain the General Navigation flow when using the JSF:

1. The framework starts with the GET request to a page, for example, index.xhtml.
2. Once we are in the JSF context, the user receives the response content and sends a POST request back to the Web server.
3. The Web server receives the request and reviews the Managed Beans involved in the request, if necessary creates an instance of them, depending on its scope and calls the setter methods of the properties of the bean to be filled.
4. The method that processes the request is executed, for this moment the other Managed Beans involved are instantiated, if any.
5. The business logic is executed, with the objective of obtaining the Model.
6. The navigation case is selected and redirected to the corresponding view.
7. The view uses the Model information to finally return the requested response to the client.

© All rights reserved Global Mentoring

© All rights reserved Global Mentoring