Seminario 2: Lenguajes y entornos de desarrollo

Nombre del grupo: “Por Asignar”

# Estudio de las guías docentes para el curso actual

A continuación se muestran en formato tabla el estudio de las guías docentes para el curso actual de las asignaturas del Grado en Ingeniería Informática, detallando los nombres de tecnologías, lenguajes y conceptos clave que hemos visto en este Seminario que aparezcan en el temario de dichas asignaturas (obviando términos generales como Web, Software, Aplicación, etc.).

| **Asignatura** | **Curso** | **Semestre** | **Tipo** | **Nombres de tecnologías, lenguajes y**  **conceptos clave** |
| --- | --- | --- | --- | --- |
| Fundamentos del Software | 1 | 1 | Básica | API |
| Estructura de Datos | 2 | 1 | Obligatoria | Java, |
| Programación y Diseño Orientado a Objetos | 2 | 1 | Obligatoria | Java, Ruby, MVC |
| Sistemas Operativos | 2 | 1 | Obligatoria | API |
| Fundamentos de Ingeniería de Software | 2 | 2 | Obligatoria | Java |
| Fundamentos de Redes | 3 | 1 | Obligatoria | TCP/IP, HTTP |
| Modelos de Computación | 3 | 1 | Obligatoria | Java |
| Aprendizaje Automático | 3 | 2 | Optativa | Python |
| Desarrollo de Software | 3 | 2 | Optativa | Ruby, Python |
| Ingeniería de Sistemas de Información | 3 | 2 | Optativa | Sockets |
| Ingeniería del Conocimiento | 3 | 2 | Optativa | Python |
| Modelos Avanzados de Computación | 3 | 2 | Optativa | Python |
| Programación Web | 3 | 2 | Optativa | Java, HTTP, PHP, HTML, CSS, JavaScript, XML, DDM, AJAX |
| Sistemas Gráficos | 3 | 2 | Optativa | JavaScript |
| Sistemas Multimedia | 3 | 2 | Optativa | Java |
| Sistemas de Información Basados en Web | 3 | 2 | Optativa | Java, API |
| Sistemas de Información para Empresas | 3 | 2 | Optativa | Java |
| Tecnologías Web | 3 | 2 | Optativa | MVC, HP, HTML, CSS, JavaScript, XML |
| Desarrollo de Aplicaciones para Internet | 4 | 1 | Optativa | HP, HTML, CSS, JavaScript, XML, Python |
| Infraestructura Virtual | 4 | 1 | Obligatoria | Python, HTTP |
| Inteligencia de Negocio | 4 | 1 | Obligatoria | Java, Python |
| Procesadores de Lenguajes | 4 | 1 | Obligatoria | back-end |
| Programación Técnica y Científica | 4 | 1 | Optativa | Python |
| Recuperación de Información | 4 | 1 | Optativa | XML |
| Teoría de la Información y la Codificación | 4 | 1 | Optativa | Python |
| Robótica Industrial | 4 | 2 | Optativa | Python |
| Nuevas Tecnologías de la Programación | 4 | 2 | Optativa | Java |
| Sistemas de Información Geográficos | 4 | 2 | Optativa | Python |

# Estudio de la herramienta Oracle ADF mediante JDeveloper

### Conceptos y tecnologías utilizados en la herramienta

* Se utiliza AJAX (Asynchronous JavaScript And XML) para desarrollar el Front-End. Es una técnica de programación que utiliza JavaScript y el objeto XMLHTTPRequest para intercambiar datos entre un navegador web y un servidor web. XMLHttpRequest es un objeto JavaScript que fue diseñado por Microsoft y adoptado por Mozilla, Apple y Google. Proporciona una forma fácil de obtener información de una URL sin tener que recargar la página completa. Una página web puede actualizar sólo una parte de la página sin interrumpir lo que el usuario está haciendo.
* Para desarrollar el Back-End se utiliza un WAF (Web Application Frameworks) que usa para la implementación de la aplicación Web el patrón de diseño MVC (Model-View-Controller). WAF protege de múltiples ataques al servidor de aplicaciones web en el backend, examina cada petición enviada al servidor, antes de que llegue a la aplicación, para asegurarse de que cumple con las reglas del firewall. El MVC se separa en tres partes: modelo (maneja datos y lógica de negocios), vista (se encarga del diseño y presentación) y controlador (enruta comandos a los modelos y vistas). Los componentes del MVC no se pueden combinar dentro de una misma clase.

### Instalación

Se ha utilizado la versión [Oracle JDeveloper 12c (12.2.1.3.0)](https://www.oracle.com/tools/downloads/jdev-v12213-downloads.html). Una vez se han descargado los archivos del enlace proporcionado anteriormente, se ejecutará el archivo jdev\_suite\_122130\_linux64.bin, que iniciará el proceso de instalación.

Una vez instalado, para añadir un Desktop Launcher en Linux, bastará con ejecutar /home/<userName>/Oracle/Middleware/Oracle\_Home/jdeveloper/jdev/bin/addjdevtodesktop en el terminal.

### Objetivo del tutorial

El objetivo del tutorial es construir una aplicación Web completa que interacciona con una base de datos. En el tutorial se trabajará con Oracle ADF Business Components, Oracle ADF Faces Rich Client Components y Oracle ADF Task Flows.

El tutorial nos proporciona los pasos para la creación de una aplicación web que proporciona una interfaz de usuario para acceder a las tablas de la base de datos. Utiliza Oracle JDeveloper 12c versión 12.2.1 y Oracle ADF para crear una aplicación completa que incluye servicios comerciales, interfaz de usuario y flujos de página. El objetivo del tutorial es ilustrar el enfoque visual y declarativo que proporciona Oracle ADF para el desarrollo de aplicaciones Java EE. Mientras trabaja en el tutorial, observe cómo ADF le evita escribir código de bajo nivel y acelera su proceso de desarrollo.

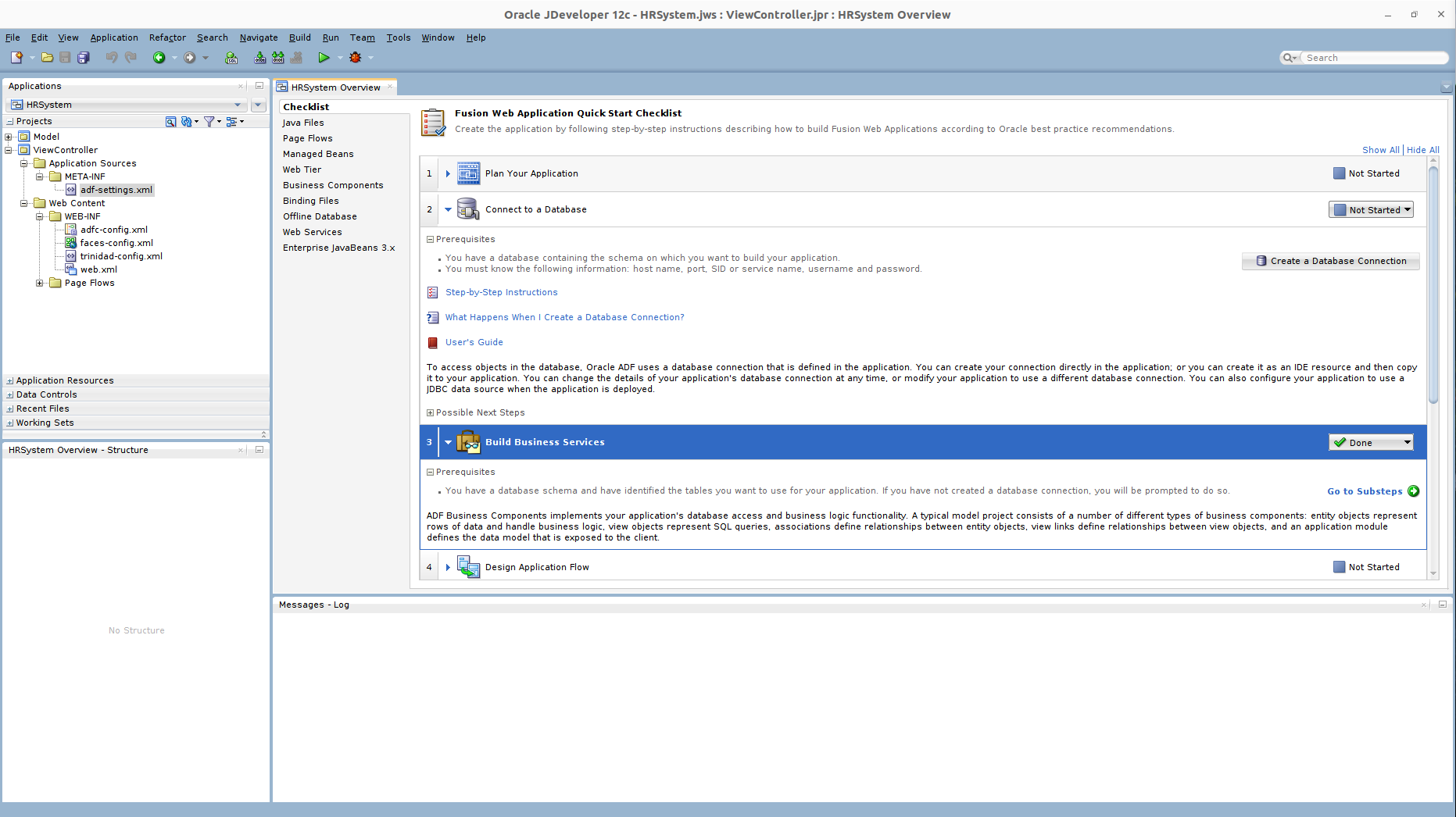
### Desarrollo del tutorial

#### Paso 1: Creación de una Fusion Web Application

JDeveloper proporciona diversas plantillas de aplicaciones que pueden usarse cuando se crean aplicaciones y proyectos. Las aplicaciones plantilla están pre-configuradas con un set básico de tecnologías requeridas para desarrollar los variados tipos de aplicaciones. El ambiente de desarrollo se creará seleccionando la plantilla que mejor se adapte a nuestras necesidades y, a posteriori, se configura para añadir cualquier tecnología adicional que se planee utilizar.

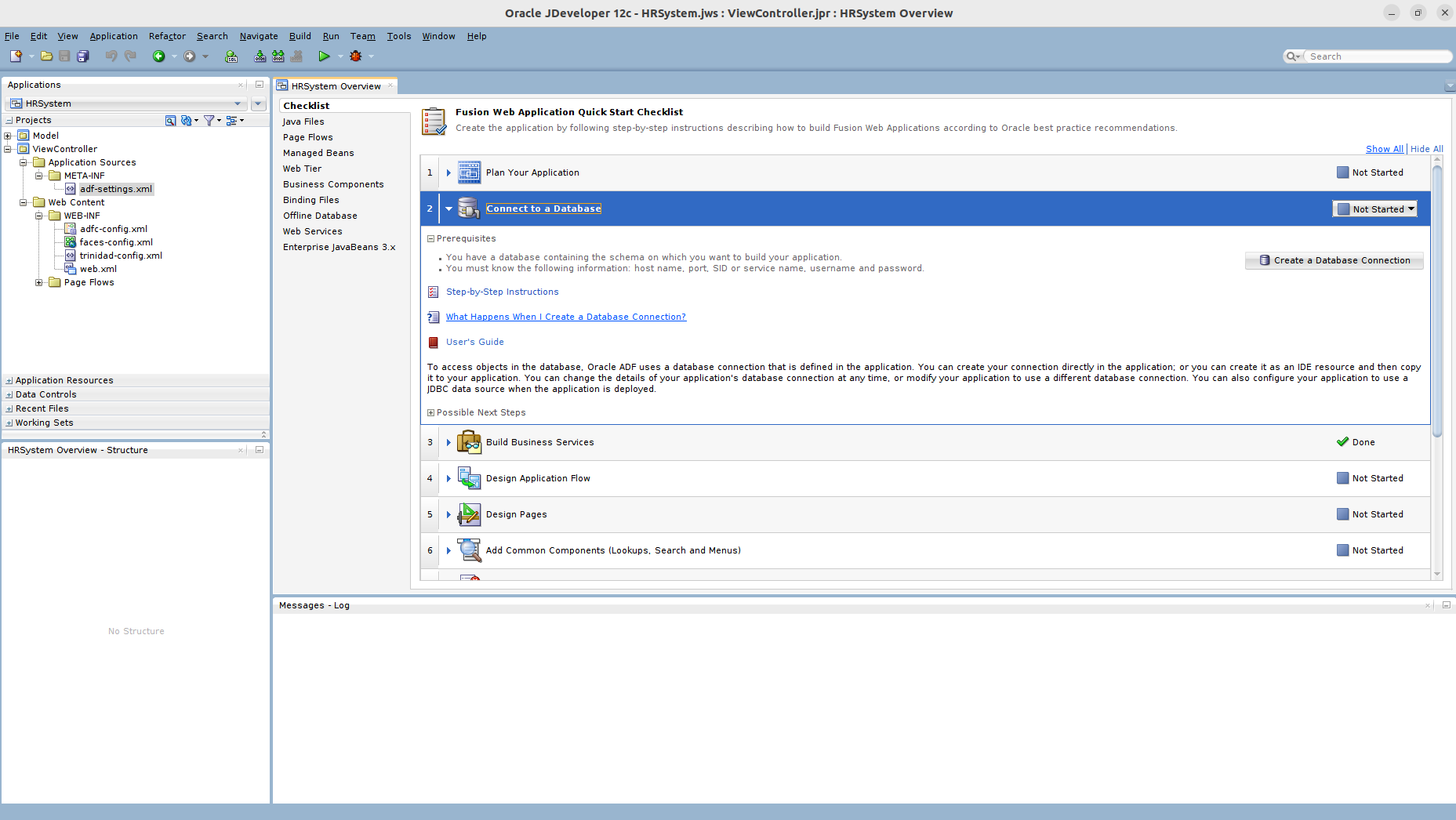
En esta primera parte del tutorial se creará una nueva *Fusion Web application* y se construirán reusable business components que accederán a la base de datos. Se utilizarán *Oracle ADF Business Components* para mapear objetos Java a tablas existentes en nuestra base de datos.

La aplicación HRSystem y los dos proyectos (Model y ViewController) se muestran en la ventana Applications situada en la parte izquierda de la pantalla. Nótese la checklist en el lado derecho de pantalla. Esta checklist se muestra por defecto cuando se crea una Fusion Web application. Está diseñada para guiarnos a través de las tareas involucradas en la construcción de este tipo de aplicación. Los siguientes pasos muestran cómo se debería usar la checklist en el desarrollo de una aplicación.

Figura 1: Aplicación HRSystem recién creada.

#### Paso 2: Build the Business Services

En esta sección, nos conectaremos a la base de datos y construiremos backend business components simples que accederán a HR schema tables.

Figura 2: Checklist de las tareas a realizar.

El paso 2 de la checklist se expande mostrando información útil como los prerrequisitos requeridos para llevar a cabo la tarea e instrucciones detalladas de cómo realizarla. Hacemos click en el botón *Create a Database Connection*. A continuación, especificamos las propiedades para conectarnos a nuestro esquema HR. Introducimos apropiadamente los *Oracle JDBC Settings* para señalar al Host correcto, JDBC Port y Service Name de nuestra base de datos. Finalmente, hacemos click en el botón *Test Connection* para verificar que la conexión se ha realizado con éxito.

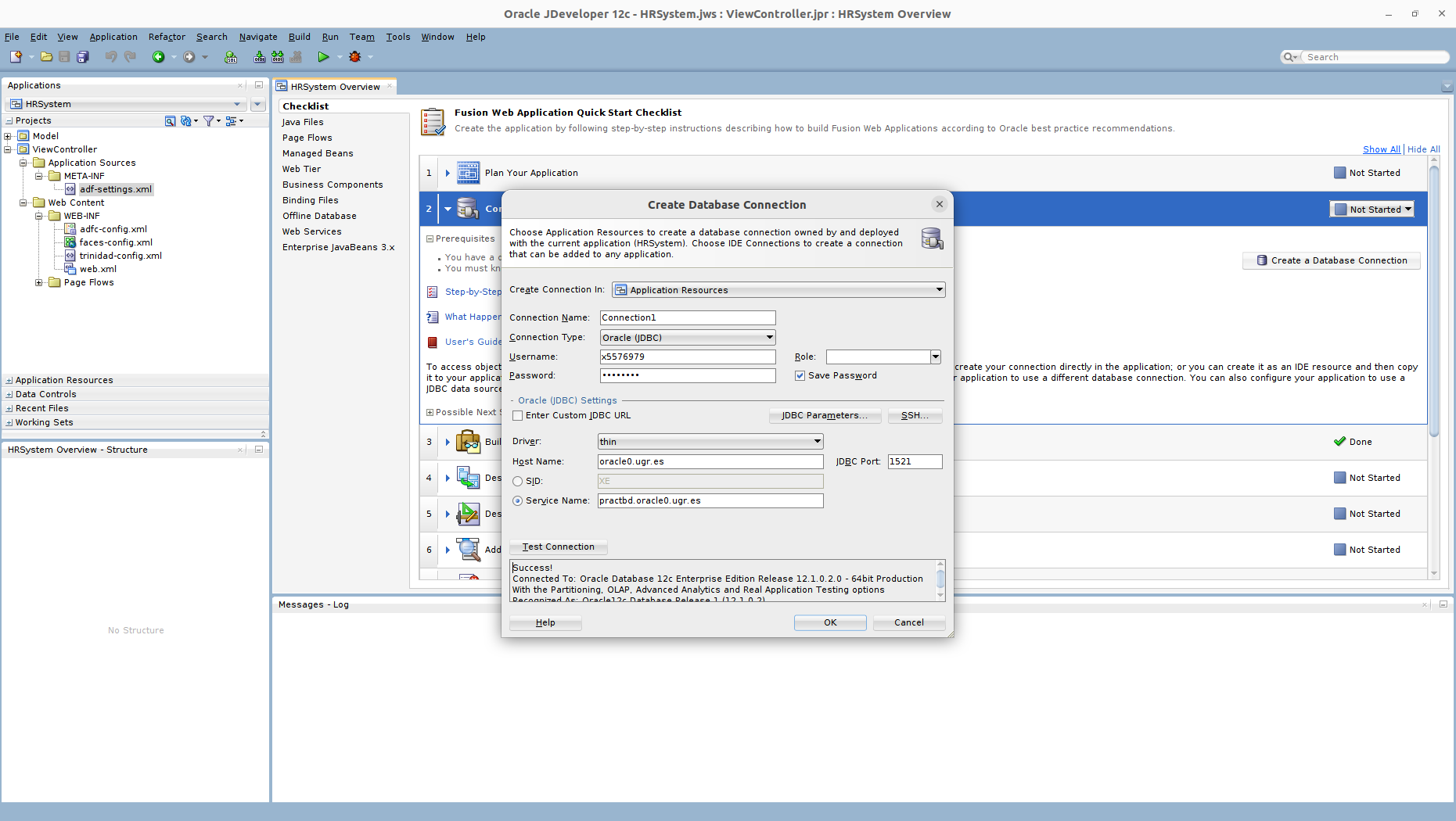


Figura 3: Creación de la conexión a la base de datos.

Ahora tendremos que ir al siguiente paso: Build Business Services. Así pues, ahora procederemos a crear Business Components en el proyecto Model.

Para comprobar si se han creado correctamente, en la ventana de Aplicaciones hacemos click derecho en *AppModule* y seleccionamos *Run* para invocar al application module tester. Esto iniciará una pequeña aplicación *Swing-based* que nos permitirá testear los *ADF Business Components* que acabamos de crear.

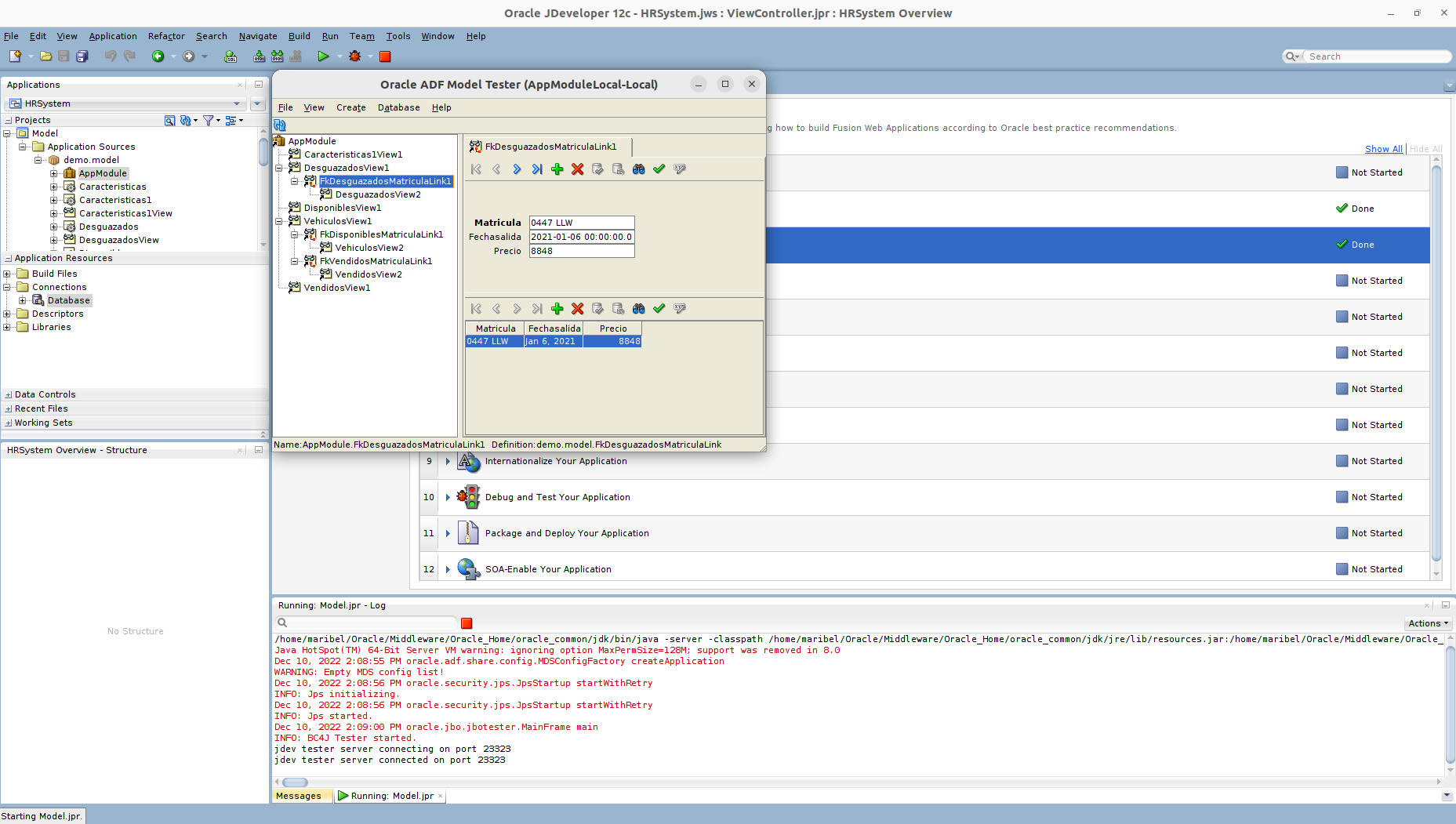


Figura 4: Oracle ADF Model Tester. Acceso a los datos del modelo.

#### Paso 3: Desarrollar una JavaServer Faces page

JavaServer Faces (JSF) es un estándar Java EE technology que simplifica el desarrollo Web. En esta parte del tutorial se crea una página JSF para acceder a los business components que hemos creado en la parte anterior.

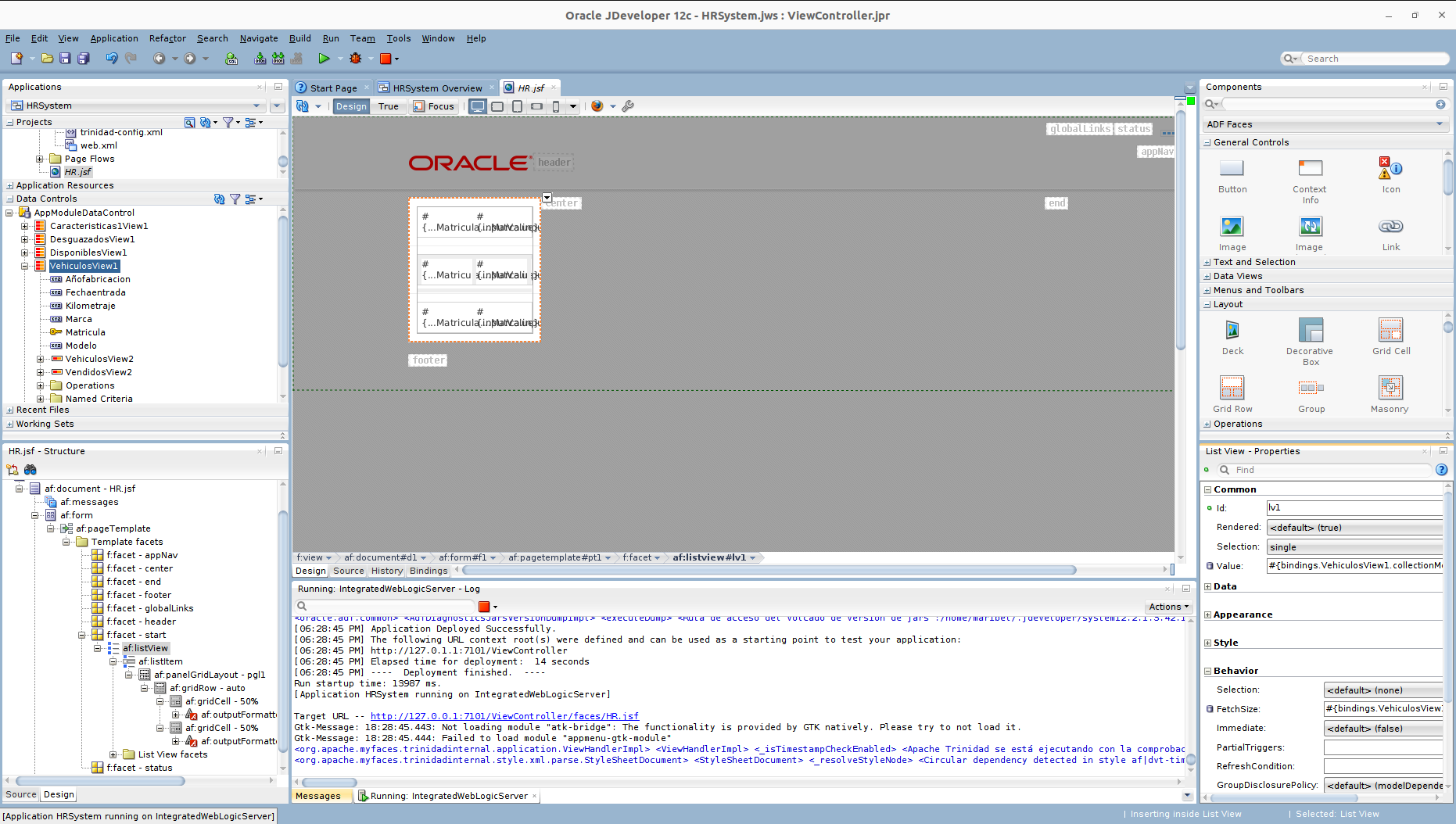
Creamos un formulario (form) que puede usarse para mostrar y modificar datos. Usaremos una master-detail relationship para mostrar la información de los empleados en un departamento. Para construir la página usaremos Oracle ADF Faces Rich Client components. Estos componentes permiten construir un Ajax-based rich Web UI sin escribir código HTML y Javascript de bajo nivel.

Hacemos click con el botón derecho en el nodo del proyecto ViewController en la ventana de Applications y seleccionamos New > Page para crear la nueva página.

La parte Web de la aplicación se desarrolla en un proyecto separado denominado ViewController project, que hemos creado en una parte previa de este tutorial cuando creamos la Fusion Web application.

Mantener una separación entre el Model layer y la interfaz de usuario mediante su desarrollo en proyectos separados hace que los business services layer sean más reutilizables.

A continuación se muestra la página JSF que hemos creado:

Figura 5: JSF que contiene un ADF List View con un Panel Grid Layout.

#### Paso 4: Testeo de la página desde un navegador

En esta sección, se testea la página tal y como está hasta el momento. Si es la primera vez que estamos ejecutando una página desde dentro de JDeveloper, debemos dejar que se instale el *integrated WebLogic Server*.

Así pues, el primer paso es instalar el integrated WLS. Desde el menú Run, se seleccionará *Start Server Instance*. En la ventana Create Default Domain popup, insertamos un valor para el campo password y lo confirmamos. La ventana Log nos mostrará un mensaje cuando el WLS se haya instalado e iniciado.

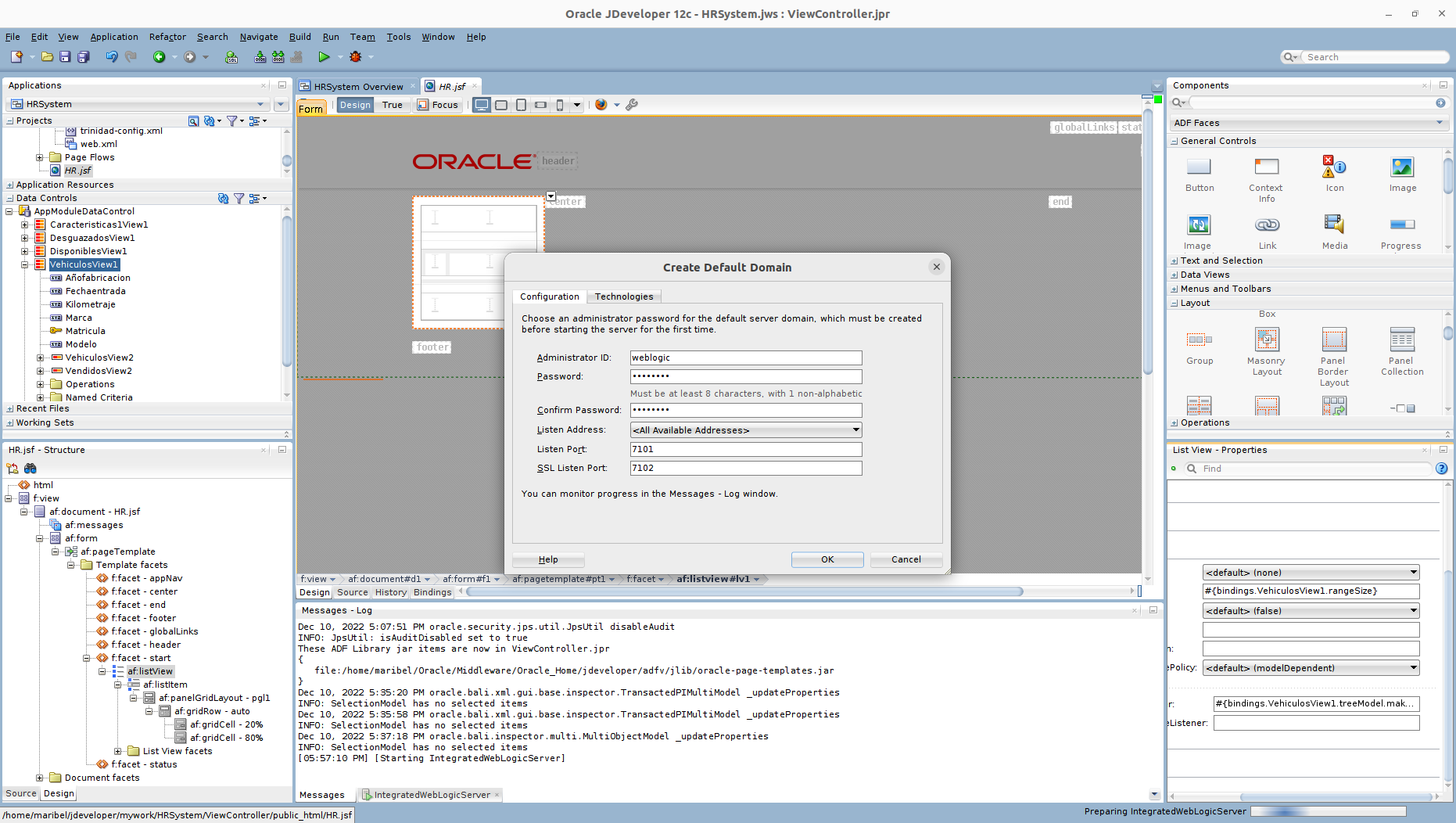


Figura 6: Instalación del integrated WebLogic Server.

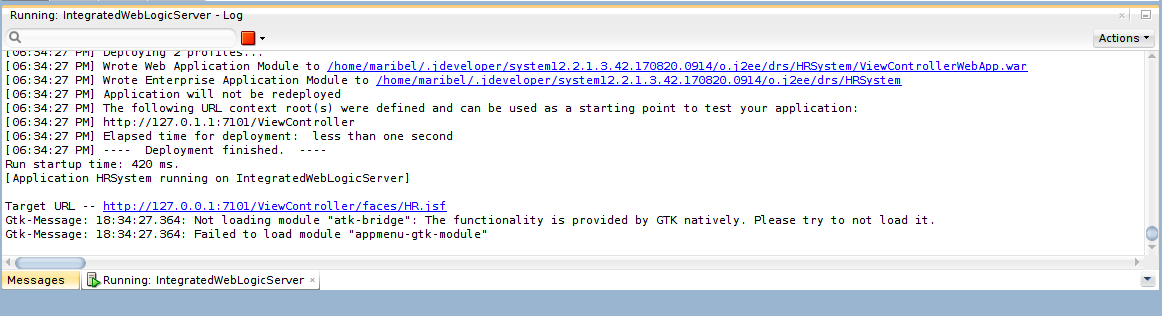
A continuación, desplega la página a WLS. Hacemos click derecho en el editor y visual y desde el menú, seleccionamos Run. La ventana Log nos mostrará la target URL cuando esté desplegada. Nuestro navegador mostrará la página y las dos columnas - Id y Name.

Figura 7: Ventana Log de Oracle JDeveloper.

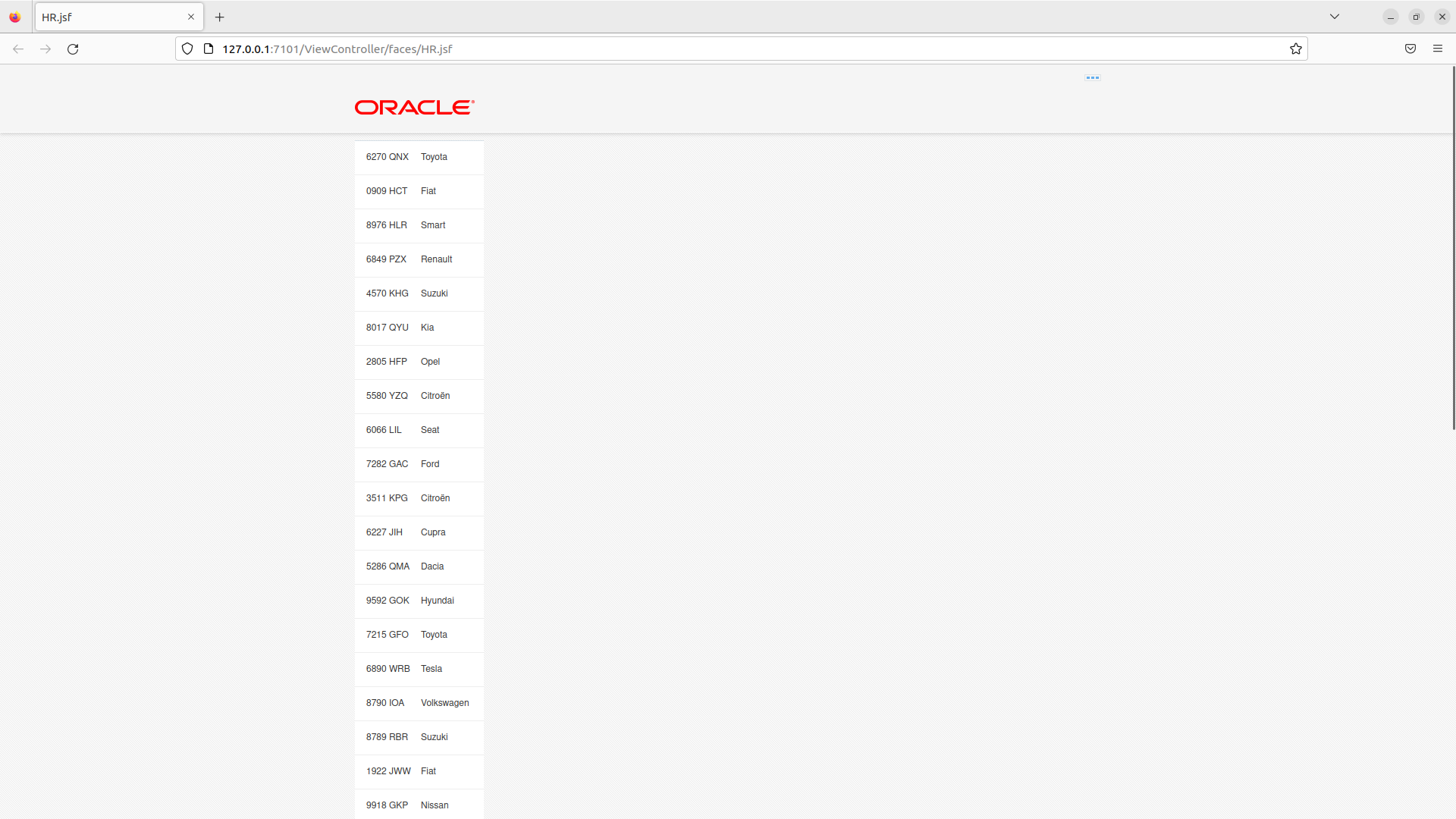


Figura 8: Visualización de la página en un navegador Web.

#### Paso 5: Modificación del layout de la página

En esta sección se modificará la página añadiendo algunos componentes para determinar el layout de la página. En primer lugar se añadirá un dashboard para establecer el layout general de la página. Además, se añadirán componentes que mostrarán datos sincronizados.

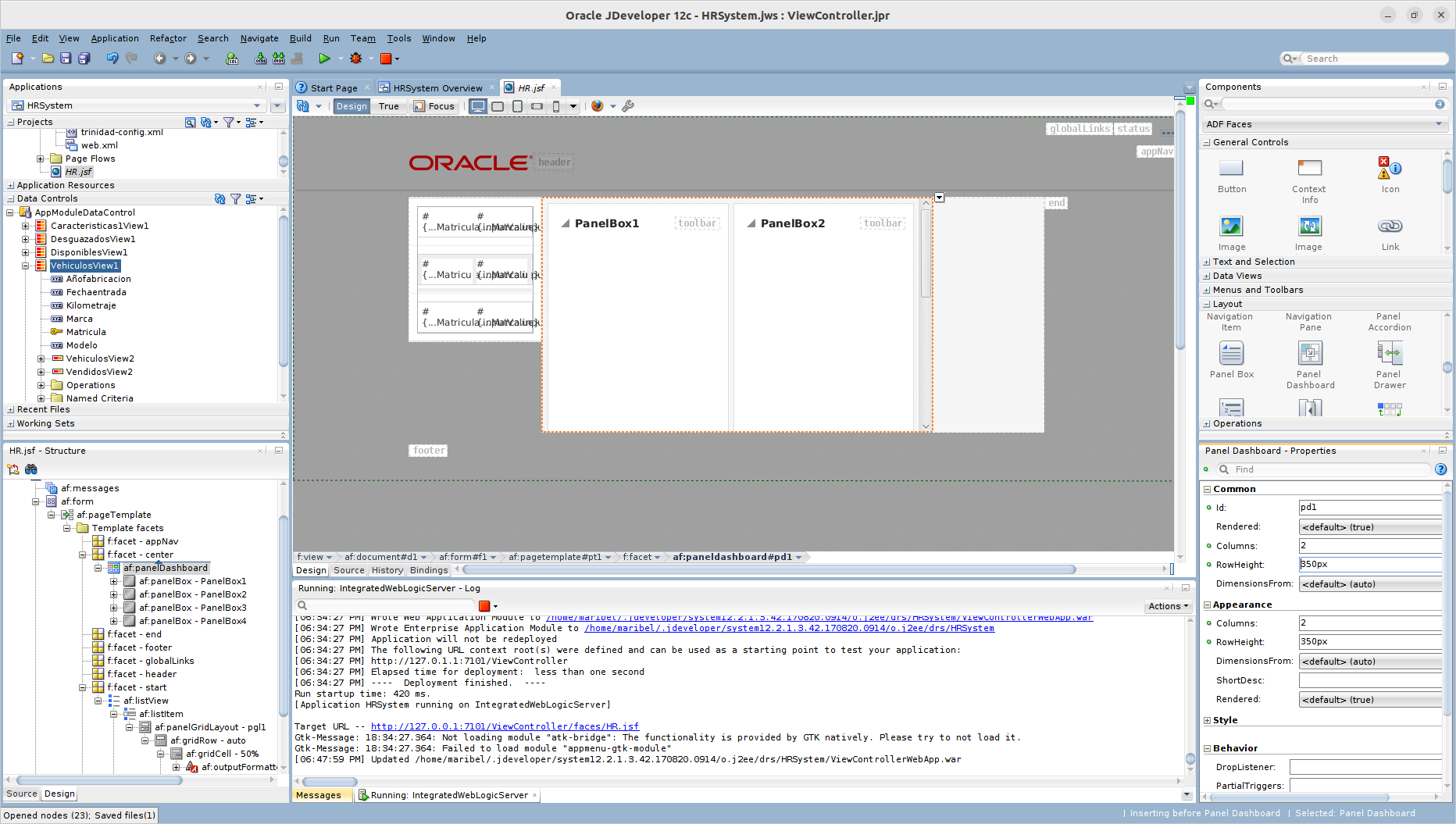


Figura 9: Interfaz JSP. Modificación de la disposición de la página.

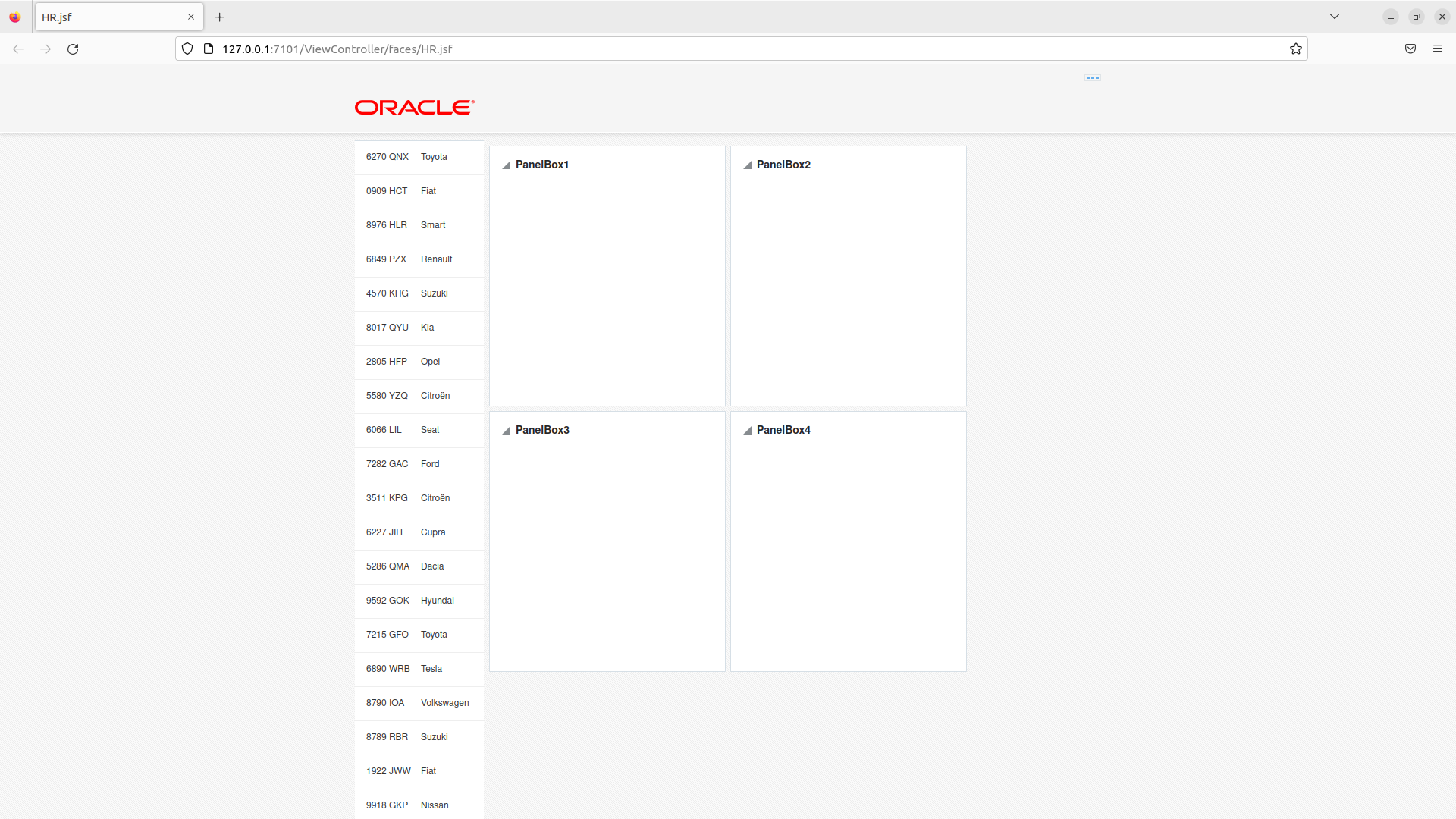


Figura 10: Interfaz gráfica. Modificación de la disposición de la página.

#### Paso 6: Refinar la pantalla principal de la aplicación y la visualización de los datos.

En esta parte del tutorial modificaremos la interfaz de usuario añadiendo una tabla de la base de datos de nuestra aplicación, permitiendo seleccionar columnas, reordenar los campos y relacionando los business components a una tabla.

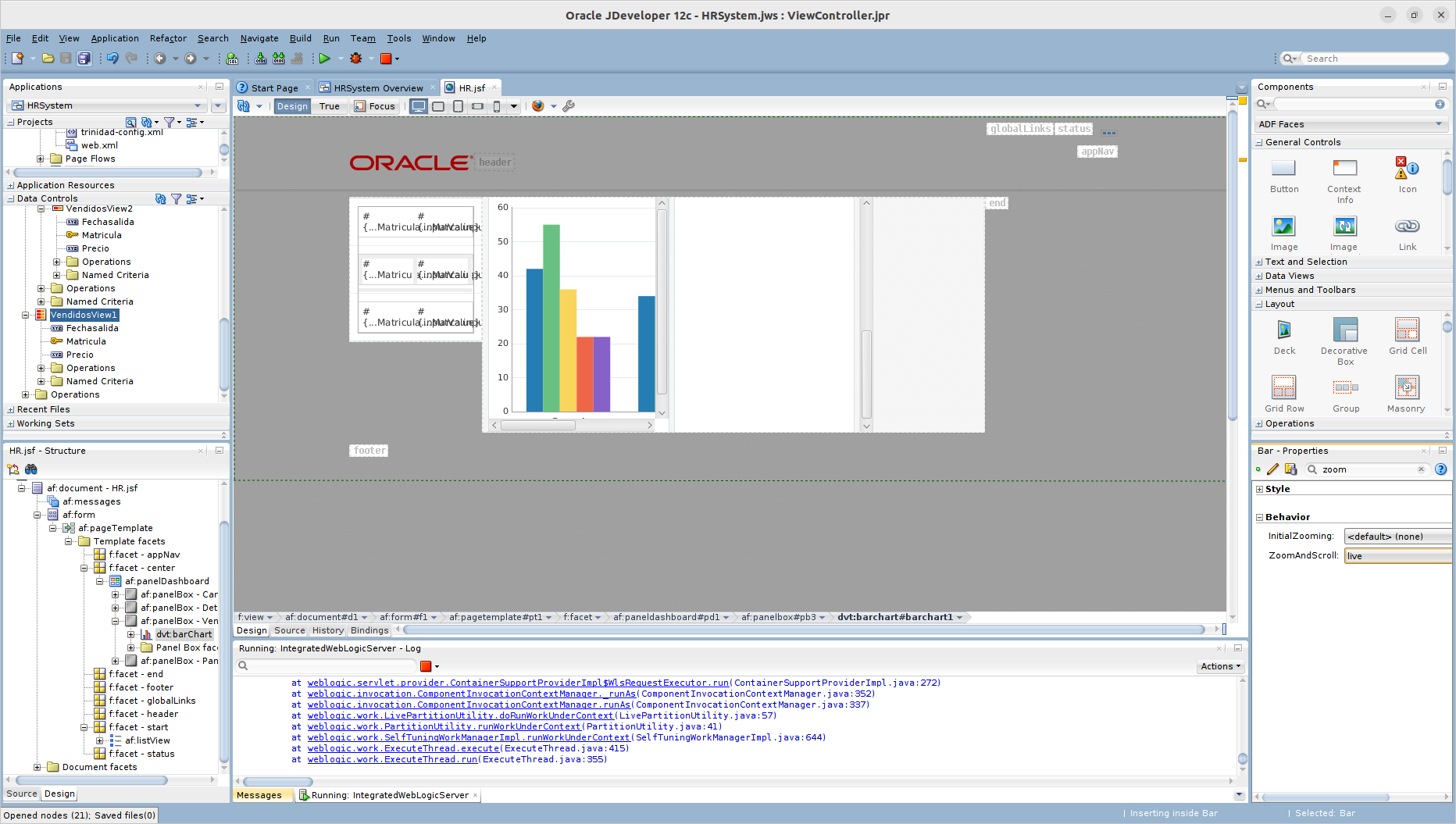


Figura 11: Interfaz JSP. Primera visualización de datos.

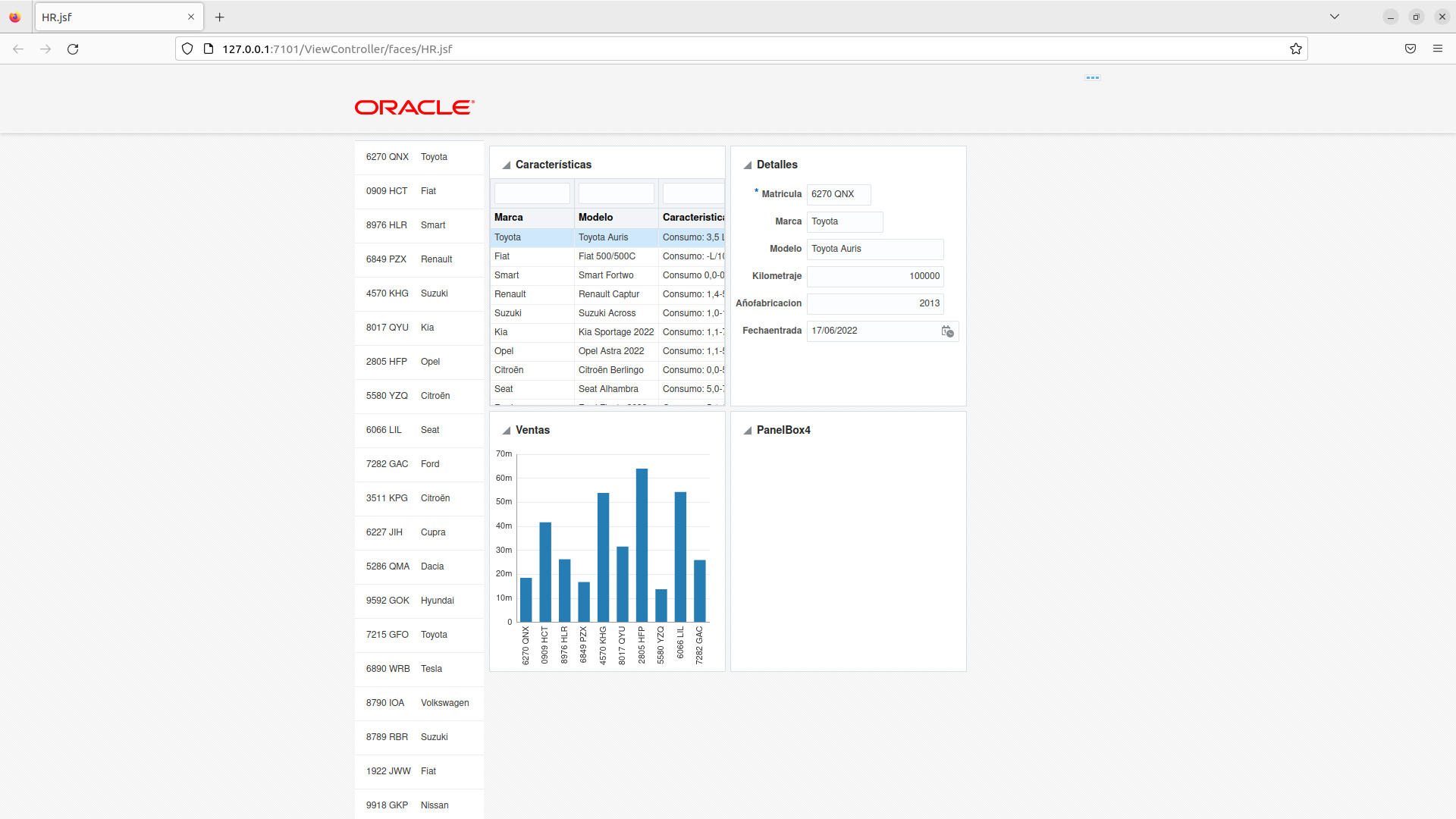


Figura 12: Interfaz gráfica. Primera visualización de datos.

#### Paso 7: Ajuste de los componentes del modelo para refinar cómo visualizar los datos.

En esta parte del tutorial, refinamos los componentes de una página de nuestra aplicación, añadiremos comprobaciones sobre las restricciones de los datos y modificaremos como se muestra la interfaz de usuario.

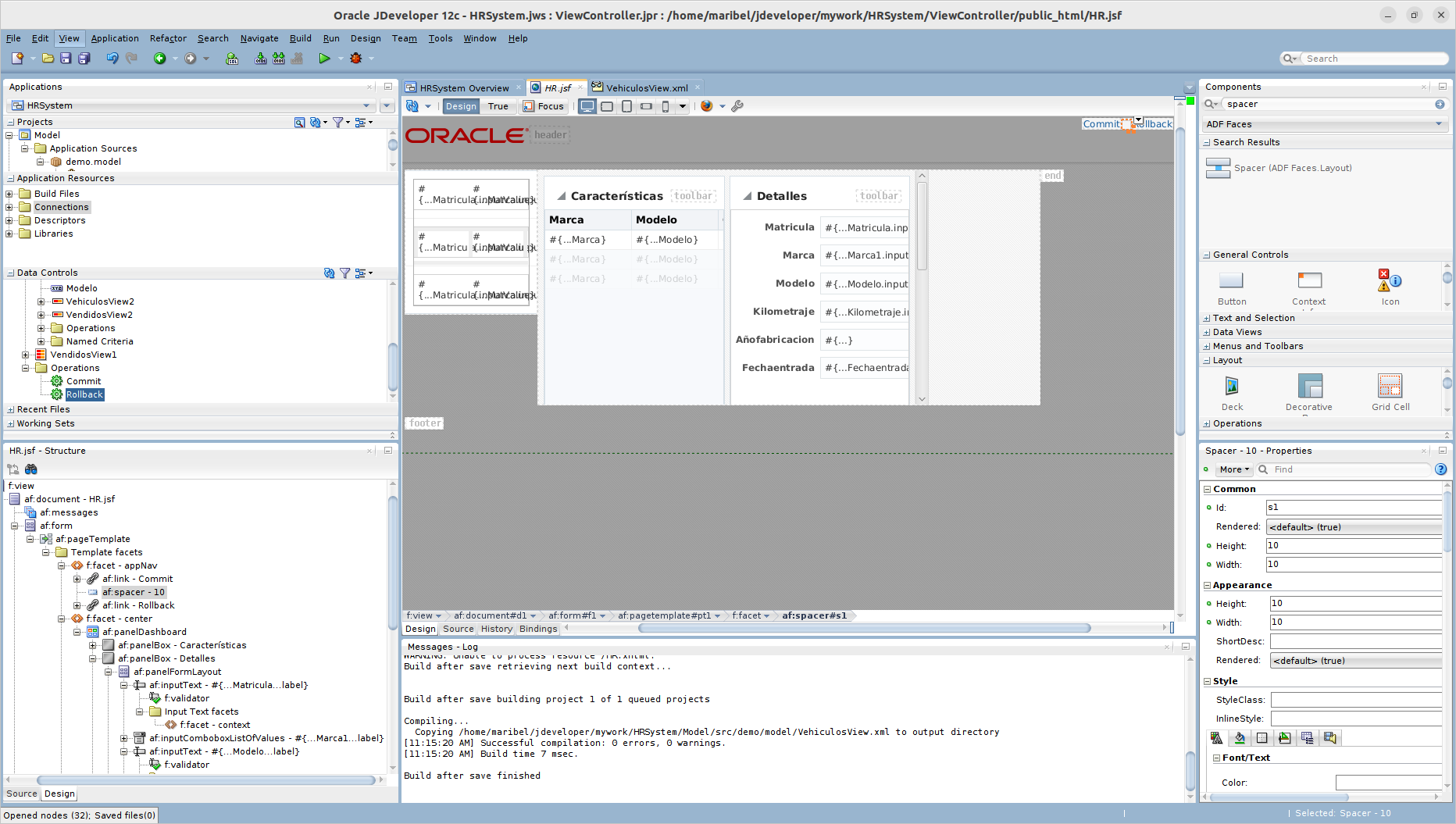


Figura 13: Interfaz JSP. Cambios en la visualización de los datos.

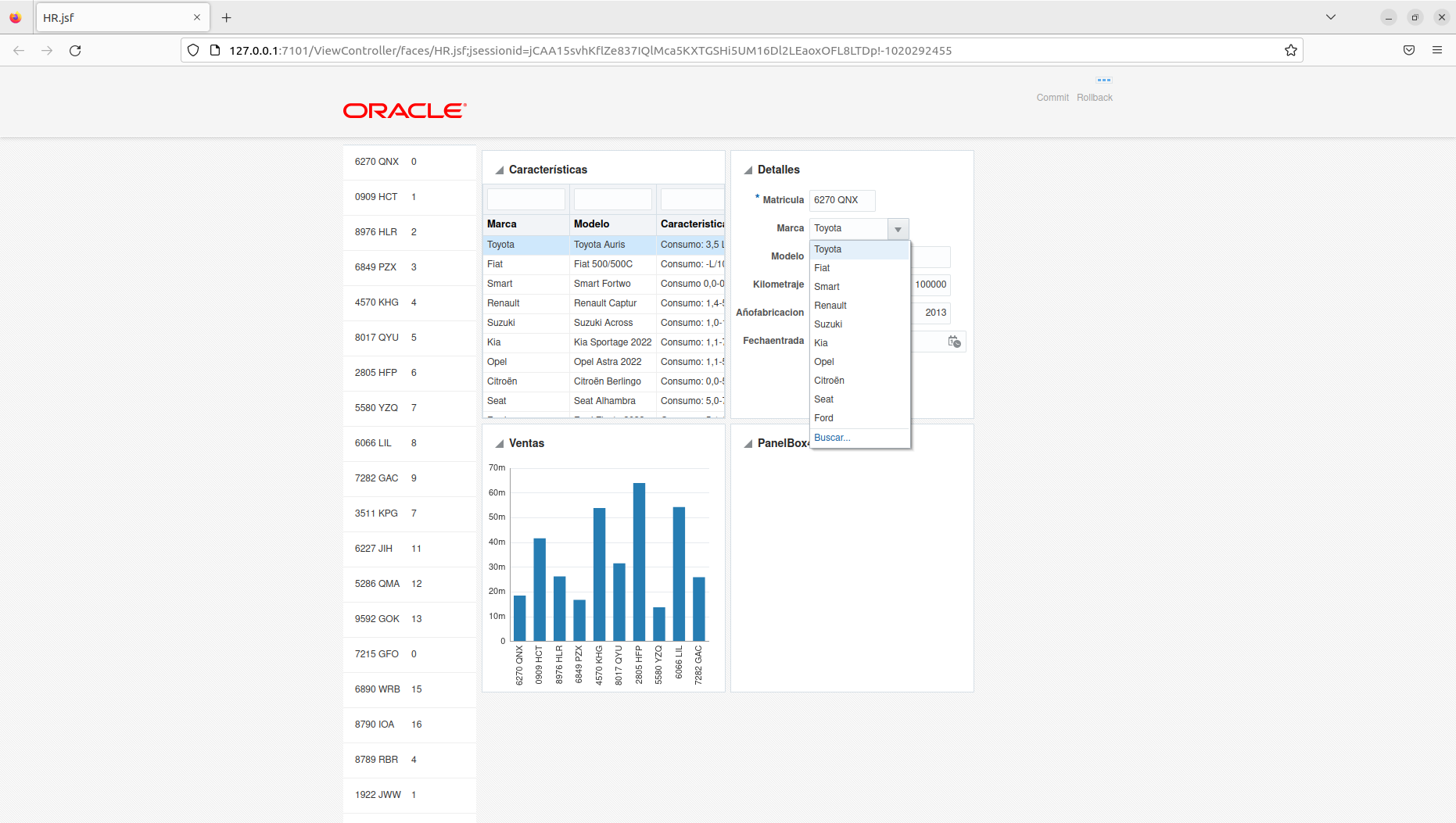


Figura 14: Interfaz gráfica. Cambios en la visualización de los datos.

#### Paso 8: Uso de un controlador ADF para determinar las relaciones entre las páginas.

Las aplicaciones Web normalmente tienen más de una página. En esta parte del tutorial, usaremos ADF Task Flow Diagrammer para definir la navegación entre dos páginas de nuestra aplicación. Después, usaremos características que proporciona AFD Faces Framework para añadir funcionalidad extra a las páginas. Finalmente, añadiremos el task flow definido a uno de los paneles de la página.

En esta sección trabajaremos con el proyecto ViewController e incluiremos task flows para determinar la navegación entre páginas. Crearemos un nuevo task flow y le añadiremos view components y control flows, para después mostrarlos en la interfaz gráfica.

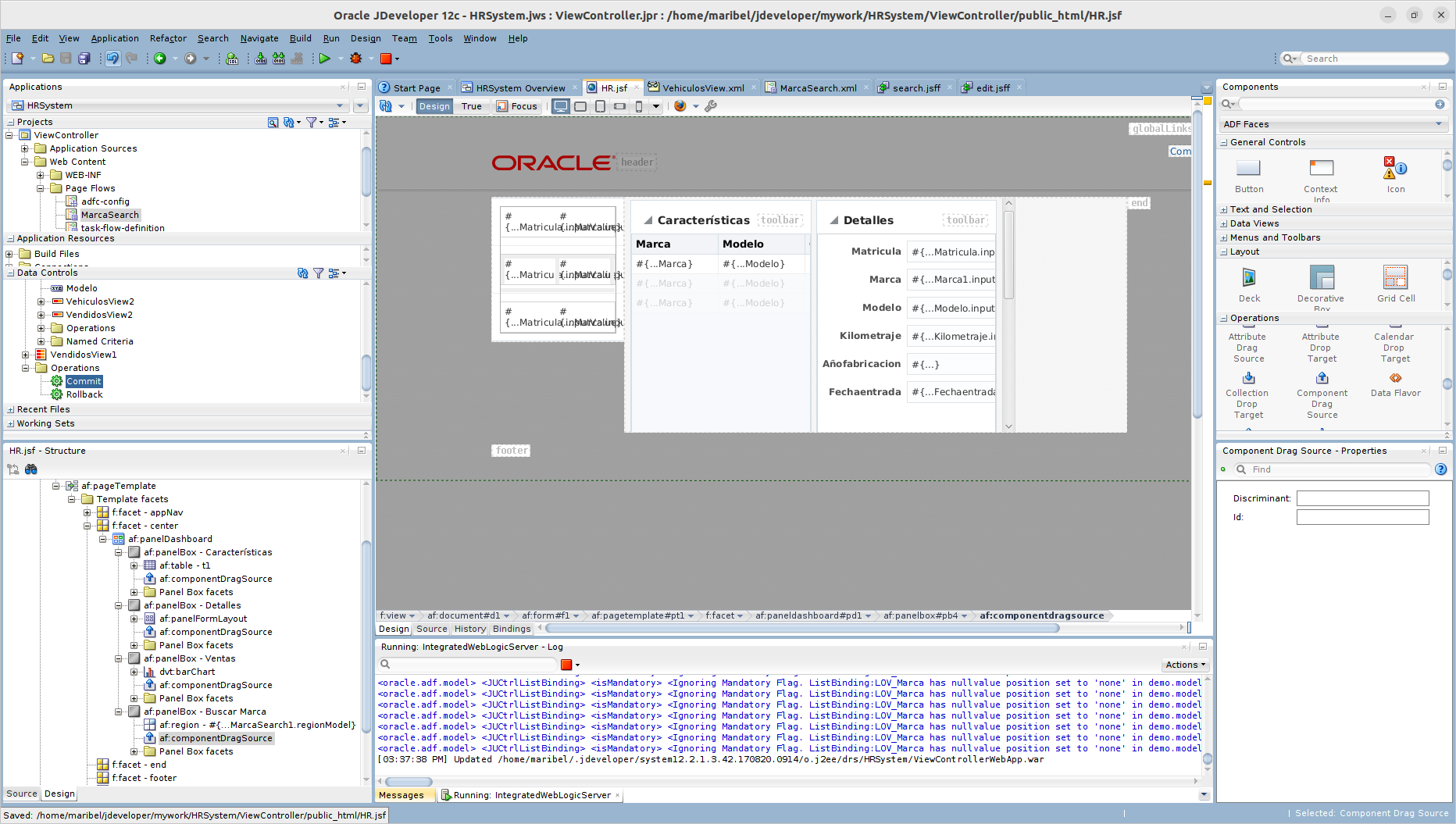


Figura 15: Interfaz JSP. Aplicación final.

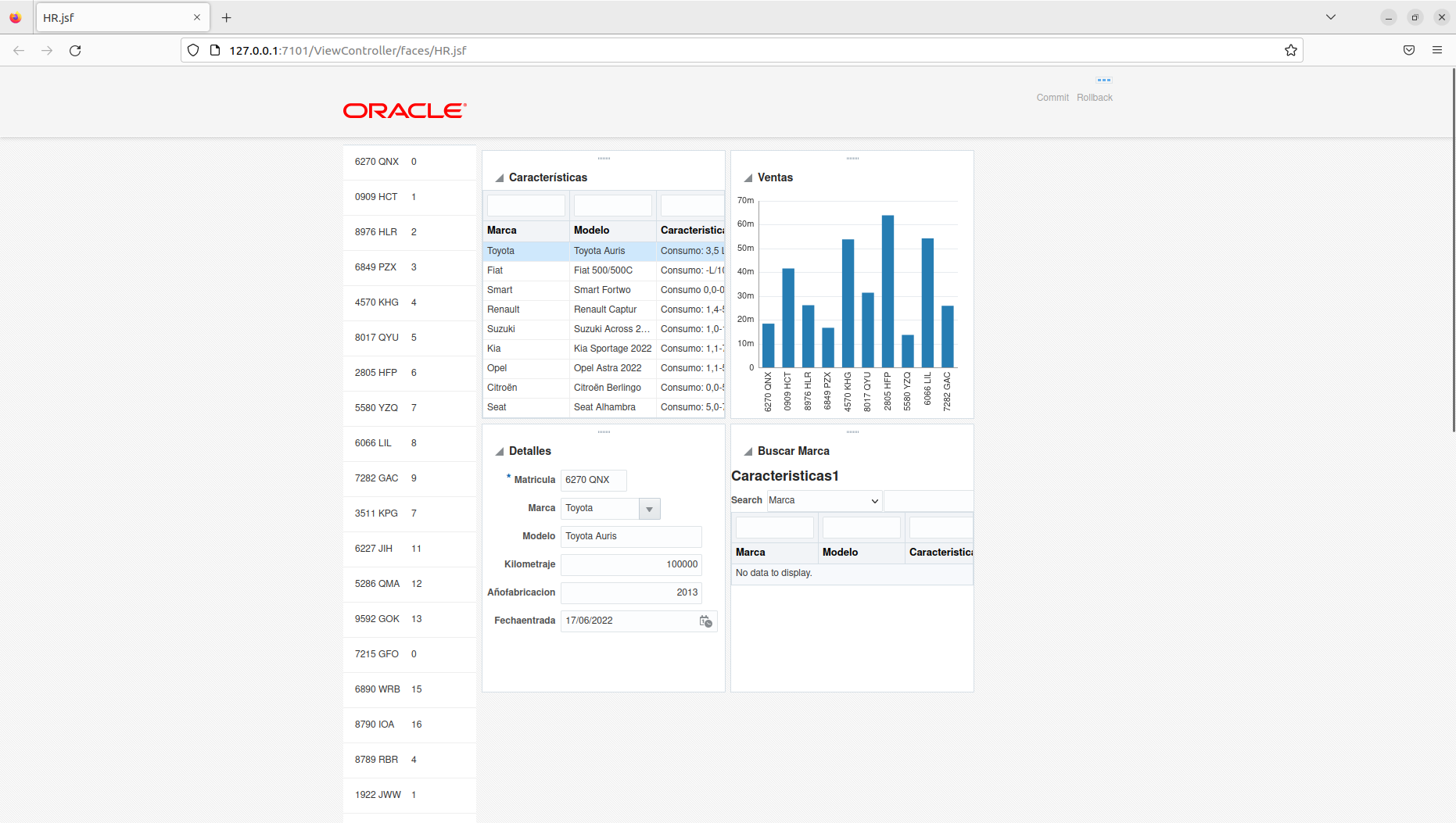


Figura 16: Interfaz gráfica. Aplicación final.

### Problemas al realizar el tutorial

En el paso 6, hemos seguido los pasos para poder hacer zoom en la gráfica que se muestra en el dashboard pero no ha funcionado.

Además, nada ocurrirá si seleccionamos alguno de los registros de la tabla que se muestra a la izquierda dado que tenemos una estructura de tablas diferente a la que se emplea en el tutorial.