

## PRÁCTICA N° 3

### Temas

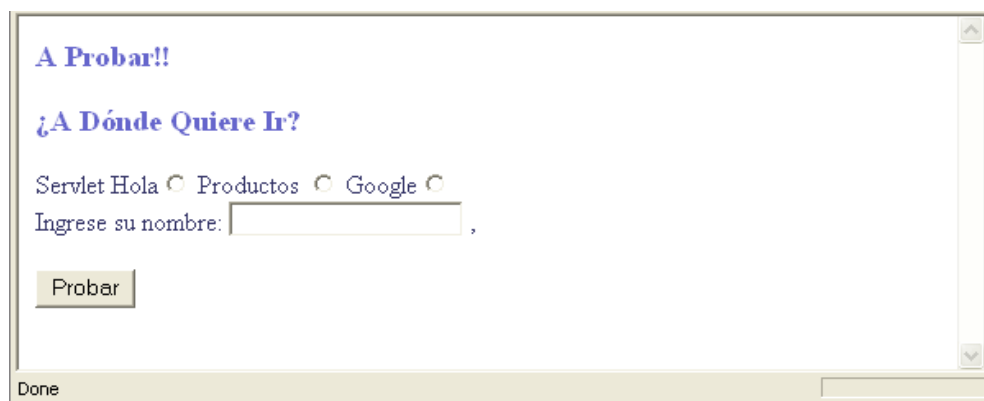
- Invocación de componentes web usando el método `forward()` de `RequestDispatcher`
- Redireccionamiento de la respuesta al cliente usando el método `sendRedirect()`
- Servlet listeners de contexto
- Servlet Context

Cree un proyecto nuevo llamado **Practica3** y asígnele como nombre del contexto (root/context path): **pruebas**.

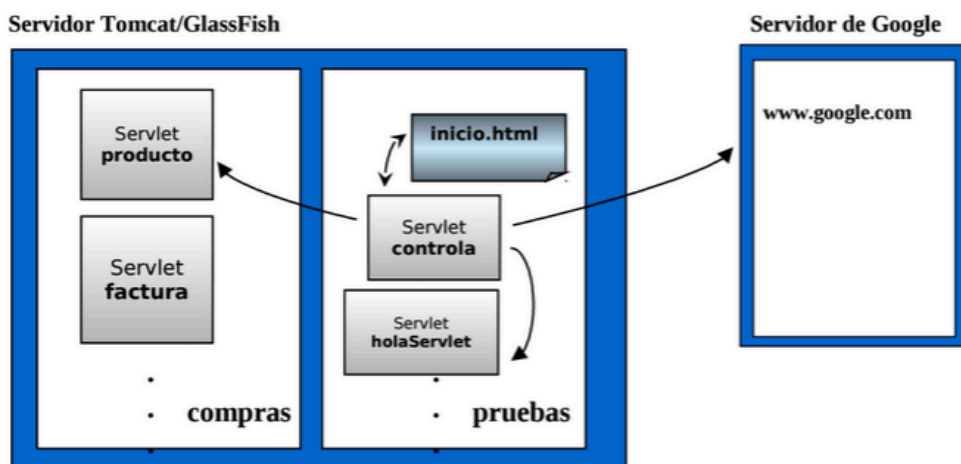
### Ejercicio 1)

Escriba un servlet llamado **controla** en el paquete **misservlets**. La funcionalidad de este servlet es leer parámetros del requerimiento http y usarlos para determinar que recurso invocar.

En este ejercicio, debe escribir una página HTML llamada **inicio.html**, que contenga, un formulario con **3 radio\_buttons** (Servlet Hola, Productos, Google), **un campo de texto** con la etiqueta **Ingrese su nombre** y un **botón** de tipo *submit* para enviar los datos ingresados. Los datos del formulario se envían al servlet **controla** para su procesamiento.



El servlet **controla** redirecciona el requerimiento de acuerdo a la selección del usuario:



Si el radio seleccionado es "**Servlet Hola**": se redirecciona a **holaservlet** (un servlet que se está ejecutando en el mismo contexto que el servlet **controla**). El servlet **/holaservlet** simplemente debe devolver una página HTML con el mensaje "Bienvenida"+nombre (tomado del parámetro del requerimiento).  
¿Qué mecanismo utilizó? ¿Existe alguna otra manera de hacerlo?

Si el radio seleccionado es "**Productos**": se debe redireccionar al Servlet **/productos**, que está en el mismo servidor que el servlet **controla** pero en el contexto **compras** (proyecto **Practica2** creado en la práctica anterior).  
¿Qué mecanismo utilizó? ¿Existe alguna otra manera de hacerlo? ¿Qué es el acceso CrossContext?

Si el radio seleccionado es "**Google**": se debe redireccionar al sitio de Google (<http://www.google.com.ar>). Este sitio está en otro Servidor.  
¿Qué mecanismo utilizó? ¿Existe alguna otra manera de hacerlo?

Si el dato ingresado no es ninguno de los de arriba, se invoca nuevamente a **inicio.html**.

## Ejercicio 2)

Realice modificaciones al ejercicio de la práctica 2 de manera de cargar los datos de las golosinas desde un archivo de texto. Para ello realice lo siguiente:

- Cree un archivo de texto llamado **stock.txt** en la carpeta WEB-INF. Inicialícelo de la siguiente manera: en cada línea escriba el nombre de una golosina y su precio unitario separado por una coma (","), Cargue por lo menos 10 productos.

### Ejemplo:

|                    |
|--------------------|
| Chocolatines, 1.25 |
| Chupetines, 0.90   |
| Caramelos, 0.10    |
| Chicles, 0.05      |
| Pastillas, 1       |

} **stock.txt**

- Agregue al archivo **web.xml** el siguiente parámetros de inicialización de la aplicación web (antes del primer tag `<servlet>.. </servlet>`):  

```
<context-param>  
<param-name>stock</param-name>  
<param-value>/WEB-INF/recursos/stock.txt</param-value>  
</context-param>
```
- Cree un *servlet listener* llamado **InicializaStock**, cuya funcionalidad es inicializar los objetos necesarios para mantener en memoria las golosinas y precios del stock. Para crear el listener, seleccione la opción **New ...** elija dentro de la categoría **Web Tier -> Servlet -> Servlet Listener**. Implemente únicamente la interface `javax.servlet.ServletContextListener`.

En el método **contextInitialized(ServletContextEvent event)** se deben crear los objetos necesarios a partir de los datos leídos del archivo de texto **stock.txt**, cuya ubicación debe obtenerse del parámetro de contexto "stock" declarado en el archivo **web.xml**.

Los objetos creados se deben ligar al contexto de la aplicación, invocando al método **setAttribute(clave, Objeto)** del objeto ServletContext. De esta forma, cada vez que desde un servlet de su aplicación, necesite usar el stock de golosinas, deberá invocar al método **getAttribute()** del ServletContext para obtener dicho objeto.

**Ayuda:** Para leer del archivo de texto use el método **getResourceAsStream()** de ServletContext que devuelve un objeto InputStream. Luego cree un objeto BufferedReader a partir del objeto InputStream obtenido y así podrá leer de a líneas el archivo de texto.

- Observe el archivo **web.xml**, ¿qué modificación o nuevo tag encuentra, luego de haber agregado el *servlet listener*?

- d. Modifique el servlet **Productos**, eliminando el código del método init() y modificando el método doGet()/doPost() de manera de obtener la información para armar la página usando los objetos del contexto de la aplicación creados en el servlet listener.

**Nota:** recuerde que el método **getAttribute(clave)** le permite recuperar del ServletContext el objeto ligado previamente con la **clave** pasada como parámetro. El objeto obtenido debe *castearlo* al tipo que corresponda.

- e. Modifique servlet **Facturar** para obtener los precios de las golosinas que el usuario compró, del objeto del contexto aplicación creado en el servlet listener.