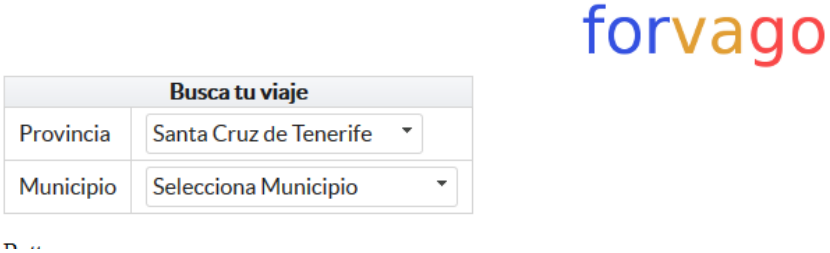
AUT06_04 Proyecto ForVago-Fase II

1. Página principal de búsqueda

Ubicada en **/search/index.xhtml**

1.1 Notas de implementación (Fase II)

Comenzaremos, en esta fase II, por implementar los *dropdown* de selección de provincia y municipio.



1.1.1 DAOs necesarios: ProvinciasDao y MunicipiosDao

Para ello necesitamos que estén disponibles los DAO: ProvinciasDao y MunicipiosDao, con, al menos, el método **selectAll** operativo.

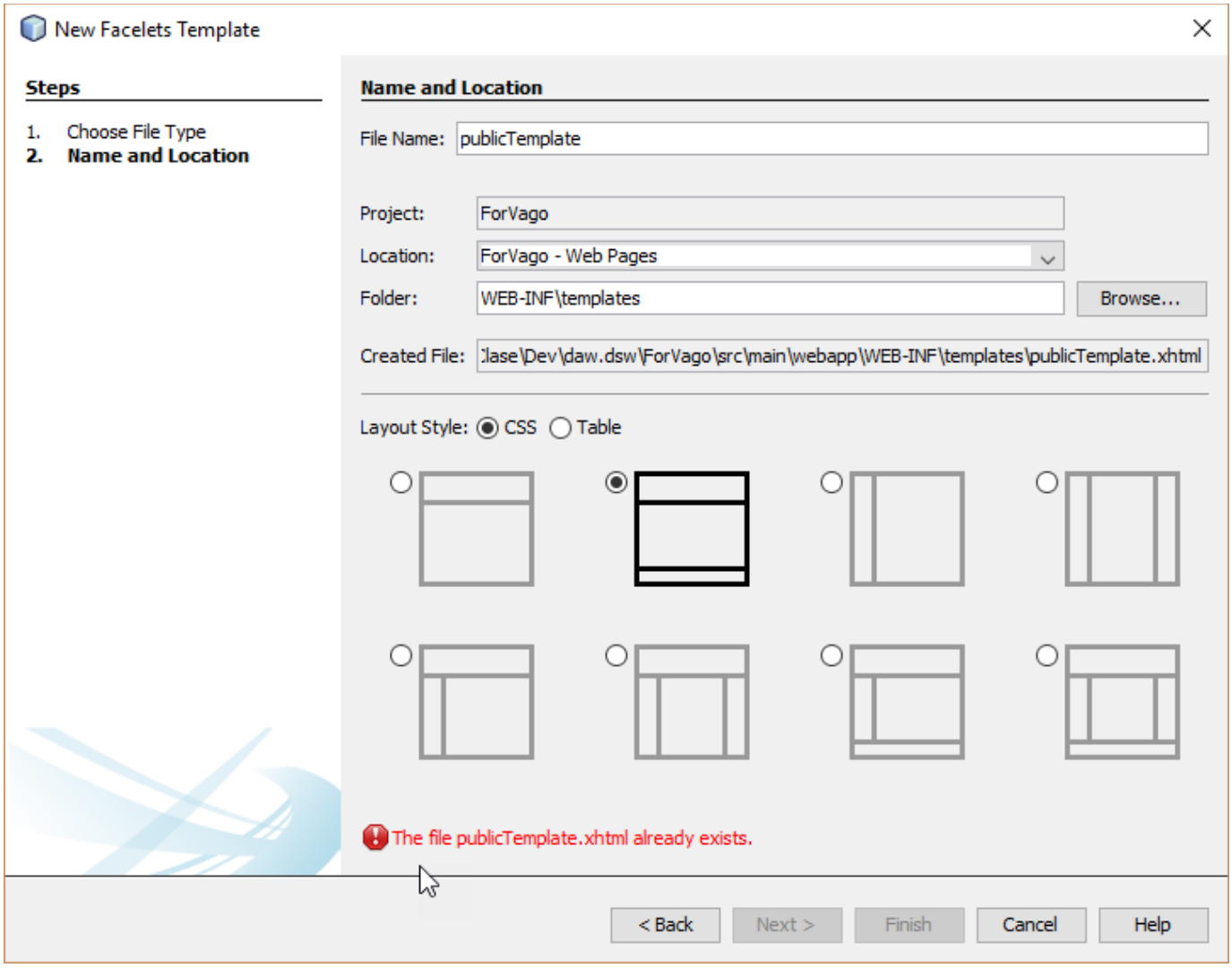
Las tablas a añadir a la base de datos se encuentran en la carpeta de recursos que está en la plataforma.

1.1.2 Layout base de la página

Vamos a crear un template, ubicado en <Web Pages>/WEB-INF/templates/publicTemplate.xhtml con una disposición de cabecera, contenido y pie.

Para ello desde el proyecto: **New > Other… > Java Server Faces > Facelets Template**

Las opciones del nuevo template son las mostradas a continuación.



En la cabecera del propio template, podemos añadir la imagen de ForVago

forvago

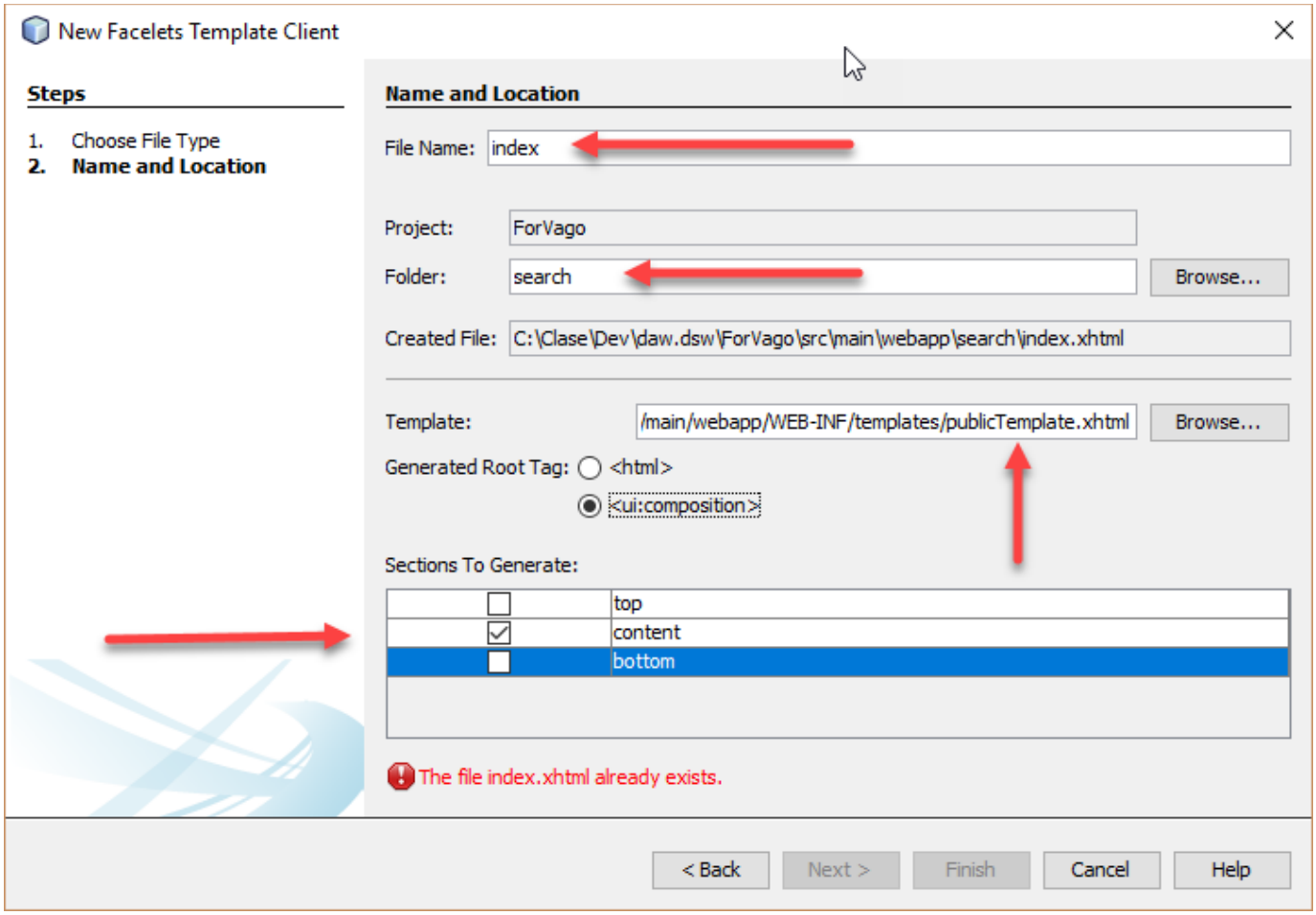
1.1.3 Cliente del template

Nuestra página de búsqueda, estará ubicada en **<WebPages>/search/index.xhtml**, pero no vamos a crearla como página JSF, sino como **Template Client**.

Para ello, desde el proyecto: **New > Other… > Java Server Faces > Facelets Template client**

- Seleccionaremos como nombre **index**
- Se creará en la carpeta **search**
- Seleccionamos el template creado en el paso anterior, buscando en WEB-INF\templates
- Seleccionamos **<ui:composition>** como tag raíz
- Indicamos que vamos a redefinir sólo el cuerpo del template.

Se



crea una página index.xhtml que utiliza el template anterior, como se ve en el código de la página.

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://ww
<ui:composition xmlns:ui="http://xmlns.jcp.org/jsf/facelets"
                template="./../WEB-INF/templates/publicTemplate.xhtml"
                xmlns:h="http://xmlns.jcp.org/jsf/html"
                xmlns:p="http://primefaces.org/ui"
                xmlns:f="http://xmlns.jcp.org/jsf/core">

    <ui:define name="content">


        <h:form>

            <p:panelGrid columns="2">
```

1.1.4 Formulario de entrada

En esta página añadiremos un formulario con dos elementos **selectOneMenu**. Podemos ver ejemplos de uso en:

http://www.primefaces.org/showcase/ui/input/oneMenu.xhtml

https://www.mkyong.com/jsf2/jsf-2-dropdown-box-example/

Las entradas a mostrar pueden añadirse una a una con tags **selectItem** o toda una colección entradas con el elemento **selectItems** (en plural)

También podemos utilizar un panelGrid para colocar los controles.

http://www.primefaces.org/showcase/ui/panel/panelGrid.xhtml

1.1.5 Managed Bean MasterDataBean

Crearemos un ManagedBean de ámbito **aplicación**, que será quien precargue y mantenga las listas de provincias y municipios: **es.cifpcm.forvago.web.common.MasterDataBean**.

Tendrá dos métodos: **+getProvincias(): List<Provincia>** y **+getMunicipios(): List<Municipio>**

Obviamente estos métodos llamarán a los DAO.

1.1.6 Managed Bean HotelSearchBean

Éste será de ámbito **request**: **es.cifpcm.forvago.web.search.HotelSearchBean**

Este bean respaldará las propiedades **-idProvincia: Short, -IdMunicipio: Short**

Además contará con los métodos **+getProvincias(): List<Provincia>** y **+getMunicipios(): List<Municipio>**

Sí, estos métodos ya están en el bean de aplicación.

El método **getProvincias**, simplemente llamará al método **getProvincias de MasterDataBean**

En cambio el método **getMunicipios** queremos que sólo devuelva aquellos municipios que correspondan a una provincia seleccionada. Para ello:

Si idProvincia es null devuelve el getMunicipios de MasterDataBean

En cambio, si IdProvincia!=null, coge la lista de Municipios de MasterDataBean y la filtra, quedándose sólo con aquellos registros donde idProvincia coincida con la seleccionada en el combo de provincias.

1.1.7 Utilizar Ajax para que cuando se cambie el combo de provincias se actualice el de municipios

Dentro del elemento <p:selectOneMenu> de cboProvincias, vamos a anidar un elemento ajax:

```
<p:ajax listener="#{hotelSearchBean.onCboProvinciasChange()}" update="cboMunicipios"/>
```

Esto lo que hace es que cuando cambia la selección de este combo, se llama a un método onCboProvinciasChange, método que crearemos en HotelSearchBean y será el que filtre la lista de municipios, cuando cambie la provincia.

Consulten en google si les hace falta más información.