

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	

Migración Reportes SSiCC a NSSiCC

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	

Tabla de contenido

1. Objetivos.....	4
2. Consideraciones Previas	5
2.1. SSiCC	5
2.2. Nomenclatura	6
2.3. Estructura	6
2.4. Modularidad del NSSiCC	7
2.6. Carpeta target del NSSiCC	9
2.7. Uso de Anotaciones.....	9
2.8. Variables de Sesión.....	9
2.9. Ubicación Clase Action	10
2.10. Métodos Bases.....	11
2.10.1. setViewAttributes:.....	11
2.10.2. setFindAttributes:	11
2.11. Uso de AJAX.....	11
2.12. Páginas WEB y templates.....	12
2.13. Componentes Personalizados	12
2.13.1. Etiqueta asociada a la caja de texto	12
2.13.2. Caja de Texto	13
2.13.3. Campo Fechas	14
2.13.4. Campo Periodo ó Campaña.....	14
2.13.5. Combox o Lista (Selección Única)	15
2.13.6. Combox o Lista (Selección Múltiple)	16
2.13.7. Campo TextArea	17
2.13.8. CheckBox	17
2.13.9. RadioButton	18
2.13.10. Grilla Datatable (Modo Consulta o de Selección simple)	18
2.13.11. CommandButton	20
2.14. Ubicación archivos XML de ibatis.....	20
2.15. Reportes.....	21
2.16. Asignando Página xhtml al Menú.....	22
2.17. Pruebas Internas	23
3. Migración de Reportes	25
3.1. Verificar existencia de Carpeta.....	25
3.2. Obteniendo Clase Action	25
3.3. Modificando Clase Action en el NSSiCC	27

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	

3.4.	Creando Clase Form en el NSSiCC	28
3.5.	Revisando métodos de la Clase Action en el NSSiCC	31
3.6.	Revisando invocaciones a Clase Service y DAO	32
3.7.	Visualizando Botones del Reporte	32
3.8.	Verificando método devuelveNombreReporte	33
3.9.	Verificando método devuelveNombreSubReporte	33
3.10.	Eliminar try catch	34
3.11.	Verificar método Reset en la Clase Form	34
3.12.	Creando página xhtml	34
3.13.	Campos Fechas	36
3.14.	Campos Periodos que invoquen Ajax	36
3.15.	Agregando los Filtros en página xhtml	37
3.16.	Creando Clase Service (Revisar método beforeExecuteReporte)	38
3.17.	Revisar código javascript en la página jsp del SSiCC	39
3.18.	Verificar existencia archivos jrxml	39

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	

1. Objetivos

El presente documento tiene como finalidad servir como Documento Base para la migración del Código Fuente correspondiente a las Pantallas de Reportes del SSiCC al Nuevo SSiCC

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	

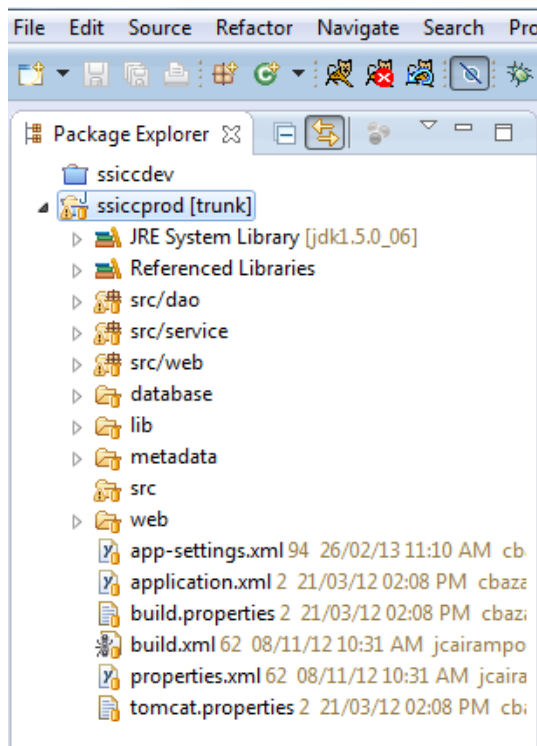
2. Consideraciones Previas

2.1. SSiCC

Debemos ingresar al proyecto SSiCC bajo el Eclipse configurado para dicho sistema y en base proyecto ssicccprod debemos realizar la migración respectiva.

Dicho proyecto se encuentra en el siguiente repositorio:

<https://pents73.galileo.ebel:8082/svn/ssicc/trunk>



Para acceder al Sistema SSiCC ingresar a la siguiente ruta (**Ejecutarlo solo bajo Internet Explorer en Modo de Compatibilidad**)

<http://pelinx3196:7777/ssicccpeesdev/login.jsp>

Usuario: ADMIN

Password: BELCORP

País: PERU ESIKA

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	



2.2. Nomenclatura

Las clases a migrar en el NSSiCC deberán respetar la nomenclatura existente en el SSiCC. Una excepción son las páginas WEB que en el SSiCC son de extensión jsp, mientras que en el NSSiCC se le colocará con el mismo nombre, pero con **extensión xhtml**

2.3. Estructura

El SSiCC trabaja bajo una estructura propietaria manejada por ANT para la generación del WAR respectivo, mientras NSSiCC se adecua a la estructura establecida para los proyectos JAVA generados por MAVEN

Debido a lo mencionado el SSiCC tiene la siguiente estructura (Existe excepciones en algunos casos):

- Clase Java de Capa WEB (Action):
biz.belcorp.ssicc.<modulo>.web.action;
- Clase Java de Capa WEB (Form):
biz.belcorp.ssicc.<modulo>.web.form;
- Clase Java de Capa Service (Interface):
biz.belcorp.ssicc.<modulo>.service;
- Clase Java de Capa Service (Implementación):
biz.belcorp.ssicc.<modulo>.service.impl;

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	

- Clase Java de Capa DAO (Interface):
biz.belcorp.ssicc.<modulo>.dao;
- Clase Java de Capa DAO (Implementación):
biz.belcorp.ssicc.<modulo>.dao.ibatis;
- Clase Java de Capa DAO (Pojo):
biz.belcorp.ssicc.<modulo>.dao.model;
- Archivo XML de Capa DAO:
biz / belcorp / ssicc / <modulo> / dao / ibatis / sql

En el NSSiCC varia la estructura de archivos debido que debe cumplir con el estándar de estructura establecido por MAVEN:

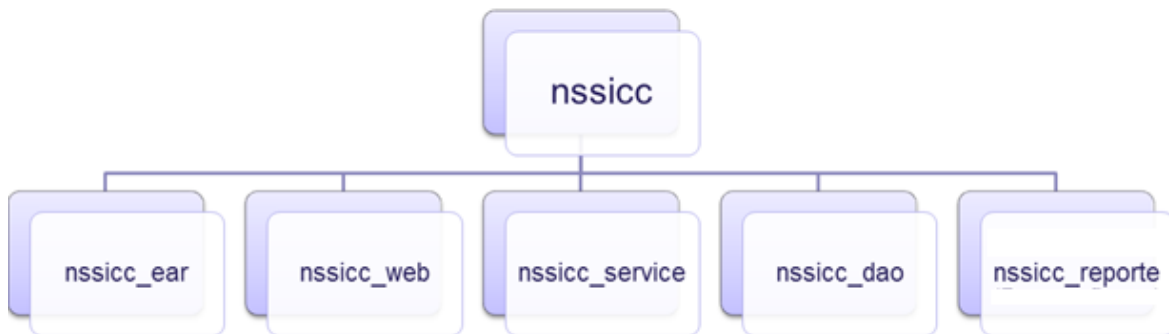
- Clase Java de Capa WEB (ManageBeans → Reemplaza al Action):
biz.belcorp.ssicc.web.<modulo>.action;
- Clase Java de Capa WEB (Form):
biz.belcorp.ssicc.web.<modulo>.form;
- Clase Java de Capa Service (Interface):
biz.belcorp.ssicc.service.<modulo>;
- Clase Java de Capa Service (Implementación):
biz.belcorp.ssicc.service.<modulo>.impl;
- Clase Java de Capa DAO (Interface):
biz.belcorp.ssicc.dao.<modulo>;
- Clase Java de Capa DAO (Implementación):
biz.belcorp.ssicc.dao.<modulo>.ibatis;
- Clase Java de Capa DAO (Pojo):
biz.belcorp.ssicc.dao.<modulo>.model;
- Archivo XML de Capa DAO:
biz / belcorp / ssicc / dao / <modulo> / ibatis / sql (Se encuentran a nivel de la carpeta resources)

2.4. Modularidad del NSSiCC

Podrán apreciar que al ver el proyecto en el Eclipse se verá varios proyectos. Sin embargo no son proyectos separados, sino son subproyectos del proyecto padre que es el nssicc.

Se ha subdivido el proyecto bajo los siguientes subproyectos:

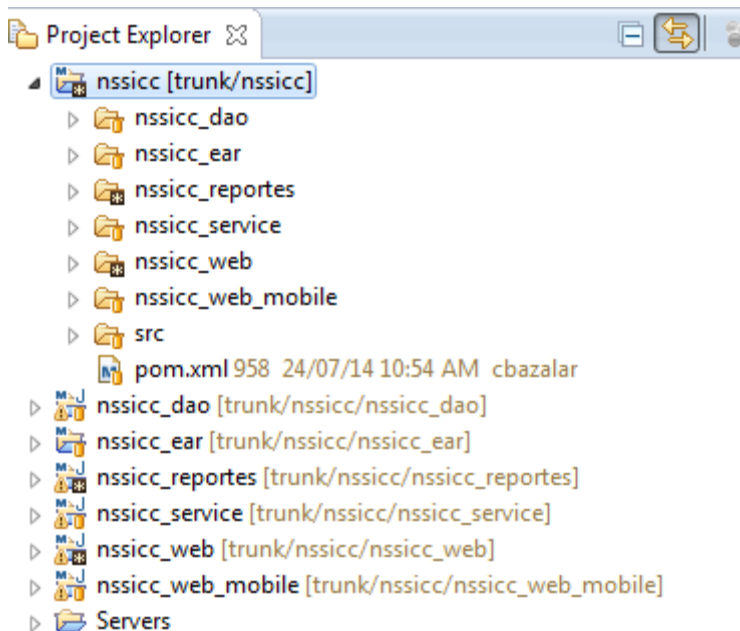
	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	



Donde:

- nssicc → Proyecto Padre
- nssicc_web → Proyecto correspondiente a la Capa Web
- nssicc_service → Proyecto correspondiente a la Capa Service
- nssicc_dao → Proyecto correspondiente a la Capa DAO
- nssicc_reportes → Proyecto que contendrá los reportes del Sistema
- nssicc_ear → Proyecto que genera EAR del proyecto
- nssicc_web_mobile → Proyecto correspondiente a la Capa Web (Mobile)

Debido a lo mencionado en el eclipse se verá de la siguiente forma:



	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	

2.6. Carpeta target del NSSiCC

Al momento de compilar las clases del NSSiCC genera la carpeta target con los archivos class generados. Dicha carpeta NO DEBERA ser sincronizada al Repositorio cuando realicen el desarrollo respectivo. Para ello deben indicar al Subversión que dicha carpeta deberá ser ignorada al momento de querer subir los cambios al Repositorio de código fuente (Ver Manual de Instalación capítulo 6.2)

Asimismo en dicha carpeta target se vuelve a grabar las paginas xhtml desarrolladas por uno.

Verificar siempre que al momento de realizar cambios en la pantalla xhtml NO se realice sobre la versión que se encuentre en la versión target, ya que luego esos cambios se perderán

2.7. Uso de Anotaciones

El NSSiCC trabaja a nivel de anotaciones tanto para el mapeo de los Manage Beans como las clases Java Spring de Capa Service y de capa DAO.

En el SSiCC las clases java Spring de Capa Service como de capa DAO se mapean en los archivos applicationContext*.xml

Tener en consideración al momento de realizar la migración de las Clases Spring que debe colocarse el id del beans en formato de anotación (Existen algunas excepciones como el mapeo de interfaces)

2.8. Variables de Sesión

El SSiCC hace bastante uso de la Session para colocar variables en ella, generalmente las listas y combobox mostradas en las pantallas.

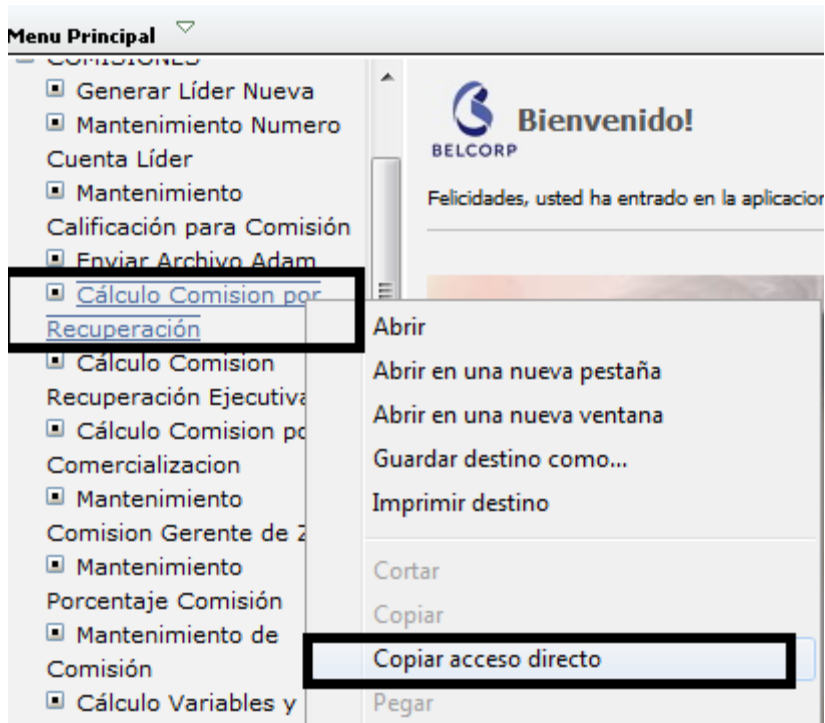
El NSSiCC no trabaja con variables de Session, para ello al momento de migrar dichas variables del SSiCC, deberán ser creadas en el NSSiCC como nuevos atributos del Manage Beans.

De esta forma evitamos el uso de la memoria de Sesión, ya que las variables quedan encapsuladas como atributos de la Clase ManageBeans.

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	

2.9. Ubicación Clase Action

Las opciones del Menú del SSiCC traen consigo una URL. Para ello al momento de migrar una opción del menú hacer click botón derecho en la opción para obtener el enlace respectivo



Por ejemplo si quisiéramos migrar la opción "Cálculo Comisión por Recuperación" obtendremos la siguiente URL:

<http://pelnx3196:7777/ssicpeesdev/viewProcesoCOMComisionRecuperacion.do?codigoSistema=COM&codigoProcesoBatch=01&codigoMenu=20010708>

Observamos que la siguiente cadena:

viewProcesoCOMComisionRecuperacion.do

Por convención del SSiCC la palabra view corresponde al método View que va invocar la cual se encontrará en la Clase Java Action **ProcesoCOMComisionRecuperacion**

Por tanto la base para comenzar a migrar será revisar la clase **ProcesoCOMComisionRecuperacion.java**

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	

2.10. Métodos Bases

Como hemos visto en el punto anterior todas las opciones del Sistema SSiCC (salvo algunas clases Action que se crearon al inicio del desarrollo del SSiCC) al hacer click en la opción del menú invocan al Método View de la clase Java Struts Action asociada, la cual realizara la carga de data inicial que se mostrará al ejecutar dicha pantalla.

Asi de la misma forma existe métodos que se utilizar para funcionalidades especificas las cual comentamos las principales:

2.10.1. *setViewAttributes:*

Método que se encontrará en todas las clases Java Action la cual es aquí donde se pondrá el código que se necesitará para mostrar la data a mostrar al momento de ejecutar la pantalla desde el menú del Sistema

2.10.2. *setFindAttributes:*

Método que devuelve una Lista con la información mostrada en la Grilla de Resultados. Dicho método es utilizado al hacer Búsqueda de Mantenimientos y Consultas

Asimismo también es invocado en ciertas Interfaces que previamente deben buscar y generar una grilla de resultados

2.11. Uso de AJAX

El SSiCC debido que usa Struts 1.x, no dispone de AJAX Nativo, para ello usa DWR para poder hacer efecto de Ajax, la cual básicamente se usa en los combos anidados.

Para el uso de AJAX en DWR, utiliza javascript para invocar a una clase definida en Spring (AjaxService.java y AjaxServiceImpl.java) para poder invocar a los métodos establecidos. Asimismo dichos métodos deben estar definidos en el archivo de configuración dwr.xml

El NSSiCC usa tecnología nativa de JSF2, debido a ello, **NO usa Javascript para el uso de AJAX**. Al momento de migrar se deberá revisar el código javascript a migrar y si hacen uso del DWR, **deberá ser migrado usando el atributo update del componente y creando el método ActionListener en la clase ManageBeans respectiva (Clase Action Migrada)**

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	

2.12. Páginas WEB y templates

El SSiCC maneja archivos jsp para las pantallas WEB mientras el NSSiCC maneja archivos xhtml para las pantallas WEB. Cuando se realice la migración de las pantallas JSP del SSiCC se deberá crear en el NSSiCC el mismo archivo pero con extensión xhtml y se deberá crear en base al template asociado.

El SSiCC no maneja template padre para las paginas jsp. Eso quiere decir que al momento de crear una página jsp similar a otra ya existente, actualmente se copia toda la página a la nueva y en la nueva se hacen los cambios.

En el NSSiCC si maneja template padre para cada tipo de página WEB a desarrollar (Ya sea un Mantenimiento, Interfaz, Consulta, Reporte, Proceso Batch, etc)

Debido a lo mencionado al momento de migrar la página jsp del SSiCC a su análogo xhtml en NSSiCC **NO se deberá copiar todo el código**, sino primero crear el archivo en base al template respectivo y **SOLO pasar las partes que son propias a la página** (Ej: Los filtros de búsqueda, ingreso de campos de mantenimiento, etc)

2.13. Componentes Personalizados

El NSSiCC trata de no utilizar directamente los componentes que disponemos de JSF2 y de Primefaces para los componentes comunes. Para ello hace uso de los componentes personalizado que se han creado

Debido a lo mencionado se deberá utilizar siempre estos componentes para estos casos (puede haber excepciones las cuales deberán ser consultadas previamente para su implementación):

2.13.1. Etiqueta asociada a la caja de texto

Se deberá utilizar el componente `ssicc:label`

Ejemplo:

```
<ssicc:label key="#{i18n['interfazSearchForm.descripcionSistema']}" />
```

En caso la caja de texto asociada su valor sea obligatorio, se deberá colocar en el componente `ssicc:label` el atributo `required="true"`

Ejemplo:

```
<ssicc:label id="labelPais" key="#{i18n['interfazForm.codigoPais']}"
required="true"/>
```

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	



Si se desea hacer referencia al objeto desde javascript o desde otro componente deberá invocarlo usando `<nombreFormulario>:<id>:label`
 Ejemplo: Si el id es labelPais
 Desde javascripts: `→ document.getElementById("labelPais:label")`
 Desde otro componente `→ :principalForm:labelPais:label`

2.13.2. Caja de Texto

Se deberá utilizar el componente `ssicc:inputText`

Ejemplo:

```
<ssicc:inputText id="descripcion" value="#{mAlias.formMantenimiento.descripcion}"
size="100" maxlength="300" />
```

En caso la caja de texto tenga valor que sea obligatorio colocar el atributo `required="true"`

```
<ssicc:inputText id="descripcion" value="#{mAlias.formMantenimiento.descripcion}"
size="100" maxlength="300" required="true" />
```

Por defecto este componente todo lo que se escriba lo pasa a mayúsculas, en caso se desee poder colocar valores tanto mayúsculas como minúsculas colocar el atributo `esMayuscula="false"`

Ejemplo:

```
<ssicc:inputText id="descripcion" value="#{mAlias.formMantenimiento.descripcion}"
size="100" maxlength="300" required="true" esMayuscula="false" />
```

Asimismo si solo se desea que ingrese valores Numéricos colocar el atributo `esNumerico="true"`

Ejemplo:

```
<ssicc:inputText id="anho" value="#{mAlias.formBusqueda.anho}" size="7"
maxlength="4" required="true" esNumerico="true" />
```

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	



Si se desea hacer referencia al objeto desde javascript o desde otro componente deberá invocarlo usando `<nombreFormulario>:<id>:input`
 Ejemplo: Si el id es anho
 Desde javascripts: → `document.getElementById("anho:input")`
 Desde otro componente → `:principalForm:anho:input`

2.13.3. Campo Fechas

Para las cajas de texto que se ingrese valores de tipo Fecha se deberá colocar el componente `ssicc:calendar`
 Ejemplo:

```
<ssicc:calendar id="fecIni" value="#{mAlias.formReporte.fechaInicio}"/>
```

Consideración: El atributo del Manage Bean asociado a dicho componente deberá ser de tipo `java.util.Date` y no `String`. Si es `String` generará ERROR



Si se desea hacer referencia al objeto desde javascript o desde otro componente deberá invocarlo usando `<nombreFormulario>:<id>:calendar`
 Ejemplo: Si el id es fecIni
 Desde javascripts: → `document.getElementById("fecIni:calendar")`
 Desde otro componente → `:principalForm:fecIni:calendar`

2.13.4. Campo Periodo ó Campaña

El SSiCC maneja el concepto de Periodo (también denominado Campaña), la cual consiste en la división del año por un grupo de días (aprox 20 días). En el año se han establecido 18 campañas, y podemos ver que existen muchas pantallas del SSiCC, donde se pide un valor de Campaña (Ejemplo: 201405)

Para cuando la pantalla pida este tipo de valores se deberá colocar el componente `ssicc:campaignCalendar2`

Ejemplo:

```
<ssicc:campaignCalendar2 id="codigoPeriodo" required="true" manage="#{mAlias}" value="#{mAlias.formReporte.codigoPeriodo}"/>
```

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	



Si se desea hacer referencia al objeto desde javascript o desde otro componente deberá invocarlo usando <nombreFormulario>:<id>:campanya
Ejemplo: Si el id es codigoPeriodo
Desde javascripts: → document. getElementById ("codigoPeriodo: campanya")
Desde otro componente → :principalForm:codigoPeriodo:campanya

2.13.5. Combox o Lista (Selección Única)

Las lista o comboBox cuya selección sea un valor único debe utilizar el componente

ssicc:selectMenu

Ejemplo:

En este caso trae una lista obtenida en el ManageBeans:

```
<ssicc:selectMenu id="codigoComision" value="#{mAlias.formReporte.codigoComision}"
required="true">

    <f:selectItems value="#{mAplicacionBean.listaComision}"
        var="lista"
        itemLabel="#{lista.descripcion}" itemValue="#{lista.codigo}"/>
</ssicc:selectMenu>
```

En este caso trae una lista obtenida en el ManageBeans y el 1er Registro es un valor en blanco:

```
<ssicc:selectMenu id="codigoComision" value="#{mAlias.formReporte.codigoComision}"
required="true">
    <f:selectItem itemLabel="" itemValue=""/>
    <f:selectItems value="#{mAplicacionBean.listaComision}"
        var="lista"
        itemLabel="#{lista.descripcion}" itemValue="#{lista.codigo}"/>
</ssicc:selectMenu>
```

Si se desea invocar a un método y refrescar usando AJAX:

Ejemplo:

```
<ssicc:selectMenu id="codigoCanal"
value="#{mAlias.formReporte.codigoCanal}" required="true"
accionListener="#{reporteINCPuntObtenidosPuntFaltantesAction.LoadRegionesCanal}"
update=":principalForm:codigoRegion:selectMany :principalForm:codigoZona:selectMany
:principalForm:codigoConcurso:select">
    <f:selectItem itemLabel="" itemValue=""/>
    <f:selectItems value="#{mAlias.siccCanalList}"
        var="lista" itemLabel="#{lista.descripcion}" itemValue="#{lista.codigo}"/>
</ssicc:selectMenu>
```

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	



Si se desea hacer referencia al objeto desde javascript o desde otro componente deberá invocarlo usando <nombreFormulario>:<id>:select
Ejemplo: Si el id es codigoCanal
Desde javascripts: → document. getElementById ("codigoCanal:select")
Desde otro componente → :principalForm:codigoCanal:select

2.13.6. Combox o Lista (Selección Múltiple)

Aquellas listas o combobox donde se pueda seleccionar más de un valor (Selección Múltiple) deberá utilizarse el componente **ssicc:selectManyMenu**

Ejemplo:

```
<ssicc:selectManyMenu id="nivelConcurso"
    value="#{mAlias.formReporte.nivelConcursoList}" required="true">

    <f:selectItems value="#{mAlias.siccNivelesConcursoList}"
        var="Lista"
        itemLabel="#{lista.Label}" itemValue="#{lista.value}"/>
</ssicc:selectManyMenu>
```

Si se desea invocar a un método y refrescar usando AJAX:

Ejemplo:

```
ssicc:selectManyMenu id="codigoRegion"
    value="#{mAlias.formReporte.codigoRegion}" required="true"
    accionListener="#{reporteINCPuntObtenidosPuntFaltantesAction.showZonasxRegion}"
    update=":principalForm:codigoZona:selectMany">
    <f:selectItems value="#{mAlias.siccRegionList}"
        var="ListaRegion" itemLabel="#{listaRegion.Label}"
        itemValue="#{listaRegion.value}"/>
</ssicc:selectManyMenu>
```



Si se desea hacer referencia al objeto desde javascript o desde otro componente deberá invocarlo usando <nombreFormulario>:<id>:select
Ejemplo: Si el id es codigoRegion
Desde javascripts: → document. getElementById ("codigoRegion:select")
Desde otro componente → :principalForm:codigoRegion:select

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	

2.13.7. Campo TextArea

Se deberá utilizar el componente `ssicc:inputTextArea`

Ejemplo:

```
<ssicc:inputTextArea id="observación" rows="3" cols="100"
value="#{mAlias.formInterfaz.observaciones}" />
```



Si se desea hacer referencia al objeto desde javascript o desde otro componente deberá invocarlo usando `<nombreFormulario>:<id>:inputtextarea`

Ejemplo: Si el id es observacion

Desde javascripts: → document. getElementById ("observacion:inputtextarea")

Desde otro componente → :principalForm:observacion:inputtextarea

2.13.8. CheckBox

Para los checkbox a mostrar en el Sistema se deberá utilizar el componente

`ssicc:selectBooleanCheckbox.xhtml`

Ejemplo:

```
<ssicc:selectBooleanCheckbox value="#{mAlias.habilitaUbigeo}"
id="habilitaUbigeo" onChange="reiniciaTerritorioConsultora()"/>
```

Si se desea invocar a un método y refrescar usando AJAX:

Ejemplo:

```
<ssicc:selectBooleanCheckbox value="#{mAlias.habilitaUbigeo}"
id="habilitaUbigeo" onChange="reiniciaTerritorioConsultora()"
accionListener="#{reporteINCGanadorasConcurso2Action.changeStatus1}"
update=":principalForm:codigoRegion:select,:principalForm:codigoZona:select,:principalForm:cb1:selectBooleanCheckbox,:principalForm:cb2:selectBooleanCheckbox" />
```



Si se desea hacer referencia al objeto desde javascript o desde otro componente deberá invocarlo usando `<nombreFormulario>:<id>:selectBooleanCheckbox`

Ejemplo: Si el id es habilitaUbigeo

Desde javascripts: → document. getElementById ("habilitaUbigeo:selectBooleanCheckbox_input")

Desde otro componente → :principalForm:habilitaUbigeo:selectBooleanCheckbox_input

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	

2.13.9. RadioButton

Se deberá utilizar el componente `ssicc:selectOneRadio`

Ejemplo:

```
<ssicc:selectOneRadio id="flagHabilitado"
  value="#{mAlias.formMantenimiento.flagHabilitado}" required="true">
  <f:selectItem itemLabel="#{i18n['interfazForm.habilitado']}" itemValue="#{SI}" />
  <f:selectItem itemLabel="#{i18n['interfazForm.deshabilitado']}" itemValue="#{NO}" />
</ssicc:selectOneRadio>
```

Si se desea invocar a un método y refrescar usando AJAX:

Ejemplo:

```
<ssicc:selectOneRadio id="tipoGeneracion"
  value="#{mAlias.formMantenimiento.tipoGeneracion}"
  ActionListener="#{interfazSearchAction.generarCodigoInterfazPorTipoGeneracion}"
  update=":principalForm:panelCodigoInterfaz :principalForm:panelUnitario1
:principalForm:panelPaquete :principalForm:panelFTP
:principalForm:panelParametrosInterfaz
:principalForm:panelGeneraArchivoServidor"
  required="true">
  <f:selectItem itemLabel="#{i18n['interfazForm.unitario']}" itemValue="#{SI}" />
  <f:selectItem itemLabel="#{i18n['interfazForm.paquete']}" itemValue="#{NO}" />
</ssicc:selectOneRadio>
```



Si se desea hacer referencia al objeto desde javascript o desde otro componente deberá invocarlo usando `<nombreFormulario>:<id>:selectOneRadio`

Ejemplo: Si el id es tipoGeneracion

Desde javascripts: → `document.getElementById("tipoGeneracion:selectOneRadio")`

Desde otro componente → `:principalForm:tipoGeneracion:selectOneRadio`

2.13.10. Grilla Datatable (Modo Consulta o de Selección simple)

Para el uso de grillas (datatable) que no se haga uso de seleccionar registro o en los casos que se pueda seleccionar solo un solo registro se deberá utilizar el componente

`ssicc:dataTable`

Ejemplo:

DataTable sin selección de registros:

```
<ssicc:dataTable id="dataTableBusqueda"
```

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	

```

mostrarTituloPanel="false"
toggleable = "true"
draggable = "false"
draggableColumns = "true"
exportable="true"
titulo="#{msg['dataTable.resultadoBusqueda']}"
value="#{mAlias.datatableBusqueda}"
fileExport="dataBusqueda"
rendered="#{not empty mAlias.listaBusqueda}"
resizableColumns="true"
mostrarMenuMantenimiento = "true"
ejecutarDBCClick = "true"
manage = "#{mAlias}"
>

```

DataTable de selección única:

```

<ssicc:dataTable id="dataTableBusqueda"
mostrarTituloPanel="false"
toggleable = "true"
draggable = "false"
draggableColumns = "true"
exportable="true"
titulo="#{msg['dataTable.resultadoBusqueda']}"
value="#{mAlias.datatableBusqueda}"
fileExport="dataBusqueda"
rendered="#{not empty mAlias.listaBusqueda}"
resizableColumns="true"
seleccionable = "true"
mostrarMenuMantenimiento = "true"
ejecutarDBCClick = "true"
manage = "#{mAlias}"
selection = "#{mAlias.beanRegistroSeleccionado}"
>

```

Para las grillas de selección múltiple en ese caso usar el componente del primefaces directamente (p:dataTable)



Si se desea hacer referencia al objeto desde javascript o desde otro componente deberá invocarlo usando <nombreFormulario>:<id>:

Ejemplo: Si el id es tipoGeneracion

Desde javascripts: → document. getElementById ("tipoGeneracion:selectOneRadio")

Desde otro componente → :principalForm:tipoGeneracion:selectOneRadio

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	

2.13.11. *CommandButton*

El SSiCC al ejecutar la botonería del Sistema ejecuta submit respectivo, mientras los botones del NSSiCC NO USA submit y utiliza AJAX para invocar al método en la clase JAVA ManageBeans

La mayoría de los botones estándar ya se encuentran implementados en los templates hechos en el NSSiCC. Sin embargo pueda darse el caso de la necesidad de migrar botones que son exclusivos a la pantalla WEB para ello se deberá utilizar el componente `ssicc:linkAjax`

Ejemplo:

```
<ssicc:linkAjax value="#{i18n['button.structfile']}"
  id="botonEjecutar"
  title="#{i18n['button.structfile']}"
  icon="/resources/images/estructura.gif"
  accion="#{interfazSearchAction.estructuraArchivo}"
  update=":principalForm:idSeleccionoRegistro :principalForm:idTieneEstructura"
  oncomplete="verificarTieneEstructura();"
/>
```



Si se desea hacer referencia al objeto desde javascript o desde otro componente deberá invocarlo usando `<nombreFormulario>:<id>:linkAjax`

Ejemplo: Si el id es botonEjecutar

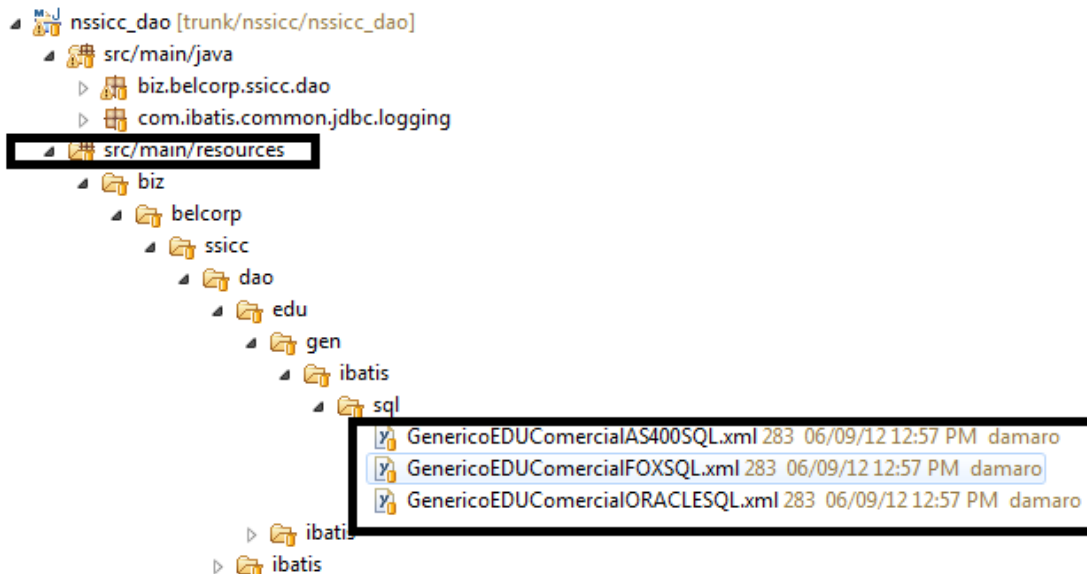
Desde javascripts: → `document.getElementById("botonEjecutar:linkAjax")`

Desde otro componente → `:principalForm:botonEjecutar:linkAjax`

2.14. Ubicación archivos XML de ibatis

En el NSSiCC los archivos JAVA y los archivos RESOURCES (xml) se encuentran en carpetas separadas. Debido a ello al migrar los archivos xml de ibatis del SSiCC al NSSiCC se deberá colocar en la carpeta resources del subproyecto nssicc_dao

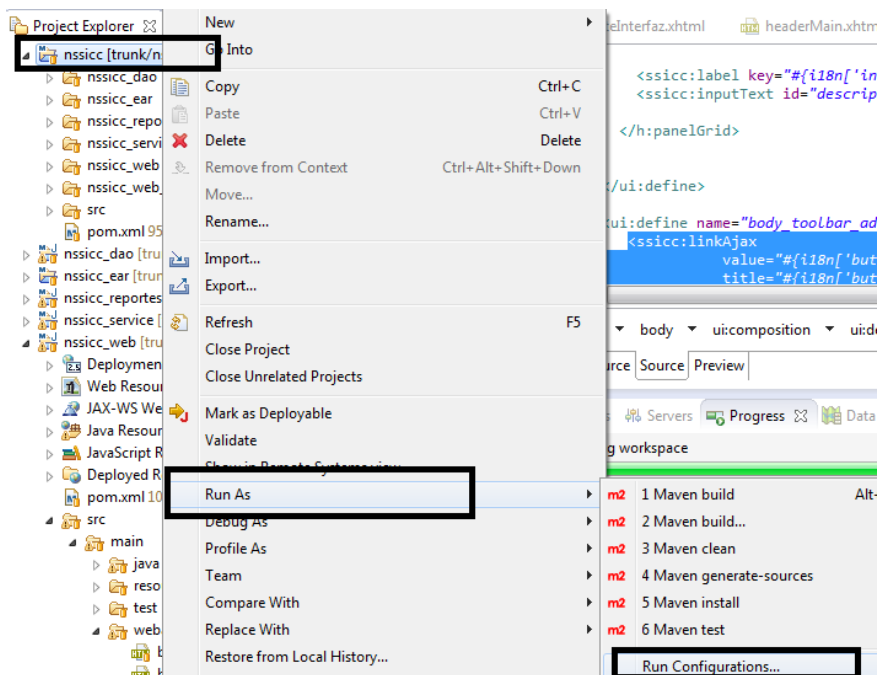
	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	



2.15. Reportes

En el SSiCC solo se coloca el jrxml ya que el archivo.jasper asociado se genera al deployar la aplicación mediante el ANT

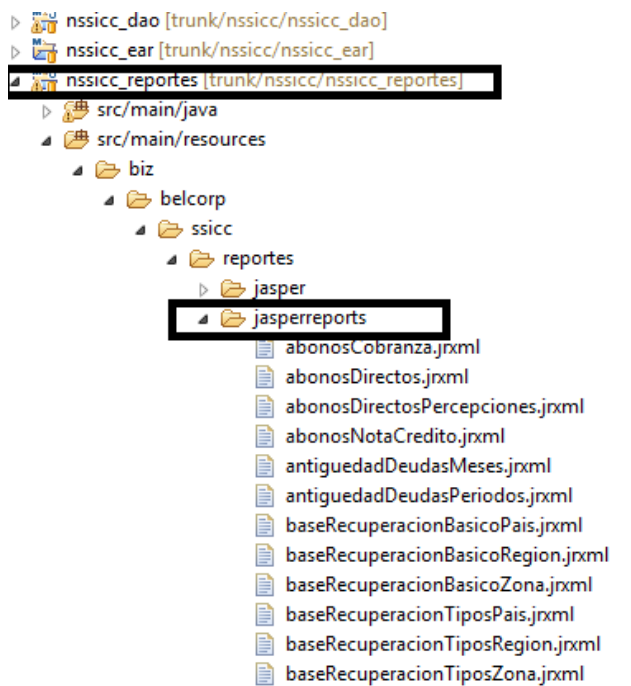
En el NSSiCC los.jasper son generados cuando se ejecuta maven a través de la sentencia mvn -install



	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	

Tanto el jrxml como el.jasper deben ser sincronizados al Repositorio SVN en el proyecto NSSiCC

Los jrxml deberán ser colocados en la carpeta *jasperreports* del proyecto *nssicc_reportes*



2.16. Asignando Página xhtml al Menú

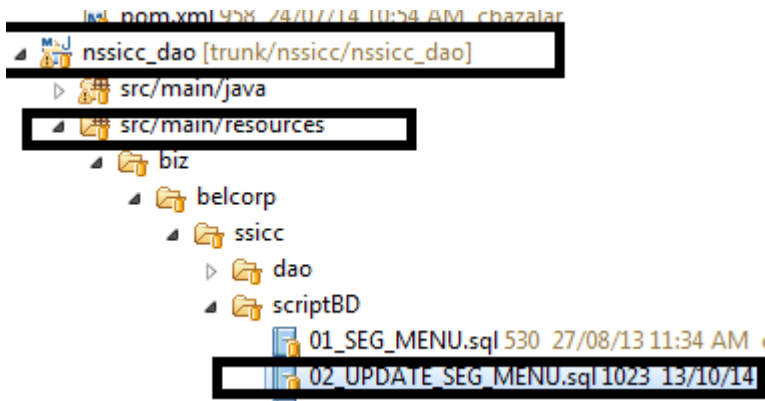
Una vez migrado una opción del menú del SSiCC al NSSiCC se deberá mapear la página xhtml en la tabla SEG_MENU

Para ello deben hacer update a dicha tabla y al campo PAG_XML por el código del Menú
Ejemplo:

```
UPDATE SEG_MENU X
SET X.PAG_XML = '/pages/scsicc/reporteSICLimitesVentaConfiguradosForm.xhtml'
WHERE X.COD_MENU = '40021815';
```

Aparte de ejecutarse dicho update en la BD, se deberá grabar dicho update en el archivo *02_UPDATE_SEG_MENU.sql* que se encuentra en el subproyecto *nssicc_dao*

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	



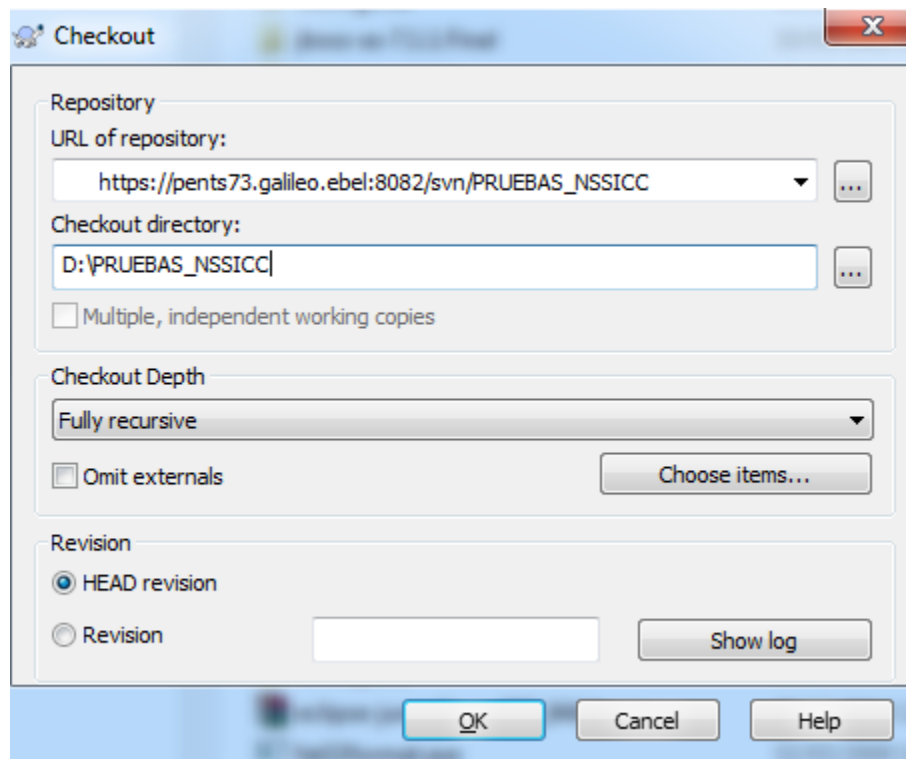
2.17. Pruebas Internas

Una vez terminado el desarrollo se deberá hacer prueba interna del mismo.

Hacer click sobre la opción SVN Checkout..., la cual le mostrara la pantalla siguiente, ingresar los datos mencionados en dicha pantalla

Repository: https://pents73.galileo.ebel:8082/svn/PRUEBAS_NSSICC

Export Directory: D:\PRUEBAS_NSSICC



	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	

Una vez terminado hacer click sobre el botón OK.

Para ello se deberá crear un archivo WORD con el siguiente formato:

<Codigo Menu> - <Nombre de Opcion del Menu>.doc

Ejemplo:

40020807 - Calculo Comisión por Comercializacion.doc

En dicho documento se deberá colocar:

- Imagen de Pantalla Original del SSiCC
- Imagen de Pantalla Migrada del NSSiCC
- Imágenes de Funcionamiento de la pantalla migrada en el NSSiCC

Una vez realizado dicho documento deberá ser sincronizado en el Repositorio de Pruebas

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	

3. Migración de Reportes

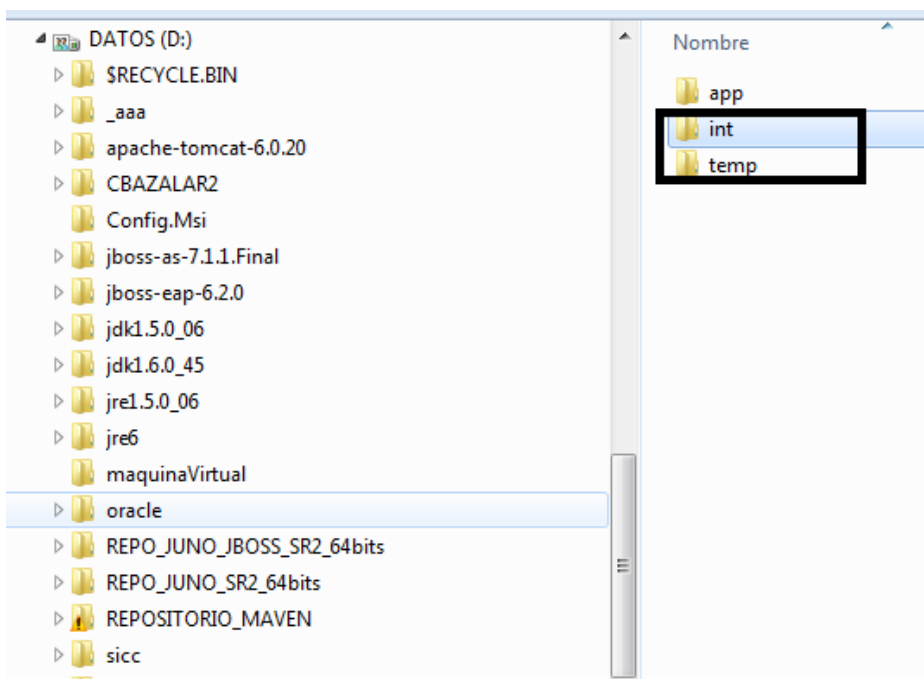
3.1. Verificar existencia de Carpeta

Inicialmente se deberá verificar que en el Disco Duro de la maquina en donde se haya colocado el JBOSS se deberá crear en la Raiz de dicho Disco duro las siguientes carpetas:

`ssicc / temp`

y

`ssicc / int`



3.2. Obteniendo Clase Action

Luego en el SSiCC se debe obtener la clase JAVA Action asociada a la opción del Menú (Ver punto 2.9 del documento)

Tomaremos por ejemplo la opción del Menú:

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	

CONSULTAS Y REPORTES / SISTEMA CONSULTAS Y REPORTES (Sc-SiCC) / COMISIONES / Reporte Pago Ejecutivas

Para ello obtendremos la url del enlace. En este caso sería:

<http://pelnx3196:7777/ssicpeesdev/viewReporteCOMComisionPagoEjecutivas.do?codigoMenu=40020815>

Por convención del SSiCC debemos buscar en base a lo resaltado la siguiente clase en el SSiCC:

ReporteCOMComisionPagoEjecutivasAction.java

Copiar tal cual al NSSiCC en el subproyecto *nssicc_web* en la misma carpeta en la que se encuentra en el SSiCC

En este caso copiaremos el archivo en el NSSiCC a la carpeta
biz/belcorp/ssicc/scsicc/web/action

De hecho veremos que la clase se mostrará con errores en el NSSiCC

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	

3.3. Modificando Clase Action en el NSSiCC

```

ReporteCOMComisionPagoEjecutivasAction.java
1 // TODO Migrar al framework de reportes
2 package biz.belcorp.ssicc.scsicc.web.action;
3
4 import java.util.Map;
5
6
7
8
9
10
11
12
13
14
15
16
17
18 /**
19  * @author <a>gfernandezo</a>
20  *
21  * @struts.action name="reporteCOMComisionPagoEjecutivasForm"
22  *               path="/viewReporteCOMComisionPagoEjecutivas"
23  *               scope="session" validate="false" parameter="method"
24  *               input="mainMenu"
25  *
26  * @struts.action name="reporteCOMComisionPagoEjecutivasForm"
27  *               path="/executeReporteCOMComisionPagoEjecutivas"
28  *               scope="session" validate="false" parameter="method"
29  *               input="view"
30  *
31  *
32  * @struts.action-forward name="view"
33  *                       path="/WEB-INF/pages/scsicc/reporteCOMComisionPagoEjecutivasForm.jsp"
34  *
35  * @struts.action-forward name="viewVirtualPDF"
36  *                       path="/WEB-INF/pages/scsicc/reporteGenerico.jsp"
37  *
38  */
39 public class ReporteCOMComisionPagoEjecutivasAction extends
40     BaseSubReporteAbstractAction {
41
42     private String formatoReporte;
43
44     /**

```

Lo primero es colocar bien el package ya que como hemos visto en el punto 2.3 del documento las rutas varían en el NSSiCC

En este caso cambiar lo mostrado en verde

package biz.belcorp.ssicc.scsicc.web.action;

Por:

package biz.belcorp.ssicc.web.scsicc.action;

Asimismo eliminar lo señalado en amarillo y colocar lo siguiente

@ManagedBean

@SessionScoped

public class ReporteCOMComisionPagoEjecutivasAction

Las clases de Reporte del SSiCC heredan de BaseSubReporteAbstractAction tal como se muestra en lo señalado en rojo

Cambiar por BaseReporteAbstractAction

Debe quedar como se muestra a continuación:

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	

```
// TODO Migrar al framework de reportes
package biz.belcorp.ssicc.web.scsicc.action;

import java.util.List;

/**
 * @author <a href="mailto:jaltamirano@sigcomt.com">Juan José Altamirano Huamán</a>
 */
@ManagedBean
@SessionScoped
public class ReporteCOMComisionPagoEjecutivasAction extends BaseReporteAbstractAction {

    private static final long serialVersionUID = -1091798983329193025L;
}
```

Cambiar la declaración de los métodos siguientes:

`protected void setViewAttributes(HttpServletRequest request, ActionForm form)`
por
`protected void setViewAttributes() throws Exception`

`protected String getReporteFileName()`
por
`protected String devuelveNombreReporte() throws Exception`

`protected String getSubReporteFileName()`
por
`protected String devuelveNombreSubReporte() throws Exception`

`protected void prepareParameterMap(Map params, ActionForm form, HttpServletRequest request)`
por
`protected Map prepareParameterMap(Map params) throws Exception`

3.4. Creando Clase Form en el NSSiCC

Buscar en el SSiCC la clase FORM asociada al ACTION. Por convención deberá tener la siguiente estructura de nombre: <NOMBRE DEL REPORTE>Form.java

En este caso buscamos el archivo java ReporteCOMComisionPagoEjecutivasForm.java

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	

```

1 package biz.belcorp.ssiCC.scsicc.web.form;
2
3 import java.io.Serializable;
4
5 /**
6  * @author <a href="mailto:jaltamirano@sigcomt.com">Juan José Altamirano Huamán</a>
7  *
8  * @struts.form name = "reporteCOMComisionPagoEjecutivasForm"
9  */
10
11 public class ReporteCOMComisionPagoEjecutivasForm extends BaseReporteForm
12     implements Serializable {
13
14 }

```

Copiarlo en la misma carpeta en el NSSiCC en el subproyecto **nssicc_web**

Lo primero es colocar bien el package ya que como hemos visto en el punto 2.3 del documento las rutas varían en el NSSiCC

En este caso cambiar lo mostrado en **verde**

package biz.belcorp.ssiCC.scsicc.web.form;
 Por:
package biz.belcorp.ssiCC.web.scsicc.form;

Asimismo eliminar lo señalado en **amarillo**
 Debe quedar como se muestra a continuación:

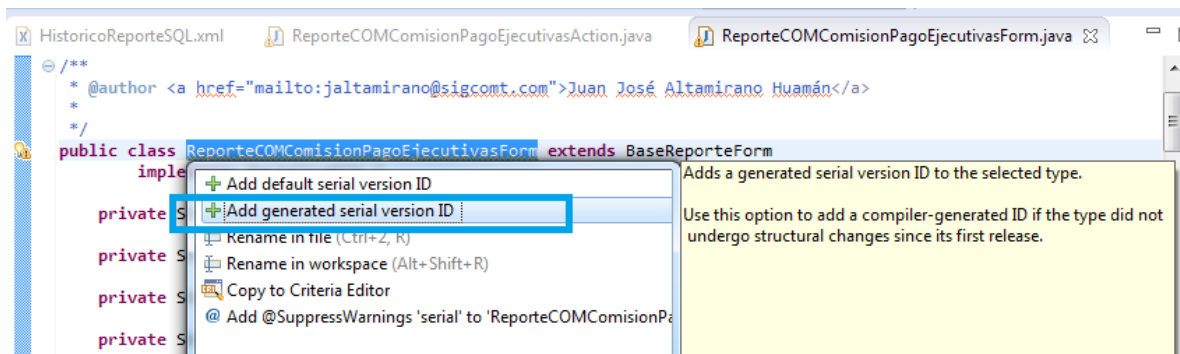
```

1 package biz.belcorp.ssiCC.reportes.web.scsicc.form;
2
3 import java.io.Serializable;
4
5 /**
6  * @author <a href="mailto:jaltamirano@sigcomt.com">Juan José Altamirano Huamán</a>
7  *
8  */
9
10 public class ReporteCOMComisionPagoEjecutivasForm extends BaseReporteForm
11     implements Serializable {
12
13 }

```

Existe un warning señalado en **azul**. Es necesario generarle el ID serial

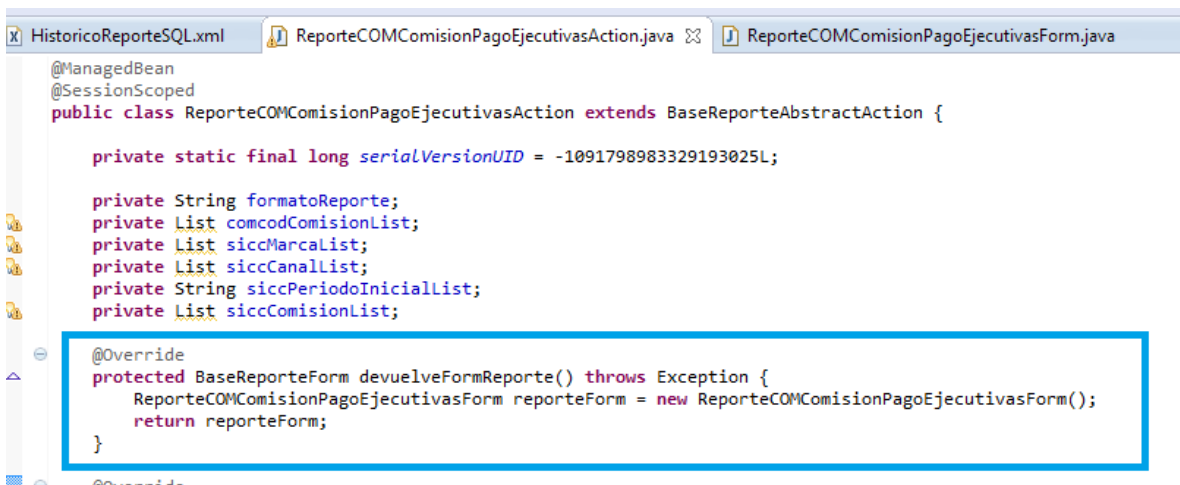
	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	



Debe quedar finalmente



Luengo en la Clase Action del NSSiCC debemos crear el método y asignarle la Clase FORM migrada tal como se muestra a continuación



	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	

3.5. Revisando métodos de la Clase Action en el NSSiCC

Revisar en los métodos que toda llamada a creación de variables de sesión deberá ser cambiada creando atributos en el manageBeans
En este caso apreciamos el método setViewAttributes

SSiCC

```

1 public class ReporteCOMComisionPagoEjecutivasAction extends
2     BaseSubReporteAbstractAction {
3
4     private String formatoReporte;
5     /*
6     * (non-Javadoc)
7     * @see biz.belcorp.ssi.web.action.BaseAbstractAction#setViewAttributes(javax.servlet.http.HttpServletRequest, org.apache.struts.action.ActionForm)
8     */
9     protected void setViewAttributes(HttpServletRequest request, ActionForm form) {
10         InterfazSiCCService service = (InterfazSiCCService) getBean("sisicc.interfazSiCCService");
11
12         HttpSession session = request.getSession(true);
13         Pais pais = getPais(session);
14         Usuario usuario = getUsuario(session);
15
16         InterfazSiCCService svc = (InterfazSiCCService) getBean("sisicc.interfazSiCCService");
17         session.setAttribute(Constants.COM_CODIGO_COMISION_LIST, svc.getListCodComision(pais.getCodigo());
18
19         session.setAttribute(Constants.SICC_MARCA_LIST, service.getMarcas());
20         session.setAttribute(Constants.SICC_CANAL_LIST, service
21             .getCanalesByCodigoISO(usuario.getIdioma().getCodigoISO()));
22         session.setAttribute(Constants.SICC_COMISION_LIST, service.getComision());
23         session.setAttribute(Constants.SICC_PERIODO_INICIAL_LIST, service
24             .getPeriodoDefaultByPaisCanal(pais.getCodigo(),
25                 Constants.CODIGO_CANAL_DEFAULT));
26         session.setAttribute(Constants.SICC_COMISION_LIST, svc.getComision());
27         log.debug("Todo Ok: Redireccionando");
28     }
29 }

```

Luego de modificarlo en el NSSiC queda como:

```

HistoricoReporteSQL.xml  ReporteCOMComisionPagoEjecutivasAction.java  ReporteCOMComisionPagoEjecutivasForm.java
@Override
protected void setViewAttributes() throws Exception {
    if (this.log.isDebugEnabled()) {
        this.log.debug("Entering 'ReporteCOMComisionPagoEjecutivasAction.setViewAttributes' method");
    }
    this.mostrarReporteXLS = true;
    // Seteo de valores por default de nuevos registros
    ReporteCOMComisionPagoEjecutivasForm form = (ReporteCOMComisionPagoEjecutivasForm) this.formReporte;
    InterfazSiCCService service = (InterfazSiCCService) getBean("sisicc.interfazSiCCService");
    Pais pais = this.mPantallaPrincipalBean.getCurrentCountry();
    Usuario usuario = this.mPantallaPrincipalBean.getCurrentUser();
    String codigoPeriodo = service.getPeriodoDefaultByPaisCanal(pais.getCodigo(), Constants.CODIGO_CANAL_DEFA

    form.setCodigoPais(pais.getCodigo());
    form.setCodigoPeriodo(codigoPeriodo);

    comcodComisionList = service.getListCodComision(pais.getCodigo());
    siccMarcalist = service.getMarcas();
    siccCanallist = service.getCanalesByCodigoISO(usuario.getIdioma().getCodigoISO());
    siccComisionList = service.getComision();
    siccPeriodoInciallist = service.getPeriodoDefaultByPaisCanal(pais.getCodigo(), Constants.CODIGO_CANAL_DE

    log.debug("Todo Ok: Redireccionando");
}

```

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	

3.6. Revisando invocaciones a Clase Service y DAO

Se deberá verificar que los métodos en el Service existan y que no haya ningún error al respecto.

```

@Override
protected void setViewAttributes() throws Exception {
    if (this.log.isDebugEnabled()) {
        this.log.debug("Entering 'ReporteCOMComisionPagoEjecutivasAction.setViewAttributes' method");
    }
    this.mostrarReporteXLS = true;
    // Seteo de valores por default de nuevos registros
    ReporteCOMComisionPagoEjecutivasForm form = (ReporteCOMComisionPagoEjecutivasForm) this.formReporte;
    InterfazSiCCService service = (InterfazSiCCService) getBean("sisicc.interfazSiCCService");
    Pais pais = this.mPantallaPrincipalBean.getCurrentCountry();
    Usuario usuario = this.mPantallaPrincipalBean.getCurrentUser();
    String codigoPeriodo = service.getPeriodoDefaultByPaisCanal(pais.getCodigo(), Constants.CODIGO_CANAL_DEFA

    form.setCodigoPais(pais.getCodigo());
    form.setCodigoPeriodo(codigoPeriodo);

    comcodComisionList = service.getListCodComision(pais.getCodigo());
    siccMarcaList = service.getMarcas();
    siccCanalList = service.getCanalesByCodigoISO(usuario.getIdioma().getCodigoISO());
    siccComisionList = service.getComision();
    siccPeriodoInicialList = service.getPeriodoDefaultByPaisCanal(pais.getCodigo(), Constants.CODIGO_CANAL_DE

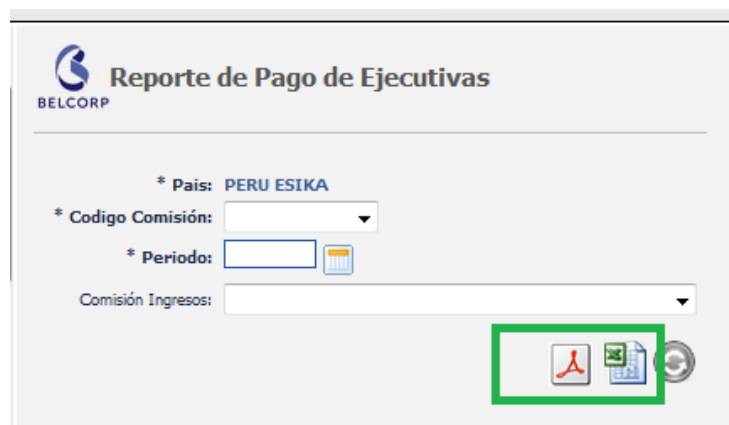
    log.debug("Todo Ok: Redireccionando");
}

```

Si hubiera error se deberá migrar las Clases Service, Clases DAO y archivo XML de ibatis. Asimismo se deberá verificar en el archivo *sql-map-config.xml* que este declarado el XML de ibatis

3.7. Visualizando Botones del Reporte

Verificar en el SSiCC que tipos de Reportes existen en la pantalla
En este caso observamos que el Reporte es emitido en PDF y en Excel



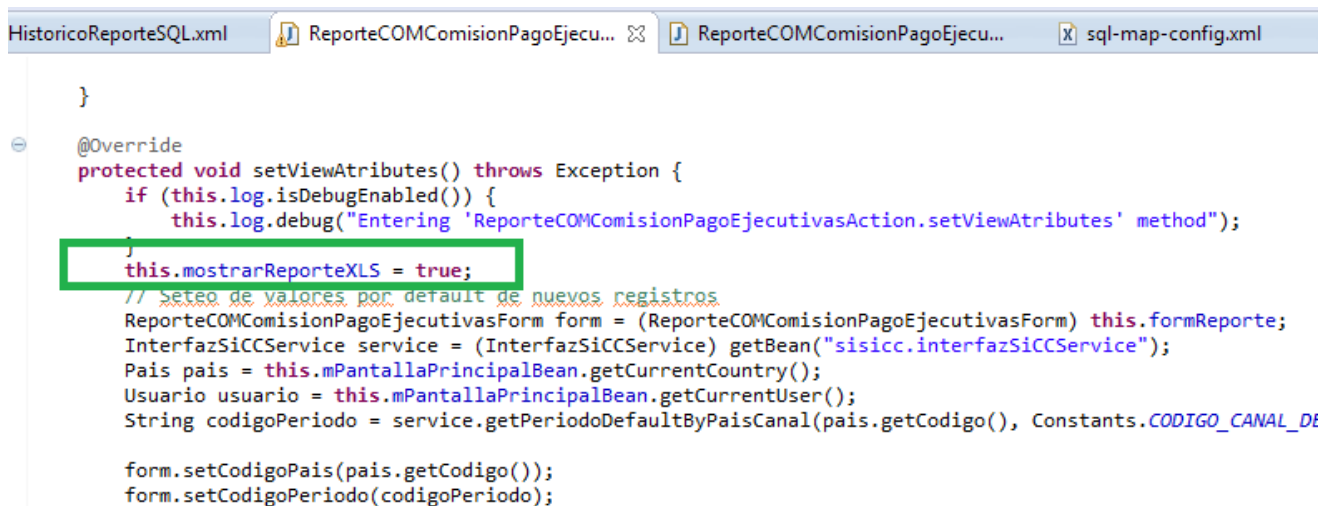
	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	

El NSSiCC tiene una serie de atributos los cuales debemos activar / desactivar para que se muestre los botones respectivos
En este caso los atributos son:

```
protected boolean mostrarReportePDF = true;
protected boolean mostrarReporteXLS = false;
protected boolean mostrarReporteXLSX = false;
protected boolean mostrarReporteCSV = false;

protected boolean mostrarReporteOCSV = false;
protected boolean mostrarReporteOJXLSX = false;
protected boolean mostrarReporteO0XLSX = false;
```

Por defecto muestra el botón PDF y los demás los oculta
En este se debe mostrar los botones PDF y Excel para ello en el método *setViewAttributtes* debemos setear dichos atributos



```
HistoricoReporteSQL.xml | ReporteCOMComisionPagoEjecu... | ReporteCOMComisionPagoEjecu... | sql-map-config.xml

}

@Override
protected void setViewAttributtes() throws Exception {
    if (this.log.isDebugEnabled()) {
        this.log.debug("Entering 'ReporteCOMComisionPagoEjecutivasAction.setViewAttributtes' method");
    }
    this.mostrarReporteXLS = true;
    // Seteo de valores por default de nuevos registros
    ReporteCOMComisionPagoEjecutivasForm form = (ReporteCOMComisionPagoEjecutivasForm) this.formReporte;
    InterfazSiCCService service = (InterfazSiCCService) getBean("sisicc.interfazSiCCService");
    Pais pais = this.mPantallaPrincipalBean.getCurrentCountry();
    Usuario usuario = this.mPantallaPrincipalBean.getCurrentUser();
    String codigoPeriodo = service.getPeriodoDefaultByPaisCanal(pais.getCodigo(), Constants.CODIGO_CANAL_DE);

    form.setCodigoPais(pais.getCodigo());
    form.setCodigoPeriodo(codigoPeriodo);
}
```

3.8. Verificando método devuelveNombreReporte

Si en dicho método `protected String devuelveNombreReporte() throws Exception` existe referencia en el código a VMLS cambiarlo por XLS
Si indica referencia a VPDF cambiarlo a PDF

3.9. Verificando método devuelveNombreSubReporte

Si en dicho método `protected String devuelveNombreSubReporte() throws Exception`

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	

existe referencia en el código a Vxls cambiarlo por XLS
Si indica referencia a VPdf cambiarlo a PDF

3.10. Eliminar try catch

En los métodos del `setViewAttributes` y `prepareParameterMap` del SSiCC existe código dentro de try catch.

Los try catch deben ser eliminados en el código migrado del NSSiCC

3.11. Verificar método Reset en la Clase Form

Verificar la existencia del método `Reset` en la Clase Form, si existe dicho método su contenido debe pasarse al método `setViewAttributes` de la Clase Action y se debe borrar el método `Reset` de la Clase Form

3.12. Creando página xhtml

Se deberá crear la pagina xhtml en base a la plantilla `templateReporte.xhtml`, con el nombre del jsp asociado al Action, para ello revisar lo siguiente en el SSiCC

```

ReporteCOMComisionPagoEjecutivasAction.java  ReporteCOMComisionPagoEjecutivasForm.java
1 // TODO Migrar al framework de reportes
2 package biz.belcorp.ssicc.scsicc.web.action;
3
4 import java.util.Map;
5
6
7
8
9
10
11
12
13
14
15
16
17
18 /**
19  * @author <a>efernandez</a>
20  *
21  * @struts.action name="reporteCOMComisionPagoEjecutivasForm"
22  *               path="/viewReporteCOMComisionPagoEjecutivas"
23  *               scope="session" validate="false" parameter="method"
24  *               input="mainMenu"
25  *
26  * @struts.action name="reporteCOMComisionPagoEjecutivasForm"
27  *               path="/executeReporteCOMComisionPagoEjecutivas"
28  *               scope="session" validate="false" parameter="method"
29  *               input="view"
30  *
31  *
32  * @struts.action-forward name="view"
33  *                       path="/WEB-INF/pages/scsicc/reporteCOMComisionPagoEjecutivasForm.jsp"
34  *
35  * @struts.action-forward name="viewVirtualPDF"
36  *                       path="/WEB-INF/pages/scsicc/reporteGenerico.jsp"
37  *
38  */
39 public class ReporteCOMComisionPagoEjecutivasAction extends
40     BaseSubReporteAbstractAction {
41     ..

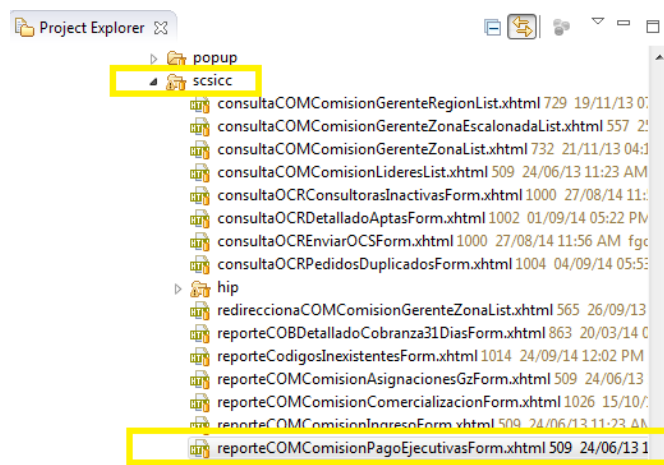
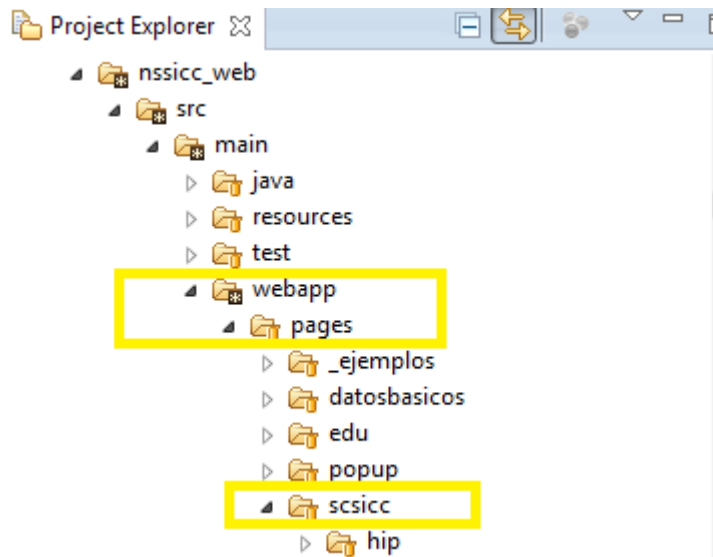
```

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	

Con el ese nombre señalado en amarillo se deberá crear la pagina en el NSSiCC

[reporteCOMComisionPagoEjecutivasForm.xhtml](#)

En la ruta indicada: [/pages/scsicc/](#) dentro del subproyecto nssicc_web



Crearlos usando el siguiente código base:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:p="http://primefaces.org/ui"
      xmlns:ssicc="http://java.sun.com/jsf/composite/componentes/ssicc"
      xmlns:bel="http://www.belcorp.biz/ssicc/components">
```

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	

```

<body>

<ui:composition template="/templates/templateReporte.xhtml">
  <ui:param name="mAlias" value="{reporteCOMComisionPagoEjecutivasAction}"/>

  <ui:define name="titulo_Pantalla">
    <h:outputText value="{i18n['reporteCOMComisionPagoEjecutivasForm.heading']}"
styleClass="clsEncabezadoPantalla" />
  </ui:define>

  <ui:define name="body_filtro_reporte">

    <h:panelGrid columns="2" >

      <ssicc:label key="{i18n['reporteCOMComisionPagoEjecutivasForm.codigoPais']}"
required="true"/>
      <ssicc:label value="{mPantallaPrincipalBean.currentCountry.descripcion}" />

    </h:panelGrid>

  </ui:define>
</ui:composition>

</body>
</html>

```

Modificar lo señalado en amarillo

3.13. Campos Fechas

Al momento de migrar los atributos de tipo Fecha del SSiCC al NSSiCC debemos tener en cuenta lo siguiente:

En el SSiCC los atributos mencionados son de tipo String

En el NSSiCC se deberá crear adicionalmente al atributo migrado de tipo String, uno de tipo java.util.Date, la cual deberá ser usado dentro del componente ssicc:calendar

Luego se deberá pasar el valor del atributo de tipo Date al atributo de tipo String, ya que dicho atributo será utilizado en la generación del Reporte. Para realizar lo mencionado utilizar el método prepareParameterMap

3.14. Campos Periodos que invoquen Ajax

Para estos casos si un Periodo invoca Ajax para luego refrescar otra parte de la pantalla (Ejemplo: Fechas de Inicio y Fecha de Fin), se deberá utilizar el componente ssicc:campaignCalendarListener

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	

3.15. Agregando los Filtros en página xhtml

Luego de creado la pagina xhtml en base al código base del punto anterior se deberá colocar los filtros propios de la pantalla.

Para ello se debe revisar el código del SSiCC y en base a ello crear lo análogo en el NSSiCC en la sección *body_filtro_reporte*

Para este caso seria:

SSiCC

```

38
39
40 <th>
41     <ssicc:label key="reporteCOMComisionPagoEjecutivasForm.codigoComision"/>
42 </th>
43 <td colspan="2">
44     <html:select property="codigoComision" styleId="codigoComision" >
45         <html:option value=""></html:option>
46         <html-el:options collection="{COM_CODIGO_COMISION_LIST}" property="{codigo}" labelProperty="de
47     </html:select>
48     <html:errors property="codigoComision"/>
49 </td>
50 </tr>
51
52 <tr>
53 <th>
54     <ssicc:label key="reporteCOMComisionPagoEjecutivasForm.codigoPeriodo"/>
55 </th>
56 <td>
57     <html:text
58         property="codigoPeriodo"
59         onkeypress="return changeKey(this, event, cancelLetters);"
60         size="6"
61         maxLength="6"
62         styleId="codigoPeriodo"/>
63     <ssicc:calendar property="codigoPeriodo"
64         type="campaign"
65         imgKey="icon.cal.img"
66         altKey="icon.cal"
67         styleClass="calIcon"/>
68     <html:errors property="codigoPeriodo"/>
69 </td>
70 </tr>

```

No está todo el código continua (Ver código fuente en el Proyecto SSiCC)

En el NSSiCC quedaría como:

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	

```

istoricoReporteSQL.xml  ReporteCOMComisionPagoEjecutivasAction.j...  ReporteCOMComisionPagoEjecutivasForm.ja...  sql-map-config.xml  reporteCOMComisionPagoEjecutivasForm.x...
<ui:param name="mAlias" value="#{reporteCOMComisionPagoEjecutivasAction}"/>
<ui:define name="titulo_Pantalla">
  <h:outputText value="#{i18n['reporteCOMComisionPagoEjecutivasForm.heading']}" styleClass="clsEncabezadoPantalla" />
</ui:define>
<ui:define name="body_filtro_reporte">
  <h:panelGrid columns="2">
    <ssicc:label key="#{i18n['reporteCOMComisionPagoEjecutivasForm.codigoPais']}" required="true"/>
    <ssicc:label value="#{mPantallaPrincipalBean.currentCountry.descripcion}" />
    <ssicc:label key="#{i18n['reporteCOMComisionPagoEjecutivasForm.codigoComision']}" required="true" />
    <ssicc:selectMenu id="codigoComisionIngreso" value="#{mAlias.formReporte.codigoComision}" required="true">
      <f:selectItem itemLabel="" itemValue="" />
      <f:selectItems value="#{mAlias.comcodComisionList}" var="Lista" itemLabel="#{Lista.descripcion}" itemValue="#{Lista.codigo}"/>
    </ssicc:selectMenu>
    <ssicc:label key="#{i18n['reporteCOMComisionPagoEjecutivasForm.codigoComisionRecu1']}" required="true" />
    <ssicc:selectMenu id="codigoComisionRecuperacion" value="#{mAlias.formReporte.comisionIngresos}" required="true">
      <f:selectItem itemLabel="" itemValue="" />
      <f:selectItems value="#{mAlias.siccodComisionList}" var="Lista" itemLabel="#{Lista.descripcion}" itemValue="#{Lista.codigo}"/>
    </ssicc:selectMenu>
    <ssicc:label key="#{i18n['reporteCOMComisionPagoEjecutivasForm.codigoPeriodo']}" />
    <ssicc:campaignCalendar2 id="codigoPeriodo" required="true" manage="#{mAlias}" value="#{mAlias.formReporte.codigoPeriodo}"/>
  </h:panelGrid>
</ui:define>

```

3.16. Creando Clase Service (Revisar método beforeExecuteReporte)

Existe un CASO ESPECIAL en la que se deberá crear una Clase SERVICE en el NSSiCC (**nssicc_reportes**) (Clase que no existe en el SSiCC)

Este caso se da cuando exista los siguientes métodos en la clase Action del SSiCC:

- *beforeExecuteReporte*
- *afterExecuteReporte*
- *beforeGrabarReporte*
- *afterGrabarReporte*
- *generarReporteOracle*
- *generarNombreArchivoReporteOracle*
- *obtieneKeyTituloReporteOracle*

Ejemplo: Si revisan la clase Action ReporteCOMComisionRecuperacionAction.java en el SSiCC verán que existe dicho método, pero en el NSSiCC no existe tal método.

El método *beforeExecuteReporte* deberá ser creado en una clase Service con el mismo nombre del Action pero sin el sufijo *Action* y se debe colocar el sufijo *ServiceImpl*

Ejemplo: ReporteCOMComisionRecuperacionServiceImpl.java

Asimismo en la clase Action se deberá crear el siguiente método devuelveBeanReporteService para hacer referencia a la clase Service creada

Ejemplo:

	Migración Código SSiCC a NSSiCC	
	Proyecto SSiCC	

```
protected String devuelveBeanReporteService(){
    return "reportes.reporteCOMComisionRecuperacionService";
}
```

Tomar como ejemplo las clases mencionadas

En este caso la CLASE FORM deberá ser movido y colocado al proyecto nssicc_reportes

3.17. Revisar código javascript en la página jsp del SSiCC

Siempre se debe revisar todo el código Javascript para detectar si hay invocaciones a AJAX. Si las hubiera deberán implementar los métodos ActionListener en la Clase Action (ManageBeans) para la implementación del AJAX

Asimismo puede ser que existan validaciones previas realizadas en Javascript por ejemplo:

Validar que la Fecha Inicial sea menor igual que la Fecha Final ingresada
Validar que el Periodo Inicial sea menor igual que el Periodo Final ingresado

Para estos casos se deberá sobrescribir el método en el NSSiCC *setValidarReporte* la cual devolverá el mensaje de Error en caso no cumpla alguna validación respectiva

Ejemplo:

```
public String setValidarReporte() {
    if (<variable fechaFin>.compareTo(<variable fechaIni>) < 0)
        return <valor mensaje error>;
    return null;
}
```

3.18. Verificar existencia archivos jrxml

Se han migrado los archivos jrxml y jasper respectivos, sin embargo puede darse el caso que no estuvieran. Si eso pasará se deberá migrarlos del SSiCC