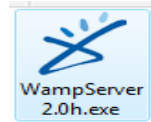


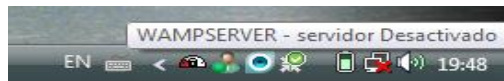
ANEXO 2

CONFIGURACIÓN DE HERRAMIENTAS UTILIZADAS

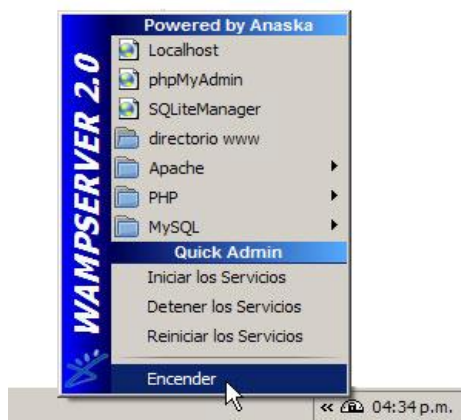
➤ WAMP SERVER



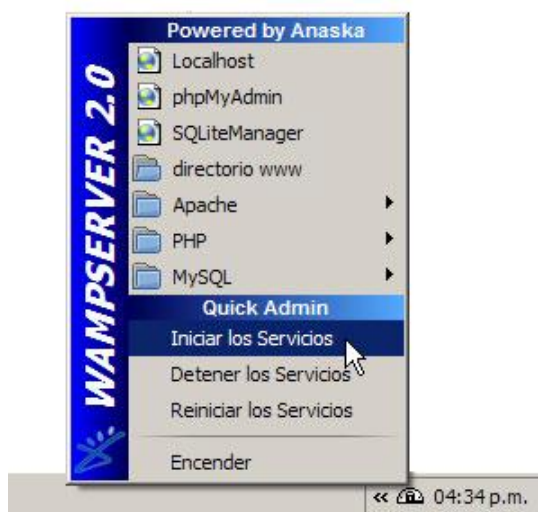
1. Instalar
2. Una vez instalado vamos a archivo de programas y ejecutamos wamp, posteriormente le vemos en la barra que se indica



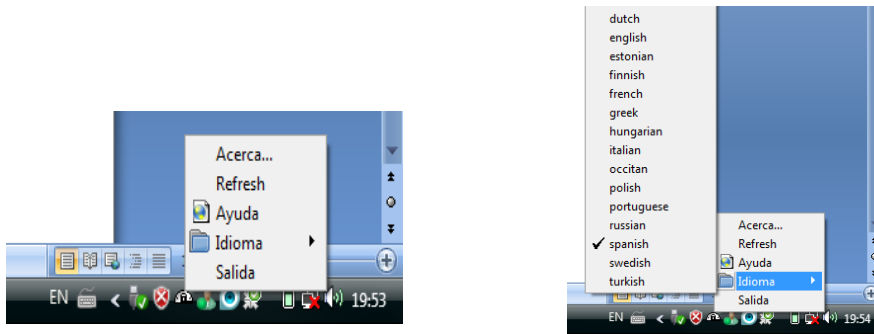
3. Para encender el servidor, para ello hacemos clic (normal) sobre el icono en nuestra barra de tareas, y le damos clic a "Encender".



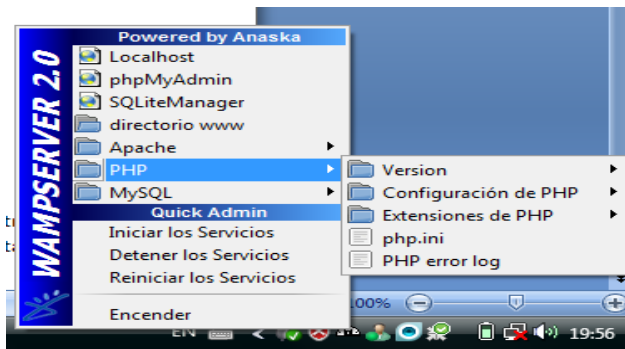
4. Para iniciar los servicios



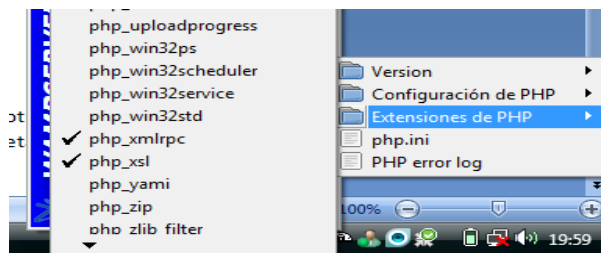
5. Luego clic derecho sobre el icono para cambiar el idioma



6. Seguidamente clic izquierdo sobre el icono

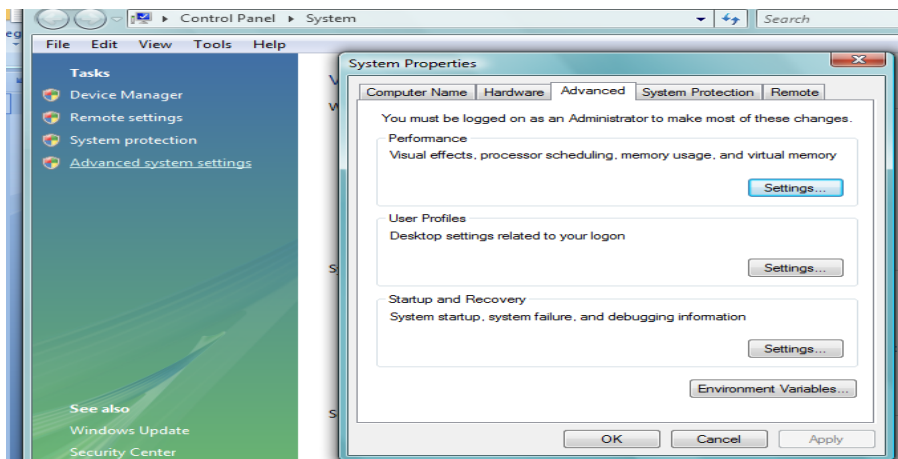


7. En extensiones de php activar xsl y xmlrpc

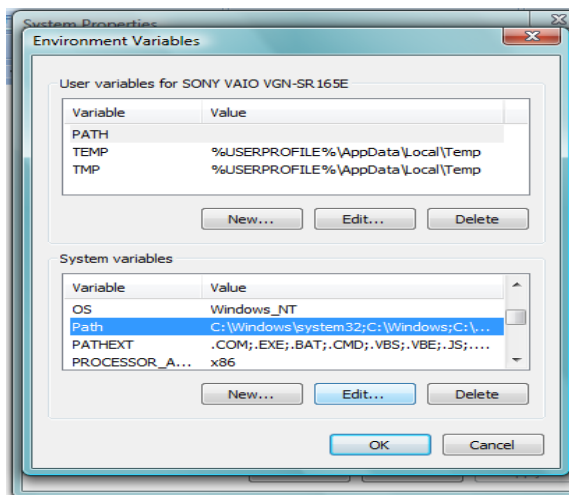


➤ CONFIGURAR EQUIPO

1. Clic derecho en MI PC o en computer escogemos propiedades y clic en donde se indica y se desplegara el cuadro de la izquierda

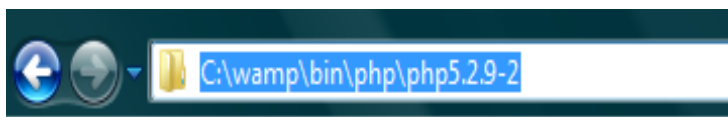


2. Clic en Environment variables y se despliega



3. En donde escogemos path y editamos: al final de lo que se presenta ahí colocamos; y copiamos lo siguiente.

bin; C:\wamp\bin\php\php5.2.9-2 que es la dirección donde se instaló wamp server.



➤ **CONFIGURAR httpd.conf**

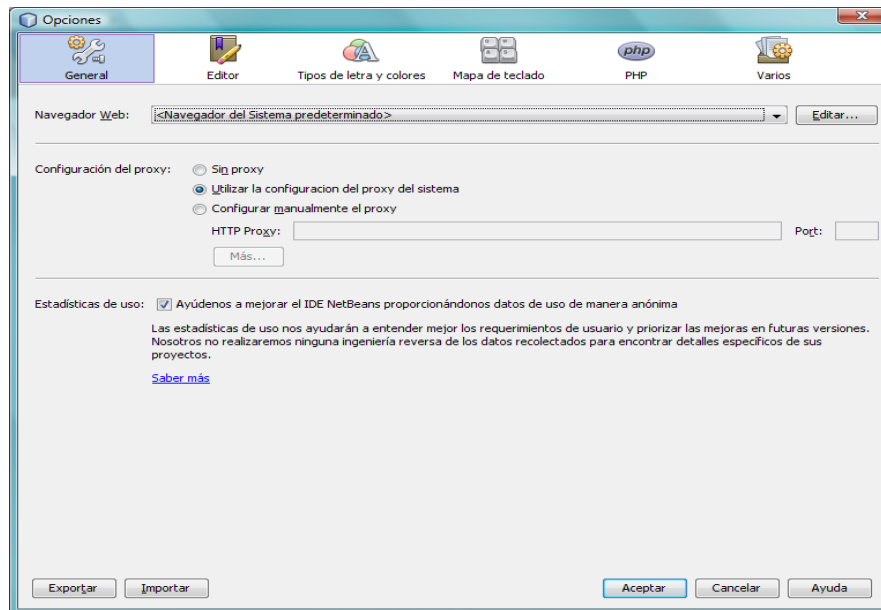
```
<VirtualHost *:9098>
  ServerName sigerhaj.localhost
  DocumentRoot "C:\wamp\www\sigerhaj\web"
  DirectoryIndex index.php
  <Directory "C:\wamp\www\sigerhaj\web">
    AllowOverride All
    Allow from All
  </Directory>
  Alias /sf "C:\wamp\www\lib\vendor\symfony\data\web\sf"
  <Directory "C:\wamp\www\lib\vendor\symfony\data\web\sf">
    AllowOverride All
    Allow from All
  </Directory>
</VirtualHost>
```


Luego de configurar el archivo guardamos y reiniciamos el servidor.

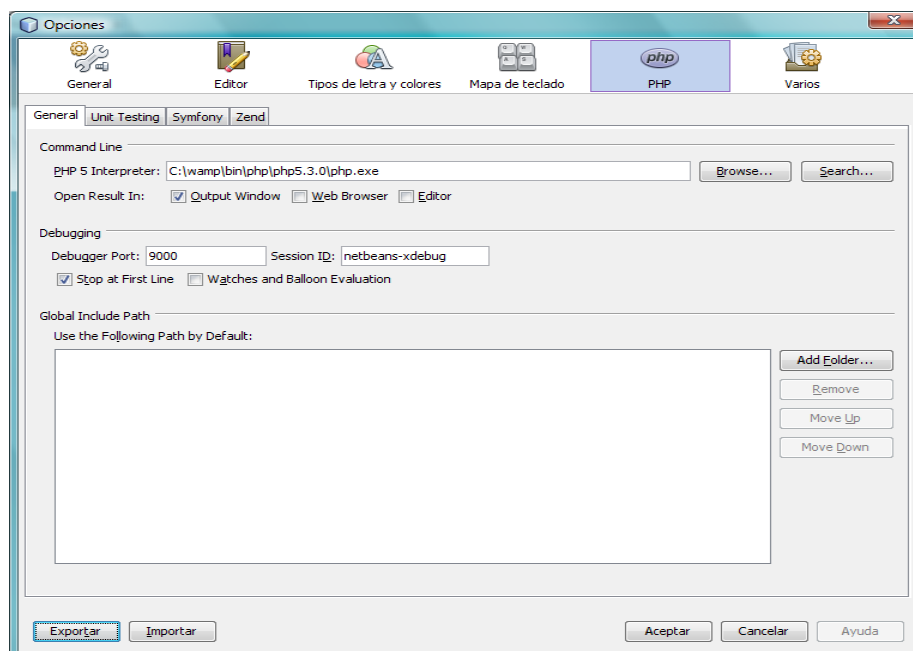
Una vez configurado el servidor, si tenemos un proyecto realizado en symfony, lo que debemos hacer, es copiar la carpeta del proyecto en la raíz del servidor, es decir en www y en el navegador colocamos: <http://localhost:9098/> para tener acceso al sistema.

➤ CONFIGURAR SYMFONY EN NETBEANS 6.9.1

- Ⓞ Debemos tener instalado un servidor web.
- Ⓞ Descargamos una versión estable de symfony desde su web oficial.
- Ⓞ Descomprimos el fichero descargado en la ruta del proyecto.
- Ⓞ Abrimos NetBeans, seleccionamos la pestaña herramientas en donde escogemos opciones en donde nos despliega la imagen siguiente.

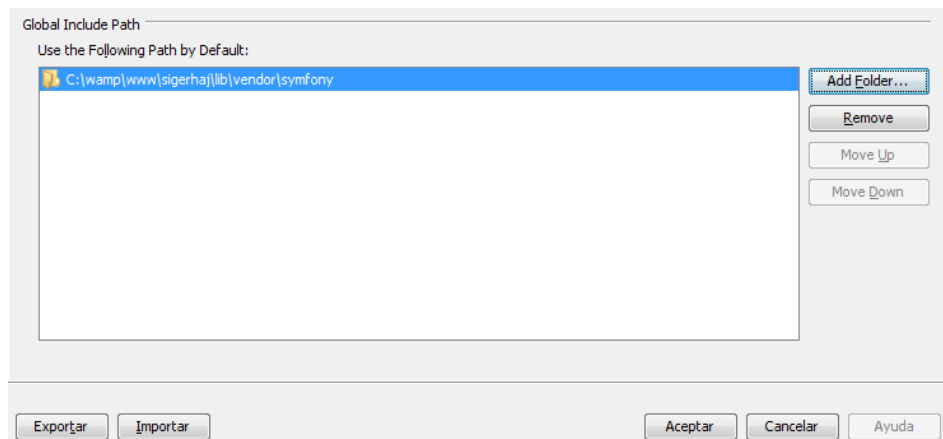


- Ⓞ En donde se debe seleccionar el icono de php  para abrir el panel de configuración de PHP indicado en la siguiente ventana.

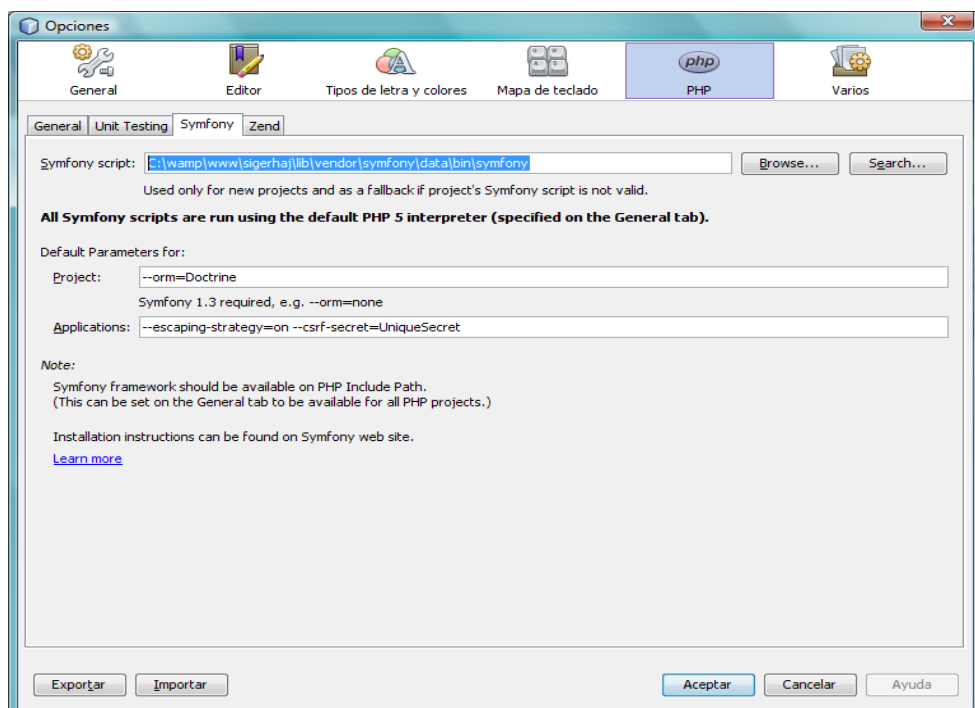


A la misma que se le suministra los valores del PHP 5 interprete indicando la ruta correspondiente y Debugger Port indicado.

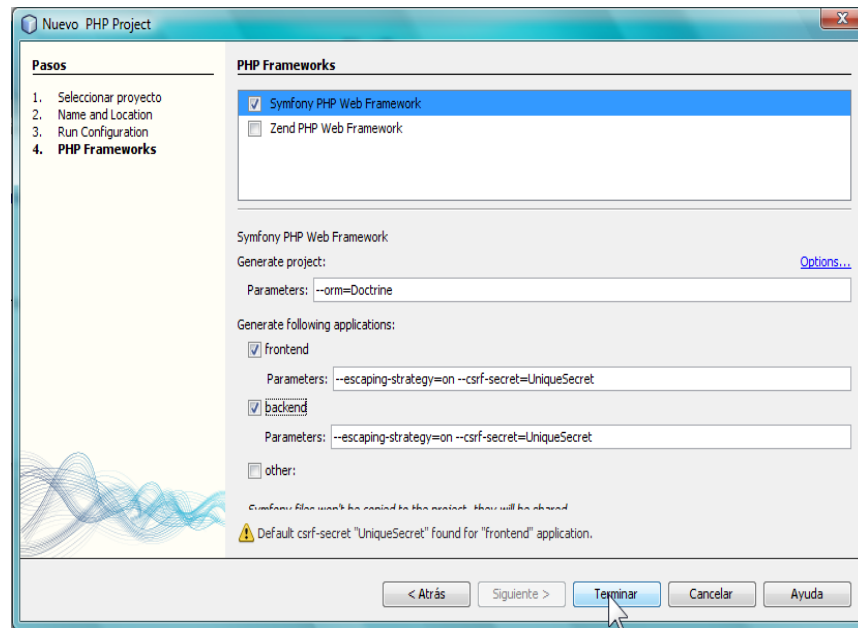
- Además, tendremos que indicar la ruta de instalación de Symfony pulsando el botón **Add Folder**, que será la carpeta que hemos descomprimido en el directorio de nuestro proyecto.



- En la pestaña **Symfony** tendremos que indicar la ruta del fichero binario del framework que se encuentra en:
C:\wamp\www\sigerhaj\lib\vendor\symfony\data\bin\symfony, y en Project: escribimos el nombre del orm con el que se desea trabajar como se indica en la ventana siguiente.



- ② Damos clic en Aceptar y antes de pulsarle en Terminar seleccionamos el checkbox de backend para que queden activos el frontend y backend



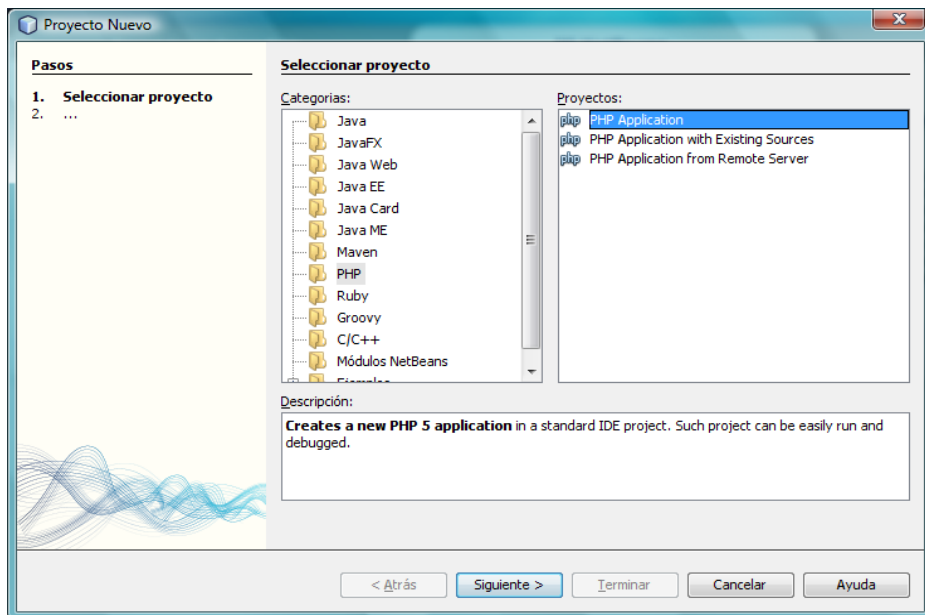
- ② Ya tendremos configurado e instalado nuestro proyecto de Symfony en NetBeans.

ANEXO 3

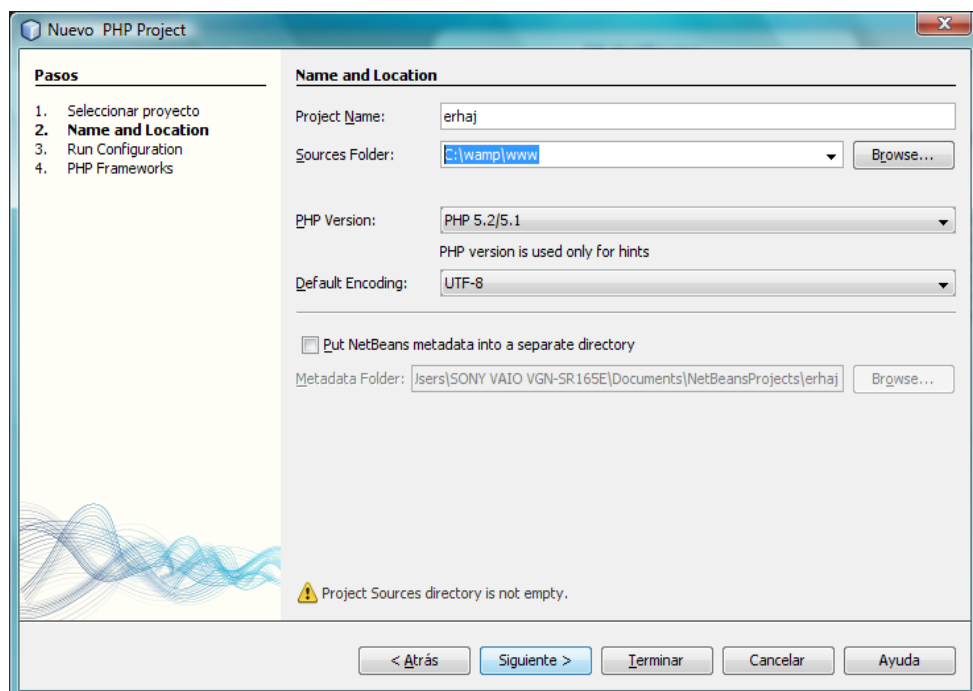
MANUAL TÉCNICO

Crear proyecto nuevo

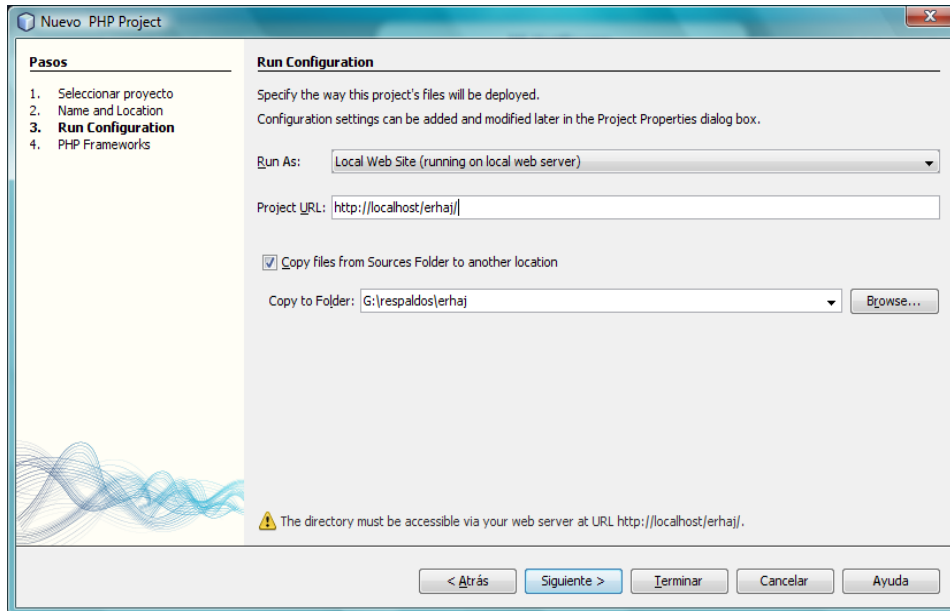
Clic en archivo y escoger Proyecto Nuevo donde se despliega la ventana abajo indicada donde se señala los pasos siguientes.



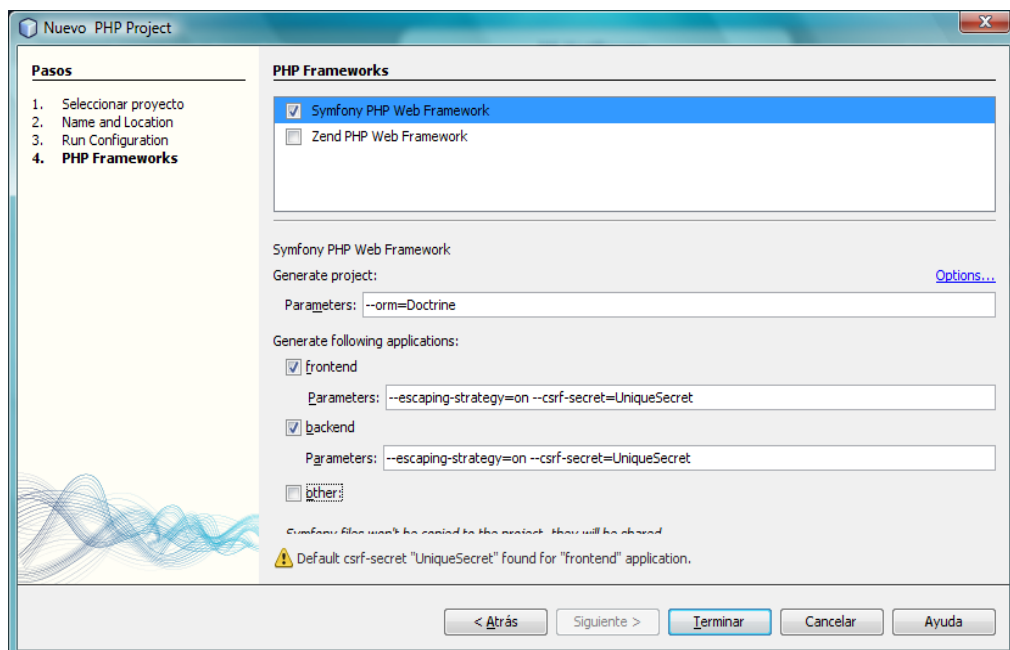
Presione sobre el botón siguiente, en donde permite desplegar la interfaz indicada a continuación:



Poner nombre al proyecto y guardar el proyecto en www directorio de Wamp y presionar en siguiente, donde aparece la siguiente ventana, en la misma que podemos escribir la ruta para guardar una copia del proyecto si se desea.



Presionamos siguiente y nos muestra la pantalla siguiente:



Aquí activamos Symfony PHP Web Framework y backend posteriormente presionamos Terminar. Netbeans creará un Proyecto de Symfony con todas las configuraciones necesarias.

Mediante un navegador se puede visualizar el proyecto colocando en la url lo siguiente : <http://localhost/erhaj/web/>



Symfony Project Created

Congratulations! You have successfully created your symfony project.

Project setup successful

This project uses the symfony libraries. If you see no image in this page, you may need to configure your web server so that it gains access to the `symfony_data/web/sf/` directory.

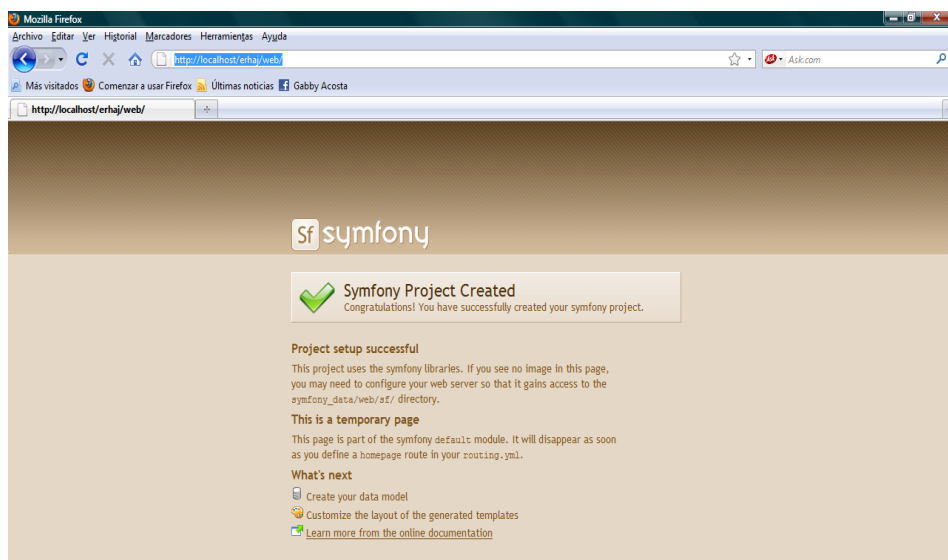
This is a temporary page

This page is part of the symfony default module. It will disappear as soon as you define a homepage route in your `routing.yml`.

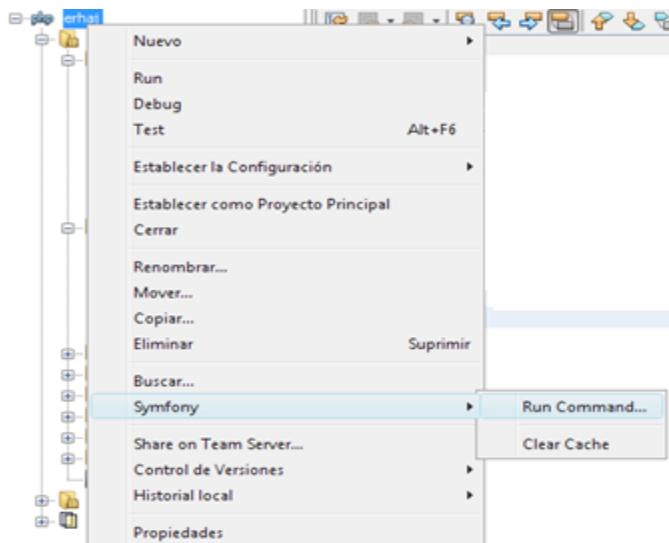
What's next

- Create your data model
- Customize the layout of the generated templates
- [Learn more from the online documentation](#)

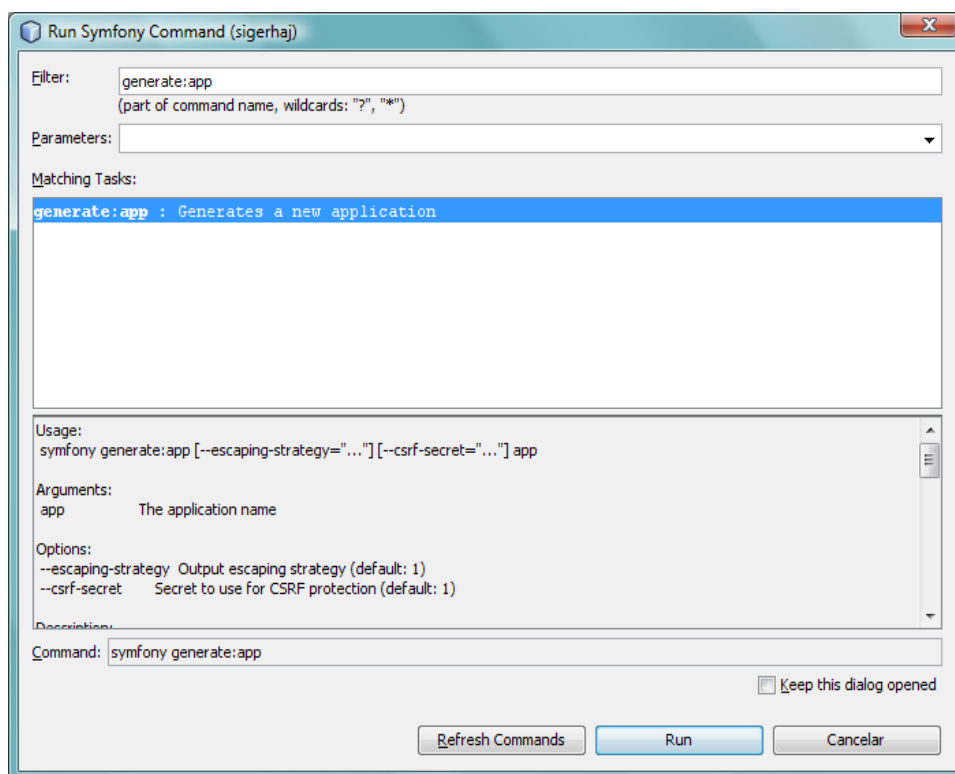
Como se puede ver que no hay ningún CSS e imágenes aquí, aunque Symfony trabaje, copiar de symfony la carpeta sf y pegar en el directorio web del proyecto y refrescar la página y se observara una agradable Bienvenida.



Creado el proyecto podemos hacer uso de los comandos, para lo cual presionamos clic derecho sobre el proyecto, escogemos Symfony, seguidamente pulsamos sobre Rub Command.



Esto abrirá una consola de mando con todo los comandos de Symfony .No tenemos que escribir el todo solamente seleccionamos el comando de la lista y le proporcionamos los parámetros requeridos que se nos indica en la parte de debajo de la ventana.



El comando ejecutado anteriormente, inicializa una aplicación. En el directorio web raíz del proyecto, se crean algunos archivos correspondientes a los controladores frontales de cada uno de los entornos por defecto: index.php, test.php (producción), test_dev.php(desarrollo).

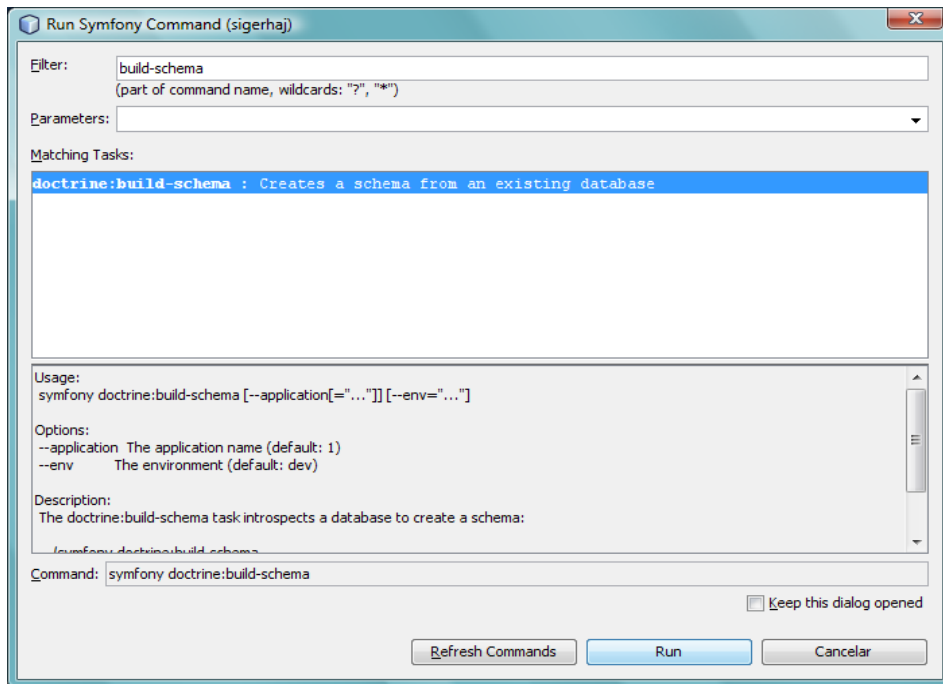
Generar módulos por default, es decir sin hacer referencia al modelo se realiza como se muestra en la imagen, para lo cual se utiliza el comando **generate:module** el cual recibe como parámetros <nombre de la aplicación > <nombre del módulo>

Para generar un nuevo módulo con las funciones CRUD, basado en una clase del modelo, es necesario primero conectar la base de datos con el proyecto, ya que se genera mediante el comando **generate-module** el cual recibe como parámetros <nombre de la aplicación > <nombre del módulo> <nombre de la clase del modelo>

Para conectar bdd es necesario configurar manualmente el archivo databases.yml, en donde se coloca el nombre de la base de datos en este caso bdd_erhaj con su respectivo username y password.

```
all:
  doctrine:
    class: sfDoctrineDatabase
    param:
      dsn:  mysql:host=localhost;dbname= bdd_erhaj
      username: erhaj
      password: admin
```

Seguidamente se ejecuta el comando



Quién genera el archivo schema.yml para la representación de una base de datos existente.

Generada la base de datos es necesario representar las clases PHP del modelo, mediante el comando **build-model**, según el modelo de datos descrito en el archivo schema.yml, las clases base del modelo se crean en el directorio `sigerhaja/lib/model/doctrine/base/` ejemplo: `BaseRhPersonal.class.php`

También se crean las clases de acceso a los datos en el directorio `sigerhaja/lib/model/doctrine/` ejemplo `RhPersonal.class.php` y `RhPersonalTable.class.php`

Para generar los formularios relacionados con un objeto del modelo de datos, se emplea el comando **build-forms**, esta tarea crea una clase por cada tabla y le añade los validadores y widgets necesarios para cada columna teniendo en cuenta la información disponible en el modelo y la relación entre las tablas.

se crea una estructura de archivos en el directorio `lib/form/doctrine/` en él que se encuentran clases php de la forma `RhPersonalForm.class.php` y otra en el directorio `lib/form/doctrine/base/`, en las que están las clases `BaseRhPersonalForm.class.php`

Para instalar un plugins se lo hace con el comando siguiente: **plugin:install**, que recibe como parámetro el nombre del plugin y para publicarlos se lo hace mediante la tarea: **plugin:publish-assets**

En la clase que se indica a continuación se coloca los plugins del proyecto.

```
class ProjectConfiguration extends sfProjectConfiguration
{
    public function setup()
    {
        $this->enablePlugins('sfDoctrinePlugin');
        $this->enablePlugins('sfTCPDFPlugin');
        $this->enablePlugins('sfProtoculousPlugin');
        $this->enablePlugins('sfjQueryUIPlugin');
    }
}
```

En symfony cada modulo creado tiene acciones y se le puede accinar un template. Para nombrar las acciones se antepone la palabra **execute** seguido del nombre de la acción en mayúsculas ejemplo: executeIndex.

Para asignar un template a la acción se coloca el nombre de la acción en minúsculas seguido de la palabra **Success** en la primera letra mayúscula como se indica en el ejemplo siguiente: listarSuccess.php

En la carpeta Config de la aplicación **test**, hay que configurar **routing.yml** con el fin de indicar a que página se direcciona al momento de ingresar a la aplicación, lo cual se lo hace como se indica a continuación

```
homepage:
url: /
param: { module: login, action: index }
```

```
default_index:
url: /:module
param: { action: index }
```

Configure el archive **security.yml**, como se indica posteriormente, con el fin de dar seguridad a la aplicación, lo que hacemos es cambiar false por true.

```
default:
is_secure: true
```

Para modificar la vista y conocer el layout principal a utilizar hay que modificar el archivo **view.yml**, donde podemos colocar el título, el lenguaje, el nombre de los java script, hojas de estilo a emplear etc.

metas:

```
title: Talento Humano
#description: symfony project
#keywords: symfony, project
language: es
#robots: index, follow
```

stylesheets:

```
[main_rh.css,menudesplegable_rh.css,style_rh.css]
```

javascripts: [functions.js]

```
has_layout: true
layout: layoutLogin
```

Para el tiempo de expiración de las sesiones debes configurar el archivo factories.yml, por defecto de las sesiones de usuario es de 1800 segundos. Puedes modificarlo en la sección user del archivo.

user:

```
class: myUser
```

param:

```
timeout: 300
```

Para efectos de demostración del cierre de sección está colocado 300.

Navegadores recomendados

Con el fin de que usted pueda disfrutar de toda la funcionalidad ofrecido por SIG_RRHH, es recomendable que emplee las versiones 6.0 o superiores de los navegadores Mozilla Firefox.

Flujo de Proceso

Cada vez que un usuario ingresa al sistema, el sistema solicitará al usuario los datos de ingreso:

- Usuario
- Password

Una vez que el usuario ingresa los datos y hace clic en “Continuar”, el sistema verificará que los valores ingresados en los campo “Usuario” y “Password” corresponden a alguno de los registros de usuarios en las bases de datos del sistema.

El sistema verificará únicamente aquellos registros en la base de datos correspondientes al user y password1 del usuario. Si el registro no existe, o alguno de ellos no es correcto, el sistema volverá a presentar la página de autenticación , la cual incluirá un mensaje de error solicitando al usuario ingresar nuevamente los datos presentados.

Conservar segura la información

- ✓ Seguridad física de los sistemas de almacenamiento de datos.
- ✓ Frecuente destrucción de los documentos temporales y no requeridos.
- ✓ Sistemas de procesamiento de datos seguros que emplean avanzados equipos de cómputo y programas que protegen su información.
- ✓ Encriptación de datos.
- ✓ Uso de roles (usuarios, contraseñas) de acceso por parte de todos los usuarios
- ✓ políticas de cambio frecuente de contraseñas.

Creación Usuarios

La creación de usuarios y la asignación de roles es por parte del Administrador del Sistema.

Usuario: donde se encuentran todos los usuarios que se han creado por parte del Administrador del sistema al que le corresponde la tabla usuario que se compone de los campos que se muestran a continuación.

usuario	
id: int	
user: varchar(50)	
password: varchar(50)	
id_rol: int	
password1: varchar(50)	
email: varchar(50)	
PRIMARY	
fk_usuario_rol	
user	

Id.- representa al identificador de la tabla

user.- Campo que permite almacenar el nombre de usuario (varchar).

password.- campo que almacena la clave o password del usuario que debe ser igual al valor almacenado en **Password1**. Cabe indicar que el valor password será encriptado al momento de ser almacenado para mayor seguridad,

email.- correo electrónico del usuario a registrarse, el cual debe ser único para cada usuario registrado.

id_rol.- clave foránea (tabla rol)

rol	
id: int	
descripcion: varchar(70)	
descripcion	
PRIMARY	

Id.- representa al identificador de la tabla rol

descripcion.- Campo que permite almacenar un varchar en el cual se colocará el nombre del rol que puede tener el usuario como puede ser: secretaria, administrador, analista, etc

Previamente al registro de un usuario ya se debe tener almacenado la información necesaria en la tabla rh_ítems menús, los cuales se desplegarán según el valor que se almacena en el campo id_rol de la tabla rh_menú.

menu	
id: int	
descripcion: varchar(50)	
id_rol: int	
fk_menu_rol	
PRIMARY	

Id.- representa al identificador de la tabla menú

descripcion.- campo de tipo varchar donde se almacena el nombre del menú el cual va ser asignado al campo **id_rol** que será escogido de la tabla rol.

items_menu	
id: int	PRIMARY
descripcion: varchar(50)	
url: varchar(80)	
id_menu: int	
imagen: varchar(50)	
fk_items_menu	

Id.- representa al identificador de la tabla ítems menú

descripcion.- campo de tipo varchar donde se almacena el nombre del ítems menú el cual va ser visualizado por el usuario mediante la plantilla a la que es dirigido después de haberse autenticado.

url.- campo de tipo varchar en donde se almacén a la url o dirección que hace referencia al módulo

id_menu.- campo que almacena el numero de referencia de la tabla menú para el ítems menú correspondiente.

Imagen.- campo en el cual se almacena el nombre de la imagen que distingue al ítems menú de otro con su respectivo formato ejemplo: guardar.jpg

Módulo <login>

Creado con el fin de Autenticar usuarios, es decir encargado de verificar que tanto el usuario y password ya se encuentren registrados en la base de datos para lo cual del template **indexSuccess.php** se captura los valores de usuario y password ingresados por el usuario, seguidamente el controlador hace un llamado al modelo, haciendo uso de una consulta **verificarUsuario(\$user, \$passwd)** en la que se envía los parámetros de ingreso para ser comprobados con los existentes en los registros de usuarios.

Hay que tomar en cuenta que al momento de guardar un registro de usuario el password es encriptado, por tanto al momento de hacer la comparación del password ingresado con el existente en el registro, al ingresado por el usuario también hay que encriptar antes de dicha comparación.

FAVOR AUTENTICARSE!!



Una vez autenticado el usuario se despliega las actividades que puede cumplir para lo cual según el rol se le asigna los privilegios previamente registrados.

Módulo <intro>

El cual se encarga de mostrar el menú que le corresponde al usuario que ha sido autenticado, haciendo uso del layout auxiliar.

Mediante esta línea de código `$usuario=$this->getUser()->getUsuario();` conocemos el usuario que se encuentra autenticado, al cual se procede identificar el rol que le ha sido asignado con el fin de conocer el menú registrado para él, con sus correspondientes ítems_menus, que mediante el uso de css, se presentan al usuario los cuales puede hacer uso con solo presionar sobre cada uno de los iconos que se presentan.

Módulo <usuario>

Acción New Para el registro de usuarios en la tabla Usuarios, módulo que verifica que el usuario no se encuentre en uso y que el password sea igual a password1 y el email no haya sido registrado por lo que se hizo uso del siguiente validador symfony:

```
$this->validatorSchema->setPostValidator(new sfValidatorAnd(array( new sfValidatorSchemaCompare('password1', sfValidatorSchemaCompare::EQUAL, 'password', array('throw_global_error' => true), array('invalid' => "Las dos contraseñas no coinciden")),new sfValidatorDoctrineUnique(array('model' => 'Usuario', 'column' => array('email')), array('invalid'=> "Este email ya está en uso")), new sfValidatorDoctrineUnique(array('model' => 'Usuario', 'column' => array('user')), array('invalid'=> "Este usuario ya está en uso"))) ); que se lo encuentra en la clase UsuarioForm.class.
```

← IR A MENU

REGISTRAR USUARIO

User analistauno
Password analistarh
Id rol 3
Password1 analistarh
Email analistauno@hotmail.com

CANCELAR GUARDAR

Formulario de registro Usuario

Si los datos ingresados cumplen con la condición anterior se guardan en la tabla usuario, pero al momento de ser guardado el password es **encriptado utilizando MD5** que encripta el password tecleado por el usuario y es imposible que partiendo desde la cadena encriptada se vuelva a la contraseña origen.

Módulo <rol>

A este módulo solo puede acceder el administrador, en el cual puede crear un nuevo rol para ser asignado al usuario, eliminarlo e actualizarlo por medio de las acciones creadas en el módulo **(CRUD)** que son generadas automáticamente por el **generate admin** al crear el módulo

Módulo <menú>

A este módulo solo puede acceder el administrador, en el cual puede crear un nuevo menú, eliminarlo e actualizarlo por medio de las acciones creadas en el módulo **(CRUD)** que son generadas automáticamente por el **generate admin** al crear el módulo. Hay que tomar en cuenta que el menú que se crea va ser asignado a un rol.

Módulo <Items_Menú>

A este módulo solo puede acceder el administrador, en el cual puede crear un nuevo Items_menú, eliminarlo e actualizarlo por medio de las acciones creadas en el módulo **(CRUD)** que son generadas automáticamente por el **generate admin** al crear el módulo. Hay que tomar en cuenta que el ítems_menú que se crea va ser asignado a un menú ya registrado.

A continuación detallaremos las tablas de la base de datos y las acciones que hace uso el **usuario arecursos1**

RH_PERSONAL.- Tabla donde se registra las Características Generales del personal



rh_personal	
id	int
cedula_pasaporte	varchar(10)
nombres	varchar(100)
apellidos	varchar(100)
libreta_militar	varchar(50)
fecha_nacimiento	date
telefono	varchar(12)
celular	varchar(9)
email	varchar(100)
estado_civil	varchar(20)
tipo_sexo	char
definicion_etnica	varchar(20)
e_parroquia_id	int
discapacidad	varchar(20)
dir_domicilio	varchar(255)
carnet_jees	varchar(100)
foto	varchar(255)
estado	varchar(20)
fecha_inscripcion	date
tipo_persona	varchar(20)

Id.- Este I.D personalizado será su identificación en el sistema.

Cédula o pasaporte.- Cedula de ciudadanía o pasaporte del personal docente, administrativo y de servicios de la Dirección Provincial de Educación de Imbabura, la cédula se compone de 10 dígitos.

Apellidos Completos.- Apellidos completos del empleado tal como consta en la cédula de ciudadanía.

Nombres Completos.- Nombres completos del empleado tal como consta en la cédula de ciudadanía.

Libreta Militar.- Número de Identificación para los empleados Hombres que han cumplido con el servicio militar.

Fecha de nacimiento.- fecha de nacimiento tal como consta en la cédula de ciudadanía. digite la fecha de nacimiento en el formato indicado (dd/mm/aaaa), primero día, luego mes y por último, año.

Teléfono.- Número de teléfono del domicilio de cada empleado, que será utilizado con fines institucionales. No digite indicativos ni espacios entre los números.

Celular.- Número telefónico personal del empleado, que será utilizado con fines institucionales

Email.- dirección electrónica necesaria para emitir comunicados.

Estado Civil.- Situación jurídica de cada empleado determinada por sus relaciones de familia, provenientes del matrimonio o del parentesco, que establece ciertos derechos y deberes, el estado civil puede ser:

- ✓ Soltero
- ✓ Casado
- ✓ Unión Libre
- ✓ Viudo
- ✓ Divorciado

Sexo.- Identificación del sexo del empleado siendo este masculino o femenino.

Definición Étnica.- Grupo humano de rasgos comunes al que pertenece el personal que labora en la DEI, dato de auto-definición por parte del mismo empleado y puede ser:

- ✓ Indígena
- ✓ Mestiza
- ✓ Negra
- ✓ Blanca

Parroquia donde Reside.- Nombre de la parroquia del lugar que actualmente vive del empleado, con el fin de conocer el cantón, provincia y país de residencia.

e_parroquia	
id: int	
descripcion: varchar(255)	
canton_id: int	
fk_parroquia_canton	
PRIMARY	

Discapacidad.- Determina si los empleados poseen alguna discapacidad reconocida oficialmente por el CONADIS, entendiéndose como discapacidad a cualquier restricción o carencia de la capacidad de realizar una actividad en la misma forma o grado que se considera normal para un ser humano.

Domicilio.- Dirección domiciliaria, calle principal seguido del número de casa y calle secundaria.

Carnet de IESS.- Numero de identificación de tener acceso a los beneficios del Instituto Ecuatoriano de Seguridad social.

Foto.- Imagen de tipo carnet correspondiente al empleado, extensión .jpg

Estado.- Para conocer el estado actual del empleado, el mismo que se registra automáticamente, según el proceso que se realice pudiendo ser:

- ✓ Pasivo
- ✓ Activo
- ✓ Renuncia
- ✓ Jubilado
- ✓ Contrato

Fecha inscripción.- Se registra automáticamente la fecha del sistema, cuando se ingresa el empleado al sistema SIG_RRHH, o cuando sufre alguna modificación en sus datos o cambio de estado.

Tipo Persona.- Campo en el cual se identifica el tipo de personal que ingresa a laborar a la Dirección Provincial de Educación de Imbabura, es decir si es:

- ✓ Docente
- ✓ Administrativo
- ✓ De Servicios


Módulo <Personal>

El cual nos permite registrar a un Docente, Administrador y personal de servicio consta de acciones, cada una de ellas permiten realizar una actividad dentro del módulo.

Acción New.- es utilizada para registrar un nuevo personal, mediante el formulario que se indica en la parte de abajo.

[← IR A MENU](#)
[CANCELAR](#)
[NUEVO](#)

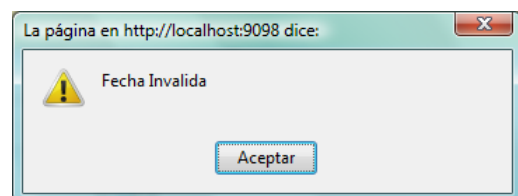
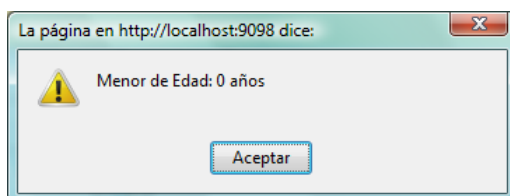
REGISTRO DE PERSONAL

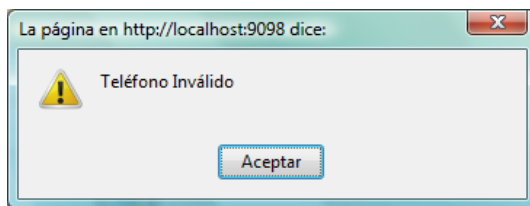
Cedula pasaporte: 0401049952 Cédula Pasaporte
 Nombres: CARLOS ALEXANDER
 Apellidos: MAGUA CAJAMARCA
 Carnet Iess: 12345678787362537
 Dir domicilio: BOYACA Y BOLIVAR
 Parroquia: SAN LUIS 
 Fecha Nacimiento: 03/05/1992
 Fotografia: C:\Users\SONY VAIO V...
 Fecha inscripcion: 03/19/2012
 Libreta Militar: 1000234566
 Definicion etnica: MESTIZA
 Discapacidad: NINGUNA
 Estado civil: SOLTERO
 Tipo sexo: MASCULINO
 Telefono: 062984340
 Celular: 097182090
 Correo Electronico: gabby@yahoo.es

Registro Personal

Parte de los datos del formulario son validados al momento del ingreso como son la fecha de nacimiento, el teléfono y el celular mediante código javascript que se encuentran en el directorio **web/js/functions.js**, en cambio otros datos son validados al momento de guardar el formulario y las validaciones de estos están añadidas en el directorio **lib/form/doctrine/RhPersonalFom.php**.

Los campos validados con código javascript al ser erróneos inmediatamente muestran mensajes como los siguientes, y no permiten que se ingrese en la casilla correspondiente, obligando de esta manera a reingresar otro valor.





Para el ingreso de la parroquia se utiliza un método de búsqueda que se encuentra en las acciones del módulo que se denomina **executeBuscarparroquia**, en el cual se incluye un parcial **_listParroquia.php**, el cual se lo muestra en el template **findparroquiaSuccess** como se indica en la figura.

Lunes 19 de Marzo de 2012

PARROQUIA

Descripción: sa 

Id	Canton	Parroquia	
3	IBARRA	SAGRARIO	<input checked="" type="checkbox"/>
4	IBARRA	SAN FRANCISCO	<input checked="" type="checkbox"/>
11	IBARRA	SALINAS	<input checked="" type="checkbox"/>
12	IBARRA	SAN ANTONIO	<input checked="" type="checkbox"/>
15	ANTONIO ANTE	IMBAYA (SAN LUIS DE COBUENDO)	<input checked="" type="checkbox"/>
16	ANTONIO ANTE	SAN FRANCISCO DE NATABUELA	<input checked="" type="checkbox"/>
17	ANTONIO ANTE	SAN JOSE DE CHALTURA	<input checked="" type="checkbox"/>
18	ANTONIO ANTE	SAN ROQUE	<input checked="" type="checkbox"/>
19	COTACACHI	SAGRARIO	<input checked="" type="checkbox"/>
20	COTACACHI	SAN FRANCISCO	<input checked="" type="checkbox"/>
31	OTAVALO	SAN LUIS	<input checked="" type="checkbox"/>
37	OTAVALO	SAN JOSE DE QUICHINCHE	<input checked="" type="checkbox"/>
38	OTAVALO	SAN JUAN DE ILUMAN	<input checked="" type="checkbox"/>
39	OTAVALO	SAN PABLO	<input checked="" type="checkbox"/>
40	OTAVALO	SAN RAFAEL	<input checked="" type="checkbox"/>
45	PIMAMPIRO	SAN FRANCISCO DE SIGSIPAMBA	<input checked="" type="checkbox"/>
50	SAN MIGUEL DE URQUQUI	SAN BLAS	<input checked="" type="checkbox"/>
52	IBARRA	LA DOLOROSA DEL PRIORATO	<input checked="" type="checkbox"/>

Listado de Parroquias

Para mostrar el calendario, se instalo el plugin **sfjQueryUIPlugin**, por tanto se modifica el campo `fecha_nacimiento` del formulario en el directorio `lib/form/doctrine/RhPersonalFom.php`, además que al momento de escoger fecha se valide mediante un javascript.

Fecha Nacimiento

Fotografia

Fecha inscripcion

Libreta Militar

Definicion etnica

Discapacidad

Estado civil

Tipo sexo

Telefono

Celular

Correo Electronico

GUARDAR

Mar 2012


Su	Mo	Tu	We	Th	Fr	Sa
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31


En el caso de la cédula, es validado al momento de guardar mediante una función php que se encuentra en **lib/Functions.php**, en el cual está el método **validarCedula(cedula)**, que recibe como parámetro la cédula ingresada por el usuario. Al ser la cédula ser incorrecta o ya estar registrada se muestra mensaje de aviso.

Acción Listar.- Permite mostrar personal activo que se encuentra registrado en la tabla **Rh_Personal**, en la ventana que se indica a continuación, donde se solicita al usuario que ingrese el número de cédula del empleado al cual se le desea hacer la modificación de algunos datos modificables, para lo cual en el modelo **RhPersonalTable.php** creamos una función de consulta denominada **getPersonal (cédula)** que recibe el parámetro indicado.

[← IR A MENU](#)
[CANCELAR](#)
[NUEVO](#)
[Listar Docentes](#)
[Listado Administrativos](#)

BUSCAR PERSONAL

Cédula: 

Cedula pasaporte	Nombres	Apellidos	Fecha nacimiento	Telefono	Email	Discapacidad	Estado	Personal
0400454005	ENMA PIEDAD	SALCEDO CUASAPUD	1949-10-17	62959894		NINGUNA	ADMINISTRATIVO	ACTIVO 

Para proceder con la edición de datos, clic en acción editar, en donde se envía el id de la persona para mediante la **acción edit** se pueda desplegar el formulario adecuado en el cual se registra los datos modificables ya que algunos campo como nombre no se puedan modificar debido a que en el parcial **_form.php** del módulo se ha restringido algunos campos como se indica.

```
<?php echo $form['cedula_pasaporte']->render(array('readonly'=>$edit?'readonly:'))?>
```

← IR A MENU CANCELAR NUEVO

EDITAR PERSONAL

Cedula pasaporte 0400454005 Cédula Pasaporte

Nombres ENMA PIEDAD

Apellidos SALCEDO CUASAPUD

Carnet Iess

Dir domicilio BOLIVAR Y ATAHUALPA

Parroquia CARANQUI

Fecha Nacimiento 03/18/1992

Fotografia C:\Users\SONY VAIO V... Examinar...
 Eliminar imagen actual

Fecha inscripcion 03/19/2012

Libreta Militar

Definicion etnica MESTIZA

Discapacidad NINGUNA

Estado civil CASADO

Tipo sexo FEMENINO

Telefono 62959894

Celular 092344567

Correo Electronico emapiedad@yahoo.es

GUARDAR

Formulario Editar Personal

RH_DATOS_FINANCIEROS.- Tabla en donde se guardara registros de los datos financieros actuales del empleado de la DEI.

rh_datos_financieros	
id: int	PRIMARY
cuenta_num: varchar(50)	
tipo_cuenta: varchar(20)	
rh_inst_bancaria_id: int	
rh_personal_id: int	
fk_datosfinancieros_instbancaria	
fk_datosfinancieros_tipocuenta	
rh_personal_id	






Id.- identificador de la tabla rh_datos_financieros

cuenta_bancaria.- campo en donde se almacenara el número de cuenta correspondiente al empleado registrado.

tipo_cuenta.- campo que registra el tipo de cuenta que tiene el empleado abierta en el banco, siendo de tipo ahorro o corriente.

rh_personal_id.- clave foránea (tabla rh_personal)

rh_inst_bancaria_id.- clave foránea (tabla rh_inst bancaria), en el campo de descripción de la tabla, se debe de ingresar el nombre de las instituciones banrias necesarias por parte del administrador, mediante el campo activo se define si se encuentra en estado activo (1) o inactivado (0).




rh_inst_bancaria	
	id: int
	descripcion: varchar(50)
	activo: tinyint(1)
	descripcion
	PRIMARY

Módulo <datosfinancieros>

Modulo que permite realizar acciones dentro del módulo como:

Acción Crear.- Permite registrar un nuevo dato financiero, para lo cual se procede a buscar el empleado mediante el número de cédula ingresado en la plantilla que se indicara posteriormente.

Al momento de enviar a buscar se activa la función **Functions.php**, la misma que comprueba si el parámetro de búsqueda ingresado es correcto. Al ser verídico se procede a verificar si el empleado al que le corresponde el número de cédula ya tiene un registro en la tabla **Rh_Datos_Financieros**, mediante la función de consulta **getCuentac(cédula)**, que se encuentra en **RhDatosFinancierosTable.php**, al encontrarse el registro el sistema mostrara un mensaje de que ya existe una cuenta, caso contrario se despliega datos relevantes después de realizar función de consulta denominada **getPersonalC(cédula)** que se encuentra en el modelo **RhPersonalTable.php** como se indica.

	IR A MENU		CANCELAR
DATOS FINANCIEROS			
Cédula:	<input type="text" value="1002561221"/>	<input type="button" value="Buscar"/>	
Nombres	Cedula	Tipo Persona	
PONCE GUAMAGSANDRA SHOMARA	1002561221	ADMINISTRATIVO	

Para proceder el ingreso presionar sobre la imagen carpeta, dando paso a la **Acción New:** a la cual se pasa el parámetro del id del empleado, campo necesario para completar el registro de los datos financieros del formulario para finalmente ser guardado mediante **function processForm**, si el formulario es válido mensaje "Datos ingresados Correctamente" caso contrario "Error Datos no Registrados"

Registro Datos Financieros.

Acción Buscar: Permite buscar al empleado mediante el número de cédula ingresado para actualizarlo. Al momento de enviar a buscar se activa la función **Functions.php**, la misma que comprueba si el parámetro de búsqueda ingresado es correcto. Al ser verídico se procede a desplegar la información relevante del empleado después de activar la función de consulta **getCuenta(cédula)** que se encuentra en el modelo **RhDatosFinancierosTable.php** como se indica en la imagen de abajo, caso contrario se mostrará un mensaje de error que dice “Cédula Inválida”










Nombres Apellidos	Cuenta num	Tipo cuenta	Inst. bancaria	Accion
SALCEDO CUASAPUD ENMA PIEDAD	11	AHORROS	BANCO DEL FOMENTO	ACTUALIZAR

Actualizar Datos Financieros

Acción Edit: a la cual se pasa el parámetro del id del empleado, con el que muestra los datos financieros existentes en el formulario, para finalmente ser modificados y guardados mediante **function processForm**, si el formulario es válido mensaje “Datos Actualizados Correctamente” caso contrario “Error Datos no Actualizados”

Editar Datos Financieros.

RH_DATOS_EDUCATIVOS.- Tabla en donde se guardara registros de los datos educativos actuales del empleado de la DEI.

rh_datos_educativos	
	id: int
	rh_personal_id: int
	e_titulo_id: int
	institucion: varchar(150)
	activo: tinyint(1)
	PRIMARY
	fk_datoseducativos_personal
	fk_mejoramiento_titulo
	fk_mejoramiento_universidad

Id.- identificador de la tabla rh_datos_educativos

rh_personal_id.- clave foránea (tabla rh_personal)

e_titulo_id.- clave foránea (tabla e_titulo) perteneciente al aplicativo SIGES.

Institución.- campo que permite almacenar el nombre de la institución en la que se ha obtenido el título.



activo.- estado del título ingresado, el cual por defecto será 1, pero este campo puede cambiar a 0, es decir pasara a un estado inactivo.

Módulo <datosedu>


Acción Crear.- Permite registrar un nuevo dato educativo, para lo cual se procede a buscar el empleado mediante el número de cédula ingresado en la plantilla que se indicara posteriormente.

Al momento de enviar a buscar se activa la función **Functions.php**, la misma que comprueba si el parámetro de búsqueda ingresado es correcto. Al ser verídico se despliega datos relevantes después de realizar función de consulta denominada **getPersonalC (cédula)** que se encuentra en el modelo **RhPersonalTable.php** como se indica.

ACTUALIZAR TITULO ACADEMICO

 IR A MENU
 CANCELAR

TITULO DEL PERSONAL

Cédula 0400454005 

Apellidos	Nombres	Cedula	Acción
SALCEDO CUASAPUD	ENMA PIEDAD	0400454005	Actual Titulo

Para proceder el ingreso del título presionar sobre Actual_Titulo, activando la **Acción New:** a la cual se pasa el parámetro del id del empleado, campo necesario para completar el registro de los datos educativos del formulario para finalmente ser guardado mediante **function processForm**, si el formulario es válido mensaje “Datos ingresados Correctamente” caso contrario “Error Datos no Registrados”

Registro Datos Educativos.

Para el ingreso del nuevo título se utiliza un método de búsqueda que se encuentra en las acciones del módulo que se denomina **execute BuscarTitu**, en el cual se incluye un parcial **_listTitulos.php**, el cual muestra en el template **findSuccess**, la búsqueda realizada como se indica en la pantalla de que se denomina: “ Buscar Títulos por el nombre.” , en donde puede colocar todo o parte del nombre del título.

Listado Títulos Encontrados

Acción Index: Permite visualizar los títulos del empleado buscado mediante el número de cédula ingresado.

Al momento de enviar a buscar se activa la función **Functions.php**, la misma que comprueba si el parámetro de búsqueda ingresado es correcto. Al ser verídico se procede a desplegar los títulos del empleado después de activar la función de consulta **getEducacionTodos(cédula)** que se encuentra en el modelo **RhDatosEducativosTable.php** como se indica en la imagen de abajo, caso contrario se mostrará un mensaje de error que dice “Cédula Inválida”

← IR A MENU CANCELAR

VISUALIZAR TITULOS

Cédula 🔍

Apellidos	Nombres	Cedula	Titulo
SALCEDO CUASAPUD	ENMA PIEDAD	0400454005	LICENCIATURA EN EDUCACION PRIMARIA
SALCEDO CUASAPUD	ENMA PIEDAD	0400454005	BACHILLER EN CIENCIAS ESPECIALIZACION INFORMATICA

Lista Títulos Empleado

Si el usuario desea regresar al menú para escoger otra actividad que desee realizar, tan solo debe pulsar sobre el icono Ir Menú el cual permite regresar al menú correspondiente al usuario.

A continuación se detalla las tablas con sus correspondientes módulos a los que puede acceder el usuario **arecursos2**

RH_ACCIONPERSONAL.- En esta tabla se ingresará registros del personal Docente, Administrativo y de Servicio de las labores que han ido desempeñado dentro de la institución, a continuación se muestra como está estructurada.

rh_accionpersonal
id: int
rh_personal_id: int
fecha_vigencia: date
rh_motivo_accion_id: int
rh_tipo_puesto_id: int
partida_presupuestaria: varchar...
is_active: tinyint(1)
explicacion_ingreso: text
tipo_situacion: varchar(1)
fecha_actual: date
rh_unidad_administrativa_id: int
rh_tipo_cargo_id: int
base_legal: mediumtext
referencia: mediumtext
num_accion: int
categoria_id: int
sueldo: varchar(100)

rh_personal_id.- se registra el id del personal sea este docente, administrativo y de servicios, al cual se le va realizar la acción de personal.

fecha_vigencia.- se registra la fecha en la que entra en vigencia la acción que se está realizando, la cual no puede ser hasta 15 días menos que la fecha actual.

rh_motivo_accion.- Se registra el motivo por la cual se realiza la acción, siendo esta detallada en la tabla que se indica a continuación:

rh_motivo_accion	
id: int	
descripcion: varchar(100)	
PRIMARY	
descripcion	

Pudiendo ser:

- ✓ Nombramiento
- ✓ Ubicación Económica
- ✓ Traslado
- ✓ Cambio
- ✓ Comisión de servicios sin sueldo
- ✓ Comisión de servicios con sueldo

rh_tipo_puesto.- indica el puesto que desempeña dentro de la institución, los mismos que son seleccionados de la tabla rh_tipo_puesto en la cual se encuentra los registros necesarios ingresados por parte del administrador.

rh_tipo_puesto	
id: int	
descripcion: varchar(255)	
PRIMARY	
descripcion	

partida_presupuestaria.- campo que permite ingresar el número de partida con la cual el personal asume desempeñar el cargo que han puesto a su disposición.

is_active.- campo donde se almacena el estado actual de la acción definida. Su valor por defecto es 1 que significa activo, el cual puede ser actualizado a su valor inactivo (0)

explicación_ingreso.- el cual alberga la explicación necesaria de la realización de la acción de personal descrita, bien puede seleccionar entre:

- ✓ Concurso de Merecimiento

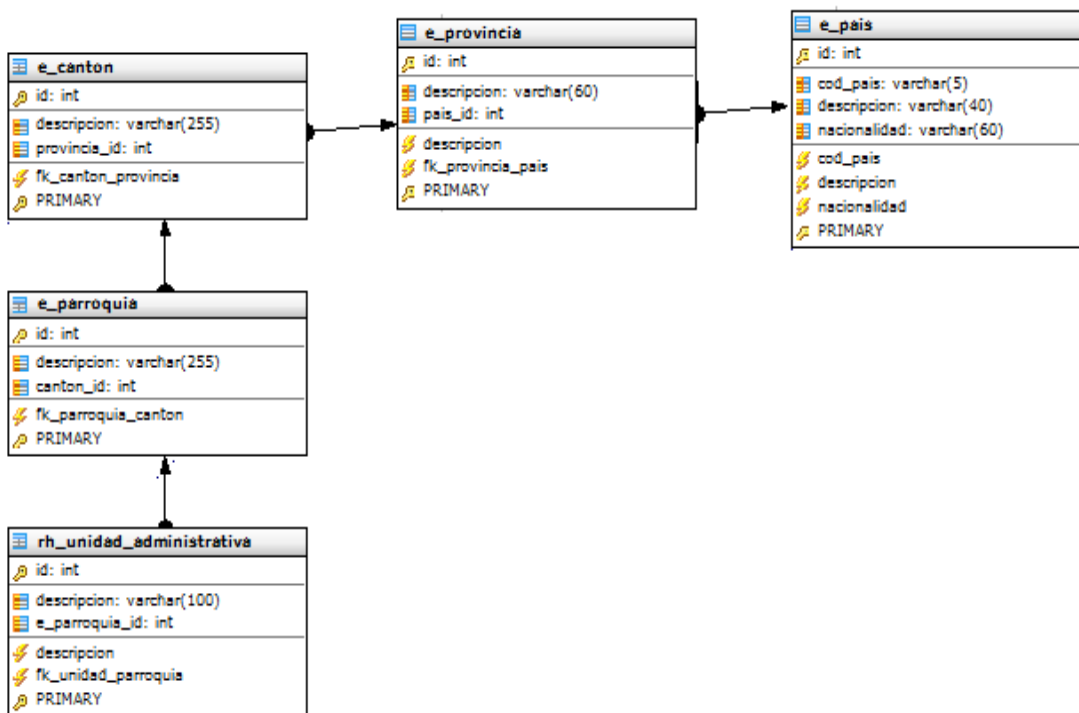
- ✓ Reemplazo por Destitución
- ✓ Facultad del Director.

tipo_situación.- Permite registrar un solo carácter el cual permite escogerlo, pudiendo ser P para las acciones propuestas es decir las que se debe legalizar, y A para acciones actuales, es decir que ya están legalizadas.

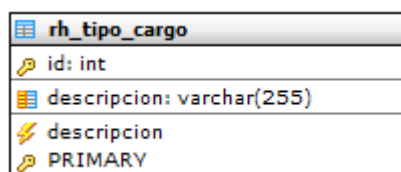
fecha_actual.- dato propio del sistema al momento del registro de la acción.

rh_unidad_administrativa_id.- clave foránea (tabla rh_unidad_administrativa).

En el campo descripción de esta tabla se coloca el nombre de la Unidad Administrativa y en e_parroquia_id; clave foránea se asigna el valor correspondiente.



rh_tipo_cargo_id.- indica el cargo que desempeña dentro de la institución, los mismos que son seleccionados de la tabla **rh_tipo_cargo** en la cual se encuentra los registros necesarios ingresados por parte del administrador.



base_legal.- campo que almacena la Ley que se rige para la realizacion de la accion de personal a registrar se selecciona entre:

base_legal	
id: int	
descripcion: varchar(255)	
PRIMARY	

referencia.- campo que almacena la informacion de referencia es decir quien a suscrito que se realice la accion de personal.

categoria_id.- clave foranea (Tabla rh_categoria_id), en la cual el campo descripcion va las categorias disponibles para el personal docente.

e_categoria	
id: int	
descripcion: varchar(50)	
PRIMARY	

Sueldo.- campo que permite albergar el sueldo, en numeros en caso de ser un decimal se almacena separando los decimales con punto.

RH_INST_TRABAJO

Tabla en la que se alberga las instituciones en las que trabaja un docente que pertenece al magisterio.

rh_inst_trabajo	
id: int	
aj_caracteristicas_institucion_id: int	
rh_accionpersonal_id: int	
fk_inst_trabajo_accion	
fk_ins_trabajo_instituciones	
PRIMARY	

id.- identificador de la tabla rh_inst_trabajo

aj_caracteristicas_institucion_id.- clave foránea (tabla aj_caracteristicas_institucion) que es propia del módulo **SIGAJ**, en la que se encuentra registrada el nombre de la institucion educativa y datos relevantes de la misma.

rh_personal_id.- clave foránea (tabla rh_personal)

Módulo <accionpersonal>


Permite registrar un nuevo dato de acción de personal, para todos los empleados de la institución.

Acción New.- permite que se despliegan dos formularios continuos en los cuales se ingresara los datos solicitados como son:

form->RhPersonal

form->RhInstTrabajo

Para registrar al personal, se utiliza un método de búsqueda que se encuentra en las acciones del módulo que se denomina **execute BuscarPersona**, en el cual se incluye un partial **_listEmpleados.php**, el cual se muestra en el template **findpersonalSuccess**, la búsqueda realizada como se indica en la pantalla de que se denomina: “ Búsqueda de Personas.” , en donde se la búsqueda se la realiza por el parámetro cédula.

BUSQUEDA DE PERSONAS				
Descripción: 1002070298 				
Id	Cedula	Nombres	Apellidos	Tipo Persona
14	1002070298	ERIBERTO ANIBAL	CAICEDO REINA	DOCENTE <input checked="" type="checkbox"/>

Datos Personal

Al momento de seleccionar el personal se toma en cuenta el tipo que es y el número de registro que tiene, mediante el cual con código javascript **/web/js/functions.js/ buscarPersonalAccion** se muestra los datos necesarios en el formulario de registro.

En caso de que sea de tipo Administrativo el formulario es el siguiente:

[← A Menú](#)
[Acción Personal](#)

NUEVA ACCION PERSONAL

Personal: ENMA PIEDAD SALCEDO CUASAPUD
Fecha Vigencia: 02 ▾ 05 ▾ 2012 ▾
Motivo Accion: NOMBRAMIENTO ▾
Explicacion ingreso: CONCURSO DE MERECIMIENTO ▾
Base legal: LEY ORGANICA DE EDUCACION INTERCULTURAL ▾
 Suscrito por la Dra. cecilia Bucheli
 jefe de recursos humanos
Referencia:
Unidad Administrativa: DIR. EDUCACION DE IMBABURA ▾
Puesto: SERVIDOR PUBLICO DE APOYO 1
Cargo: ANALISTA DE RECURSOS HUMANOS 1
Partida presupuestaria: 34
Remuneracion: 984.34
Fecha actual: 03/26/2012
Tipo situacion: PROPUESTA ▾

Si el tipo del personal es Docente se visualiza el siguiente formulario.

NUEVA ACCION PERSONAL

Personal: ENMA PIEDAD SALCEDO CUASAPUD
Fecha Vigencia: 02 ▾ 01 ▾ 2012 ▾
Motivo Accion: CAMBIO ▾
Explicacion ingreso: CONCURSO DE MERECIMIENTO ▾
Base legal: LEY ORGANICA DE EDUCACION INTERCULTURAL ▾
 Dr. carlos Mata responsable de la
 zona 1
Referencia:
Unidad Administrativa: DIR. EDUCACION DE IMBABURA ▾
Puesto: TECNICO DOCENTE
Cargo: PROFESOR DE MATEMATICA
Partida presupuestaria: 5
Fecha actual: 03/21/2012
Tipo situacion: PROPUESTA ▾

PARA PERSONAL DOCENTE Y DE SERVICIOS

Aj características institucion: JOSE JOAQUIN OLMEDO

Id	Institucion	Eliminar
1	MIGUEL DE CERVANTES	Eliminar
2	JOSE JOAQUIN OLMEDO	Eliminar

Para registrar el puesto respectivo, se utiliza un método de búsqueda que se encuentra en las acciones del módulo que se denomina **executeBuscarPuesto**, en el cual se incluye un parcial **_listPuestos.php**, el cual se muestra en el

template **findpuestoSuccess**, la búsqueda se la realiza por medio de una plantilla se denomina: “ Búsqueda de Personas.” , en donde la búsqueda se la realiza por el nombre del puesto o parte de él.

Para registrar el cargo respectivo, se utiliza un método de búsqueda que se encuentra en las acciones del módulo que se denomina **executeBuscarCargo**, en el cual se incluye un partial **_listCargos.php**, el cual se muestra en el template **findcargoSuccess**, la búsqueda realizada como se indica en la pantalla se denomina: “ Búsqueda de Cargo.” , en donde la búsqueda se la realiza por el nombre del puesto o parte de él.

En caso de Docentes para registrar la institución, se utiliza un método de búsqueda que se encuentra en las acciones del módulo que se denomina **executeBuscarInstitucion**, en el cual se incluye un partial **_listituciones.php**, el cual se muestra en el template **findSuccess**, la búsqueda realizada como se indica en la pantalla se denomina: “ Búsqueda de Instituciones.” , en donde la búsqueda se la realiza por el nombre del puesto o parte de él, dato que es guardado en sección mediante el método **test/lib/myUser.class.php/addDetalles** en caso de trabajar en varias instituciones (Docentes Especiales), el cual nos permite generar un historial, si se desea eliminar alguna de las instituciones del historial se lo realiza mediante la **Acción DeleteDetalle**.

Historial Institución

Una vez ingresado los datos solicitados en cada formulario, se procede a guardar tomando en cuenta que la acción se registrará en su campo como tipo_situación por defecto propuesta y estado activo (1) ya legalizada hay que editarla mediante **Accion Edit.-** se cambia su valor de tipo_situación propuesta a actual y se procede a guardar, dando lugar a inactivar o (0) a las acciones anteriores del empleado por medio de función **updateAcccionPersonal(id_persona)** que se encuentra en **lib/model/doctrine/rh_accion_personalTable.class..php**.

RH_LABORES_DOCENCIA

Tabla en la que se almacenara la fecha de ingreso al magisterio de los docentes.

rh_labores_docencia	
id: int	
rh_personal_id: int	
fecha_ing_magisterio: date	
PRIMARY	
rh_personal_id	

id.- identificador de la tabla rh_labores_docencia

rh_personal_id.- clave foránea (tabla rh_personal), permite id del personal docente de planta, es decir que no se encuentre en contrato.

fecha_ing_magisterio.- fecha en la cual el docente pasa a ser parte del ministerio.

Módulo <magisterio >

Acción New.- Para registrar al personal, se utiliza un método de búsqueda que se encuentra en las acciones del módulo que se denomina **execute BuscarPersona**, en el cual se incluye un partial **_listEmpleados.php**, el cual se muestra en el template **findpersonalSuccess**, la búsqueda realizada como se indica en la pantalla de que se denomina: “ Búsqueda de Personas.” , en donde se la búsqueda se la realiza por el parámetro cédula.

BUSQUEDA DE PERSONAS					
Descripción:		1002070298			
Id	Cedula	Nombres	Apellidos	Tipo Persona	
14	1002070298	ERIBERTO ANIBAL	CAICEDO REINA	DOCENTE	<input checked="" type="checkbox"/>

Datos Personal

Para completar el registro ingresar la fecha de ingreso al magisterio perteneciente al docente seleccionado, la cual mediante código javascript se encuentra valida en el directorio **web/js/functions.js**

RH_LABORES_ADMINISTRATIVOS

Tabla en la que se almacenara la fecha de ingreso al magisterio de los docentes.

rh_labores_administrativos	
id: int	
rh_personal_id: int	
fecha_ingministerio: date	
fk_datoslaborales	
PRIMARY	


id.- identificador de la tabla rh_labores_docencia

rh_personal_id.- clave foránea (tabla rh_personal), permite id del personal administrativo de la DEI, es decir que no se encuentre en contrato.

fecha_ingministerio.- fecha en la cual pasa a ser parte del Magisterio Nacional de Educadores.

Módulo <ministerio>

Acción New.-Para registrar al personal, se utiliza un método de búsqueda que se encuentra en las acciones del módulo que se denomina **execute BuscarPersona**, en el cual se incluye un partial **_listEmpleados.php**, el cual se muestra en el template **findpersonalSuccess**, la búsqueda realizada como se indica en la pantalla de que se denomina: “ Búsqueda de Personas.” , en donde se la búsqueda se la realiza por el parámetro cédula.

BUSQUEDA DE PERSONAS					
Descripción: 1002070298 					
Id	Cedula	Nombres	Apellidos	Tipo Persona	
14	1002070298	ERIBERTO ANIBAL	CAICEDO REINA	DOCENTE	<input checked="" type="checkbox"/>

Datos Personal

Para completar el registro ingresar la fecha de ingreso al ministerio, la cual mediante código javascript se encuentra valida en el directorio **web/js/functions.js**

Módulo <jubilaciones>


Acción Index.- Se ejecuta esta acción donde se despliega la pantalla en donde se solicita el ingreso del número de cédula del docente al que se desea actualizar su estado de activo a jubilado. Se valida el parámetro de búsqueda mediante el método **validarCedula(cedula)** de la clase **Functions.php**, Si la cédula es correcta se muestra el registro, en el cual al dar clic sobre el icono da paso a la **Acción Jubilación**, en donde se procede a cambiar el estado del Docente con la función **cambiarEstadoJubilado(\$cedula)** que recibe el parámetro de la búsqueda que está en el directorio **lib/model/RhPersonalTable.class.php**, también se cambia el estado de la acción de personal actual, con la función **updateAcccionActiva(\$id)** que recibe el

parámetro del identificador del personal que está en el directorio **lib/model/RhAccionPersonal.class.php**.

[← Ir a Menu](#) [Listado](#)

BUSCAR PERSONAL PARA COLOCAR ESTADO JUBILADO

Cédula







Apellidos	Nombres	Cedula	Cargo	Puesto	Jubilación
CAICEDO REINA	ERIBERTO ANIBAL	1002070298	PROFESOR DE INFORMATICA	TECNICO DOCENTE	

Acción Listar.- la cual nos permite mostrar el listado de los jubilados por medio de la función de consulta iniciada en la misma y de contabilizarlos mediante la función de consulta acción **getJubiladoCount** que se encuentra en el directorio **lib/model/RhPersonal.class.php**.

En esta acción también se encuentra el método de cómo se realiza el paginado del listado anterior como se muestra en el template **listarSuccess.php**

RH_RENUNCIA

Permite registrar la renuncia del personal de la Dirección de Educación de Imbabura.

rh_renuncia
 id: int
 rh_personal_id: int
 fecha_renuncia: date
 fecha_exp: date
 motivo_renuncia: varchar(50)
 PRIMARY

id.- identificador de la tabla

rh_personal_id.- clave foránea (Tabla rh_personal)

fecha_renuncia.- fecha de la resolución de la renuncia, puede emitirse durante 15 días

fecha_exp.- fecha de expedición o emisión de la acción de personal renuncia

motivo_renuncia.- campo de tipo varchar para almacenar el motivo de renuncia, seleccione: destitución, sanción.

Módulo <renuncias>

Acciones relacionadas con las renuncias del personal de la institución.

Acción Index.- muestra una plantilla en la que se solicita que ingrese el número de la cédula de identidad con el fin de buscar al empleado para registrar los datos necesarios para cambiar su estado de activo a renuncia. El parámetro es validado mediante el método **validarCedula(cedula)** de la clase **Functions.php**, Si la cédula es correcta se muestra el registro, en el cual al dar clic sobre el enlace renuncia, se da paso a la **Acción New**, en donde se procede a ingresar los datos restantes, la fecha de renuncia es validada al momento de ingresarla mediante código javascript de la función **verificarFechaR()**, que se encuentra alojado en el directorio **web/js/functions.js**.

The screenshot shows a web interface with a navigation bar at the top containing buttons for 'A Menú', 'Nueva Renuncia', 'Renuncias Administrativas', and 'Renuncias Docentes'. Below this is a yellow header with the text 'BUSCAR PERSONAL'. A search input field contains the ID '1001766177' and has a search icon. Below the search field is a table with the following data:

Apellidos	Nombres	Cedula	Tipo Persona	Acción
CRIOLLO SEVILLANO	ROSA GIOVANY	1001766177	DOCENTE	Renuncia

Una vez ya registrados todos los datos solicitados en el formulario y dar clic en guardar se activa la función **processForm**, el cual según conocer tipo persona que se registra mediante la función de consulta **tipoPersona (\$id)**, se procede a direccionar a la acción correspondiente para indicar el correspondiente template de impresión.

Acción Listado.- Acción que ejecuta distintas funciones de consulta como es **getRenunciaCountd()**, la cual permite contabilizar los registros de renuncia de docentes, que se encuentra alojada en el modelo **RhPersonal**. En esta misma acción se realiza la consulta que permite desplegar el listado de los docentes con estado renuncia, así como la paginación del mismo

Cambiar el estado del Docente con la función **cambiarEstadoRenuncia(\$cedula)** que recibe el parámetro de la búsqueda que está en el directorio **lib/model/RhPersonalTable.class.php**, también se cambia el estado de la acción de personal actual, con la función **updateAcccionActiva(\$id)** que recibe el parámetro del identificador del personal que está en el directorio **lib/model/RhAccionPersonal.class.php**.

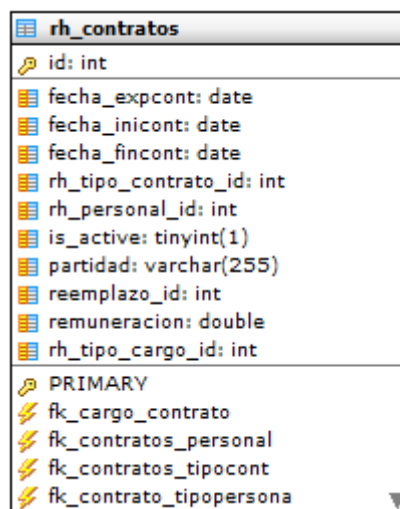
Acción Listadoa.- Acción que ejecuta distintas funciones de consulta como es **getRenunciaCounta()**, la cual permite contabilizar los registros de renuncia de administrativos, que se encuentra alojada en el modelo RhPersonal. En esta misma acción se realiza la consulta que permite desplegar el listado de los administrativos con estado renuncia, así como la paginación del mismo

Cambiar el estado del Administrativo con la función **cambiarEstadoRenuncia(\$cedula)** que recibe el parámetro de la búsqueda que está en el directorio **lib/model/RhPersonalTable.class.php**, también se cambia el estado de la acción de personal actual, con la función **updateAcccionActiva(\$id)** que recibe el parámetro del identificador del personal que está en el directorio **lib/model/RhAccionPersonal.class.php**.

A continuación se indica los módulos a los cuales tiene acceso el usuario arecursos3 con sus respectivas tablas y acciones de estos.

RH_CONTRATOS

Tabla que permite registrar información de contratos del personal, así como también la renovación de los mismos siempre y cuando no estén activos, la finalización terminado el tiempo estipulado.



rh_contratos	
id: int	
fecha_expcont: date	
fecha_inicont: date	
fecha_fincont: date	
rh_tipo_contrato_id: int	
rh_personal_id: int	
is_active: tinyint(1)	
partidad: varchar(255)	
reemplazo_id: int	
remuneracion: double	
rh_tipo_cargo_id: int	
PRIMARY	
fk_cargo_contrato	
fk_contratos_personal	
fk_contratos_tipocont	
fk_contrato_tipopersona	

Id.- identificador de la tabla

fecha_expcont.- campo en el que automáticamente se registra la fecha actual del sistema.

fecha_inicont.- campo en donde se almacena la fecha de inicio del contrato, el cual es mayor o igual a la fecha actual.

fecha_fincont.- campo que permite el registro de la fecha de finalización del contrato el cual es mayor a fecha inicio.

rh_tipo_contrato.- clave foránea (tabla rh_tipo_contrato), permite el almacenamiento de atributo descripción, en el cual se encuentra registrado el nombre del tipo de contrato, el que tiene un estado 1 activo y 0 inactivo.

rh_tipo_contrato	
id: int	
descripcion: varchar(100)	
activo: tinyint(1)	
descripcion	PRIMARY

rh_personal_id.- clave foránea (tabla rh_personal), donde se muestra el nombre y apellido del personal

is_active.- atributo que permite tiene por defecto el valor de 1, el cual indica que está en estado activo, pero que tras realizar alguna acción puede tomar el valor de 0, es decir se inactiva.

partida.- campo en donde se almacenara la partida presupuestaria individual asignada por el departamento financieros.

reemplazo.- clave foránea (tabla rh_personal), donde se muestra el nombre y apellido del personal

remuneración.- atributo que hace referencia al sueldo que se paga por el contrato a realizase.



rh_tipo_cargo_id- clave foránea (tabla rh_tipo_cargo), selección del cargo a desempeñar.

Módulo <contratos>

Acción New

En el caso de que sea un contrato nuevo.- permite ingresar los datos solicitados en las plantillas indicadas de acuerdo al empleado que se ingresa mediante un método de búsqueda que se encuentra en las acciones del módulo que se denomina **executeBuscarPersona**, en el cual se incluye un partial

`_listEmpleados.php`, el mismo que se muestra en el template `findpersonalSuccess`, la búsqueda realizada como se indica en la pantalla de que se denomina: “ Búsqueda de Personas.” en donde se la realiza por medio del parámetro cédula, siempre y cuando la persona tiene su estado pasivo. Al momento de seleccionar el personal se toma en cuenta el tipo persona para que mediante código javascript alojado en `/web/js/functions.js/` función `buscarPersonal`, permita desplegar los campos correspondientes en cada formulario.


BUSQUEDA DE PERSONAS				
Descripción: 1002070298 				
IdCedula	Nombres	Apellidos	Tipo Persona	
141002070298	ERIBERTO ANIBAL	CAICEDO REINA	DOCENTE	

Datos Personal


En caso de que sea de tipo personal Administrativo el formulario es el siguiente:

REGISTRAR CONTRATOS	
 A Menú	 Contrato
Contratado	FARINANGO RAUL TUQUERREZ 
Fecha Expedicion	03/25/2012
Fecha Inicio	03 ▾ 05 ▾ 2012 ▾
Fecha Finilizacion	02 ▾ 02 ▾ 2013 ▾
P.Presupuestaria	76
Sueldo	500.65
Contrato de:	OCASIONALES ▾
Puesto	ANALISTA DE ACOMPAÑAMIENTO Y PROM 
Reemplaza a:	<input type="text"/> 
 GUARDAR	

Al ser Docente el formulario desplegado es el que se muestra a continuación

[← A Menú](#)


REGISTRAR CONTRATOS

Contratado LUIS FERNANDO CAMUES HERRERA 

Fecha Expedicion 03/25/2012


Fecha Inicio 02 ▾ 04 ▾ 2012 ▾


Fecha Finilizacion 02 ▾ 04 ▾ 2013 ▾


P.Presupuestaria 94



Sueldo 600.23

Contrato de: OCASIONALES ▾

Puesto ANALISTA DE ARTICULACION POLITICA COM 

Reemplaza a: 

Aj características institucion CENTRO ARTESANAL ALPACHACA 


Id	Institucion	Eliminar
1	JOSE MARIA URBINA	
2	CENTRO ARTESANAL ALPACHACA	

En el caso de que sea el formulario para Docentes para registrar la institución, se utiliza un método de búsqueda que se encuentra en las acciones del módulo que se denomina **executeBuscarInstitucion**, en el cual se incluye un parcial **_listituciones.php**, el cual se muestra en el template **findSuccess**, la búsqueda realizada como se indica en la pantalla se denomina: “ Búsqueda de Instituciones.”, en donde la búsqueda se la realiza por el nombre del puesto o parte de él, dato que es guardado en sección mediante el método **test/lib/myUser.class.php/addDetalles** en caso de trabajar en varias instituciones (Docentes Especiales), el cual nos permite generar un historial, si se desea eliminar alguna de las instituciones del historial se lo realiza mediante la **Acción DeleteDetalle**.

Lunes 19 de Marzo de 2012

PARROQUIA

Descripción: sa 

Id	Canton	Parroquia	
3	IBARRA	SAGRARIO	
4	IBARRA	SAN FRANCISCO	
11	IBARRA	SALINAS	
12	IBARRA	SAN ANTONIO	
15	ANTONIO ANTE	IMBAYA (SAN LUIS DE COBUENDO)	
16	ANTONIO ANTE	SAN FRANCISCO DE NATABUELA	
17	ANTONIO ANTE	SAN JOSE DE CHALTURA	
18	ANTONIO ANTE	SAN ROQUE	
19	COTACACHI	SAGRARIO	
20	COTACACHI	SAN FRANCISCO	
31	OTAVALO	SAN LUIS	
37	OTAVALO	SAN JOSE DE QUICHINCHE	

Búsqueda de Institución

PARA PERSONAL DOCENTE Y DE SERVICIOS

Aj características institucion JOSE JOAQUIN OLMEDO

Agregar

IdInstitucion	Eliminar
1 MIGUEL DE CERVANTES	Eliminar
2 JOSE JOAQUIN OLMEDO	Eliminar
3 JOSE JOAQUIN OLMEDO	Eliminar

GUARDAR

Historial Institución

Una vez ingresado los datos solicitados en cada formulario, se procede a guardar donde se invoca a la acción **processForm**, en donde se valida el formulario en caso de tener valores no validos se muestra el error correspondiente, y si son correctos se guarda y se muestra la información en el template correspondiente con el fin de ser legalizado, para los docentes se direcciona a la **acción Imprimirdoc**, para los administrativos a la **acción Imprimir** en donde se realiza las consultas necesarias como: **getContratoMax(id_persona)**, que permite retornar el ultimo identificador del contrato de la persona, **getRetornarContrato(id_contrato)** el cual permite visualizar campos importantes del último contrato realizado que se encuentran en el model **RhContratosTable.php**, también la consulta para conocer el Jefe de Recursos Humanos mediante **getJefeRecursosHumanos()**, función ubicada en el model **RhAccionPersonalTable.php** y la funcion **getDirector()**, para conocer el director de Educacion Actual de Imbabura y en el model **RhPersonalTable.php** la funcion **cambiarEstadoCont(\$id_person)**, para cambiar el estado de pasivo a contrato. Pero para el caso de ser contrato para docentes también se realiza la consulta para determinar en la o las instituciones en donde se contrata sus servicios mediante la función **getInstitucionesc(id_contrato)** localizada en el model **de RhInstContratosTable.php**

Acción Index

La que nos permite visualizar los contratos que puede renovar el empleado al que se le realiza la búsqueda mediante la función de consulta **getContratosByCi(cedula)**, que recibe como parámetro la cédula de identidad si esta es incorrecta se muestra un mensaje de error , en el caso que la búsqueda no encuentre datos, el sistema da la opción de volver a ingresar nuevamente de realizar la búsqueda, como se indica en la plantilla que se muestra a continuación.

LISTA DE CONTRATOS					
Cédula: 1002088076					
Id	Nombres	Apellidos	Fecha expcont	Tipo Contrato	
51	LUIS FERNANDO	CAMUES HERRERA	2012-03-25	1	

Contratos a Renovar

Clic en el icono renovar, que activa a la **Acción New**, en donde se realiza la función de consulta **getContratoRenovar(id_persona)**, ubicada en el modelo **RhContratosTable.php** solo si existe valores que retornen a realizar la consulta se realiza la renovación de contrato escogido ya que son subidos a sección para ser utilizados en la renovación.

Acción Contrato

En la que se encuentra alojada la consulta que muestra en el template correspondiente a la acción todos los contratos existentes, en el cual se puede dar por finalizado presionando sobre el icono

LISTADO DE CONTRATACIONES		
Cédula	Contratado	Puesto Institucional
1002088076	CAMUES HERRERA LUIS FERNANDO	ANALISTA DE ARTICULACION POLITICA CON OTRAS FUNCIONES DEL ESTADO
1002209888	TUQUERREZ FARINANGO RAUL	ANALISTA DE ACOMPAÑAMIENTO Y PROMOCION DE ACTORES SOCIALES Y POLITICOS

Acción Inactivar

Al momento de realizar dicha acción se realiza las siguientes consultas **cambiarEstadoContrato()**; para cambiar el estado de los contratos de activo (1) a inactivo (0) funcion alojada en el modelo **RhContratosTable.php**, de igual mader para cambiar el estado del personal de contratado a pasivo mediante la consulta **cambiarEstadoContrato()**, ubicada en el modelo **RhPersonalTable.php**, posteriormente mostrando un mensaje de notificación.

Acción Listado

La cual al ser ejecutada da paso a que se realice la consulta indicada en este mismo lugar que permite mostrar un listado del personal contratado, con su respectiva paginación y el número de los mismos como se indica a continuación,

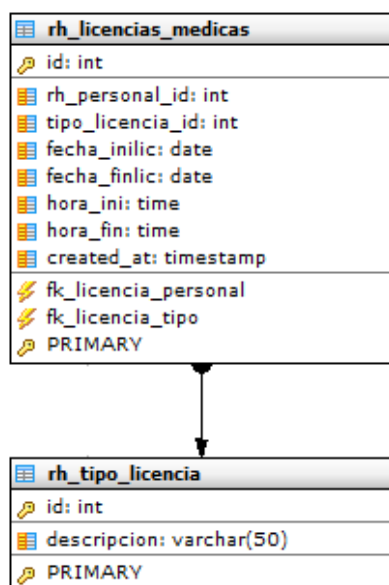


al que tenemos la opción de imprimir.

A continuación se presentan las tablas y módulos correspondientes al usuario arecursos4 y las acciones que se realizan en cada uno de los módulos.

RH_LICENCIAS_MEDICAS

Tabla que permite almacenar datos relevantes de un licencia médica, a continuación se detalla cada campo.



id.- identificador de la tabla

rh_personal_id.- clave foránea (tabla rh_personal), hace referencia a los id de cada personal registrado.

tipo_licencia.- clave foránea (tabla rh_tipo_licencia)

fecha_inilic.- permite registra la fecha de inicio de la licencia

fecha_fin.- fecha que indicara la finalización de la licencia_médica

hora_ini.- tiempo en horas de inicio de la licencia médica

hora_fin.- tiempo en horas de finalización de la licencia médica

El campo descripción de la tabla **rh_tipo_licencia** permite que se almacenen el nombre de cada motivo de los permisos.

Módulo licenciasmedicas

En este módulo se realizan acciones que corresponden a la administración de las licencias medicas del módulo indicado.

Acción New

Formulario de registro de Licencias médicas en el cual se debe registrar los datos solicitados como se indica en la pantalla siguiente:

A screenshot of a web form titled "REGISTRO DE LICENCIAS MEDICAS". At the top left is a button "A Menú". Below the title, there is a "Personal" field with the value "ENMA PIEDAD SALCEDO CUASAPUD" and a search icon. Below that are several fields: "Licencia para" with a dropdown menu showing "LACTANCIA", "Fecha Inicio" with the value "03/26/2012", "Fecha Finalización" with the value "03/30/2012", "Hora Inicio" with a dropdown showing "9" and "17", and "Hora Finalización" with a dropdown showing "11" and "17". At the bottom left is a "GUARDAR" button.

Formulario Licencia Médica

Para registrar al personal, se utiliza un método de búsqueda que se encuentra en las acciones del módulo que se denomina **execute BuscarPersona**, en el cual se incluye un partial **_listEmpleados.php**, el cual se muestra en el template **findpersonalSuccess**, la búsqueda realizada como se indica en la pantalla de que se denomina: " Búsqueda de Personas." , en donde se la búsqueda se la realiza por el parámetro cédula.

A screenshot of a search results page titled "BUSQUEDA DE PERSONAS". It features a search bar with the text "Descripción: 1002070298" and a search icon. Below the search bar is a table with the following data:

Id	Cedula	Nombres	Apellidos	Tipo Persona	
14	1002070298	ERIBERTO ANIBAL	CAICEDO REINA	DOCENTE	<input checked="" type="checkbox"/>

Datos Personal

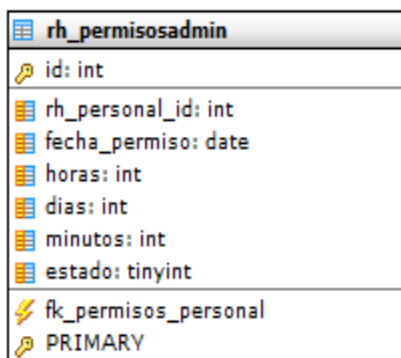
Fechas ingresadas mediante el calendario JQuery, que son validadas al momento de su ingreso mediante código javascript que se encuentra en el directorio **web/js/functions.js** mediante las funciones **verificarFechaL(fecha_actual)** y **verificarFechaLic()**, por medio de las cuales se comprueba la fecha de inicio no

sea menor a la del sistema y la fecha final que no debe ser menor a la fecha inicio respetivamente.

Una vez ingresado los datos solicitados en el formulario, se procede a guardar donde se invoca a la acción **processForm**, el cual valida el formulario en caso de tener valores no validos se muestra el error correspondiente, y si son correctos se guarda y se muestra la información para lo que se direcciona a la **acción Imprimir**, en donde se realiza las consultas necesarias indicadas en la misma acción, para luego ser visualizadas en la plantilla **imprimirSuccess.php**

RH_PERMISOS_ADMIN

Tabla en la que se almacena información perteneciente a permisos ocasionales de administrativos, los cuales descontados de las vacaciones.



rh_permisosadmin	
id: int	PRIMARY
rh_personal_id: int	
fecha_permiso: date	
horas: int	
dias: int	
minutos: int	
estado: tinyint	
fk_permisos_personal	

id.-identificador de la tabla

rh_personal_id.- clave foránea (tabla rh_personal), identificador del administrador que posee permiso.

Fecha_permiso.- ingreso de fecha permiso no menor que la fecha actual.

horas.-campo en el que se guarda una hora que va dentro de un rango de 1-7 horas

días.- campo de almacenamiento de los días que van desde de 1-30 minutos

minutos.- campo que registra desde 1 a 59 minutos

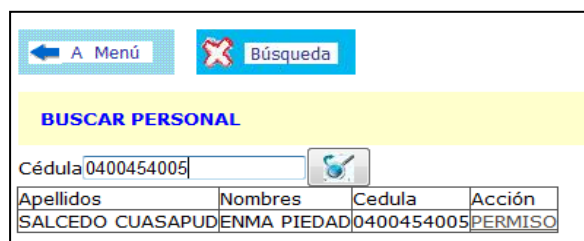
estado.- el valor almacenado en este campo por defecto es 1 indicando que está activo, al momento de ser utilizados los registros activos pasaran a un estado 0, es decir inactivos.

Módulo permiso

Se registra los permisos ocasionales del personal Administrativo mediante las acciones indicadas en el mismo como son:

Acción Index

Permite buscar al personal administrativo al cual se le registrara el permiso ocasional solicitado, por lo que en la pantalla que se indicara a continuación se debe ingresar la cédula de identidad correcta en el caso de ser incorrecta se muestra un mensaje de error, caso contrario se procede a mostrar el registro de la consulta **getPersonalActivosAd(\$cedula)** que recibe como parámetro la cédula función alojada en el modelo **RhPersonalTabla.php**, siempre y cuando la función de consulta **getDescuentos(\$id)** del modelo **RhPermisosadminTable.php** y los correspondientes cálculos realizados en la acción el número de días **días_descuento** no supere a los 15.



Apellidos	Nombres	Cedula	Acción
SALCEDO	CUASAPUDENMA PIEDAD	0400454005	PERMISO

Búsqueda Administrativa

Una vez indicado el registro y al presionar sobre el link permiso se ejecuta la

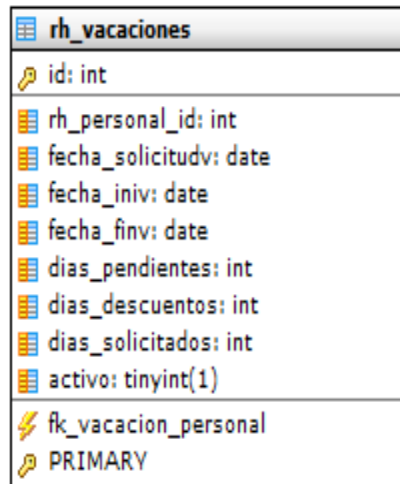
Acción New:

La cual recibe el número de identificación del empleado y las fechas serán ingresadas mediante el calendario JQuery, que son validadas al momento de su ingreso mediante código javascript que se encuentra en el directorio **web/js/functions.js** mediante las funciones **verificarFechaP(fecha_actual)** y **verificarFechaPer()**, por medio de las cuales se comprueba la fecha de inicio no sea menor a la del sistema y la fecha final que no debe ser menor a la fecha inicio respetivamente. La hora es vaildad por la funcion **validarhora()**, los minutos **validarminuto()** y los días **validardia()**.

Una vez ingresado los datos solicitados en el formulario, se procede a guardar donde se invoca a la acción **processForm**, el cual valida el formulario en caso de tener valores no validos se muestra el error correspondiente, y si son correctos se guarda el registro en la tabla **rh_personaladmin**

RH_VACACIONES

Tabla que almacena los registros de vacaciones anuales para los administrativos de la institución. A los cuales se les resta los permisos ocasionales registrados en el año correspondiente a la vacaciones.



The screenshot shows the structure of the 'rh_vacaciones' table. It includes a primary key 'id' of type 'int'. Other fields include 'rh_personal_id' (int), 'fecha_solicitudv' (date), 'fecha_iniv' (date), 'fecha_finv' (date), 'dias_pendientes' (int), 'dias_descuentos' (int), 'dias_solicitados' (int), and 'activo' (tinyint(1)). There is also a foreign key 'fk_vacacion_personal' and a 'PRIMARY' constraint.

Field Name	Field Type
id	int
rh_personal_id	int
fecha_solicitudv	date
fecha_iniv	date
fecha_finv	date
dias_pendientes	int
dias_descuentos	int
dias_solicitados	int
activo	tinyint(1)

id.- identificador de la tabla

rh_personal_id.- clave foránea (tabla rh_personal), personal administrativo al cual se le registrará las vacaciones

fecha_solicitud.- fecha cuando se registra las vacaciones y es igual a la fecha actual del sistema.

fecha_iniv.- fecha en la inicio de las vacaciones del empleado administrativo

fecha_finv.- fecha de finalización de las vacaciones solicitadas por el empleado en el año en curso.

días_pendientes.- días disponibles, es decir días de los que puede hacer uso el administrativo para sus vacaciones.

días-descuentos.- días gastados durante los permisos ocasionales solicitados por parte del empleado.

días solicitados.- días solicitados en vacaciones

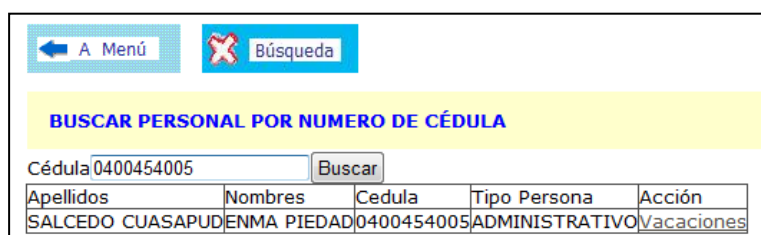
activo.- indica el estado actual de las vacaciones 1 activo y 0 inactivo.

Módulo Vacaciones

En el cual se puede realizar acciones como:

Acción Búsqueda

Permite buscar al personal administrativo al cual se le registrara el permiso ocasional solicitado, por lo que en la pantalla que se indicara a continuación se debe ingresar la cédula de identidad correcta en el caso de ser incorrecta se muestra un mensaje de error, caso contrario se procede a obtener la fecha de inscripción mediante la función de consulta **getFechaIns(\$id)** localizada en el modelo **RhPersonalTable.php**, la cual es utilizada en el método de la clase **Functions.php** **Functions::tiempo_transcurrido(fecha)**; con el fin de conocer el tiempo transcurrido desde la inscripción hasta el día de hoy. En el caso que sea menor que uno esto quiere decir que aun no cumple el año de trabajo poder solicitar vacaciones, si es mayor o igual a uno se procede a mostrar el registro de la consulta **getPersonalActivosAd(\$cedula)** que recibe como parámetro la cédula función alojada en el modelo **RhPersonalTabla.php**, siempre y cuando la función de consulta **getDescuentos(id_persona)** del modelo **RhPermisosadminTable.php**.



Apellidos	Nombres	Cedula	Tipo Persona	Acción
SALCEDO	CUASAPUDENMA PIEDAD	0400454005	ADMINISTRATIVO	Vacaciones

Búsqueda Administrativo

Una vez indicado el registro y al presionar sobre el link vacaciones se ejecuta la **Acción New:**

La cual recibe el número de identificación del empleado y las fechas serán ingresadas mediante el calendario JQuery, que son validadas al momento de su ingreso mediante código javascript que se encuentra en el directorio **web/js/functions.js** mediante las funciones **verificarFechaV(fecha_actual)** y **verificarFechaVa()**, por medio de las cuales se comprueba la fecha de inicio no sea menor a la del sistema y la fecha final que no debe ser menor a la fecha inicio pero tampoco debe superar a los días de vacaciones disponibles para el empleado indicado, los días pendientes van a ser modificados en esta acción mediante las operaciones realizadas ahí para lo cual se toma en cuenta la función

de consulta **getPendientes(id_persona)** que está estructurada en el módulo RhVacacionesTabla.php , los cuales no serán números negativos. En cambio para los días descuentos se utiliza la consulta **getDescuentos(id_persona)** alojada en el modelo **RhPermisosadminTable.php** y tras operaciones realizadas en la acción descrita se calcula los días descuentos los cuales no serán negativos ni mayores a 15 días.



REGISTRO DE VACACIONES	
Fecha iniv	03/26/2012
Fecha finv	03/30/2012
Dias pendientes	30
Dias descuentos	0
Dias solicitados	
<input type="button" value="GUARDAR"/>	

Formulario Ingreso Vacaciones

Una vez ingresado los datos solicitados en el formulario que se pueden registrar , se procede a guardar donde se invoca a la acción **processForm**, el cual valida el formulario en caso de tener valores no validos se muestra el error correspondiente, y si son correctos se calcula los días solicitados utilizando un método de la clase **Functions.php** llamado **restarFechas** (fecha1, fecha2), que recibe como parámetro dos fechas para este caso la fecha de inicio de vacaciones y la fecha de finalización de las mismas, el cual retorna un valor que va ser asignado a días solicitados. Cabe indicar que también se modifica el valor de los días pendientes ya que será los días descuentos menos los solicitados, finalmente guarda el registro en la tabla **rh_vacaciones**.

Posteriormente se cambia el estado de los permisos ocasionales realizados en el periodo del año correspondiente a la vacación emitida mediante la función de consulta **anularpermisos(id_personal)**, ubicada en el modelo **RhPermisosadminTable.php**.

Finalmente mostrar la plantilla **ImprimirSuccess** con los datos de la vacación registrada mediante la ejecución de la **Acción Imprimir**.