



Manitos mágicas



Deja que tus manos se expresen por ti

MANUAL TÉCNICO

APLICACIÓN WEB PARA LA ENSEÑANZA CON EL LENGUAJE DE SEÑAS, MÓDULO AVANZADO PARA NIÑOS DEL TERCER AÑO DE EDUCACIÓN BÁSICA DE LA UNIDAD EDUCATIVA DE SORDOS DEL GOBIERNO PROVINCIAL DE IMBABURA

MIRIAN CATALINA FRANCO CHICAIZA
UNIVERSIDAD TÉCNICA DEL NORTE FICA – CISIC
IBARRA - 2015

1. Introducción

Según la constitución ecuatoriana, sobre la educación, en la sección sexta, art.47 Se garantiza la educación para las personas con necesidades especiales, visto como grupo prioritario de atención en varios campos entre los cuales está la equiparación de oportunidades laborales, educación, becas e inclusión social.

En la actualidad la tecnología forma parte del campo educativo en todo tipo de instituciones, siendo de vital importancia incluirla y potenciarla en instituciones enfocadas a una o varias necesidades especiales para ayudar a realizar las tareas de aprendizaje a los estudiantes. La utilización de la multimedia en la educación ha despertado el interés educativo en los niños con capacidad diferente logrando así su interacción con el docente y la aplicación informática, ayudando al desarrollo de sus destrezas con rapidez a diferencia de los métodos tradicionales de enseñanza.

El presente proyecto ha sido desarrollado con la finalidad de convertirse en apoyo del proceso de enseñanza-aprendizaje del lenguaje de señas en el nivel básico y el avanzado para los niños del tercer año de educación básica de la Unidad Educativa de Sordos del gobierno provincial de Imbabura mediante el uso de una aplicación web.

2. Objetivos

Proporcionar una guía para el lector del desarrollo de la aplicación web MANITOS MÁGICAS.

3. Metodología utilizada para el desarrollo de la aplicación

La metodología utilizada para el desarrollo de la aplicación es la OOHDM, es una metodología orientada a modelos para el desarrollo de aplicaciones hipermedia y en particular de aplicaciones Webs, esta aplicación propone cinco fases Definición u Obtención de Requerimientos, Diseño o Modelo Conceptual, Diseño Navegacional, Diseño de Interfaces Abstractas, Implementación.

4. Diseño de la base de datos

La base de datos que se utilizó fue PostgreSQL 9.0, es un gestor de bases de datos objeto-relacional, distribuido bajo licencia BSD, muy conocido y usado en entornos de software libre, soporta un conjunto de funciones avanzadas. Esta base de datos permite un diseño amplio y concreto de las tablas y los campos que contiene la base de datos.

A continuación se describe las tablas que conforman la Base de datos.

Perfil: La clase Perfil contiene toda la información de los perfiles con el que cuenta la aplicación. Esta es su estructura.

PERFIL	
perfilusuario :	String
descripción :	String

Usuario: La clase Usuario contiene la información personal de cada uno de los usuarios registrados en el sistema y está estructurada de la siguiente manera.

USUARIO	
idusuario	: string
clave	: string
nombre	: string
apellido	: string
sexo	: char
pais	: string
correo	: string
capacidad especial	: char
fecha nacimiento	: date
fecha registro	: date
perfil usuario	: perfil

Módulo: La clase Módulo contiene la información de los módulos de aprendizaje con los que cuenta la aplicación y esta es su estructura:

MÓDULO	
idmodulo	: int
	:
nombre modulo	string
	:
descripcionm	string

Ingreso: La clase Ingreso contiene la información acerca de los ingresos realizados por los usuarios a la aplicación.

INGRESO	
idingreso	: int
	:
idmodulo	: modulo
idusuario	: usuario
fechaingreso	: date

Unidad: La clase Unidad contiene la información de las unidades que se estudian en determinado módulo. La siguiente Figura muestra su estructura:

UNIDAD	
idunidad	: int
idmodulo	: modulo
nombreunidad	: string
descripcionu	: string

Contenido: En la clase Contenido se encuentra la información de los contenidos a estudiar dentro de una Unidad elegida y está estructurada de la siguiente manera:

CONTENIDO	
idcontenido	: int
idunidad	: unidad
nombrecontenido	: string
descripcioncontenido	: string

Tema: Esta clase contiene cada uno de los temas del contenido seleccionado para estudiarlo y aprender, su estructura es la siguiente:

TEMA	
idtema	: string
sonido	:string
texto	: string
imagen	: string
video	: string
idcontenido	: contenido
descripciontema	: string

Reporte: Esta clase contiene información de los reportes de notas obtenidas en cada evaluación realizada por el usuario, está estructurada de la siguiente manera:

REPORTE	
idreporte	: int
idusuario	: usuario
calificacion	: int
idunidad	: unidad

Pregunta: La clase Pregunta contiene todas las preguntas que aparecerán de manera randómica en las evaluaciones de cada una de las unidades estudiadas, esta es su estructura.

PREGUNTA	
idpregunta	: int
pregunta	: string
puntaje	: int
respuesta	: string
pathvideo	: string
idunidad	: unidad
imagen1	: string
imagen2	: string
imagen3	: string

Frases: Esta clase contiene las frases más utilizadas en el vivir diario, su estructura es:

FRASES	
idfrases	: string
idunidad	: unidad
nombrefrases	: string

descripcion	: string

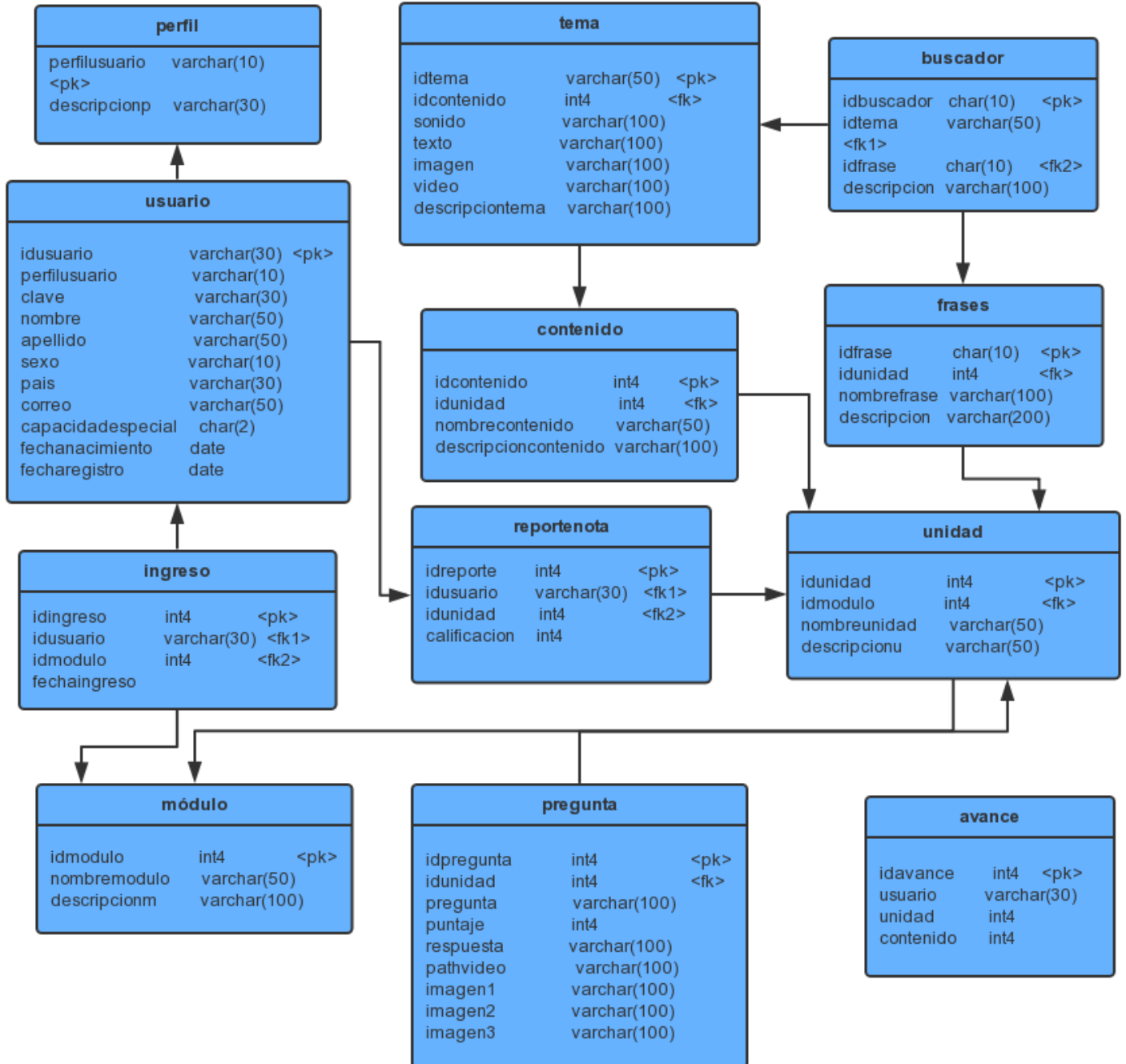
Buscador: En la clase Buscador se puede encontrar de forma rápida las palabras que se encuentran en el sistema, su estructura es la siguiente:

BUSCADOR	
idbuscador	: string
idtema	: tema
idfrases	: frases
descripcion	: string

Avance: Esta clase contiene información sobre contenidos estudiados en cada Unidad.

AVANCE	
idavance	: int
usuario	: string
unidad	: int
contenido	: int

DIAGRAMA ENTIDAD RELACIÓN



CODIFICACIÓN DE LAS PANTALLAS

El diseño de la aplicación se realizó con jsf y primefaces. A continuación se detalla el código de algunas pantallas.

Codificación

CÓDIGO DE LOS LOGOS DE LA APLICACIÓN

```
<div id="parallax" class="clear">
    <div class="parallax-layer" >
        
    </div>
    <div class="parallax-layer" >
        
    </div>
    <div class="parallax-layer">
        
    </div>
    <div class="parallax-layer">
        
    </div>
</div>
```



CÓDIGO PARA EL INGRESO A LA APLICACIÓN

```
public String verificar1(){

    System.out.println("usuario ANTES DE INGRESAR"+id.toUpperCase());
    try {

if(HibernateUtilDAO.buscausuario("Usuario","idusuario="+id.toUpperCase()+"", "")){
    System.out.println("el usuario existe"+id);
    Usuario c=(Usuario)HibernateUtilDAO.findByClavePrimaria(Usuario.class,
id.toUpperCase());
    id1 = c.getIdusuario().toString();
    System.out.println("id primario"+id1);
    pass1 = c.getClave().toString();
    System.out.println("pass primario"+id1);
    if (id.toUpperCase().equals(id1) && pass.equals(pass1)) {
    System.out.println("comparacion");
    perfil = c.getPerfil().getPerfilusuario();
    if (perfil.equals("ADM")) {
        usuario = id1;
        System.out.println("el usuario con mayusculas"+id1);
        int registros = 0;
        Object reg = 0;
        try {
            reg = HibernateUtilDAO.maximo();
```

```
        registros = Integer.parseInt(reg.toString());
    } catch (Exception ex) {
        Logger.getLogger(login.class.getName()).log(Level.SEVERE, null, ex);
    }
    System.out.println("numero de registros de ingresos" + registros);

    Usuario u = new Usuario();
    System.out.println(login.usuario);
    try {
        u = (Usuario) HibernateUtilDAO.findByClavePrimaria(Usuario.class,
login.usuario);
    } catch (Exception ex) {

Logger.getLogger(adminitracionUsuarios.class.getName()).log(Level.SEVERE, null, ex);
    }

    Ingreso in = new Ingreso();
    in.setIdingreso(registros + 1);
    Modulo m = new Modulo();
    try {
        m = (Modulo) HibernateUtilDAO.findByClavePrimaria(Modulo.class, 3);
    } catch (Exception ex) {

Logger.getLogger(adminitracionUsuarios.class.getName()).log(Level.SEVERE, null, ex);
    }
    in.setModulo(m);
    in.setUsuario(u);
    System.out.println(fechaingreso + "akiiiiiiiiiiiiiiiiiiii");
    in.setFechaingreso(fechaingreso);
    try {
```

```
        HibernateUtilDAO.insertarObjeto(in);//INSERTA INGRESO DE
USUARIO AL MODULO
    } catch (Exception ex) {

Logger.getLogger(adminitracionUsuarios.class.getName()).log(Level.SEVERE, null, ex);
    }

    return "AdminBienvenida";
}
usuario = id1;
HttpServletRequest request = (HttpServletRequest) FacesContext

.getCurrentInstance().getExternalContext().getRequest();
        HttpSession sesionOk = request.getSession();
        sesionOk.setAttribute("usuario", usuario);
        sesionOk.setMaxInactiveInterval(3000);
        Object soy=sesionOk.getAttribute("usuario");
        System.out.println("ingresó:"+soy);
    return "loginCliente_1";
}
}
} catch (Exception ex) {
    Logger.getLogger(login.class.getName()).log(Level.SEVERE, null, ex);
}

    System.out.println("el usuario No existe");
    FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage(FacesMessage.SEVERITY_INFO, "Sample info message", "DATOS
Incorrecto"));

return "index";
```

```
}
```

```
<h:form style="width: 100%; height: 450px;background-repeat: no-repeat;background-size:cover; background-position: bottom">
```

```
<center> <h:commandButton image="../imagenes/palo.png" style=" height:371px; width:235px " onmouseover="this.src='imagenes/palo1.png';" onmouseout="this.src='imagenes/palo.png'" ></h:commandButton></center>
```

```
<h:commandButton class="btnmod" image="/imagenes/mb.png" action="#{administracionUsuarios.avanzaModuloBasico(1)}" style="position: absolute;left: 41%;z-index: 2;height:67px; width:199px; top:371px"/>
```

```
<h:commandButton class="btnmod" image="/imagenes/ma.png" action="#{administracionUsuarios.avanzaModuloAvanzado(2)}" style="position: absolute;left: 39%;z-index: 2;height:67px; width:199px; top:447px"/>
```

```
</h:form>
```



CÓDIGO PARA IR A UNA UNIDAD

```
public String iracontenido(int contenido) { //direcciona a los subcontenidos del contenido seleccionado
```

```
HttpServletRequest request = (HttpServletRequest) FacesContext
    .getCurrentInstance().getExternalContext().getRequest();
HttpSession sesionOk = request.getSession();
Object soy = sesionOk.getAttribute("usuario");
System.out.println("yo soy:" + soy);
sesionOk.setAttribute("estudiando contenidos", soy);
sesionOk.setMaxInactiveInterval(3000);
try {
    activacertificado(1);
    activarbotonevaluar(contenido);
} catch (Exception ex) {
    Logger.getLogger(adminitracionUsuarios.class.getName()).log(Level.SEVERE,
null, ex);
}
return direccionarunidad(contenido);
}
```

```
<p:panel style="background: #BDBDBD; width: 80px; height: 120px; border:
transparent; top: 20px; margin-top: 20px;left: 100px; margin-left: 100px; border-radius:
10px ">
```

```
<h:commandButton class="boton" alt="Geografía" title="Geografía"
image="/resources/imagenes/iconos/geoico.png"
onmouseover="style.backgroundColor='#0078ff'"
onmouseout="style.backgroundColor='white'"
style="background: white; height: 100px; cursor: pointer;
top: -5px; margin-top: -5px;border: transparent;
width: 56px; left: -12px; margin-left: -12px"
action="#{adminitracionUsuarios.iracontenido(8)}" >
</h:commandButton> </p:panel>
```



CÓDIGO PARA MOSTRAR LOS TEMAS

```
public String mostrar25(String t) {
    HttpServletRequest request = (HttpServletRequest) FacesContext
        .getCurrentInstance().getExternalContext().getRequest();
    HttpSession sesionOk = request.getSession();
    Object soy = sesionOk.getAttribute("usuario");
    System.out.println("yo soy:" + soy);
    sesionOk.setAttribute("usuario", soy);
    sesionOk.setMaxInactiveInterval(3000);
    Tema tem = null;
    try {
        tem = (Tema) HibernateUtilDAO.findByClavePrimaria(Tema.class, t);
    } catch (Exception ex) {
        Logger.getLogger(tema.class.getName()).log(Level.SEVERE, null, ex);
    }
    codigo = tem.getIdtema();
    sonido = quitaespacios(tem.getSonido());
    texto = tem.getTexto();
    imagen = tem.getImagen();
    video = quitaespacios(tem.getVideo());
    System.out.println("..." + video + "...");
}
```



```
contenido = tem.getContenido().getIdcontenido();  
descripcion = tem.getDescripciontema();  
  
return "paises";  
  
}
```

```
<h:commandButton class="botong" value="Panamá"  
action="#{tema.mostrar25('PANAMA')}}" style="background:  
url('../resources/css/blue3.png') center; "/>
```



CODIGO ACTIVA EL BOTÓN EVALUAR

```
public void activarbotonevaluar(int unidad) throws Exception {  
    HttpServletRequest request = (HttpServletRequest) FacesContext  
        .getCurrentInstance().getExternalContext().getRequest();
```

```
HttpSession sesionOk = request.getSession();
Object soy = sesionOk.getAttribute("usuario");
System.out.println("yo soy:" + soy);
sesionOk.setAttribute("activando boton evaluar para", soy);
sesionOk.setMaxInactiveInterval(3000);
if ((HibernateUtilDAO.buscausuario("Avance", "usuario=" + soy.toString() + "",
""))) {
// if((HibernateUtilDAO.buscausuario("Avance", "usuario="+login.usuario+"", "")))
// {
    try {
        listacontenidoporusuario = HibernateUtilDAO.findByFiltro("Avance",
"unidad=" + unidad + " and usuario=" + soy.toString() + "", "");
// listacontenidoporusuario=HibernateUtilDAO.findByFiltro("Avance",
"unidad="+unidad+" and usuario="+login.usuario+"", "");
//
        System.out.println("lista de contenidos x unidad " + unidad + " del usuario " +
soy.toString() + "eeees:" + listacontenidoporusuario.size());
    } catch (Exception ex) {

Logger.getLogger(adminitracionUsuarios.class.getName()).log(Level.SEVERE, null, ex);
    }
    try {
        listacontenidosporunidad = HibernateUtilDAO.findByFiltro("Contenido",
"idunidad=" + unidad + "", "");
    } catch (Exception ex) {

Logger.getLogger(adminitracionUsuarios.class.getName()).log(Level.SEVERE, null, ex);
    }
    if (listacontenidoporusuario.size() == listacontenidosporunidad.size()) {
//// estado=true;
        estado = false;
    }
}
```

```
        System.out.println("estado de boto evaluar habilitado " + estado);
    } else {
//        estado=false;
        estado = true;
        System.out.println("estado de boto evaluar1 deshabilitado " + estado);
    }
}
}
```

```
<h:commandButton value="evaluar" title="Para evaluar sus conocimientos debe estudiar
todos los contenidos" action="#{evaluacion.irEvaluacionM(8)}" alt="Evaluar"
image="../resources/imagenes/geografia1/evag.png"
style="height: 150px; top: 50px; margin-top: 50px"
disabled="#{administracionUsuarios.estado}"
onmouseover="this.src='../resources/imagenes/geografia1/evag1.png';"
onmouseout="this.src='../resources/imagenes/geografia1/evag.png';"></h:commandButton
>
```



CÓDIGO DEL BUSCADOR

```
public void buscarpalabra() throws Exception{
    HttpServletRequest request = (HttpServletRequest) FacesContext
        .getCurrentInstance().getExternalContext().getRequest();
    HttpSession sesionOk = request.getSession();
    Object soy = sesionOk.getAttribute("usuario");
    System.out.println("yo soy:" + soy);
    sesionOk.setAttribute("usuario", soy);
    sesionOk.setMaxInactiveInterval(3000);
    System.out.println("FUERA");
    System.out.println("palabra:"+palabra);
    if (palabra!= null){
        if((HibernateUtilDAO.buscador("Tema", "idtema="+palabra.toUpperCase()+"", "")))
        {
            System.out.println("entra");

t=HibernateUtilDAO.findByFiltro("Tema", "idtema="+palabra.toUpperCase()+"", "");
            muestra="visible";
            imagen=t.get(0).getImagen();
            texto=t.get(0).getTexto();
            video=quitaespacios(t.get(0).getVideo());
            sonido=quitaespacios(t.get(0).getSonido());

            System.out.println("pruebaaa"+imagen+ texto+video+sonido);
        }
    }
    else
    {

        JOptionPane.showMessageDialog(null, "palabra no Existe");
        muestra="hidden";
    }
}
```

```
}  
}else  
{  
  
    JOptionPane.showMessageDialog(null, "palabra no Existe");  
    muestra="hidden";  
}  
}
```

```
<h:commandButton value="BUSCAR" styleClass="boton grande blue"  
action="#{buscadorAvanzado.buscarpalabra()}"  
style="height: 30px; width: 100px; font-size: 13px; font-weight: bold;  
color:black;  
cursor: pointer ">  
</h:commandButton>
```



AL USUARIO FINAL

Con este manual técnico se pretende que sea fácil de usar, claro para que los programadores, diseñadores, analistas de software deseen realizar alguna acción descrita anteriormente al sistema