

# Capítulo 5

## 5.1. Referencia Virtual Afectivo v 2.0

El objetivo del sistema de Referencia Virtual que se encuentra dentro de la Biblioteca de la Universidad de las Américas, es probar la participación de los usuarios y referencistas a través de un medio digital, sin embargo, este proyecto no contiene componentes que el ayuden a probar esa interactividad, entre usuarios y referencistas.

Como comentamos anteriormente, se toma como base la idea general de este sistema, pero se le han añadido y eliminado algunas funciones. A continuación se muestra la Tabla 5.1 la cual presenta los cambios que se ha realizado en el sistema de Referencia Virtual y dando como resultado Referencia Virtual Afectivo versión 2.0

	Referencia Virtual	Referencia Virtual versión 2.0
1	Usuarios en línea (sistema asíncrono)	Usuarios en línea con oportunidad de platicar con otro usuario (sistema síncrono)
2	Ver preguntas, Ver respuestas	Integraran en la interfaz principal
3	Actualizar Datos	Actualizar Mail
4		Calificar Respuesta

Tabla 5.1 Muestra de manera general los servicios con los que cuenta el Sistema de Referencia Virtual y los nuevos componentes que se desea que contenga el Sistema de Referencia Virtual v. 2.0 a fin de que se cumpla la afectividad en esta interfaz.

## 5.2 Análisis del Sistema Referencia Virtual Afectivo versión 2.0.

En este apartado veremos el análisis que se llevó a cabo para la formación del Sistema de Referencia Virtual Afectivo versión 2.0

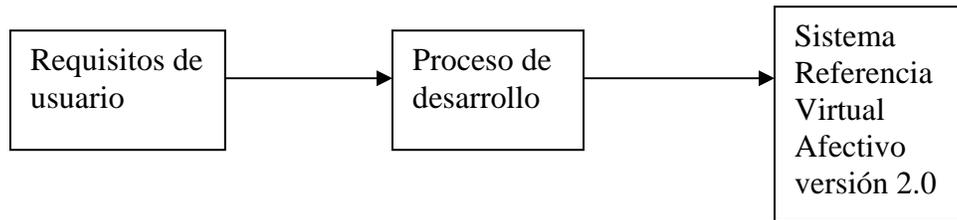


Figura 5.1 Análisis del Sistema Referencia Virtual Afectivo versión 2.0

### 5.2.1 Requisitos: Panorama General

Este proyecto tiene por objeto crear un sistema que contenga interfaces afectivas que serán utilizadas dentro del contexto de Bibliotecas Digitales U\_DL\_A. Logrando así que las interfaces afectivas promuevan una mayor interacción entre usuarios y bibliotecarios Referencistas.

#### Funciones del Sistema

- Usuario puede:
  - \* ingresar al sistema
  - \* hacer preguntas
  - \* pertenecer al grupo de bibliotecarios referencistas
  - \* cambiar su correo electrónico, la información llegue al correo que use con mayor frecuencia.

\*conversar usuarios en línea

- Bibliotecario referencista puede:
  - \* contestar preguntas de usuarios
  - \*conversar usuarios en línea.

Dentro de las funciones del sistema algunas de ellas son evidentes para los usuarios, mientras otras son ocultas. A continuación se muestra la Tabla 5.2 donde se especifican cada una de las funciones con su categoría respectiva

#### Funciones del Sistema

Número	Funciones del sistema	Categoría
1	El usuario ingresa al sistema.	Evidente
2	Registro entrada al sistema del usuario y se agrega en la base como usuario en línea.	Oculto
3	El usuario puede enviar preguntas y también puede ver otras preguntas hechas por otros usuarios.	Evidente
4	Almacena pregunta en la base de datos realizadas por el usuario.	Oculto
5	El usuario puede agregarse al grupo de bibliotecarios referencistas.	Evidente
6	El usuario puede cambiar su correo electrónico.	Evidente
7	Se registran los cambios que el usuario ha realizado mientras interactúa con el sistema, como el agregarse al grupo de bibliotecarios referencistas y el cambio de correo electrónico.	Oculto
8	El usuario tiene la oportunidad de comunicarse con otros usuarios que están en línea.	Evidente
9	Establecer comunicación con el servidor que subministra el servicio de chat.	Oculto
10	El bibliotecario referencista puede responder preguntas hechas por usuarios.	Evidente
11	Registro de preguntas constestadas y sin contestar.	Oculto
12	Salida del sistema del usuario.	Evidente
13	Cambio de estado de usuario cuando ha salido del sistema	Oculto

Tabla 5.2 Funciones del Sistema Refencia Virtual Afectiva versión 2.0, asociada con su categoría

### **5.3 Diagrama de contexto**

El digrama que se muestra a continuación (Tabla 5.3) determina de forma general como los usuarios interactúan con el sistema y con los bibliotecarios, obteniendo como salida la comunicación afectiva entre ellos.

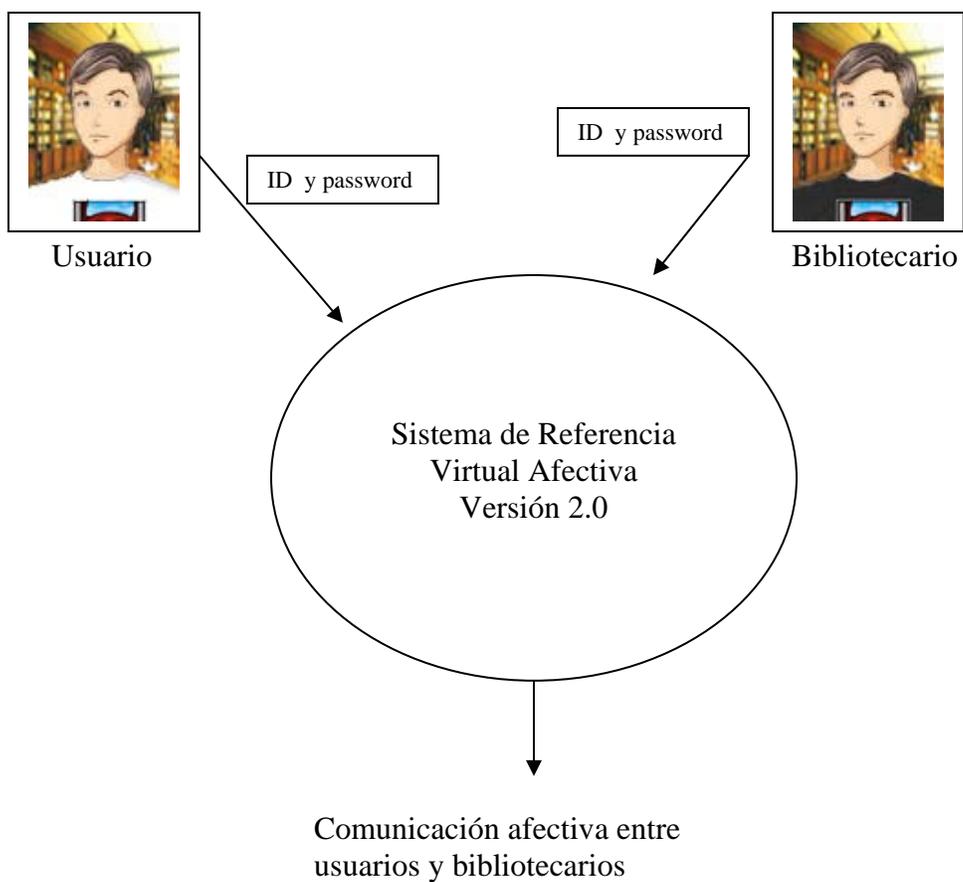


Figura 5.2 Diagrama de Contexto de Referencia Virtual Afectiva versión 2.0. Muestra el contexto general del sistema.

## 5.4 Casos de Uso

### 5.4.1 Diagrama de Caso de Uso

Este diagrama muestra como el usuario interactúa con los servicios que proporciona el sistema.

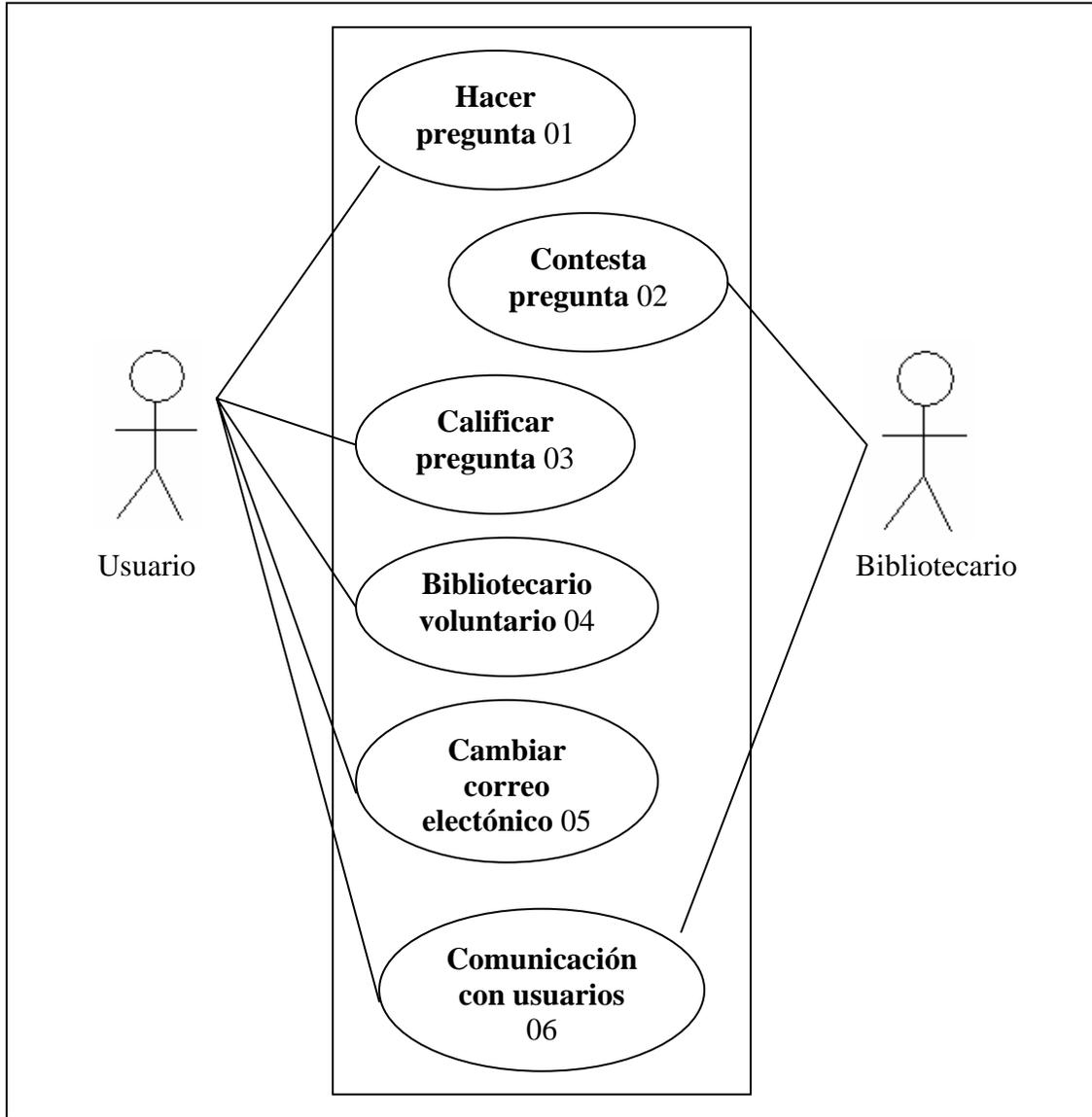


Figura 5.3 Diagrama general de caso de uso del Sistema de Referencia Virtual Afectiva versión 2.0. Muestra interacción entre usuario, el sistema y el bibliotecario

### 5.4.2 Casos de Uso

#### Caso de Uso: “Hacer Pregunta”

<b>N°:</b>	01
<b>Autor:</b>	Blanca Leticia Tecuanhuehue Calderón

<b>Participantes:</b>	Usuario
<b>Precondiciones:</b>	Ninguna
<b>Postcondiciones:</b>	Almacena pregunta en el sistema
<b>Descripción:</b>	El usuario decide hacer una pregunta, para ello se necesita llenar una forma, en la que indique la pregunta, el tema y el periodo de búsqueda.

### Caso de Uso: “Contesta Pregunta”

<b>N°:</b>	02
<b>Autor:</b>	Blanca Leticia Tecuanhuehue Calderón
<b>Participantes:</b>	Usuario, Bibliotecario Referencista
<b>Precondiciones:</b>	Pregunta almacenada en el sistema
<b>Postcondiciones:</b>	Almacena respuesta en el sistema
<b>Descripción:</b>	El bibliotecario revisa preguntas que se han hecho y contesta las preguntas, las cuales se muestran a los usuarios para que sean calificadas por los usuarios que hicieron la pregunta

### Caso de Uso: “Calificar Pregunta”

<b>N°:</b>	03
<b>Autor:</b>	Blanca Leticia Tecuanhuehue Calderón
<b>Participantes:</b>	Usuario
<b>Precondiciones:</b>	Respuesta almacenada en el sistema
<b>Postcondiciones:</b>	Ninguna
<b>Descripción:</b>	El usuario al recibir la respuesta de su pregunta, puede calificar la respuesta. Esto se hace con la finalidad de conocer si la respuesta fue la esperada, si fue así se reconoce el desempeño del bibliotecario o viceversa.

### Caso de Uso: “Bibliotecario Voluntario”

<b>N°:</b>	04
------------	----

<b>Autor:</b>	Blanca Leticia Tecuanhuehue Calderón
<b>Participantes:</b>	Usuario
<b>Precondiciones:</b>	Respuesta almacenada en el sistema
<b>Postcondiciones:</b>	Ninguna
<b>Descripción:</b> El usuario al recibir la respuesta de su pregunta, puede calificar la respuesta. Esto se hace con la finalidad de conocer si la respuesta fue la esperada, si fue así se reconoce el desempeño del bibliotecario o viceversa.	

#### Caso de Uso: “Cambiar correo electrónico”

<b>N°:</b>	05
<b>Autor:</b>	Blanca Leticia Tecuanhuehue Calderón
<b>Participantes:</b>	Usuario
<b>Precondiciones:</b>	Ninguna
<b>Postcondiciones:</b>	Ninguna
<b>Descripción:</b> Si el usuario quiere participar como Bibliotecario Voluntario, tiene la oportunidad de hacerlo, lo que se necesita es que especifique en que área está especializado o tiene mayor conocimiento	

#### Caso de Uso: “Comunicación con usuarios”

<b>N°:</b>	06
<b>Autor:</b>	Blanca Leticia Tecuanhuehue Calderón
<b>Participantes:</b>	Usuario, otros usuarios y Bibliotecario Referencista
<b>Precondiciones:</b>	Al menos dos usuarios estén conectados
<b>Postcondiciones:</b>	Ninguna
<b>Descripción:</b> el usuario puede observar que hay uno o varios usuarios conectados, puede ver la información de cada uno de los participantes, además de poder invitarlos a comunicarse por medio del chat.	

## 5.5 Matriz de funcionalidad

En la siguiente tabla se definen las funciones que desarrollan en el sistema de Referencia Virtual Afectiva versión 2.0. En los renglones se encuentran las entidades y en las columnas la función que realiza cada una de ellas.

Concepto	1	2	3	4	5	6
Usuario	Registro usuario	Registrar pregunta	Referencista voluntario	Actualizar mail	Calificar respuesta	* comunicars e con otros usuarios
Bibliotecario referencista	Ingresar al sistema	Ejecutar contestar	* comunicarse con otros usuarios	Ver calificaciones de las respuestas		

Tabla 5.3 Matriz de funcionalidad

## 5.6 Diagramas de actividades

A continuación se presentará los diagramas (Figura 5.4 - Figura 5.10) de las entidades definidas anteriormente para la generación del nuevo sistema de Referencia Virtual Afectivo versión 2.0. Específicamente en la Figura 5.7 observamos la itegración de los compoentes afectivos que hemos desarrollado, los cuales se describen posteriormente. Cabe destacar que es en este proceso (Calificar Respuesta) donde se prueba la afectividad del sistema, pues permitir una retroalimentación y dependiendo de su desempeño afectará significativamente el comportamiendo del bibliotecario referencista.

### 1.1 Registro de usuario

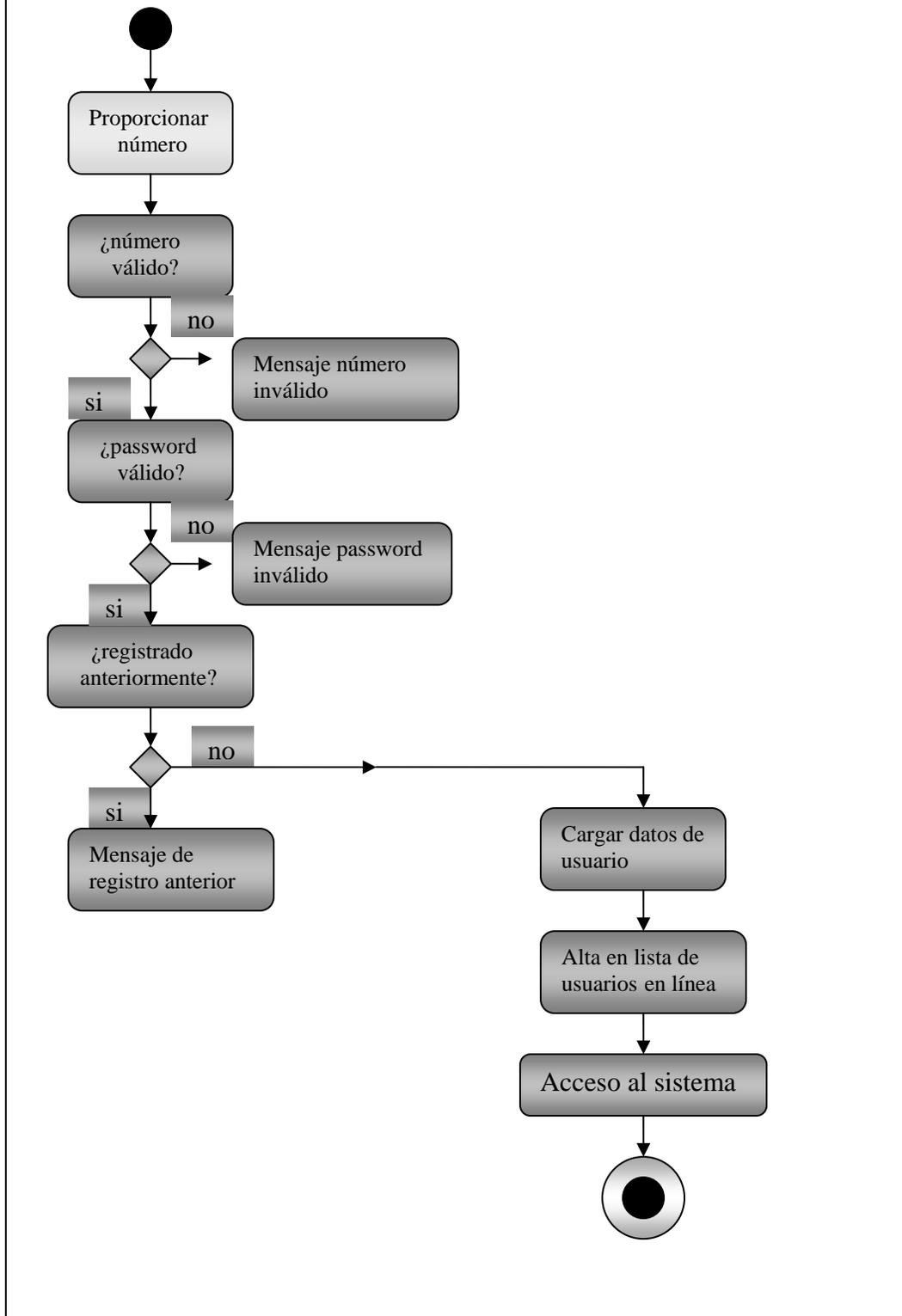


Figura 5.4 Registro del usuario

## 1.2 Registrar Pregunta

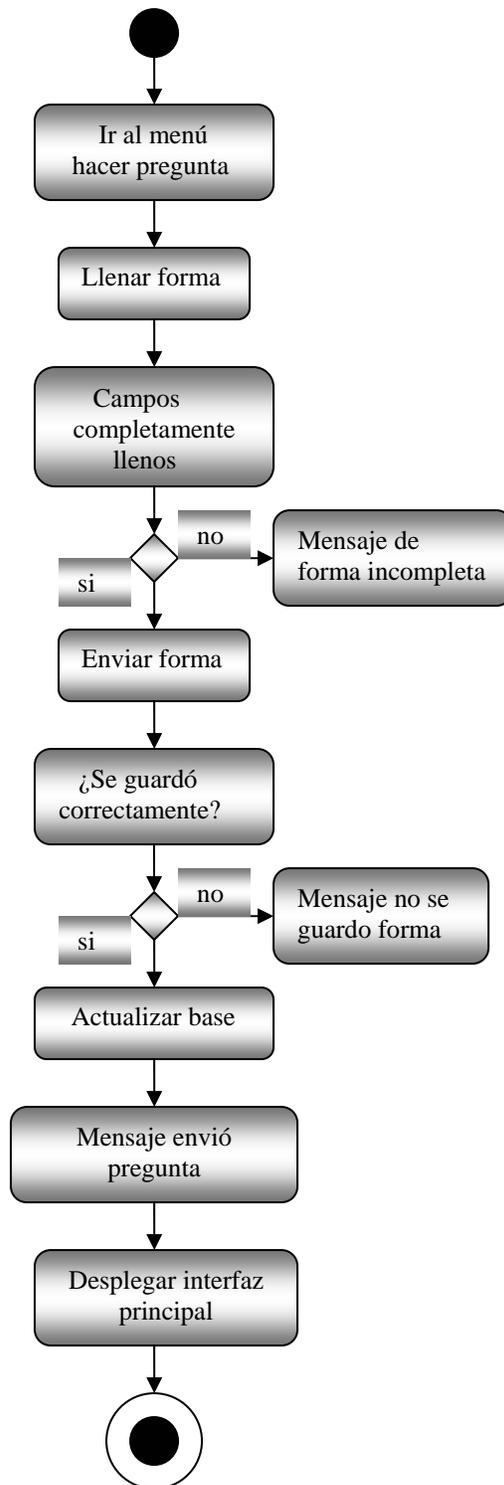


Figura 5.5 Registrar pregunta

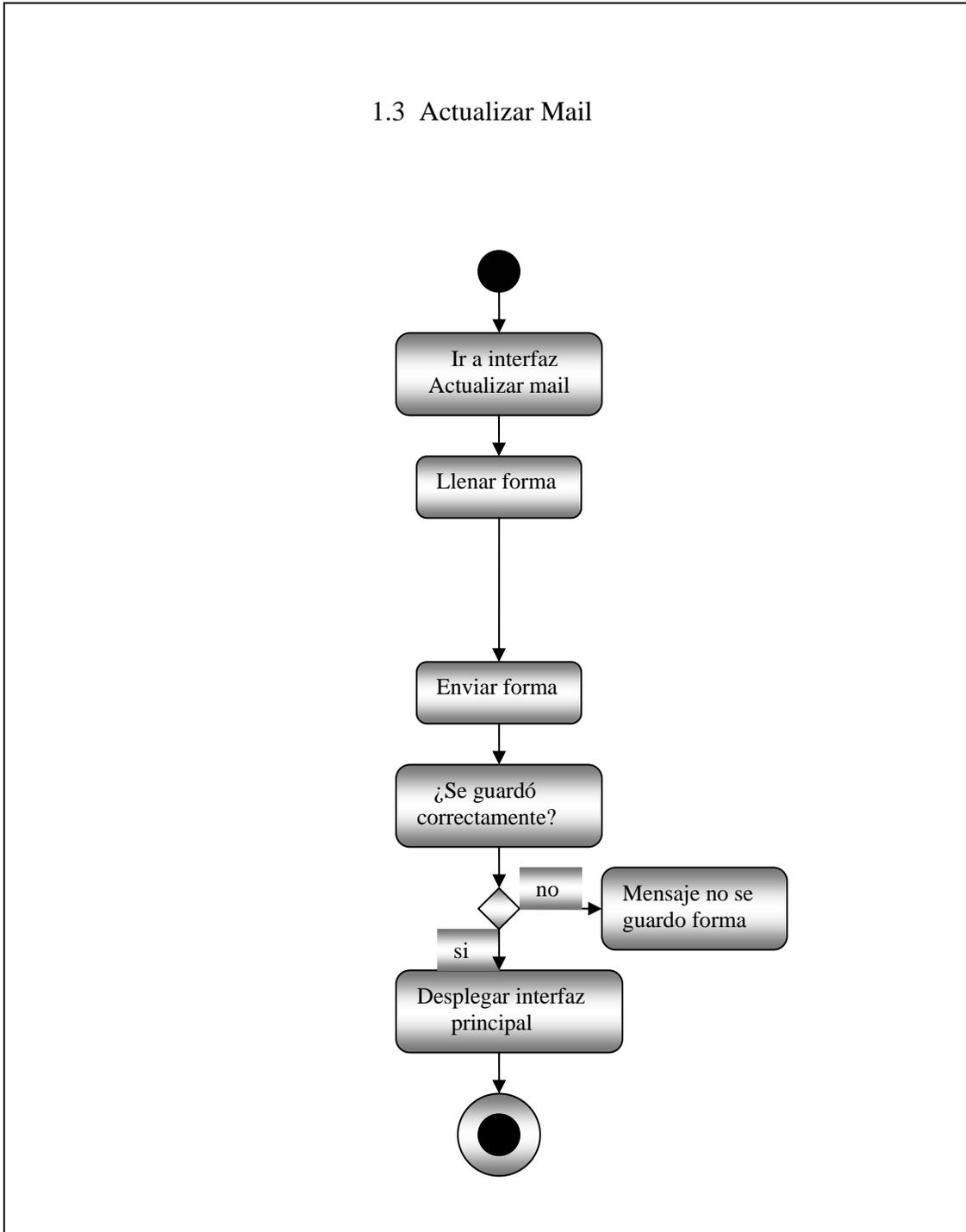


Figura 5.6 Actualiza Mail

#### 1.4 Calificar Respuesta

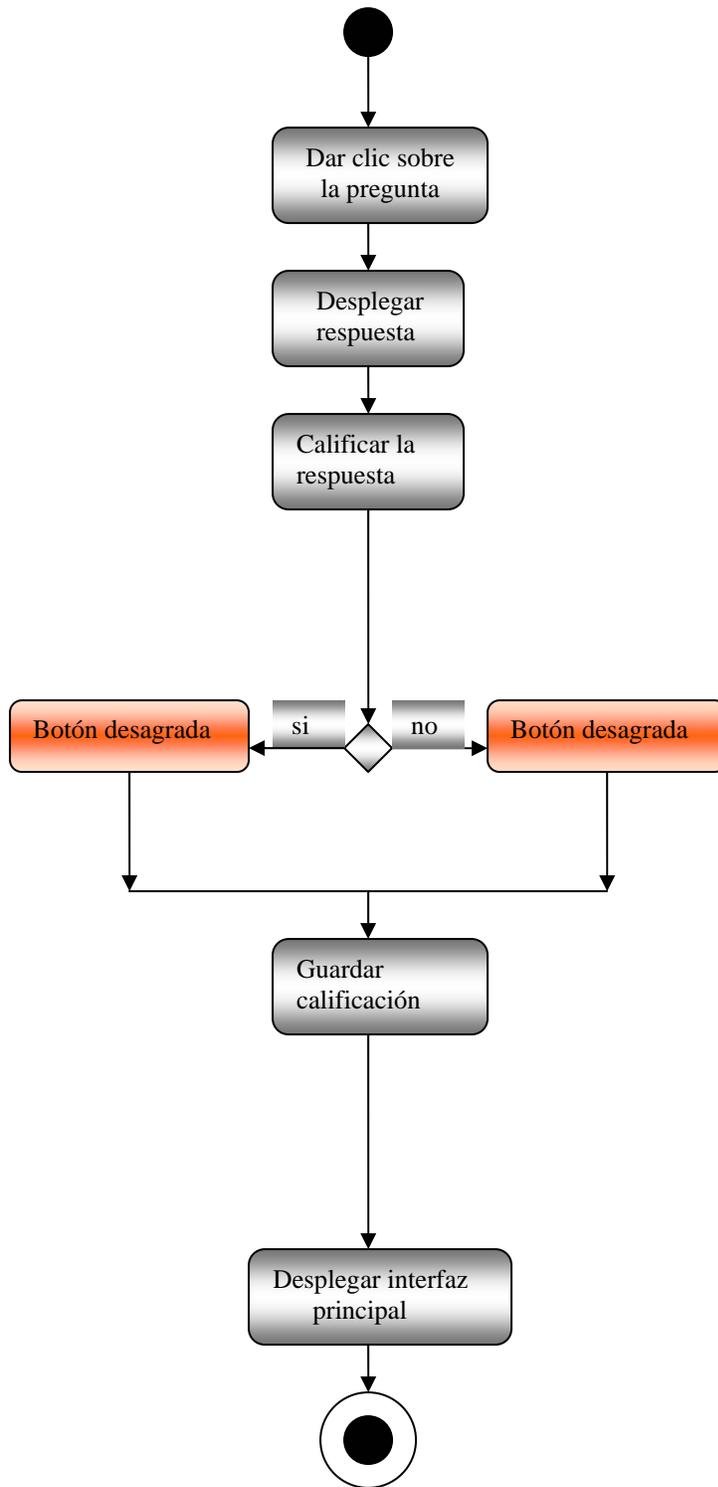


Figura 5.7 Califica Respuesta

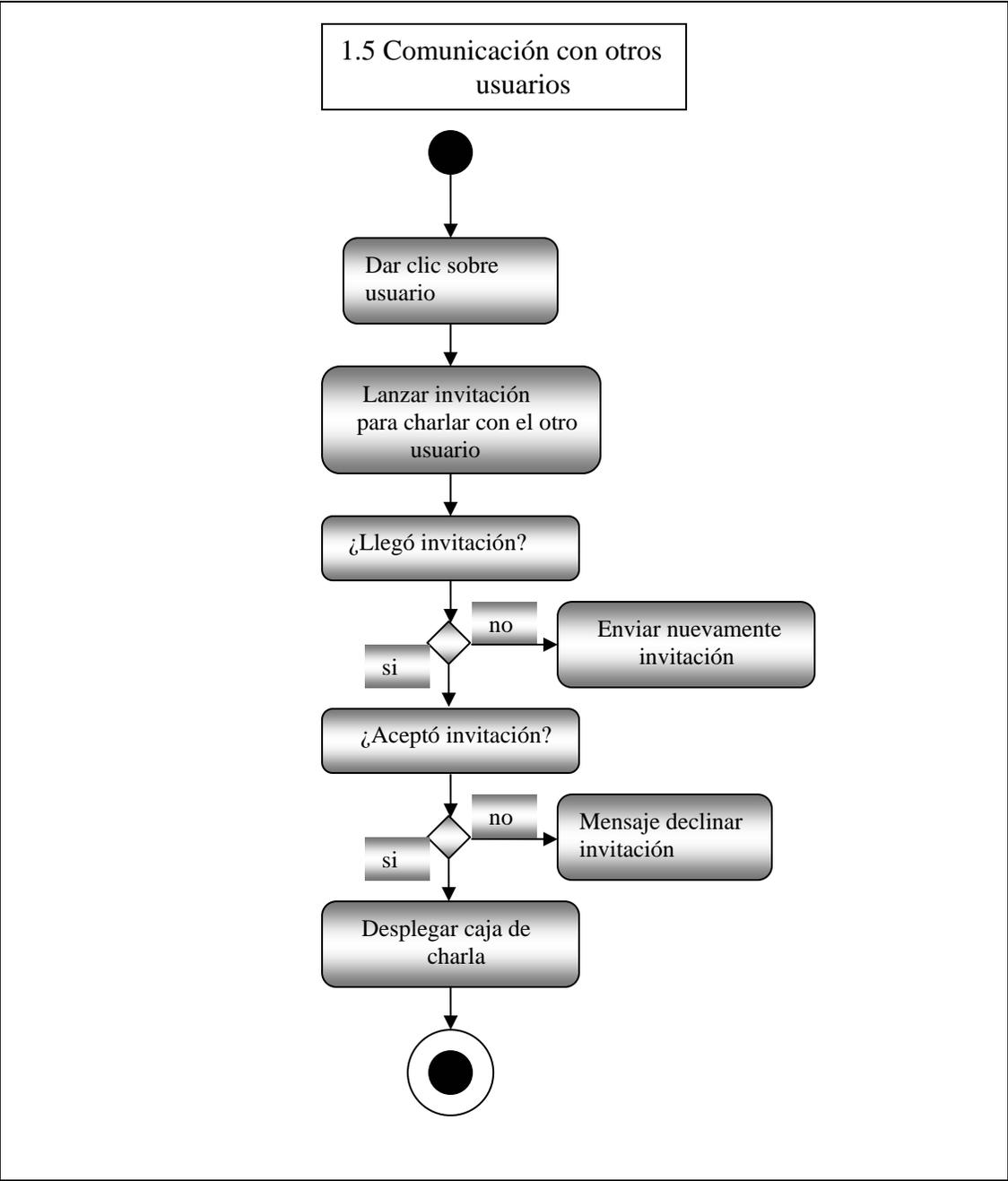


Figura 5.8 Comunicación otros usuarios

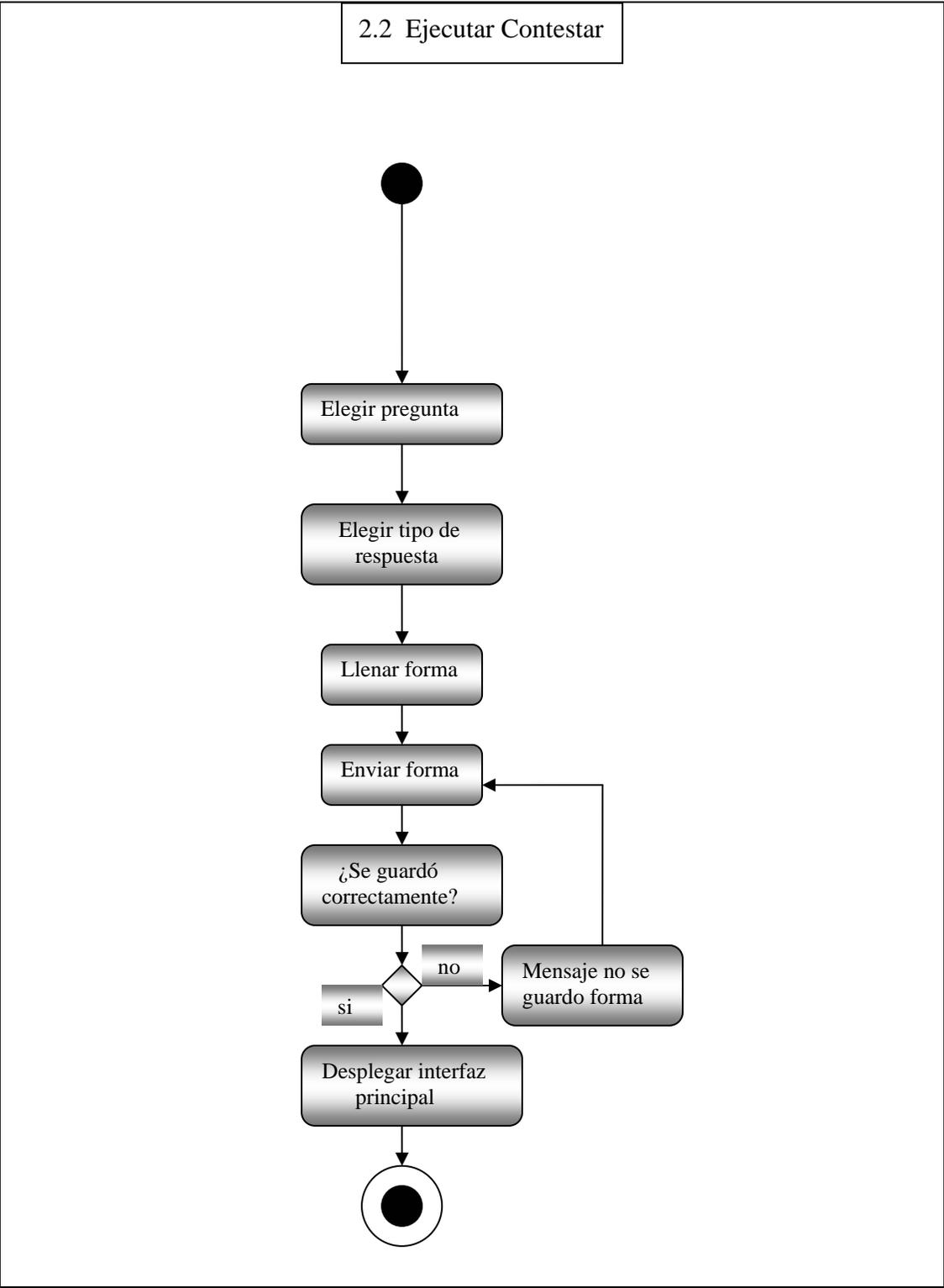


Figura 5.9 Ejecuta contestar

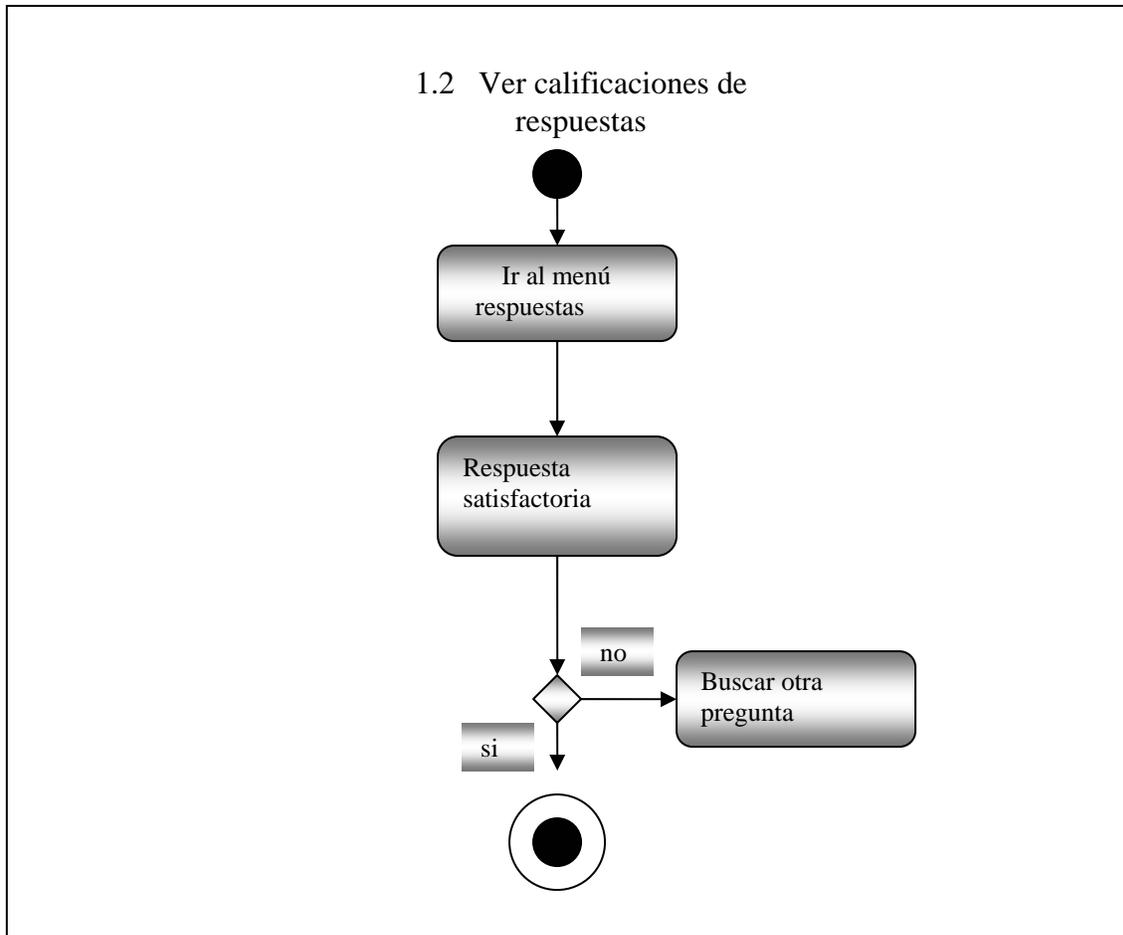


Figura 5. 10 Ver calificación de respuestas

## 5.7 Mapeo de emociones del modelo OCC

De las categorías emocionales del modelo OCC, consideramos que las más apropiadas para integrarlas a nuestro prototipo serían las referentes a las acciones de agentes de software o usuarios, las que son de atribución para otro agente, (admiración y reproche) y para el mismo agente (orgullo y vergüenza).

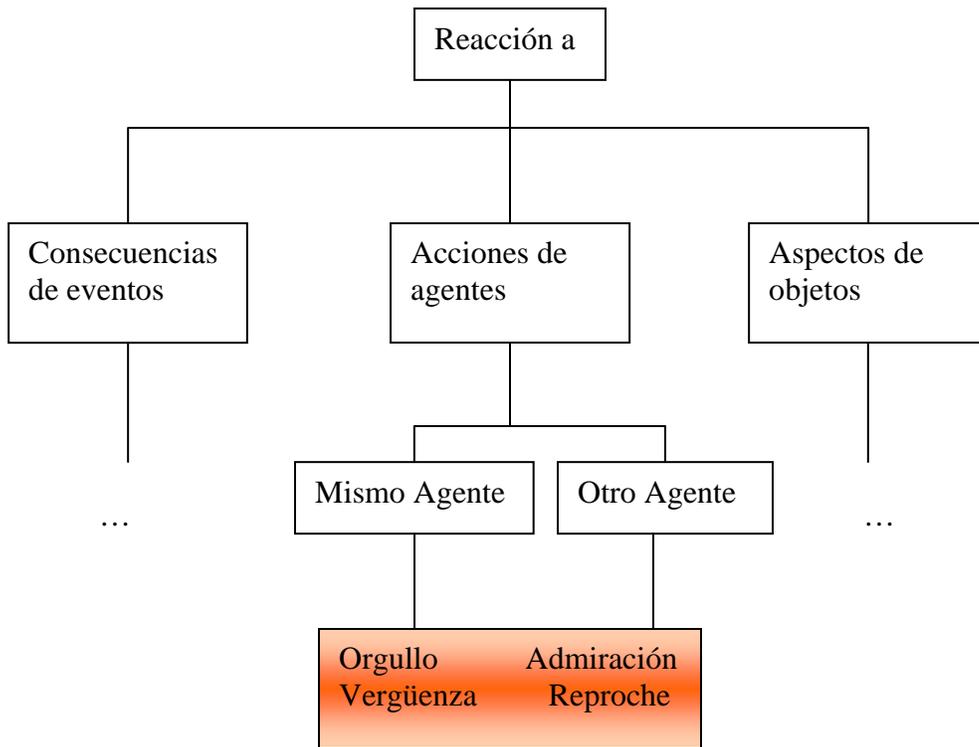


Figura 5.11 Emociones específicas del modelo OCC para integrarlas en el Sistema de Referencia Virtual Afectivo versión 2.0

Este tipo de emociones pueden presentarse cuando un usuario envía una pregunta, el bibliotecario referencista proporciona alguna respuesta, y esta respuesta puede ser calificada, provocando así el refuerzo o la reprogramación del comportamiento del agente o bibliotecario.

A continuación se muestran dichos componentes.

### **Admiración (¡Qué bien lo hiciste!)**

/\*\* Esta clase dibuja el componente "Admiración".

```

* */

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Boton_Amiracion extends JPanel
{
    public Boton_Amiracion() {
        int comportamiento =0;
        /** Recupera imagen. */

        ImageIcon presion = new ImageIcon( "admiracion.jpg" );

        /** Dibuja boton */

        JButton boton = new JButton( "Qué bien lo hiciste", presion);

        boton.setSize(200,200);
        boton.setPressedIcon( presion );
        boton.setRolloverEnabled( true );
        add( boton );

        /**
         * Este método especifica como responder a las acciones del usuario,
         * en este caso para el botón Admiración.
         *
         * @param e El evento registrado.
         */
        boton.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
                System.out.println("Admiracion");
                comportamiento ++;
                if (comportamiento > 0)
                    System.out.println("El comportamiento del agente fue satisfactorio");

            }
        });
    }

    /**
     * Este es el método inicializador, dibuja el contenedor, en este caso el Panel
     * y la ventana. */

    public static void main( String args[] ) {

```

```

Boton_Admiracion panel = new Boton_Admiracion();
JFrame ventana = new JFrame();
ventana.getContentPane().add( panel, BorderLayout.CENTER );
ventana.addWindowListener( new WindowAdapter() {

    public void windowClosing( WindowEvent evt ) {
        System.exit( 0 );    }
    });
ventana.setSize( 270,160 );
ventana.setTitle( "Admiración" );
ventana.setVisible( true );
}
}

```

### **Reproche (¡Qué mal lo hiciste!)**

```

/** Esta clase dibuja el componente "Reproche".
 * Dentro del modelo OCC atribución es la respuesta que se puede tener
 * para con otro agente, desaprobando su comportamiento, para reprogramar el
 * agente.
 *
 * @author Blanca L. Tecuanhuehue C.
 * @version 1.0*/

```

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Boton_Reproche extends JPanel
{
    public Boton_Admiracion() {

        /** Recupera imagen. */

        ImageIcon presion = new ImageIcon( "reproche.jpg" );

        /** Dibuja boton */

        JButton boton = new JButton( "Qué mal lo hiciste", presion);

        boton.setSize(200,200);

```

```

boton.setPressedIcon( presion );
boton.setRolloverEnabled( true );
add( boton );

/**
 * Este método especifica como responder a las acciones del usuario,
 * en este caso para el botón Admiración.
 *
 * @param e El evento registrado.
 */
boton.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        System.out.println("Reproche");
        comportamiento ++;
        if (comportamiento == 0)
            System.out.println("El comportamiento del agente fue insatisfactorio,
                reprogramar al agente");

    }
});
}

/**
 * Este es el metodo inicializador, dibuja el contendor, en este caso el Panel
 * y la ventana. */

public static void main( String args[] ) {
    Boton_Admiracion panel = new Boton_Reproche();
    JFrame ventana = new JFrame();
    ventana.getContentPane().add( panel, BorderLayout.CENTER );
    ventana.addWindowListener( new WindowAdapter() {

        public void windowClosing( WindowEvent evt ) {
            System.exit( 0 );
        }
    });
    ventana.setSize( 270,160 );
    ventana.setTitle( "Reproche" );
    ventana.setVisible( true );
}
}

```

De igual forma se desarrollan los componentes de reacción hacia el mismo agente,

Orgullo y Vergüenza.

## 5.8 Clases

Las clases que componen el Sistema de Referencia Afectiva versión 2.0, son las siguientes.

La clase AccesoPrincipal.java abre la conexión a la base de datos Vref, recoge los datos proporcionados por el usuario (id y password), y da de alta el usuario en la tabla de usuarios en línea a menos que el usuario ya haya sido dado de alta, en este caso manda un aviso de “usuario anteriormente dado de alta”. Por otro lado hace una búsqueda de las preguntas realizadas hasta el momento, llama al método que cierra la conexión a la base y finalmente manda a la paginaPrincipal.jsp esta información.

```
AccesoPrincipal.java
```

```
bandera: boolean  
nombre: String  
id_usuario: String  
pregunta: Vector  
activo: Vector  
id_pregunta: Vector  
cues: Queries
```

```
doPost(HttpServletRequest req,  
HttpServletResponse res)  
fuera(HttpServletRequest req,  
HttpServletResponse res)  
cierra()
```

AlmacenaPregunta.java, en primer lugar esta clase hace la apertura de la base y recupera el id del usuario que desea enviar una pregunta, una vez hecho esto, almacena los siguientes parámetros, pregunta, tema, (otro tema, si es que no esta especificado en tema),

fecha inicial y final del periodo de búsqueda, se envía a la base y a esta pregunta se le asigna una clave, si no hubo error en el almacenamiento de la pregunta, se despliega una página informativa del envío correcto de la pregunta, se cierra la conexión a la base, y finalmente regresa a la paginaPrincipal.jsp

```
AlmacenaPregunta.java

bandera: boolean
id: String
pregunta: String
tema: String
otro_tema: String
fecha_inicial: String
fecha_final: String
clave: String
pregunta: Vector
cues: Queries

doPost(HttpServletRequest req,
HttpServletResponse res)
fuera(HttpServletRequest req,
HttpServletResponse res)
cierra()
```

ActualizaMail.java, abre la conexión a la base para modificar el correo electrónico del usuario que desea actualizarlo. Para ello recupera el id del usuario, y el nuevo mail proporcionado, por último escribe en la base el nuevo correo y cierra la conexión a la base.

```
ActualizaMail.java
```

```
id: String  
userId: String  
correo: String  
userName: String  
carrera: String  
mail: String
```

```
doPost(HttpServletRequest req,  
HttpServletResponse res)  
fuera(HttpServletRequest req,  
HttpServletResponse res)  
cierra()
```

Chat.java, cuando un usuario ha detectado que hay algún usuario con el cual desea platicar, lo invita, esta clase recupera el id del usuario para ingresarlo en el chat e iniciar la conversación con estos usuarios por medio de la página ChatApplet.jsp

```
Chat.java
```

```
id: String  
session: HttpSession  
servlet: Context
```

```
doPost(HttpServletRequest req,
```

ConexiónVref, esta clase es la que hace la conexión a la base, que se encuentra en jdbc:mysql://scanner.udlap.mx:9097/vref, por medio de un driver:

org.gjt.mm.mysql.Driver, los parámetros requeridos son el login y el password para el acceso a la base.

```
ConexiónVref.java

url: String
driver: String
login: String
password: String
conexion: Connection
select: Statement
conec: ConexiónVref

ConexionVref()
AbrirConexion()
ResultSet()
cerrarConexion()
ejecutarInsert()
```

Si hay preguntas, estas son desplegadas en la pagina principal, y si algún usuario o referencista desean tener más información sobre esta pregunta, tienen la posibilidad de darle clic sobre la pregunta y desplegará el id del usuario que realizó la pregunta, así como la pregunta, y si es que ya fue contestada, muestra su respectiva respuesta, se puede calificar la respuesta por medio de los *componentes afectivos*. La presentación de la información de la pregunta es realizada por la clase LeePregunta.java

```
LeePregunta.java

id: String
tema: String
recuperaPregunta: String
cadenaPregunta: String
entero: int
preguntas: Vector
cues: Queries

doPost(HttpServletRequest req,
HttpServletResponse res)
fuera(HttpServletRequest req,
HttpServletResponse res)
```

MuestraInformacion.java, se encarga de desplegar la información de los usuarios que se encuentran en línea, para ello abre la conexión a la base, obtiene el índice del usuario del cual se esta solicitando la información, y con ese índice ejecuta el querie a la base, la cual devuelve la información (id, nombre completo, carrera y correo electrónico).

```
MuestraInformacion.java

usuario: Vector
recuperaid: String
usser: String
userId: String
userName: String
carrera: String
mail: String
entero: int
cues: Queries

doPost(HttpServletRequest req,
HttpServletResponse res)
fuera(HttpServletRequest req,
HttpServletResponse res)
cierra()
```

Queriesv.java, esta clase realiza los queries a la base.

```
Queriesv.java

mensajeError: String
conex: ConexionVref

Queriesv()
setMensajeError()
booleanBuscaUser()
obtieneNombre()
obtieneCarrera()
obtieneMail()
obtienePregunta()
lineaUsuario()
idPregunta()
preguntaP()
temaPregunta()
verIdPregunta()
actualizaMail()
insertaUsuario()
insertaEspecializacion()
insertaPregunta()
borraUsuario()
```

Si un usuario desea integrarse al grupo de referencistas, pasa su estado a de usuario normal al de referencista voluntario, la clase de RVoluntario.java abre la conexión a la base, el usuario debe darse de alta en la especialidad que considera de su mayor conocimiento y a partir de ese momento tiene la oportunidad de contestar preguntas hechas por los otros usuarios.

```
RVoluntario.java
```

```
id: String  
especializacion: String  
recuperaPregunta: String  
bandera: boolean  
cues: Queries
```

```
doPost(HttpServletRequest req,  
HttpServletResponse res)  
fuera(HttpServletRequest req,  
HttpServletResponse res)  
cierra()
```

Usuario.java, esta clase contiene todas las características de un usuario, así como su id, nombre, nombre secundario en caso de tenerlo, apellidos paterno y materno, password y correo electrónico

```
Usuario.java
```

```
id: String  
nombre: String  
nombre_s: String  
Apaterno: String  
Amaterno: String  
password: String  
correo: String
```

```
Usuario(id, nombre, apaterno,  
amaterno, pass, mail)  
setId(id)  
setNombre(nombre)  
setNombre_s(nombre)  
setApaterno(apaterno)  
setAmaterno(amaterno)  
setPassword(pass)  
setCorreo(correo)  
getId()  
getNombre()  
getNombre_s()  
getApaterno()  
getAmaterno()  
getPassword()  
getCorreo()
```

Finalmente la clase Salir.java, esta abre nuevamente la conexión a la base, y al usuario que ha pedido su salida del sistema lo da de baja de la tabla de usuarios en línea.

```
Salir.java  
  
id: String  
bandera: boolean  
cues: Queries  
  
doPost(HttpServletRequest req,  
HttpServletResponse res)  
fuera(HttpServletRequest req,  
HttpServletResponse res)  
cierra()
```

## 5.9 Diagrama de Clases

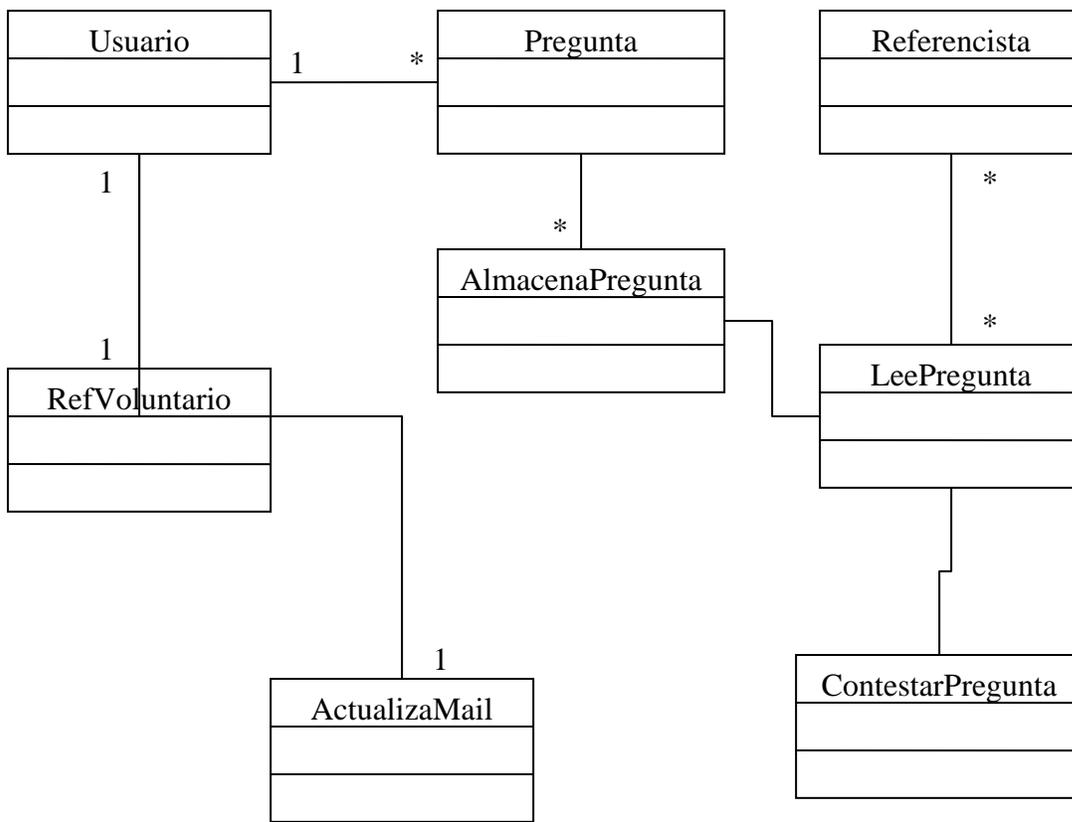


Figura 5.11 Diagrama de Clases

## 5.10 Pantallas del sistema

A continuación se muestran algunas pantallas del sistema.

La primera página muestra la bienvenida al sistema

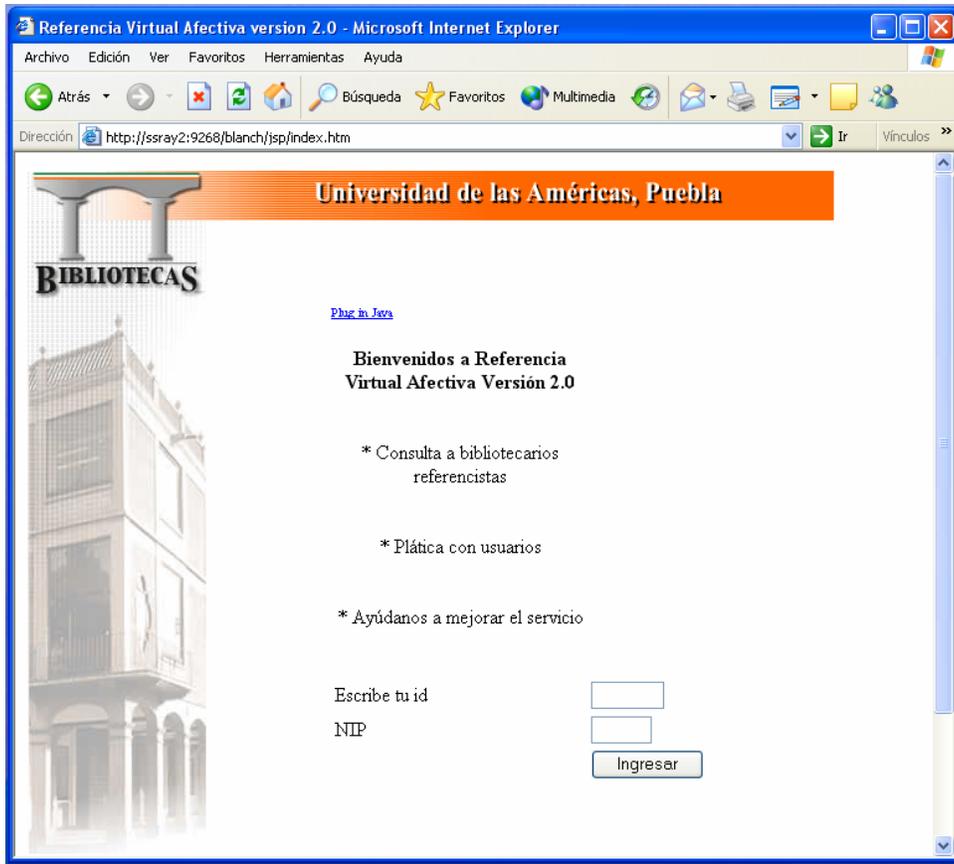
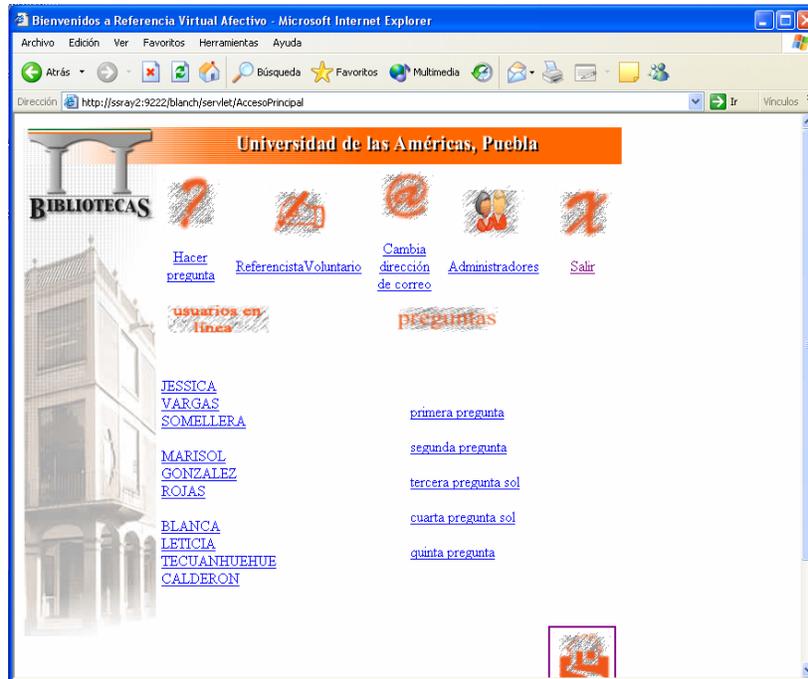


Figura 5.12 Página de Bienvenida

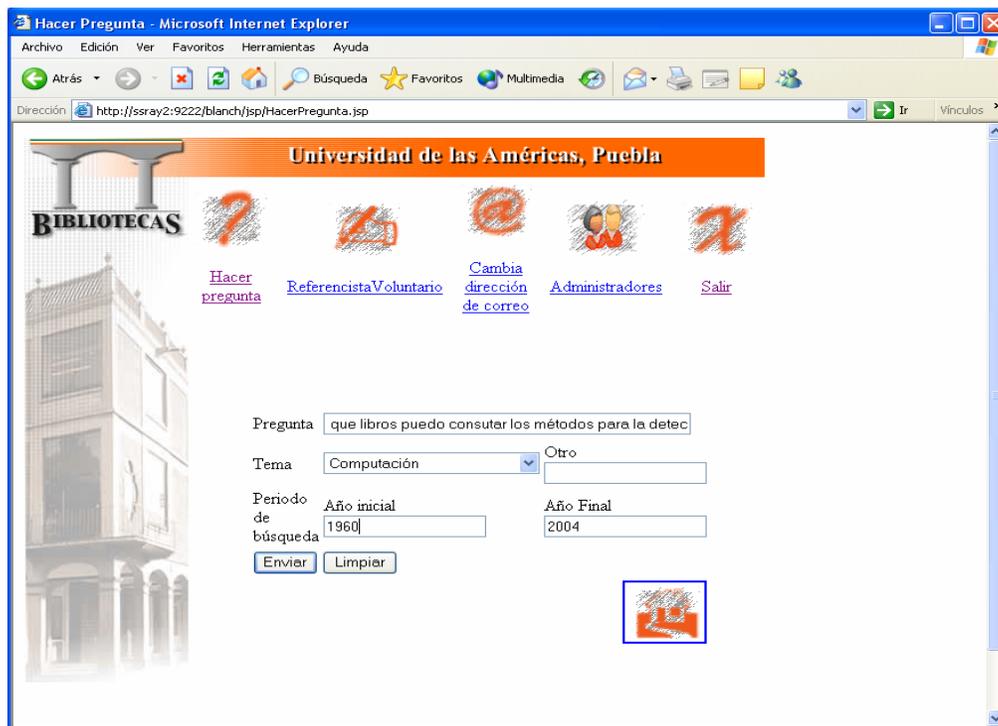
Si algún usuario ingresa mail sus datos, se el envía la página de acceso denegado, por haber ingresado mail sus datos



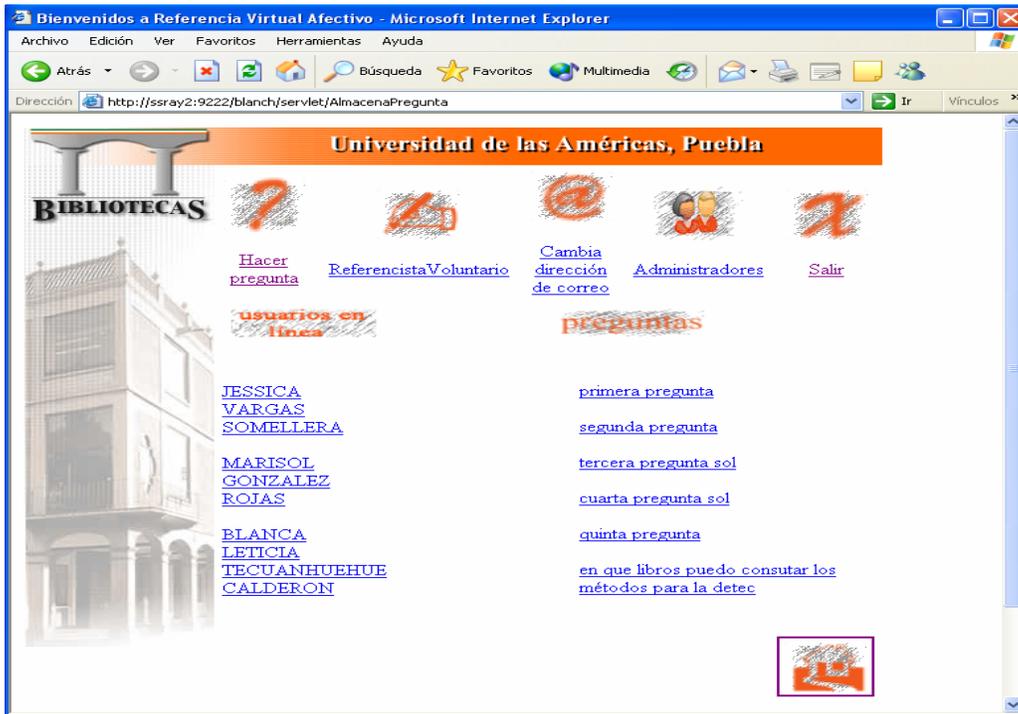
Página principal, muestra las opciones que tiene cada usuario por realizar, las cuales son:  
Hacer Pregunta, Referencista Voluntario, Cambiar Correo, Consultar Administradores y  
finalmente Salir.



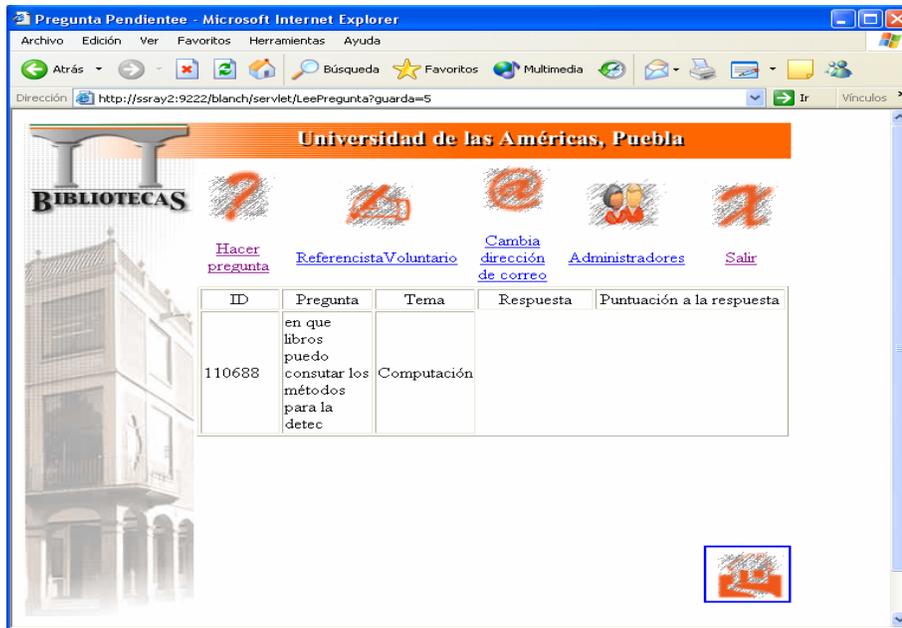
El usuario que desea Hacer Pregunta, tiene que llegar la forma que pide la pregunta, el tema, así como el periodo que desea que se haga la búsqueda.



Envía pregunta a principal



Damos clic sobre la pregunta, envía a una página que muestra, quien hizo la pregunta y la pregunta. Como es una pregunta recién enviada no hay nada en el campo *Respuesta*, ni en el campo de *Calificación*.

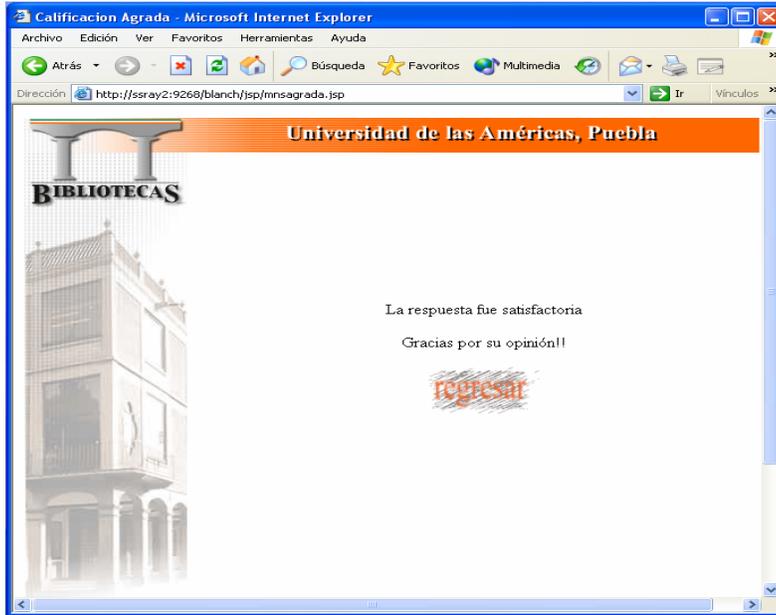


Cuando una pregunta es respondida se muestra la respuesta, así como la posibilidad de calificarla, como muestra la pantalla siguiente:

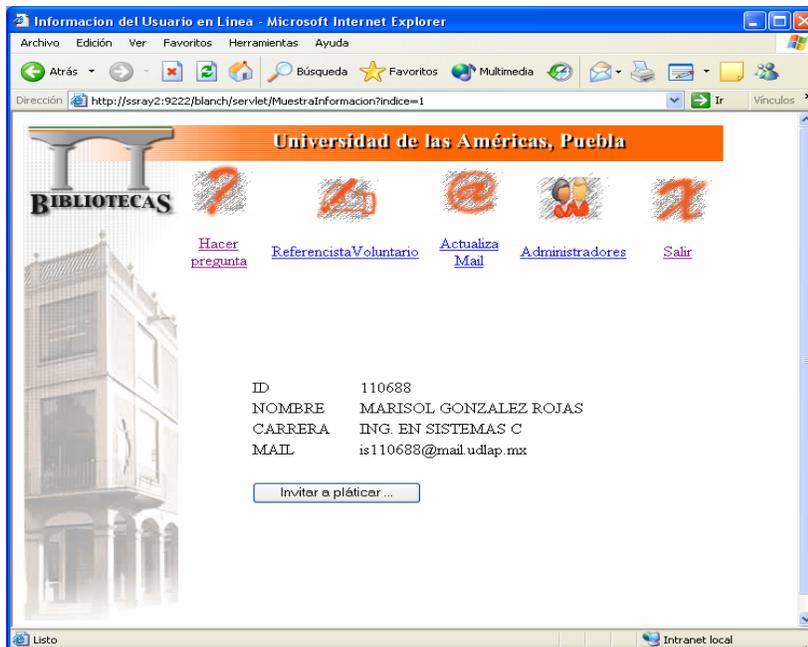


Si la respuesta fue satisfactoria entonces, se dirige a esta página. Y se muestra una página parecida a esta, a diferencia del mensaje “Respuesta no satisfactoria... Buscar otra” si no

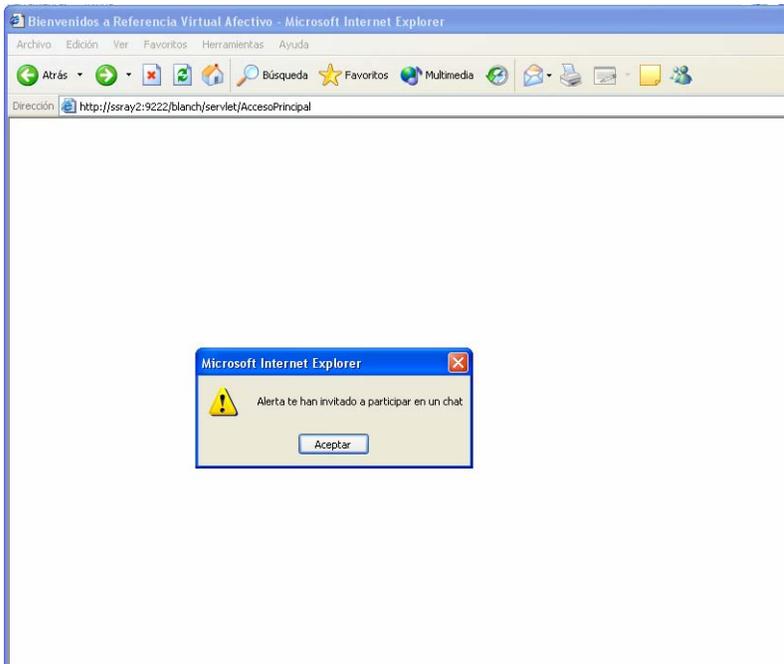
fue satisfactoria.



Cuando damos clic sobre un usuario en línea vemos la siguiente página, la cual presenta la información del usuario, así como un botón que nos permite invitar a platicar al usuario del cual estamos viendo su información al “Chat Afectivo”



Si el usuario oprimió el botón, se lanza una invitación al usuario del cual se está observando su información.



Finalmente se muestra el chat

