Módulo 6: Java EE

Antonio J. Díaz Rodríguez
Alberto Jiménez Ruiz
Juan Manuel Rodríguez González
Manuel Valls Vorndran
Francisco Romera Luzón

vodafone
MASTERINFTEL
IX Edición
UNIVERSIDAD DE MÁLAGA

Índice

1. Introducción

- Desarrollar un proyecto de Teleasistencia, el cuál está dirigido a personas con problemas de comunicación, es decir, con deficiencias auditivas y/o en la voz. Para la impleentación de la aplicación se ha utilizado el protocolo PaSOS.

- Este colectivo podrá comunicarse mediante mensajes que llegarán a la central de Teleasistencia.

- Aplicaciones interesantes: Deportistas de alta montaña como escaladores y senderistas.

2. Objetivos

- Desarrollar interfaz Cliente - Centralita

- Usar el protocolo paSOS.

*SAU11&LD20160303&LH060654&LN1008052067&LT153052067&PB50&LA12002&DT10 #

- Utilizar tecnologías Java EE.

3. Terminal

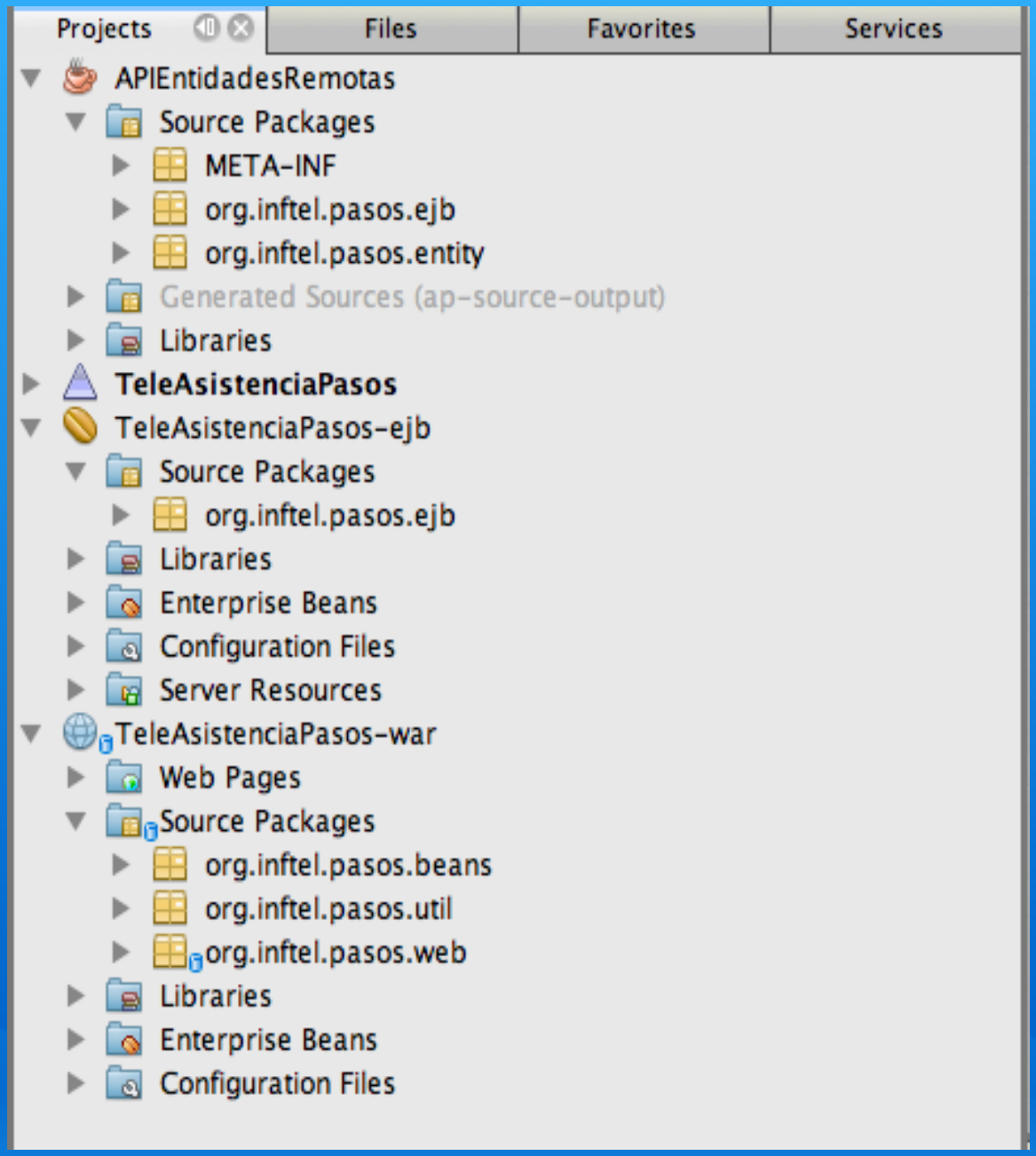
- Simulación de terminal en formulario Web.

- Envío de la trama a un Servlet de:
  - Alarma de usuario
  - Fecha
  - Hora
  - Longitud
  - Latitud
  - Nivel de batería
  - Altitud
  - Temperatura
  - Imei

- Descomposición de la trama en otro Servlet.

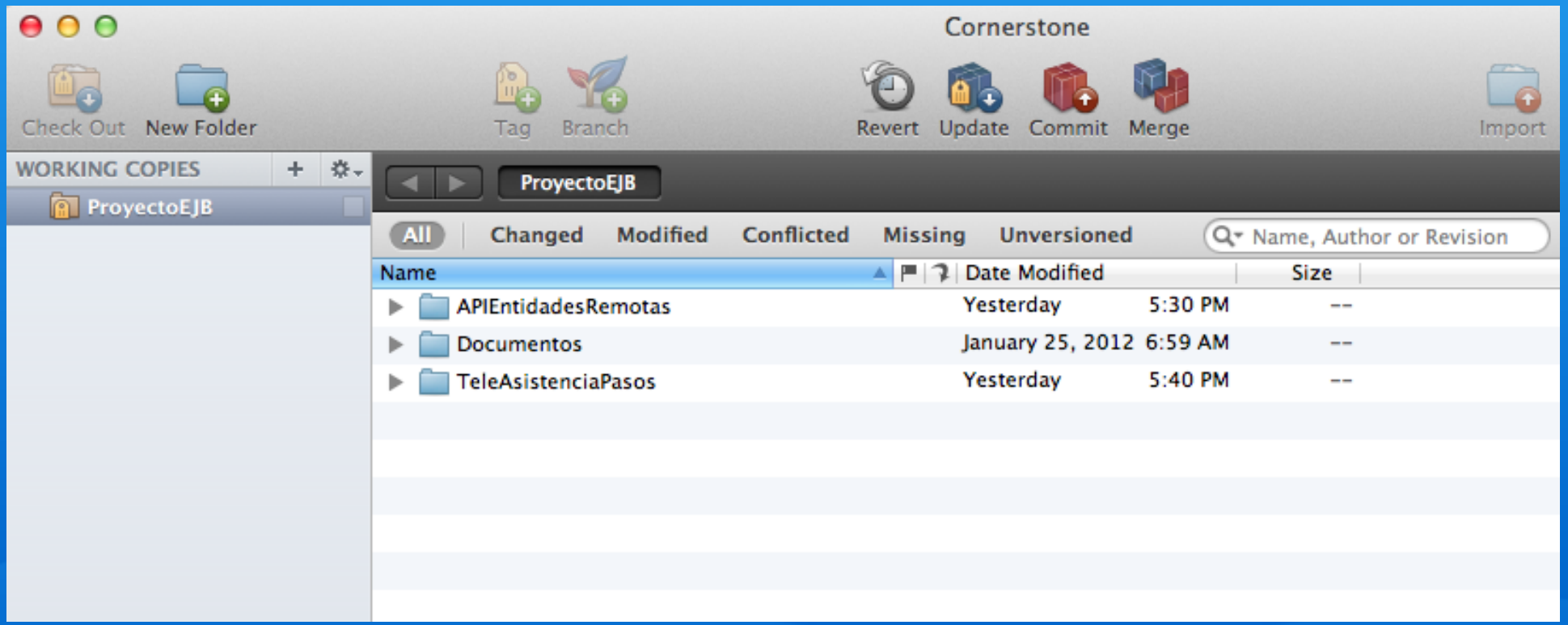
4. Estructura del Proyecto

- Estructura ideada para el proyecto:



4. Estructura del Proyecto

- Trabajo en grupo facilitado por Google Code y SVN:



4.Estructura del Proyecto

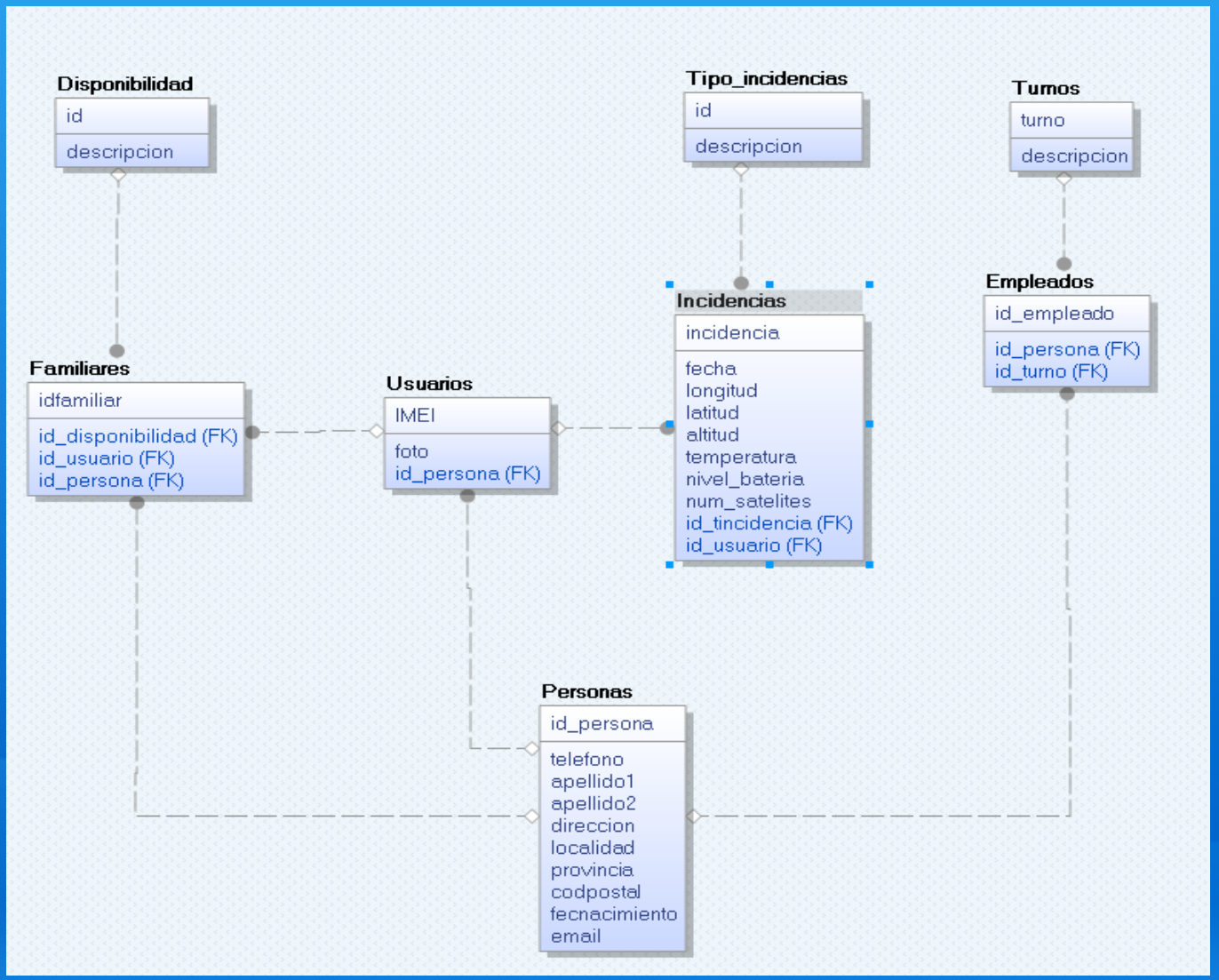
Vista → HTML, CSS, JSP

Controlador → Servlets

Modelo → JPA, EJB

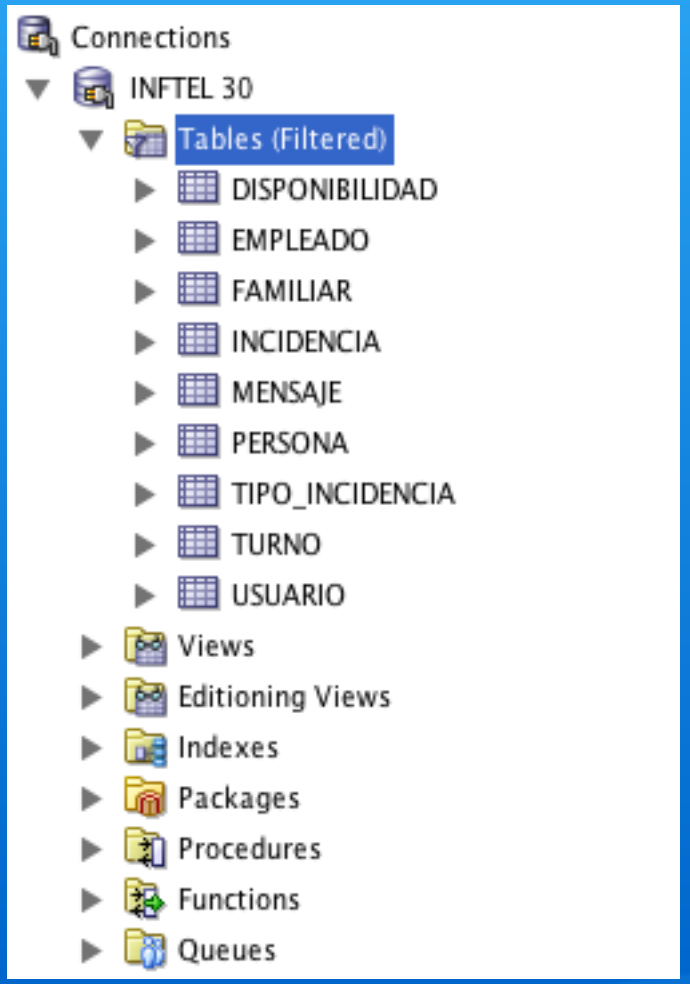
5. Tecnologías utilizadas

- ER-Win: Para generar el script principal de la Base de Datos:



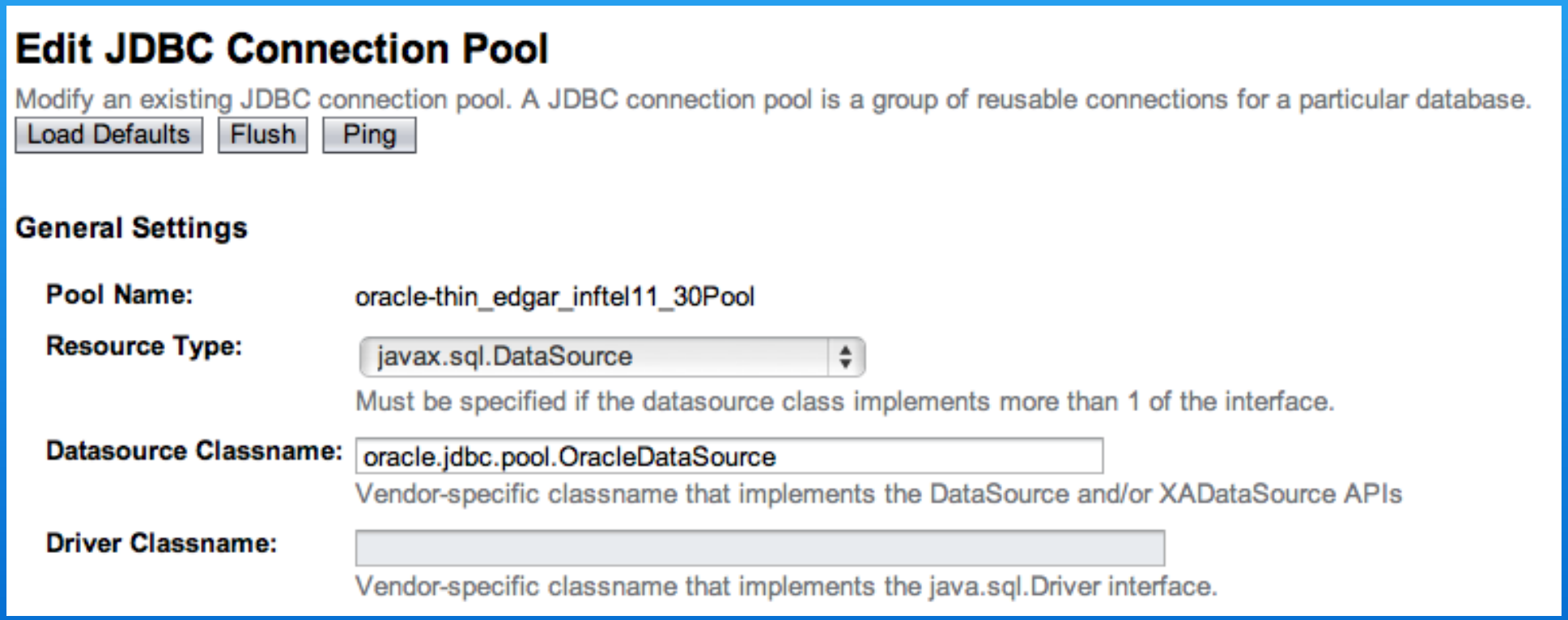
5. Tecnologías utilizadas

- Oracle: Como gestor de la Base de Datos. Conexión remota con el servidor de INFTEL.



5. Tecnologías utilizadas

- Glassfish: Servidor de Aplicaciones




```
WARNING: RAR5035:Unexpected exception while destroying resource from pool oracle-thin_edgar_inftel11_30Pool.
WARNING: RAR5035:Unexpected exception while destroying resource from pool oracle-thin_edgar_inftel11_30Pool.
WARNING: RAR5035:Unexpected exception while destroying resource from pool oracle-thin_edgar_inftel11_30Pool.
```

5. Tecnologías utilizadas

- JPA: Tecnología para abstraer el acceso a la BD mediante Entidades

- Entidades creadas por Netbeans.
- Posteriormente las Entidades se adaptaron a sus necesidades.

```
@Id
@Basic(optional = false)
@NotNull
@Column(name = "ID_USUARIO")
@GeneratedValue(strategy = GenerationType.AUTO, generator="personas_seq_gen")
@SequenceGenerator(name="personas_seq_gen", sequenceName="PERSONAS_SEQUENCE", allocationSize=1)
private BigDecimal idUsuario;
```

5. Tecnologías utilizadas

- JPA: Tecnología para abstraer el acceso a la BD mediante Entidades

```
@JoinColumn(name = "ID_PERSONA", referencedColumnName = "ID_PERSONA")
@ManyToOne(cascade=CascadeType.PERSIST)
private Persona idPersona;
```

```
@Override
public String toString() {
    String info = "";
    info += "NOMBRE: " + nombre;
    info += "\nAPELLIDO1: " + apellido1;
    info += "\nAPELLIDO2: " + apellido2;
    info += "\nDIRECCION: " + direccion;
    info += "\nLOCALIDAD: " + localidad;
    info += "\nPROVINCIA: " + provincia;
    info += "\nCODIGO POSTAL: " + codpostal;
    info += "\nTELEFONO: " + telefono;
    info += "\nFECHA DE NACIMIENTO: " + fecnacimiento;
    info += "\nEMAIL: " + email;
    return info;
}
```

5. Tecnologías utilizadas

- Subir imágenes utilizando JPA:

```
public static Usuario formToBeanUser(HttpServletRequest req) {

    boolean isMultipart = ServletFileUpload.isMultipartContent(req);
    Usuario usuario = new Usuario();
    Persona persona = new Persona();
    usuario.setIdPersona(persona);

    if (isMultipart) {
        // Patrón factoría para manejar ficheros procedentes de formularios
        FileItemFactory factory = new DiskFileItemFactory();
        ServletFileUpload upload = new ServletFileUpload(factory);

        try {
            // Sólo cojo la primera imagen
            List<FileItem> items = upload.parseRequest(req);
            for (FileItem item : items) {
                if (item.isFormField()) {
                    usuario = actualizaUsuario(usuario, item);
                } else {
                    InputStream is = item.getInputStream();
                    long sizeLong = item.getSize();
                    int size;

                    if (sizeLong < Integer.MAX_VALUE) {
                        size = (int) sizeLong;
                        byte[] buffer = new byte[size];
                        is.read(buffer);
                        usuario.setFoto(buffer);
                    }
                }
```

5. Tecnologías utilizadas

- Subir imágenes utilizando JPA:

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

    BigDecimal identi = new BigDecimal(request.getParameter("id"));
    Usuario usuario = usuarioFacade.find(identi);
    byte[] data = usuario.getFoto();
    InputStream in = new ByteArrayInputStream(data);
    BufferedImage image = ImageIO.read(in);

    // Send back image
    ServletOutputStream sos = response.getOutputStream();
    JPEGImageEncoder encoder = JPEGCodec.createJPEGEncoder(sos);
    encoder.encode(image);
}
```

```
<c:choose>
    <c:when test="${personaBean.usuario.foto != null}">
        <img src="http://localhost:8080/TeleAsistenciaPasos-war/ServletAvatar?id=
            ${personaBean.usuario.idUsuario}&rnd=<%=System.currentTimeMillis()%>" />
    </c:when>
    <c:otherwise>
        <img src="images/foto.jpg"  />
    </c:otherwise>
</c:choose>
```

Tecnologías utilizadas

Obtención de la ubicación del usuario (bug en mapa)

Tecnologías utilizadas

Obtención de la ubicación del usuario (bug en mapa)

- Implementación:
    - Bean de sesion local
    - Servlet para consultar dicho bea
    - Jquery para obtener los resultados

Tecnologías utilizadas

SERVLET

```
Incidencia incidencia = recibirCoordenadas.
getCoordenadaIncidencia(BigDecimal.valueOf(1));

BigDecimal lat = incidencia.getLatitud();
BigDecimal lon = incidencia.getLongitud();

out.println("<resultado><respuesta>SI"+
"</respuesta><latitud>" + lat +
"</latitud><longitud>" + lon +
"</longitud></resultado>");
```

Tecnologías utilizadas

BEAN

```
@Override
public Incidencia getCoordenadaIncidencia(BigDecimal idUsuario) {
    Usuario u = new Usuario();
    Persona p = new Persona();
    p.setIdPersona(idUsuario);
    u.setIdPersona(p);
    Collection<Incidencia> linc = u.getIncidenciaCollection();
    Incidencia inc;
    if(linc.size() > 0) {
        inc = linc.iterator().next();
    } else {
        inc = new Incidencia();
        inc.setLatitud(BigDecimal.valueOf(0));
        inc.setLongitud(BigDecimal.valueOf(0));
    }
    return inc;
}
```

Tecnologías utilizadas

JQUERY

```
$(document).ready(function(){
var myLatlng = new google.maps.LatLng(12.456781, 12.5450253);
var myOptions = {
    zoom: 14,
    center: myLatlng,
    mapTypeId: google.maps.MapTypeId.ROADMAP
}
map = new google.maps.Map(document.getElementById("map_canvas"), myOptions);
comprobarRecibido();
});
//-----------------------------------------------------------
var comprobarRecibido = function(){
$.post(
"ServletEsperarUsuario",
function(xml) {
    var respuesta = $("respuesta", xml).text();
    if(respuesta == "NO") {
        setTimeout(comprobarRecibido, 4000);
    } else if(respuesta == "SI") {
        var latitud = $("latitud", xml).text();
        var longitud = $("longitud", xml).text();
        var latlng = new google.maps.LatLng(latitud, longitud);
        ...
    }
```

Tecnologías utilizadas

- Uso de Servlets.
- Uso de páginas JSP para utilizar contenido dinámico.
- Beans de sesion.
- Base de datos oracle.

Dificultades y problemas

Dificultades

- Implementar un chat sin bajar a nivel de socket.
- Propuestas estudiadas:
  - Beans de mensajerias
  - Comunicacion con la base de datos
  - Utilizar API de chat.

Problemas

- NetBeans.
- Glashfish.
- Oracle.

Lineas futuras

- Implementar una aplicacion movil para el usuario.

- Implementar otros tipos de alarmas, y la consecuente gestion de estas en la aplicación web.

- Mejorar aplicacion (chat, estadisticas por usuario, l
de incidencia en mapa...)

Demostración