

MANUAL TÉCNICO DE INFO SALUD PLUS



Trabajo fin de ciclo: Desarrollo de Aplicaciones Web

Alumno: Ricardo Díaz González

Fecha de entrega: 20 de junio de 2017

Centro: IES Albarregas

ÍNDICE

1. Introducción.....	Pág 3
2. Tecnologías utilizadas.....	Pág 3
a) Tecnologías del lado servidor.....	Pág 3
b) Tecnologías del lado cliente.....	Pág 5
3. Diseño de la base de datos.....	Pág 6
4. Estructura del proyecto.....	Pág 7
5. El proceso de despliegue del proyecto.....	Pág 8
6. Guía de estilos.....	Pág 10
7. Diagrama de clases.....	Pág 12

1. Introducción.

En este manual pretendemos reflejar como se ha construido InfoSaludPlus y como puede ponerse en funcionamiento sin que sea necesario tener grandes conocimientos de despliegue de aplicaciones web.

También pretende facilitar la comprensión y la asimilación del funcionamiento de la aplicación para cualquier técnico, que a posteriori, pudiera modificarla o ampliarla.

2. Tecnologías utilizadas.

Para realizar esta aplicación hemos utilizado diferentes tecnologías para cumplir con todos los requisitos necesarios.

Para realizar el proyecto hemos necesitado un entorno de desarrollo. Hemos elegido netbeans por su potencia y por haber trabajado antes con él.

2.1 Tecnologías del lado servidor.

Java EE o Java Web: El proyecto ha sido desarrollado en tecnología Java EE o Java Web. Es una plataforma de programación que permite utilizar una arquitectura de n capas, utilizando el lenguaje de programación java.

```
public String aniadirCentro() {  
    try {  
        pueblo.setProvincia(provincia);  
        direccion.setPueblo(pueblo);  
        centro.setDireccion(direccion);  
        igd.add(centro);  
        listCentros.add(centro);  
        contadorListCentros++;  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
    return "";
```

(Fragmento de código java en el que añadimos un centro a la base de datos)

Frameworks de Java

Java Server Faces: es un framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuarios en aplicaciones Java EE..

Richfaces: es una biblioteca de código abierto basada en [Java](#) que permite crear aplicaciones web con Ajax.

Construye sobre el framework de Java Server Faces. Sobre él, implementa unos filtros para permitir peticiones Ajax en la página. La singularidad del planteamiento que ofrece es que la petición Ajax provoca una ejecución en el servidor y finalmente una renderización parcial o total de la página del navegador. Así, el control de lo que sucede está en el servidor.

```
<h:form>
  <rich:panel>
    <h:outputText styleClass="msgOk" value="El informe se ha generado correctamente."/>
    <div class="botones">
      <a4j:commandButton value="Cerrar" styleClass="boton"
        immediate="true" render="menuPrincipal"
        onclick="#{rich:component('confirmInforme')}.hide();" />
    </div>
  </rich:panel>
</h:form>
```

(modal de confirmación de generación de un informe con JSF y Richfaces)

Hibernate: es una herramienta de mapeo objeto-relacional para java, que facilita el mapeo de atributos entre una base de datos relacional y el modelo de objetos de una aplicación construida en java.

```
public void update(T objeto) {
    try {
        iniciaSesion();
        sesion.update(objeto);
    } catch (HibernateException he) {
        this.manejaExcepcion(he);
    } finally {
        this.cierraSesion();
    }
}
```

(Ejemplo de un método para actualizar un registro de una tabla en hibernate)

JPA: Es una api de persistencia desarrollada para Java EE.

```
@Id
@Column(name = "idUserario")
@GeneratedValue(strategy = IDENTITY)
protected int id;
```

(Una clave primaria auto_increment de una tabla declarada con jpa)

2.2 Tecnologías del lado cliente.

JavaScript: es el lenguaje de programación por excelencia del lado cliente, que apenas hemos utilizado ya que richfaces te abstrae de casi todo el lado cliente.

```
onclick="document.getElementById('citaReservadaForm').style.display = 'table'"
```

(Utilizamos muy poco js, y muy aisladamente, pues richfaces lo usa implícitamente)

Jquery: es un framework de JavaScript, que apenas hemos utilizado ya que richfaces te abstrae de casi todo el lado cliente.

```
oncomplete="#{rich:component('modalConfirmCita')}.show()"
```

(Con Jquery nos ocurre lo mismo que con Java Script)

Bootstrap: es un framework de diseño web y de programación del lado cliente, abarca o mezcla ambas cosas. Lo hemos utilizado principalmente para hacer la aplicación responsive, con sus famosas rejillas de 12 columnas. En alguna ocasión hemos utilizado otras propiedades o incluso alguna parte de la web puede estar diseñada con este framework, ya que es muy potente y da agilidad y vistosidad al diseño.

```
<h:form id="citaReservadaForm" class="col-sm-6 col-sm-offset-1" styl
<div class="panel panel-primary">
  <div class="panel-heading"><h:outputText value="Cita previa
  <div class="panel-body">
```

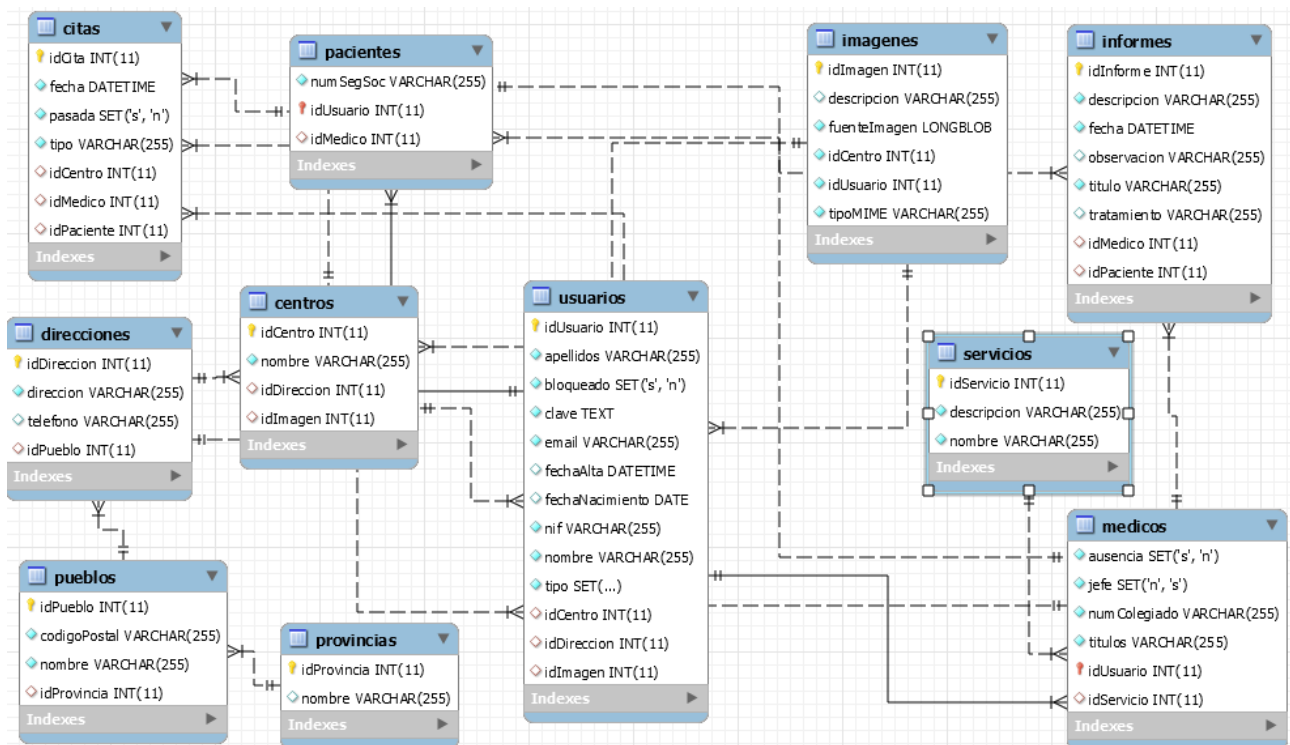
(Estas son las clases propias de bootstrap)

3. Diseño de la base de datos.

Nuestra base de datos, llamada infoSaludPlus, tiene las siguientes tablas:

Tables_in_infosaludplus
centros
citas
direcciones
imagenes
informes
medicos
pacientes
provincias
pueblos
servicios
usuarios

El modelo relacional de nuestra base de datos (infoSaludPlus):



Vemos todas las tablas relacionadas entre ellas. Y podemos apreciar como Pacientes y Médicos heredan de Usuarios. Aunque esto lo conseguimos con hibernate y las clases Java, que lo veremos más abajo cuando veamos el diagrama de clases.

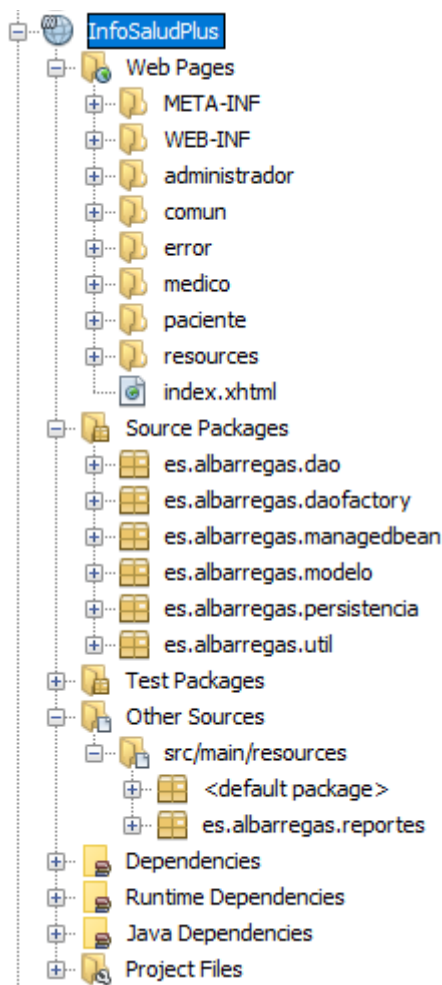
4. Estructura del proyecto.

Para construir y gestionar nuestro proyecto, utilizamos la herramienta [maven](#).

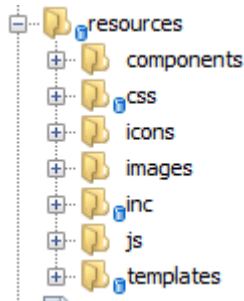
La estructura del proyecto es la siguiente:

El proyecto, InfoSaludPlus, tiene una estructura en árbol donde se aloja todo el contenido. Dentro del Web Pages se encuentran todos los archivos relacionados con las vistas, así como algunos archivos de configuración.

Dentro de WebPages, hemos separado agrupado las vistas, en carpetas, según su tipo de usuario: administrador, médico, paciente, común (para usuarios anónimos). Y además, error, para las páginas de error.



Además, también tenemos resources, donde desde las vistas se cargarán algunos recursos necesarios (imágenes, ficheros javascript, includes...).



En la carpeta templates están las plantillas de InfoSaludPlus. En nuestra aplicación nos ha sido suficiente con una sólo plantilla, pero podríamos haber utilizado más de una.

En components, podríamos haber guardado componentes jsf propios, pero en esta ocasión no hemos creado ninguno.

5. El proceso de despliegue del proyecto.

Lo primero que debemos hacer para que funcione nuestra aplicación es importar la base de datos. Para ello deberemos tener instalada una versión de mysql igual o superior que la 5.5 (en versiones anteriores puede funcionar, pero no está probado).

Para importar la base de datos nos vamos a la consola, que en windows podemos acceder a ella pulsando windows + R, se nos abre el "Ejecutar", ponemos "cmd" sin comillas y pulsamos aceptar. (en Linux y Mac a través de "Terminal", o el programa que uséis.)

Ahora, dando por supuesto que hemos instalado mysql server y que tenemos configurado correctamente el path, pondríamos

```
mysql -uusuario -p - -default_character_set utf8 baseDeDatos < ficheroBaseDatos.sql
```

(poniendo el - - default_character_set utf8, nos aseguramos que importe en utf8, sino

puede haber problemas. Y es conveniente hacer lo mismo al exportar).

Lo siguiente será tener instalada la máquina virtual de Java (JVM) en su versión 1.8, imprescindible para correr aplicaciones java. Cada Sistema Operativo tiene su propia JVM. La encontraréis sin ningún problema en (<https://www.java.com/es/download/>).

Para desplegar nuestro proyecto necesitamos el archivo .war de la aplicación.

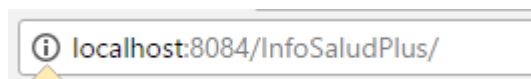
Este archivo se genera en el target del proyecto, en la siguiente imagen lo vemos mejor:

Equipo > OS (C:) > proyectosJava > JSF > InfoSaludPlus > target			
Nombre	Fecha de modifica...	Tipo	
classes	20/06/2017 9:57	Carpeta de archivos	
endorsed	19/06/2017 16:26	Carpeta de archivos	
generated-sources	19/06/2017 16:26	Carpeta de archivos	
InfoSaludPlus-1.0	19/06/2017 20:52	Carpeta de archivos	
maven-archiver	19/06/2017 16:26	Carpeta de archivos	
maven-status	19/06/2017 16:26	Carpeta de archivos	
surefire	19/06/2017 16:26	Carpeta de archivos	
test-classes	20/06/2017 9:57	Carpeta de archivos	
InfoSaludPlus-1.0.war	19/06/2017 16:26	Archivo WAR	

Y debemos copiarlo y pegarlo en la carpeta "webapps", dentro de Tomcat 8, como vemos en la siguiente imagen:

Equipo > OS (C:) > apache-tomcat-8.0.37 > webapps			
Nombre	Fecha de modifica...	Tipo	
manager	27/09/2016 16:37	Carpeta de archivos	
InfoSaludPlus-1.0.war	19/06/2017 16:26	Archivo WAR	

Ahora nos vamos al navegador. Ponemos <http://localhost:8084/nombreDelWar> donde localhost es nuestra máquina, y 8084 es nuestro puerto por donde escucha Tomcat 8. Hay que tener en cuenta que el puerto por defecto de escucha de Tomcat es el 8080, pero nosotros lo hemos configurado para que escuche por el 8084.



6. Guía de estilos

Fotos y logos:

La página web tendrá un logo situado a la izquierda del nombre, en la cabecera. Este logo será común a todas las secciones de la web.

Las imágenes-logos no serán inferiores a 50x50px, para que puedan verse con facilidad. Las imágenes que se muestren en InfoSaludPlus serán responsivas, es decir, se adaptarán al tamaño de la pantalla

Tipografías:

- Título de la página: estará en H1, en tipo de letra *Helvetica Neue*, con tamaño 36px (predefinido por h1) y en mayúsculas, con doble espaciado entre las letras y cada palabra.
- Subtítulo de la página: estará en H2, en tipo de letra Helvetica Neue, itálica, con tamaño en 30px y en minúsculas menos la primera letra de cada palabra.
- Texto de las secciones en 11 pt, tipo de letra Arial.
- El footer tendrá la información accesible con tamaño 14pt, en tipo de letra Arial, con estilo en negrita y subrayado.

Colores de la web:

- Título: #317eac
- Subtítulo: #333333
- Fondo cuerpo de la página: rgb(240,240,255)
- Pie de página – RRSS: #2c2c2c
- Fondo pie de página – Enlaces: #222
- Fondo pie de página – Derechos : #006290

Fondos:

Tomaremos como colores de fondo, colores pasteles, para no cansar la vista del usuario,

ya que los colores pasteles son los que mejor perciben los usuarios, sin producir fatiga visual.

El color de fondo del cuerpo será un azul claro. Pudiendo ser usado el gris claro o el amarillo claro en otros casos.

Es muy importante que el color de la fuente resalte bien con el color de fondo, para que el texto sea totalmente legible y los usuarios no se vean obligados a forzar la vista para comprender el texto o el contenido de la página.

Los colores de fondo que empleamos los podemos ver en el apartado anterior.

Letras:

Para que la página sea robusta, al igual que usamos sólo 2 o 3 colores de fondo, y 2 o 3 colores de letras, también usaremos sólo unos pocos tipos de letras. Por tanto, en nuestra web sólo usaremos los tipos de letra Helvética, para título y subtítulo. Arial, y sans-serif, para el resto del texto.

Iconografía:

Los iconos que se usen serán propios o libres, y deberán tener una claridad suficiente para guiar al usuario en su cometido. También deben tener un "alt" y un "title", para identificarlos mejor.

Estructura:

InfoSalud tiene una estructura bien definida en todas sus páginas.

En todas ellas tiene una cabecera, un navegador, un cuerpo principal y un pie de página.

Después, en las áreas privadas de cada usuario habrá también un menú vertical situado a la izquierda.

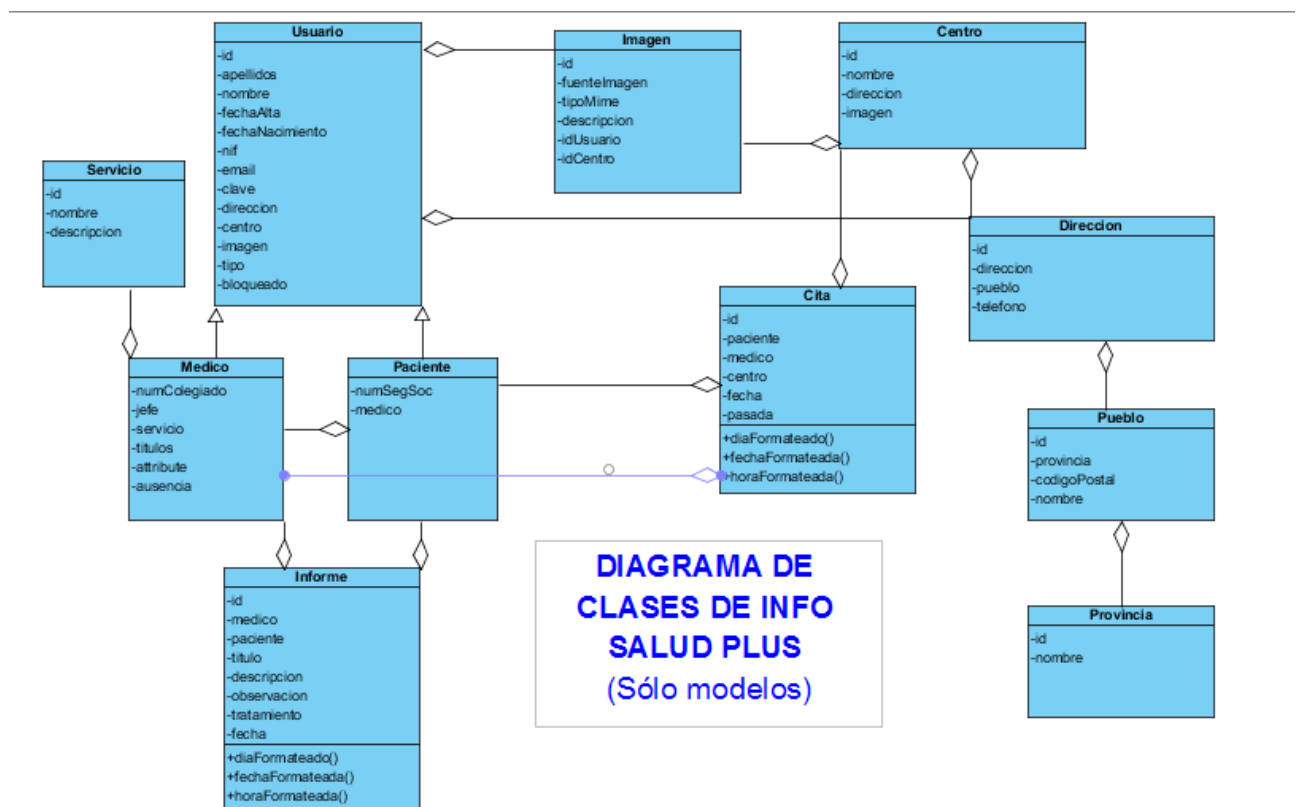
Entre elementos de la misma capa siempre habrá un margen de un mínimo de 10px.

7. Diagrama de clases

El diagrama de clases representa la estructura de las clases del proyecto, visualizando así sus atributos, métodos y relaciones.

Por cuestiones de tamaño, sólo he incluido los modelos. Ya que si hubiera metido los managed beans y las clases de utilidades y de acceso a base de datos el diagrama sería de un tamaño desproporcionado y lo que nos interesa es ver el concepto, como interactúan las clases principales en nuestra aplicación.

También se adjuntará la imagen a parte en el proyecto para poderlo ver con mayor nitidez.



Quizá lo más interesante del diagrama, es ver que Pacientes y Médicos heredan de Usuarios. Toman todos sus atributos, y aparte Paciente y Médico tienen sus atributos propios o específicos.