

# **Práctica 3:**

## Aplicación web de Sistema Tutor

Universidad de Málaga - E.T.S Ingeniería Informática

Desarrollo de Servicios Telemáticos

*Antonio J. Galán Herrera*

<b>Indicaciones previas</b>	<b>2</b>
<b>Manual de usuario</b>	<b>3</b>
<b>Funcionamiento</b>	<b>4</b>
Página Principal	5
Páginas de Login y de Registro	5
Página del perfil del usuario	5
Página de creación de un examen	5
Páginas de examen, pregunta y estadísticas	5
<b>Discusión crítica</b>	<b>6</b>
Bondades	6
Defectos	6

# Indicaciones previas

**Esta práctica es funcional pero no cumple todos los requisitos del enunciado.**

Todo el proyecto ha sido descrito a través de **javadocs** que indican el funcionamiento y/o la finalidad de los métodos que no son `doGet()` ni `doPost()` de los Servlets.

El proyecto se ha creado y construido usando el entorno **IntelliJ** y usando el servidor GlassFish 5.0.1, pero cabe destacar que solo **IntelliJ Ultimate** permite la creación de proyectos de Java EE.

Por otra parte, el proyecto debía entregarse como proyecto de NetBeans, así que al hacer la conversión a este entorno, la ruta de contexto cambia: **los ficheros usuarios.txt y los exámenes se crean y se tratan en la ruta donde está alojado el servidor de GlassFish de NetBean 8.2.** Esto solo debe tenerse en cuenta si el proyecto es el de la entrega.

Por último, se ha incluido en formato HTML el código que emplean los Servlets de la aplicación para generar las diferentes páginas; se encuentran en la carpeta *plantillas*, y no son usados por la aplicación en ningún momento, su única función es proporcionar una lectura más cómoda de la página que generan los Servlets.

# Manual de usuario

Para ejecutar la aplicación, el proyecto debe abrirse con el entorno adecuado:

- NetBeans (8.2 recomendado) en el caso de la entrega.
- IntelliJ Ultimate en caso contrario (formato original).

Una vez ejecutada, se mostrará inicialmente la página principal, *index.html*, que emplea el estilo *estilos.css* (igual que todas las demás) y se podrá elegir entre hacer login o registrarse.

La página principal tiene un logo que puede modificarse pulsando los botones -1 y +1.

Si se registra un nuevo usuario, se almacenarán sus credenciales de usuario y contraseña en el fichero *usuarios.txt*. Posteriormente, estará disponible para el login. Si se hace un login, se buscará en el fichero *usuarios.txt* y si las credenciales son correctas, se accederá al perfil de usuario.

Dentro del perfil se podrá elegir crear un examen nuevo, que conducirá a otra página mostrando la información del examen generado e incrementará la lista de exámenes disponibles para el usuario, o podrá elegir un examen para realizarlo.

Cada examen generado es almacenado en la carpeta *exámenes* con un nombre secuencial *examenX.txt* donde X es un número a partir de la cantidad de exámenes que ya haya creados.

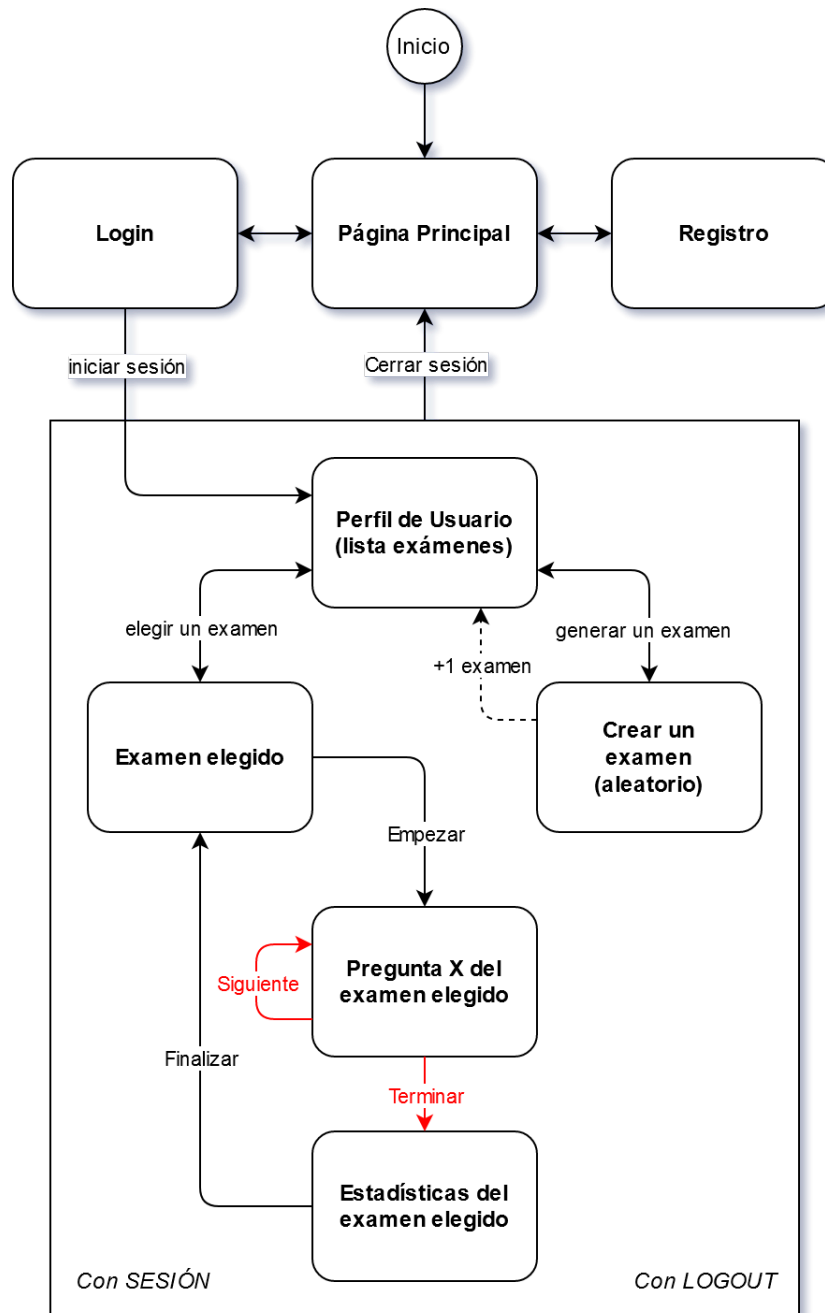
Si se elige un examen, se accederá a una página indicando que se ha seleccionado un examen y que va a iniciarse el proceso de las preguntas. Esta página conduce a una serie de páginas con una pregunta cada una.

El proceso de leer las preguntas de un examen y generar una página para cada una, así como el de procesar las respuestas y generar las estadísticas no están implementadas. No obstante, se crearon métodos de clase para hacer lo descrito.

# Funcionamiento

Salvo la página principal que se genera usando un fichero HTML, todas las demás páginas son implementadas mediante Servlets.

Todas las páginas a partir del perfil del usuario cuentan con un botón de regreso a la página principal, indicado en el diagrama como las páginas dentro del cuadrado.



Se muestran en **rojo** las funciones no implementadas.

Todo lo demás, en negro (páginas, enlaces y procesos), si está implementado.

## Página Principal

Contiene el nombre de la aplicación, un logo, una animación, y 2 enlaces hacia las páginas de Login y de Registro.

## Páginas de Login y de Registro

Ambas páginas contienen un formulario: la primera comprueba si un usuario está registrado (verifica los campos en el fichero usuarios.txt) y la segunda añade un nuevo usuario y su contraseña al fichero usuarios.txt.

Ambas clases tratan errores de coherencia de datos e inicializan el fichero usuarios.txt si este no existe (se ejecuta la aplicación por primera vez).

## Página del perfil del usuario

Muestra el usuario que se ha registrado y diversos enlaces.

Esta página genera una carpeta con 3 exámenes iniciales en el caso de que dicha carpeta no exista (se ejecuta la aplicación por primera vez).

Los enlaces conducen a una página para generar un nuevo examen y cada examen creado en la carpeta mencionada, genera su propio enlace para ser resuelto.

Habrán tantos enlaces a exámenes, como exámenes haya en dicha carpeta.

## Página de creación de un examen

Genera un examen con un número de entre 5 y 10 preguntas al azar y con respuestas aleatorias entre las opciones «Sí», «No» y «A veces», mostrando toda esa información en la pantalla.

## Páginas de examen, pregunta y estadísticas

Tras seleccionar un examen del perfil del usuario, se genera una página avisando de que va a realizarse un examen.

La generación dinámica de preguntas no ha sido implementada y por tanto, tampoco se generan estadísticas (aunque existe un método en la clase Examen que las calcule).

# Discusión crítica

## Bondades

Todo el código se ha modularizado en función del tipo de Servlet y se han añadido javadocs a aquellos métodos que no son `doGet()` ni `doPost()`.

Se ha aplicado un CSS minimalista y que ofrece buena visibilidad.

Se han dejado las plantillas HTML usadas en los Servlets para visualizar mejor el código.

Se han implementado todas las clases necesarias para cubrir todas las necesidades del proyecto.

El programa inicializa los ficheros si estos no existen, de forma que no es necesario introducir ningún fichero manualmente en el primer arranque.

Los datos de los exámenes se almacenan en ficheros que se crean de forma secuencial y están contenidos en una carpeta para un fácil acceso.

## Defectos

No se ha usado JavaScript.

No se ha implementado el proceso de resolución de un examen.

Pese a que los Servlets están bien modularizados, se usan mucho.