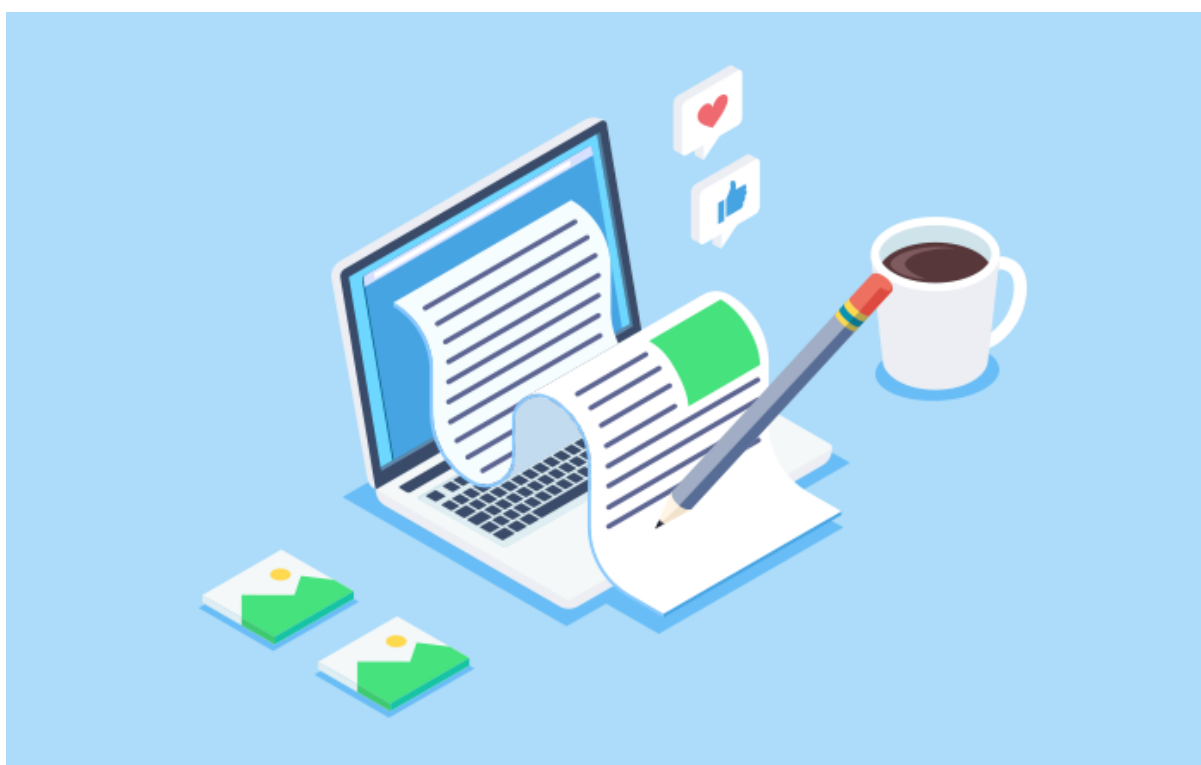


Documentacion Blog Servlets

Lucas Garau



Introducción/Índice:

Esta aplicación es una aplicación de blogs gestionada en un servidor tomcat con base de datos mysql dockerizada con una conexión phpmyadmin para gestionar, esta documentación está separada en 3 partes principales dirigidas a diferentes lectores para saber y entender cómo funciona:

1.Manual de instalación:

Orientado al gestor/persona que mantendrá o instalará la aplicación de los pasos a seguir en caso de poner la aplicación en marcha por primera vez y prepararla para ser usada.

2.Manual de uso:

Manual dedicado a el uso que le darán los usuarios a la aplicación y explicación lógica de el uso con un paso a paso de los menús y todas las posibilidades de la misma.

3.Estructura:

Indicaciones de como se ha creado la aplicación y dirigido a un programador/analista en la que se explica cómo ha sido creada la aplicación , que herramientas y recursos se han utilizado.

1. Manual de Instalación:

En este apartado veremos lo necesario para hacer funcionar el programa.

1.1 Descargar el programa:

El programa puede ser descargado desde el enlace de drive donde viene incluido todo lo necesario para que funcione.

1.2 Base de datos

La aplicación se hospeda en un servidor tomcat primero de todo tendremos que ir a la pagina oficial de tomcat <https://tomcat.apache.org/download-90.cgi> y descargarnos la ultima versión.

Una vez descargada la guardamos en un sitio seguro dado que la requerimos para hacer deploy de nuestra explicación y como hacerlo se explicara en un paso siguiente.

1.2.2 En mi caso he usado IntelliJ ultimate instalado por JetBrains toolbox

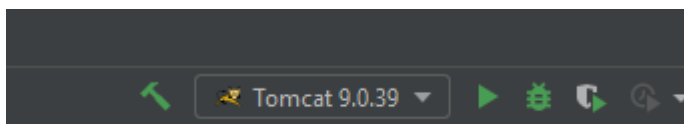
<https://www.jetbrains.com/es-es/toolbox-app> simplemente descarge el formato que le ponga y instale , despues se le abra un menu y instale IntelliJ ultimate

1.3 Base de Datos:

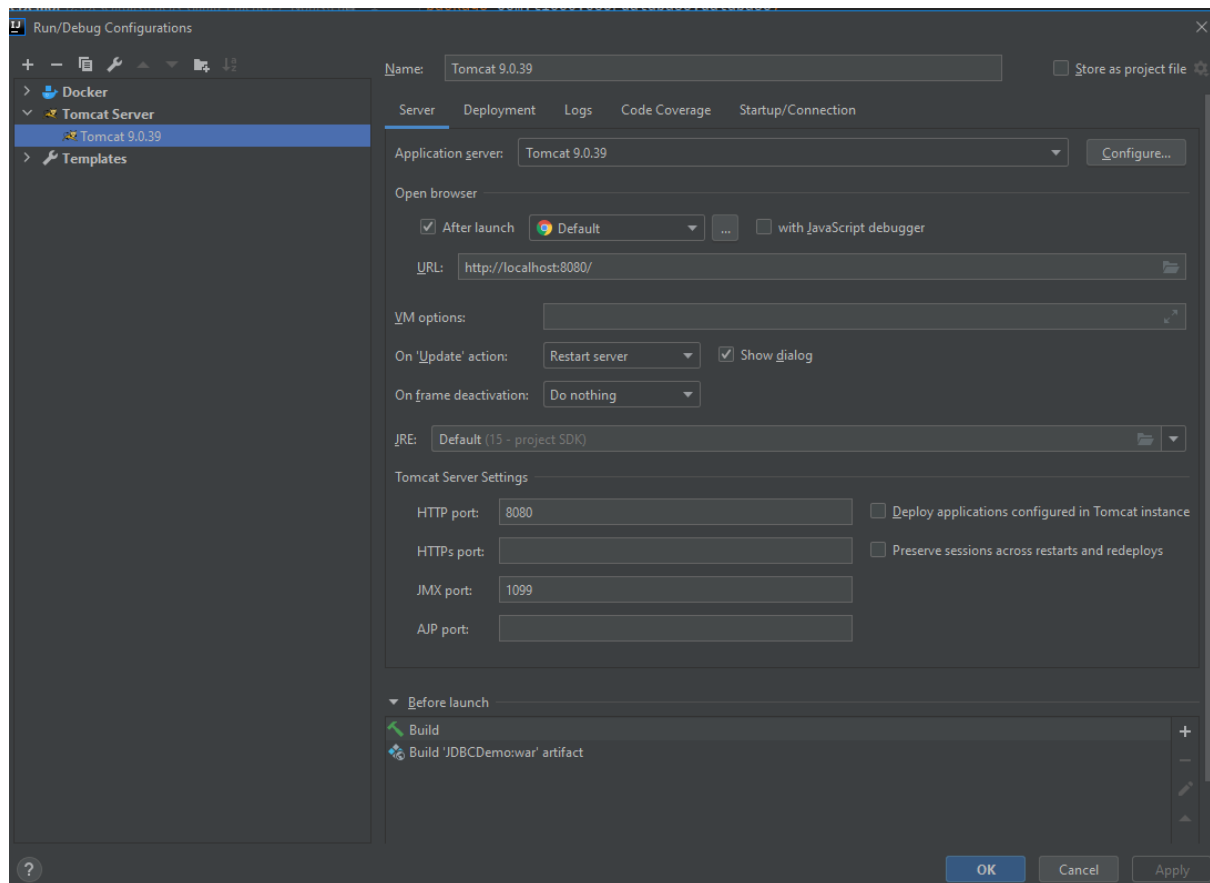
La base de datos viene se encuentra en la carpeta DB y en el root de la aplicación incluye un docker-compose que la pone en marcha fácilmente en el caso de que se quiera hacer uso de otra en la clase Database podemos cambiar el url de la que proviene por defecto el usuario es root y contraseña root y se hospeda en el puerto 3036 el propio docker compose incluye también un phpmyadmin en el puerto 8081 es accesible desde <http://localhost:8081> al hacer un docker-compose up de el docker.

1.4 Hacer Deploy

En el caso de usar IntelliJ ultimate para hacer deploy he utilizado el preset para tomcat que incluye.



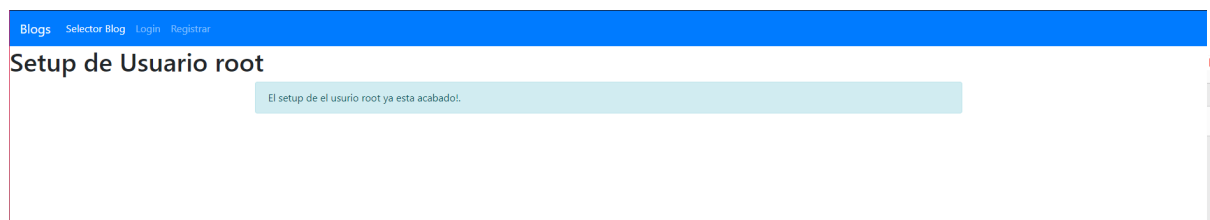
Solo ha sido necesario incluir el war de forma manual con el plus de la parte de el preset.



Opcional:

1.5 Usuario Root de la aplicación

De forma opcional el administrador de el sitio web puede crear el usuario Root el cual tiene todos los permisos de la aplicación (crear blogs, crear post en blogs que no son suyos, borrarlos y moderar) esto es posible en <http://localhost:8080/setup> el cual solicitara que le pongas una contraseña al usuario en caso de ya existir soltara este mensaje una vez creado esto ya está finalizado la instalación del servidor. El usuario root en caso de usar la DB anterior tiene la contraseña Root123.

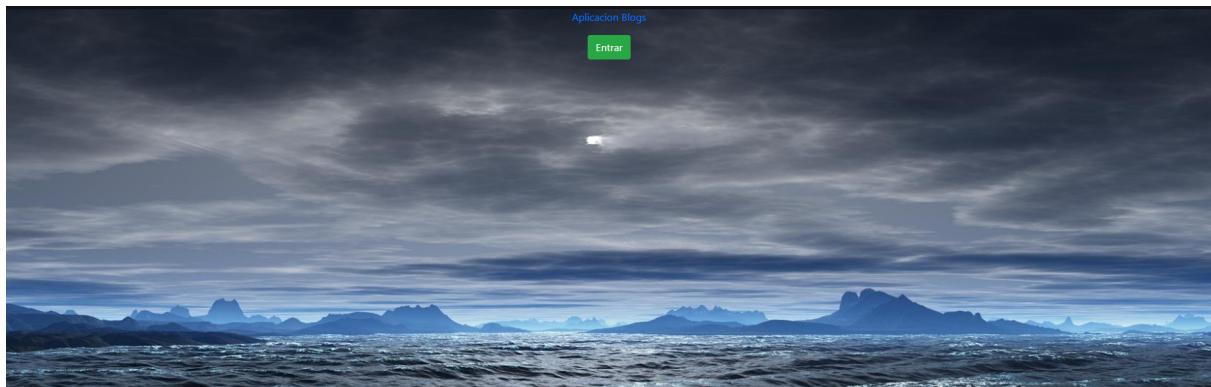


2.Manual de uso

En este apartado explicaremos el funcionamiento general de la aplicación para un usuario de ella.

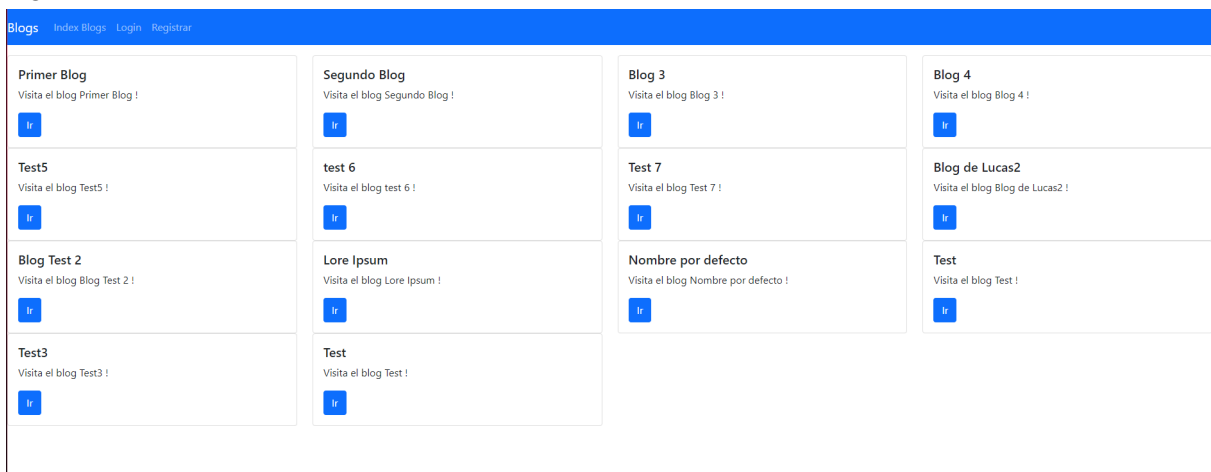
2.1 Inicio

Al iniciar la aplicación veremos un menú simple bastante explicativo al darle a entrar nos dirigirá al índice de blogs.

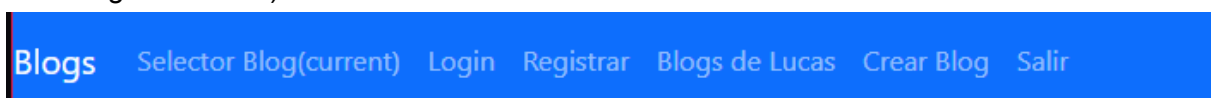


2.2 Indice de blog

Todos los blogs creados por los usuarios se encontrarán posteados aquí los blogs son accesibles por usuarios tanto logueados como no en la barra superior en todas estas pestañas el usuario no logueado siempre tiene la opción de volver al índice , loguearse o registrarse.



Cuando un usuarios e registra y se loguea sus opciones se expanden con la opcion de ir a su pagina privada de blogs , crear un blog nuevo o dejar la sesion y salir(estos hara que se ha redirigido al inicio).



2.3 Indice de blogs propios

La parte privada es muy parecida a la anterior, la diferencia es que muestra solo los blogs que son creados por ti y que por ende tienes permisos de gestionar para un facil acceso.

Blogs Selector Blog(current) Login Registrar Blogs de Lucas Crear Blog Salir

Blogs de Lucas:

Primer Blog Visita tu blog Primer Blog ! Ir	Segundo Blog Visita tu blog Segundo Blog ! Ir	Blog 3 Visita tu blog Blog 3 ! Ir
Blog 4 Visita tu blog Blog 4 ! Ir	Test5 Visita tu blog Test5 ! Ir	test 6 Visita tu blog test 6 ! Ir
Nombre por defecto Visita tu blog Nombre por defecto ! Ir	Test Visita tu blog Test ! Ir	Test3 Visita tu blog Test3 ! Ir

En caso de no tener blogs te pone este mensaje.

Blogs de root:

Parece que no tienes aun blogs prueba a hacer alguno!

2.4 Login

Es un menú de login en el que solicita la información que diste en el registro en el caso de que algo esté mal se pondrá un mensaje acorde . Si introduces la información correcta eres re enviado al menú anterior de blogs personales.

Login

Name:

Pass:

Submit

Please enter username

2.5 Registro

Es el registro que todos los usuarios han de hacer en él se solicita un nombre de usuario el cual tiene que ser único y una contraseña que se codifica, en el caso de que falten campos o no se cumplan los requisitos sale un mensaje explicativo.

[Blogs](#) [Selector Blog](#) [Login](#) [Registrar](#) [Blogs de root](#) [Crear Blog](#) [Salir](#)

Registro Usuario

Nombre:


Contraseña(Entre 5 y 25 con Mayuscula,miniscula y numero):

Submit

Usuario no valido

Nombre:

Contraseña(Entre 5 y 25 con Mayuscula,miniscula y numero):

 Completa este campo

2.6 Registro de blog: Para registrar un blog solo es requerido el nombre de él mismo una vez creado ya se inserta automáticamente en la base de datos.

[Blogs](#) [Selector Blog](#) [Login](#) [Registrar](#) [Blogs de root](#) [Crear Blog](#) [Salir](#)

Registro Blog

Nombre:

Submit

2.7 Ver blogs

Esta es la pestaña que se ve al pulsar un blog en caso de ser el usuario creador del blog o el usuario root. Las opciones de el blog de añadir un post o aprobar comentarios anónimos los post creados con su fecha de publicación y una pequeña preview de lo que contienen , además de las opciones de borrarlo o editarlo.

Blogs [Selector Blog](#) [Login](#) [Registrar](#) [Blogs de root](#) [Crear Blog](#) [Salir](#)

Primer Blog

[Añadir Post +](#) [Aprobar Comentarios 2](#)

Lore Ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam nec neque fermentum mi placerat porta. Phasellus eu vulputate erat. Cras id lorem quis dolor molestie vestibulum. F

Aa

- Elemento 1
- Elemento 2
- Elemento 3

Vestibulum quis lobortis justo. Pellentesque cursus, lectus vitae gravida ornare, neque nunc eleifend erat, vitae molestie nibh mi id sapien. Nam justo tortor, tempus ut sapien quis, i

In hac habitasse platea dictumst. Aliquam accumsan, ipsum quis vestibulum scelerisque, lorem nisi elementum turpis, ut fringilla est nibh ac lacus. Suspendisse imperdiet tempus o

Nunc aliquam, mauris vehicula facilisis efficitur, massa mi varius purus, eget iaculis nunc nibh quis mi. Maecenas ac fermentum metus, iaculis cursus nibh. Nam dictum aliquam finil

Aliquam erat volutpat. Aenean scelerisque sed enim at imperdiet. Duis vestibulum at sem vitae elementum. Praesent in ipsum risus. Pellentesque sed nisi congue, elementum turpi

Publicado Thu May 20 13:34:45 CEST 2021

[Borrar Post](#) [Editar Post +](#)

Blog test

1. First item
2. Second item
3. Third item
 1. Indented item
 2. Indented item
4. Fourth item

Publicado Sun May 30 19:36:34 CEST 2021

[Borrar Post](#) [Editar Post +](#)

2.8 Nuevo Post al darle a crear un nuevo post en nuestro blog nos saldrá este pequeño formulario en el que se pedirá el título de nuestro post y el cuerpo de lo que contiene tanto el cuerpo como título serán renderizados con una librería markdown así que el usuario lo conoce puede darle un poco de estilo a su post.

Blogs [Selector Blog](#) [Login](#) [Registrar](#) [Blogs de root](#) [Crear Blog](#) [Salir](#)

Nuevo Post

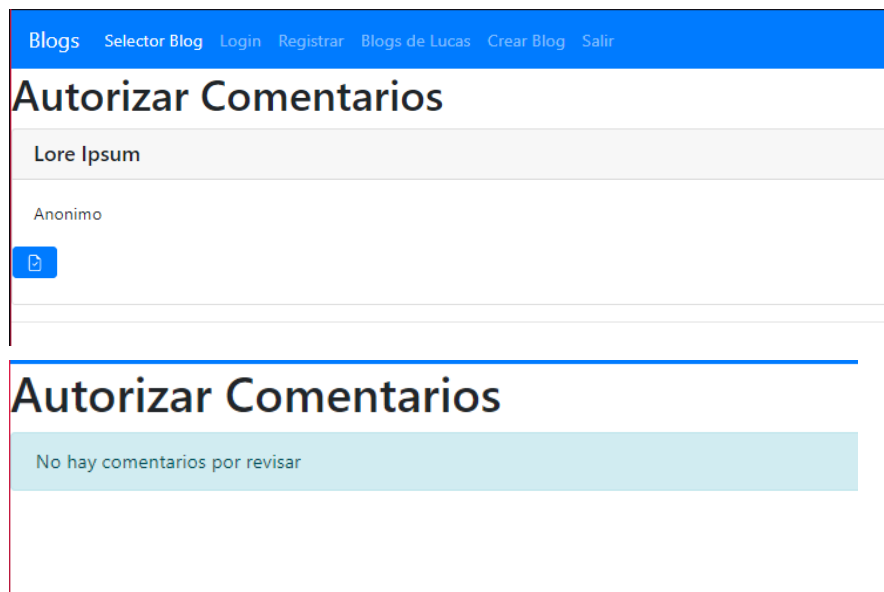
Título Post:

Cuerpo Post:

2.9 Autorizar comentarios

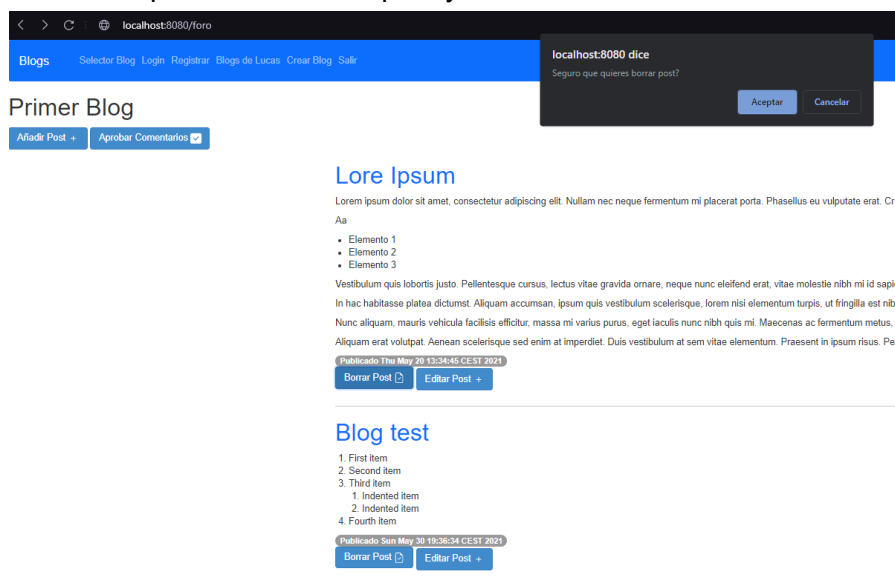
La pestaña autorizar comentarios nos muestra los comentarios anónimos creados en nuestros post (por defecto serán ocultos para evitar comentarios no deseados) y el administrador del blog podrá aceptarlos en el caso de que este de acuerdo, el comentario indica de qué post es y lo que incluye.

En el caso de no tener comentarios por revisar verá un comentario acorde.



2.10 Borrar Post

En el caso de querer borrar un post al darle a borrar post te saldrá una confirmación y al darle a aceptar se borrará el post y todos los comentarios asociados al mismo.



2.11 Modificar Post

En el caso de darle a modificar saldrá el post en sí con cómo había sido escrito es decir con su formato markdown original , en este se puede ver como tiene incluido contenido malicioso como scripts pero estos son eliminados a la hora de mostrarlo y están aquí sólo de demostración.

Blogs

Selector Blog

Login

Registrar

Blogs de Lucas

Crear Blog

Salir

Update Post

Título Post:

Lore Ipsum <script> alert("hep"); </script>

Cuerpo Post:

<script> alert("hep"); </script>

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam nec neque fermentum mi placerat porta. Phasellus eu vulputate erat. Cras id lorem quis dolor molestie vestibulum. Pellentesque sapien lectus, suscipit luctus lectus eget, facilisis ultrices urna. Quisque ornare nec nulla et aliquet. Cras tempor, libero sed volutpat mollis, turpis diam lacina justo, sit amet gravida eros felis a ligula. Morbi id quam lacus. Morbi consequat pellentesque lorem, a facilisis quam commodo non. Nunc id tortor eu lacus imperdiet volutpat eget in sem. Cras non hendrerit tellus, ac tempor ligula. Aenean id bibendum odio. Aenean quis lorem id est dignissim porttitor et id augue. Proin varius magna id elit gravida, sed finibus velit tempus. Sed rutrum tincidunt augue, at egestas neque. Praesent aliquam lacus enim.

Aa

- Elemento 1

- Elemento 2

- Elemento 3

Vestibulum quis lobortis justo. Pellentesque cursus, lectus vitae gravida ornare, neque nunc eleifend erat, vitae molestie nibh mi id sapien. Nam justo tortor, tempus ut sapien quis, consequat bibendum purus. Donec vel est at ligula porta dapibus eget sed lacus. Vestibulum ac volutpat urna, nec feugiat nibh. Cras faucibus sit amet turpis eget fringilla. Donec nec euismod risus, aliquet accumsan velit.

In hac habitasse platea dictumst. Aliquam accumsan, ipsum quis vestibulum scelerisque, lorem nisi elementum turpis, ut fringilla est nibh ac lacus. Suspendisse imperdiet tempus odio, vel condimentum mauris hendrerit sed. Donec ligula risus, eleifend a imperdiet a, pellentesque sed dolor. Donec semper in tortor ut tincidunt. Phasellus id odio venenatis, molestie tortor vitae, commodo arcu. Etiam non augue urna. Nulla venenatis suscipit cursus. Vivamus egestas sapien a metus suscipit, et congue massa consectetur. Integer in interdum diam, et finibus mi. Donec eget elit felis. Morbi tortor diam, sollicitudin vitae fermentum in, auctor euismod lorem. Aliquam in neque eget nibh consectetur pulvinar. Fusce imperdiet in lorem et ultrices.

Nunc aliquam, mauris vehicula facilisis efficitur, massa mi varius purus, eget laculis nunc nibh quis mi. Maecenas ac fermentum metus, laculis cursus nibh. Nam dictum aliquam finibus. Aenean id sodales nunc. Curabitur neque lorem, ornare sed neque et, aliquet tincidunt justo. In consequat, nulla et dapibus facilisis, sapien nulla dapibus sem, at semper enim ex vitae est. Suspendisse scelerisque porttitor nisi, a molestie elit tristique sit amet. Suspendisse scelerisque eu nisi nec sollicitudin. Aliquam porttitor sem urna.

Aliquam erat volutpat. Aenean scelerisque sed enim at imperdiet. Duis vestibulum at sem vitae elementum. Praesent in ipsum risus. Pellentesque sed nisl congue, elementum turpis et, mattis augue. Nullam et lacus et lectus malesuada egestas. Orci varius natoque penatibus et magnis dis parturient

Submit

2.12 vista post

Al darle un post lo veremos agradando para facilitar la lectura de post mas grandes con su fecha de creación y el markdown utilizado.

Blogs

Selector Blog

Login

Registrar

Blogs de Lucas

Crear Blog

Salir

Lore Ipsum

Date | Thu May 20 13:34:45 CEST 2021

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam nec neque fermentum mi placerat porta. Phasellus eu vulputate erat. Cras id lorem quis dolor molestie vestibulum. Pellentesque sapien lectus, suscipit luctus lectus eget, facilisis ultrices urna. Quisque ornare nec nulla et aliquet. Cras tempor, libero sed volutpat mollis, turpis diam lacina justo, sit amet gravida eros felis a ligula. Morbi id quam lacus. Morbi consequat pellentesque lorem, a facilisis quam commodo non. Nunc id tortor eu lacus imperdiet volutpat eget in sem. Cras non hendrerit tellus, ac tempor ligula. Aenean id bibendum odio. Aenean quis lorem id est dignissim porttitor et id augue. Proin varius magna id elit gravida, sed finibus velit tempus. Sed rutrum tincidunt augue, at egestas neque. Praesent aliquam lacus enim.

• Elemento 1

• Elemento 2

• Elemento 3

Vestibulum quis lobortis justo. Pellentesque cursus, lectus vitae gravida ornare, neque nunc eleifend erat, vitae molestie nibh mi id sapien. Nam justo tortor, tempus ut sapien quis, consequat bibendum purus. Donec vel est at ligula porta dapibus eget sed lacus. Vestibulum ac volutpat urna, nec feugiat nibh. Cras faucibus sit amet turpis eget fringilla. Donec nec euismod risus, aliquet accumsan velit.

In hac habitasse platea dictumst. Aliquam accumsan, ipsum quis vestibulum scelerisque, lorem nisi elementum turpis, ut fringilla est nibh ac lacus. Suspendisse imperdiet tempus odio, vel condimentum mauris hendrerit sed. Donec ligula risus, eleifend a imperdiet a, pellentesque sed dolor. Donec semper in tortor ut tincidunt. Phasellus id odio venenatis, molestie tortor vitae, commodo arcu. Etiam non augue urna. Nulla venenatis suscipit cursus. Vivamus egestas sapien a metus suscipit, et congue massa consectetur. Integer in interdum diam, et finibus mi. Donec eget elit felis. Morbi tortor diam, sollicitudin vitae fermentum in, auctor euismod lorem. Aliquam in neque eget nibh consectetur pulvinar. Fusce imperdiet in lorem et ultrices.

Nunc aliquam, mauris vehicula facilisis efficitur, massa mi varius purus, eget laculis nunc nibh quis mi. Maecenas ac fermentum metus, laculis cursus nibh. Nam dictum aliquam finibus. Aenean id sodales nunc. Curabitur neque lorem, ornare sed neque et, aliquet tincidunt justo. In consequat, nulla et dapibus facilisis, sapien nulla dapibus sem, at semper enim ex vitae est. Suspendisse scelerisque porttitor nisi, a molestie elit tristique sit amet. Suspendisse scelerisque eu nisi nec sollicitudin. Aliquam porttitor sem urna.

Aliquam erat volutpat. Aenean scelerisque sed enim at imperdiet. Duis vestibulum at sem vitae elementum. Praesent in ipsum risus. Pellentesque sed nisl congue, elementum turpis et, mattis augue. Nullam et lacus et lectus malesuada egestas. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Sed aliquam magna risus, eget fermentum dolor rutrum quis. Cras in purus at lorem maximus fringilla. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Curabitur erat orci, commodo at massa at, imperdiet convallis sapien. Pellentesque at magna lorem. Etiam lectus lacus, volutpat dapibus lectus et, posuere pellentesque quam. Nunc varius mattis lobortis. Aliquam aliquam scelerisque nisl, nec pulvinar mauris blandit et. Phasellus enim magna, luctus vel dolor ut, vehicula dignissim nunc.

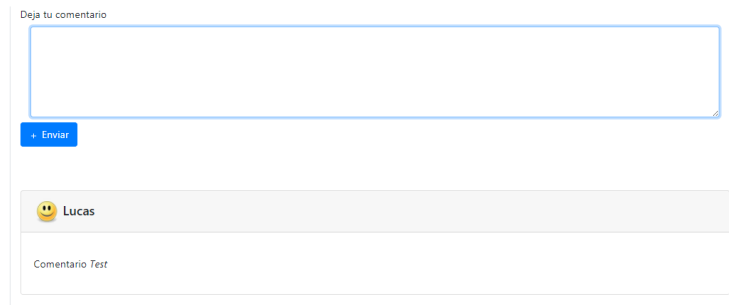
Comments

Deja tu comentario

2.13 vista Comentarios

Debajo de cada post tenemos una sección de comentarios relacionados a este , los usuarios tanto anónimos como logueados pueden dejar sus opiniones del post y leer la de los demás.

Comments



The screenshot shows a web interface for comments. At the top, there is a text input field with the placeholder text "Deja tu comentario". Below the input field is a blue button with a white plus sign and the text "Enviar". Below the button is a comment entry. The entry has a grey header bar containing a yellow smiley face icon and the name "Lucas". Below the header bar is a text area containing the text "Comentario Test".

3.Estructura

En esta parte explicaremos cómo funciona internamente la aplicación y qué recursos han sido usados.

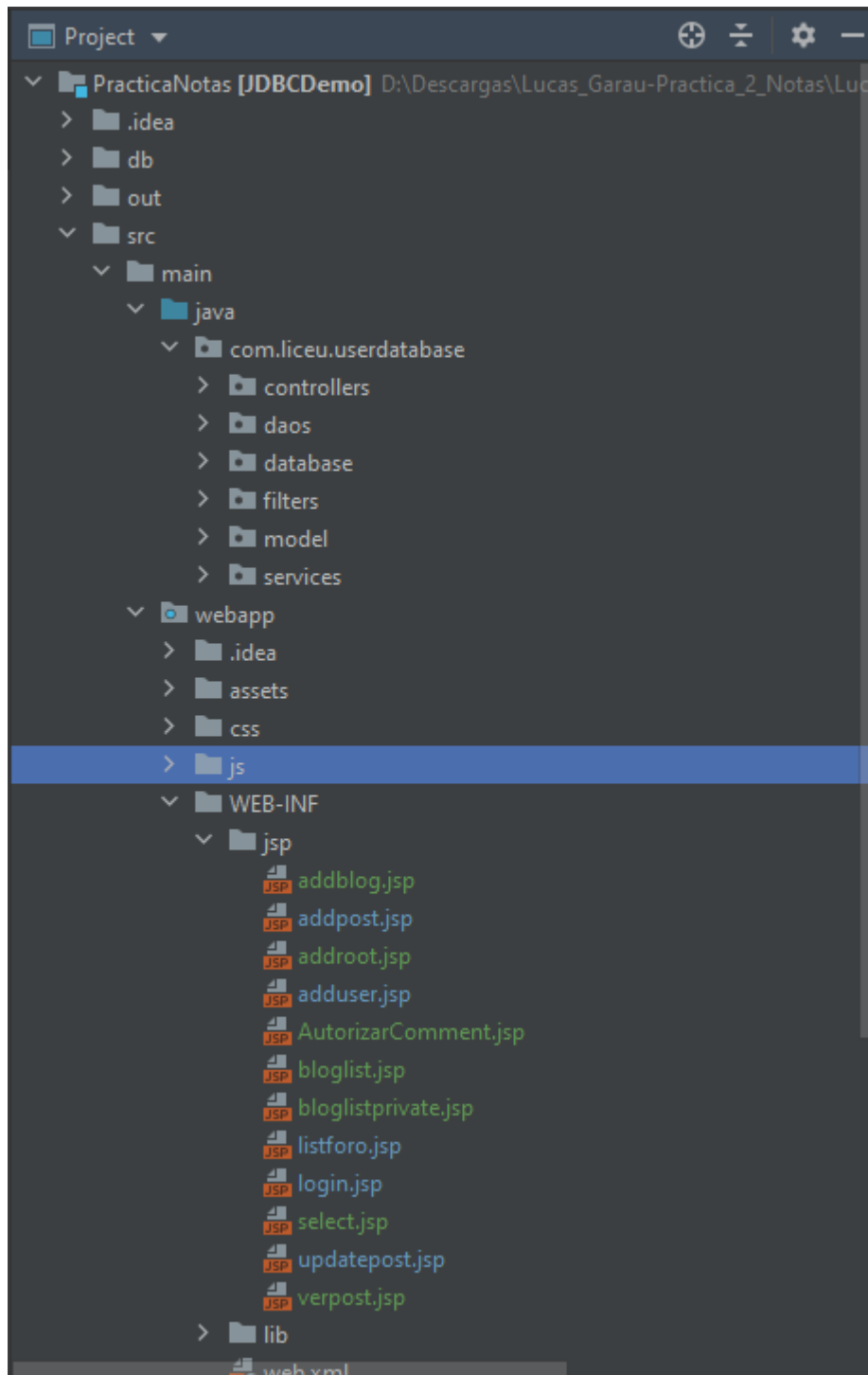
3.1 Docker Mysql

El encargado de tener permanencia de información en este caso a sido este docker de mysql incluido con el proyecto basta con hacer un docker-compose up en la carpeta del proyecto para que funcione.

```
1  services:
2
3  db:
4      image: mysql:8
5      volumes:
6          - ./db:/var/lib/mysql
7      environment:
8          MYSQL_ROOT_PASSWORD: root
9          MYSQL_DATABASE: blog
10     ports:
11         - "3306:3306"
12
13     phpmyadmin:
14         depends_on:
15             - db
16         image: phpmyadmin/phpmyadmin
17         ports:
18             - '8081:80'
19         environment:
20             PMA_HOST: db
21             MYSQL_ROOT_PASSWORD: root
22
```

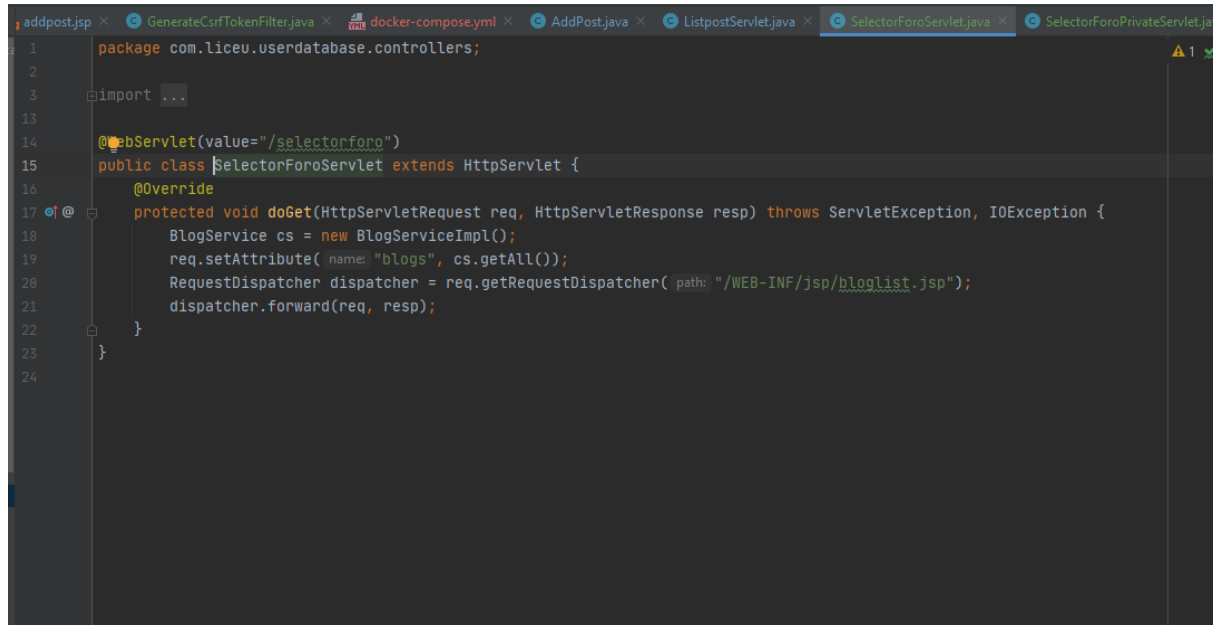
3.2 Estructura de la aplicación:

La aplicación se ha utilizado una estructura por capas para separar la lógica , del acceso a datos y la parte visual las diferentes capas :



3.2.1: Controladores

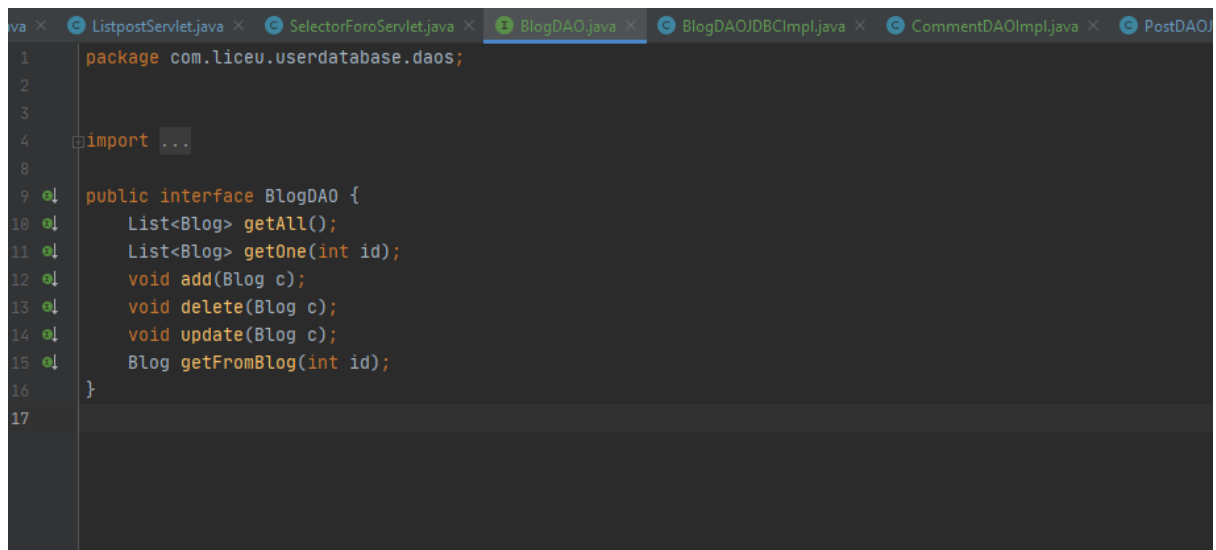
Los controladores son los encargados de recibir las instrucciones del usuario y mostrarle la vista solicitada extraen la información ofrecida por el usuario y se la pasa a los servicios ellos mismos no modifican esa información una vez ejecutan la petición pasan al usuario a la vista, hay un controlador por cada vista.



```
1 package com.liceu.userdatabase.controllers;
2
3 import ...
4
5 @WebServlet(value="/selectorforo")
6 public class SelectorForoServlet extends HttpServlet {
7     @Override
8     protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
9         BlogService cs = new BlogServiceImpl();
10        req.setAttribute("name: \"blogs\", cs.getAll());
11        RequestDispatcher dispatcher = req.getRequestDispatcher("path: \"/WEB-INF/jsp/bloglist.jsp");
12        dispatcher.forward(req, resp);
13    }
14}
```

3.2.2 DAO

Son los encargados de interactuar con la base de datos en el proyecto y tenemos uno por cada modelo además de su implementación, reciben la información del service y les notifica si la operación con la base de datos a funcionado o no en el caso de que no disparara un error hasta la vista para mostrárselo al usuario.



```
1 package com.liceu.userdatabase.daos;
2
3 import ...
4
5 public interface BlogDAO {
6     List<Blog> getAll();
7     List<Blog> getOne(int id);
8     void add(Blog c);
9     void delete(Blog c);
10    void update(Blog c);
11    Blog getFromBlog(int id);
12}
13
14
15
16
17
```

3.2.3 Database

Poseemos dos clases database una anterior que usa Sqlite y la más actual que utiliza Mysql esta clase simplemente le da la conexión a el programa con la base de datos en caso de querer usar una conexión diferente no incluida se añadirá una clase base de datos.

```
BlogDAO.java x BlogDAOJDBCImpl.java x CommentDAOImpl.java x PostDAOJDBCImpl.java x SelectorForoPrivateServlet.java x Database.java x DatabaseMySQL.java x
1 package com.liceu.userdatabase.database;
2
3 import ...
4
5
6 public class Database {
7     private static Connection connection;
8
9     @
10    public static Connection getConnection() {
11        try {
12            String url = "jdbc:mysql://localhost:3306/blog?user=root&password=root";
13            if (connection == null) {
14                Class.forName("com.mysql.cj.jdbc.Driver");
15                connection = DriverManager.getConnection(url, "root", "root");
16            }
17            return connection;
18        } catch (Exception e) {
19            e.printStackTrace();
20        }
21        return null;
22    }
23 }
```

3.2.4 Filters

La aplicación incluye dos filtros uno de login que impide el acceso a paginas especificas de la parte privada y otro de CSRF para las peticiones ambos van incluidos en el paquete filtros.

```
AOImpl.java x PostDAOJDBCImpl.java x SelectorForoPrivateServlet.java x Database.java x CheckCsrfTokenFilter.java x GenerateCsrfTokenFilter.java x LoginFilter.java x
1 package com.liceu.userdatabase.filters;
2
3 import ...
4
5
6
7
8
9
10 @WebFilter(value = {"/addblog", "/selectorforoprivate", "/addpost", "/autocomments", "/updatepost"})
11 public class LoginFilter implements Filter {
12
13     @Override
14     public void doFilter(ServletRequest req, ServletResponse res, FilterChain chain) throws ServletException, IOException {
15         HttpServletRequest request = (HttpServletRequest) req;
16         HttpServletResponse response = (HttpServletResponse) res;
17         HttpSession session = request.getSession(create: false);
18         String loginURI = request.getContextPath() + "/login";
19
20         boolean loggedIn = session != null && session.getAttribute(name: "user") != null;
21         boolean loginRequest = request.getRequestURI().equals(loginURI);
22
23         if (loggedIn || loginRequest) {
24             chain.doFilter(request, response);
25         } else {
26             response.sendRedirect(loginURI);
27         }
28     }
29 }
30 }
```

3.2.5 Modelos

La aplicación incluye 4 modelos necesarios para la misma estos son los objetos que manejamos en ella :

```
Blog.java x
1 package com.liceu.userdatabase.model;
2
3 public class Blog {
4     private int id;
5     private String Nom;
6     private int userid;
7
8     public Blog() {
9     }
10
11     public Blog(int id, String nom, int userid) {
12         this.id = id;
13         Nom = nom;
14         this.userid = userid;
15     }
16
17     public int getId() { return id; }
18
19     public void setId(int id) { this.id = id; }
20
21     public String getNom() { return Nom; }
22
23     public void setNom(String nom) { Nom = nom; }
24
25     public int getUserid() { return userid; }
26
27     public void setUserid(int userid) { this.userid = userid; }
28 }
29
30
31
32
33
34
35
36
37
38
39
40
41
```

3.2.6 Servicios

Los servicios son utilizados por el controlador para que modifiquen la información y se la pasen a los datos ellos también manejan el control de errores en caso de que algo no suceda bien entre las capas.

```
Blog.java x BlogService.java x
1 package com.liceu.userdatabase.services;
2
3 import ...
4
5
6
7
8 public interface BlogService {
9     boolean addBlog(String BlogName, int userid);
10    Blog find(String username, String password);
11    List<Blog> getAll();
12    List<Blog> getOne(int id);
13    Blog getfromblog(int id);}
14
15
```

3.2.7 JSP

Para cada vista hemos usado un jsp por cada página utilizada en la pagina estas son servidas por los controladores.

