

Pagina Index.html

Pag 1	Nombre: index	Descripción: Pagina inicial	Tipo: Single Page Application
-------	---------------	-----------------------------	-------------------------------

Body

Este contiene de fondo la imagen **image_1.png** dentro de la carpeta **image** de templates

Su estructura es la siguiente:

header	nav	header
reservada	section	
	footer	

- Dentro de la cabecera estará nuestro **nav** con las opciones de menú
- En **section** se mostrara nuestro contenido
- en **footer** se encontrara nuestro pie de pagina
- La parte **reservada** se utilizara posteriormente para las opciones de usuario logeado que se mostrara en otro HTML.

Barra superior

1. Logo
 - Link a pagina principal
2. Home
 - Link a indice
3. Servicios
 - Listado de los servicios activos
4. Contacto
 - Abre el correo predeterminado del usuario y crea un nuevo correo con la dirección del destinatario
5. About
 - Muestra una descripción de lo que es un banco de tiempo
6. Acceso Login
 - Formulario de identificacion
7. Acceso Registro
 - Formulario de registro

Logo

El modelo de Logo de la pagina ha sido creado con Cinema 4D (entorno 3D) y renderizado en secuencia de cuadros en canal alfa y con animación de rotación en 90 frames. Posteriormente se exporto a Photoshop en capas para generar una GIF animado.

C4D es capaz de renderizar y exportar a un formato mp4, pero al pasarlo a GIF pierde bastante calidad, por eso se ha usado la tecnica anteriormente comentada.

Características

Los accesos mediante las opciones de Servicios, about, login y registro son de tipo Single Page Application. Solo puede convivir uno a la vez y en cualquiera de los casos, si se cliquea fuera de su contenedor, vuelve al estado de inicio. En este estado se mostrará un carrusel con diferentes imágenes con información.

El contralador de este proceso estará alojado en la carpeta **js** del proyecto con el nombre **scriptSPA.js**. Este recibirá las siguientes **id's** desde el HTML y direccionará los correspondientes **div's**.

href	id petición	file controller	file css	div html
servicios	selecService	scriptSPA.js	styleServicios.css	id= " cont_servicios "
about	aboutform	scriptSPA.js	style.css	class=" about "
login	loginform	scriptSPA.js	stylelogin.css	class=" login "
registro	registerform	scriptSPA.js	styleregistro.css	class=" cont_register "
carrusel	presente sin las llamadas anteriores	scriptSPA.js	stylecarrusel.css	class=" cont_carrusel "

En la anterior tabla también se muestra el nombre del archivo css que le da estilo a cada elemento. Estos archivos estarán alojados en la carpeta CSS en el template.

Formulario Servicios

En el contenido de section se mostraran 8 apartados para la selección de categorías del servicio. Estos apartados están identificados con su nombre y un icono (diseñado con Photoshop) representativo de cada servicio. Su función es por tanto la de pedir al controller el listado de trabajos que esperan ser realizados de esa categoría.

El listado resultante se visualizara dentro del contenido de la section. Estos no podrán ser aceptados si previo Login.

Formulario Login

La identificación de usuario sera a través de un correo y una contraseña, previamente registrada. Con ello daremos paso a un segundo HTML llamado **sesion.html**.

Los dos campos son obligatorios por lo que dará mensajes de error si se intenta enviar sin estar completados. Si no se refresca la pagina, los datos que se habían rellenado pero aun no se han enviado, se mantendrán en el formulario.

Formulario Registro

Primer contenedor de registro

Las entradas disponibles son las siguientes

- Nombre
- Contraseña
- Confirmar contraseña
- Email
- Teléfono (opcional)
- Ubicación
- Fecha nacimiento (selección manual o por calendario predefinido)

```
<div class="cont_register">
  <div class="register1">
```

*y requiere de un controlador que crea una ArrayList de servicios. Este esta realizado JavaScript dentro de la carpeta **js** con el nombre de **AddService.js** *

NOTA: **se ha rechazado la idea de usar este JS por la manipulación de la Array al trabajar con el controller de JAVA**

Footer

Como pie de pagina tendremos un contenedor con los diferentes Logotipos del centro del estudio y de las subvenciones que han ayudado a crear este curso, y un pequeño scroll de texto de las partes implicadas en el desarrollo del proyecto.

Este footer sera común en todas las paginas de contenga este proyecto

Pagina sesion.html

Pag 2	Nombre: sesion.html	Descripción: funcionalidades del usuario registrado	Tipo: Link/Servlet
-------	---------------------	---	--------------------

Body

Sobre el mismo fondo de imagen **image_1.png** dentro de la carpeta **image** de templates

Su estructura es la siguiente:

header	nav	header
opciones de sesion	section	
	footer	

Barra superior

1. Logo
 - Link a pagina principal
2. Home
 - Link a indice
3. Servicios
 - Listado de los servicios activos
4. Contacto
 - Abre el correo predeterminado del usuario y crea un nuevo correo con la dirección del destinatario
5. About
 - Muestra una descripción de lo que es un banco de tiempo
6. Acceso Logout (cierra sesión)

Panel de Opciones de sesión

Este panel esta dividido en varias partes

1. Zona superior destinada a la información del usuario

- Nombre usuario
- Computo de horas acumuladas
- Foto de perfil (pendiente de habilitar en futuras versiones)
- Lista de categorías de servicios que el usuario esta dispuesto ha realizar, representadas por su icono correspondiente
- Editar perfil. En la zona de contenido aparecerá el formulario de registro con los datos del usuario rellenados por defecto.

2. Servicios

- Servicios activos . Volverá a salir en el contenido los iconos de selección de categoría, con la diferencia que se pueden aceptar

3. Pendiente Aceptación

- Por Usuario. Servicios que donde otro usuario esta dispuesto a prestar el servicio que solicitaste, y espera que aceptes para iniciarlo
- Por Demandante. Es el caso contrario, eres tu el que espera ser aceptado para iniciar el trabajo-

4. Pendientes realización

- Trabajos Pendientes. Servicios ya aceptados que solo espera que los realices. Estos esta vinculados con una barra de porcentaje de trabajos realizados y pendientes

5. Histórico

- Trabajos Realizados. Historial de trabajos que ya has concluido.

Cada apartado tiene a su izquierda un pequeño icono indicativo de sus cualidades. En realidad se trata de una fuente de letra con estilo gráfico. En este link están todas las posibilidades posibles [fontawesome\[1\]](#)
También se ha añadido la función de mostrar el numero de peticiones pendientes dentro de un circulo a la derecha de cada una de ellas

Pagina addjobs.html

Segundo contenedor de registro

Este bloque añade los servicios que el usuario esta dispuesto a realizar

Los campos son los siguientes:

1. Servicios (campos predefinidos en un select)

- Carpintería
- Construcción
- Educación
- Electricista
- Fontanero
- Mecánico
- Reparaciones
- Servicios doméstico

2. Descripción del servicio (breve comentario para especificar sobre el trabajo)

3. Disponibilidad (indica las variantes de disponibilidad de usuario por servicio)

4. Botón de borrar (elimina el registro de ese servicio)

5. Botón de añadir servicio (a cada pulsación se podrá añadir un servicio)

6. Botón enviar(enviara el formulario de los dos contenedores)

El contenedor en HTML es

Codigo interesante: Se van sumando los trabajos aceptados, mientras que en la parte inferior siempre tienes la opción de incrementarlos

Cada registro ya aceptado tiene la posibilidad de ser borrado.

Se usará un botón de inicio de sesión para finalizar el proceso

```

<!-------ADD JOBS ----->

<!-- Respuesta del controlador -->

<div class="cont_addjobs">
  <div class="register2">
    <div class="re_right">
      <table>
        <tr>
          <td class="job_text">SERVICIOS</td>
          <td class="job_text">DESCRIPCION</td>
          <td class="job_text">DISPONIBILIDAD</td>
        </tr>
      </table>
      <form method="post" action="listjobs">
        <table id="tableJobs">
          <tbody>
            <tr th:each="userJobs:${jobs}">
              <td class="jobs" th:text="${userJobs.category}"></td>
              <td class="jobs" th:text="${userJobs.description}"></td>
              <td class="jobs" th:text="${userJobs.disponibilidad}"></td>
              <td><a th:href="@{deleteJob(id=${userJobs.id})}"
                class="btn btn-danger btn-sm"><b>X</b></a></td>
            </tbody>
          </table>
        </form>

<!-------formulario de un nuevo registro de trabajo----->

<form class="form-inline" method="post" action="addJob">
  <label for="category"></label> <select name="category"
    id="category" required>
    <option value="carpinteria">Carpinteria</option>
    <option value="construccion">Construccion</option>
    <option value="educacion">Educacion</option>
    <option value="electricista">Electricista</option>
    <option value="fontanero">Fontanero</option>
    <option value="mecanico">Mecanico</option>
    <option value="reparaciones">Reparaciones</option>
    <option value="domesticos">S.Domesticos</option>
  </select> <label for="description"></label><input type="text"
    name="description" id="description" required> <label
    for="disponibilidad"></label><input type="text"
    name="disponibilidad" id="disponibilidad" required>

    <input type="image" src="images/enviar.png" alt="Enviar"
      width="38" height="38" >
</form>

```

```

    <form method="post" action="sesionuser">
    <div class="iniciar_sesion">
    <input type="image" src="images/iniciarSesion.png" alt="Enviar"
        width="42" height="42" >
    </div>
    </form>
    </div>
</div>
</div>

```

Pagina jobsCategory.html

Este Html se encarga mostrar el listado de trabajos en curso según la categoría que el usuario, con sesión, que desea consultar

Se ha usado un contenedor similar al resto para mantener la imagen corporativa, donde se muestran además de los trabajos en curso, los datos de los usuarios.

De esta sección hay que destacar la utilización de la estructura del for each.

En la mayoría de ejemplo este se usaba dentro de una tabla insertado dentro de un 'tr'. ¿ Pero que pasa si es necesario que actúe por igual en diferentes 'tr' ?. La solución la encontramos en la siguiente instrucción de thymeleaf que funciona como un bloque de 'tr'

```
<tr:block th:each="userJobs:${jobs}">
```

Listado de ficheros empleados en el front-end

Nombre	Ubicación	Funciones principales
index.html	timeBank /src /main /resources /templates	pg inicio, login y primer contenedor registro
addjobs.html	timeBank /src /main /resources /templates	segundo contenedor registro
sesion.html	timeBank /src /main /resources /templates	perfil usuario y acceso a trabajos
jobsCategory.html	timeBank /src /main /resources /templates	listado de servicios por categoria
bootstrap.min.css	timeBank /src /main /resources /static /css	librería de componentes
jquery-ui.css	timeBank /src /main /resources /static /css	librería de componentes
Style.css	timeBank /src /main /resources /static /css	ccs para estructura principal
Stylecarrusel.css	timeBank /src /main /resources /static /css	css para mostrar el carrusel

Stylelogin.css	timeBank/src/main/resources/static/css	css para contenedor del login
Styleregistro.css	timeBank/src/main/resources/static/css	css para los contenedores de registro
StyleServicios.css	timeBank/src/main/resources/static/css	css para los contenedores los trabajos
scriptSPA	timeBank/src/main/resources/static/js	JS para la funcionalidad de single page application

Ubicación de las imágenes. timeBank/src/main/resources/static

Imagenes	
about3.png	gencat.png
carrusel1.png	soc.png
carrusel2.png	union_europea.png
carrusel3.png	PQTM.png
albañil.png	image_fondo.png
mecanico2.png	pestaña3.png
reparaciones.png	person_1.png
carpintero.png	enviar.png
fontanero2.png	iniciaSesion.png
electricista.png	servicios domesticos.png
cabecera3.png	educacion.png
cabecera4.png	pastilla8.png
logo animado.gif	delete.png

Links

[1] fontawesome, <https://fontawesome.com/v4.7.0/icons/>