

**CFGS  
DESARROLLO DE  
APLICACIONES WEB**

# **PROYECTO FIN DE CICLO**



**FAT DOWN**  
NO HAY EXCUSAS

**GRUPO  
DW2A  
TUTOR  
MARCOS PUENTE**

**DANIEL ESTEBAN UCLÉS  
ALEJANDRO RELAÑO TORTAJADA  
AMANDA SEGURA GONZÁLEZ**

# FAT DOWNLOAD

## ÍNDICE

**4 EL PROYECTO**

**5 ANÁLISIS**

**5 ALCANCE DEL  
PROYECTO**

**5 REQUISITOS  
FUNCIONALES**

**5 REQUISITOS NO  
FUNCIONALES**

**6 DISEÑO**

**6 ESTRUCTURA - BACK END**

**10 ESTRUCTURA - FRONT END**

**12 ARQUITECTURA Y  
DIAGRAMA DE CLASES**

**14 LÓGICA DE NEGOCIO**

<b>15</b>	LÓGICA DE LA APLICACIÓN
<b>17</b>	TECNOLOGÍAS
<b>19</b>	GUÍA DE ESTILOS
<b>21</b>	IMPLEMENTACIÓN
<b>23</b>	PRUEBAS
<b>25</b>	CONCLUSIONES
<b>26</b>	ANEXOS
<b>26</b>	ANEXO 1
<b>26</b>	ANEXO 2
<b>27</b>	ANEXO 3
<b>28</b>	ANEXO 4
<b>28</b>	ANEXO 5

# EL PROYECTO **FAT DOWN**

**La aplicación web de fitness Fat Down surge como respuesta a un mundo en constante cambio al que debemos adaptarnos. La pandemia causada por el COVID-19 nos ha hecho reflexionar sobre la verdadera importancia de mantenerse activo no solo como el medio para mejorar nuestra salud física, sino también mental.**

## **ELIGE CREA PERSONALIZA ¡Y A SUDAR!**

Fat Down es una propuesta diferente, con consejos saludables para todo el mundo y con vídeos de rutinas para un entrenamiento completo, así como con ejercicios especialmente pensados y descritos por entrenadores que el usuario podrá añadir como desee a sus rutinas personalizadas.



**Durante los momentos más duros del confinamiento, nuestros hogares se convirtieron en nuestro lugar de trabajo, nuestra clase, nuestro gimnasio... En definitiva, nuestro refugio.**

**Muchos se dedicaron a perfeccionar su destreza repostera, preparando cada día un pastel diferente, o también a aprender nuevas habilidades, como tocar el ukelele.**

**Otros, en cambio, encontraron en el ejercicio físico una vía de escape ideal para sobrellevar el día a día y, por qué no, mejorar su fuerza y resistencia. Fue entonces cuando las aplicaciones de ejercicios y vídeos de entrenamiento se convirtieron en la mejor guía para conseguir nuestros objetivos.**

# ANÁLISIS

## ALCANCE DEL PROYECTO

**Los objetivos iniciales del proyecto aquí presentado consisten en el desarrollo de una aplicación multiusuario.** En ella los usuarios tendrán acceso a una variedad de ejercicios y contenidos con la que los usuarios podrán crear sus propias rutinas de ejercicio físico.

## REQUISITOS FUNCIONALES

- **Sistema multiusuario:** Cualquier persona debe ser capaz de darse de alta y, de este modo, acceder a su espacio personal donde disfrutar de los servicios que ofrece la plataforma.
- **Sistema de autorización:** La aplicación dispondrá de distintos roles que regularán el acceso a los contenidos para los usuarios.
- **Ejercicios:** El núcleo del proyecto. Los usuarios podrán buscar en la plataforma ejercicios previamente creados por usuarios administradores para después personalizar sus rutinas.
- **Rutinas:** Permiten agrupar los ejercicios para dar lugar a entrenamientos consistentes y coherentes para los usuarios, pudiendo determinar tanto la duración como las repeticiones de los ejercicios seleccionados.
- **Vídeos:** Soporte visual adicional disponible para que el usuario complemente la práctica física. Vídeos enlazados a YouTube.
- **Sistema de favoritos:** Los usuarios dispondrán de una lista personal donde almacenar fácilmente los videos de su interés.

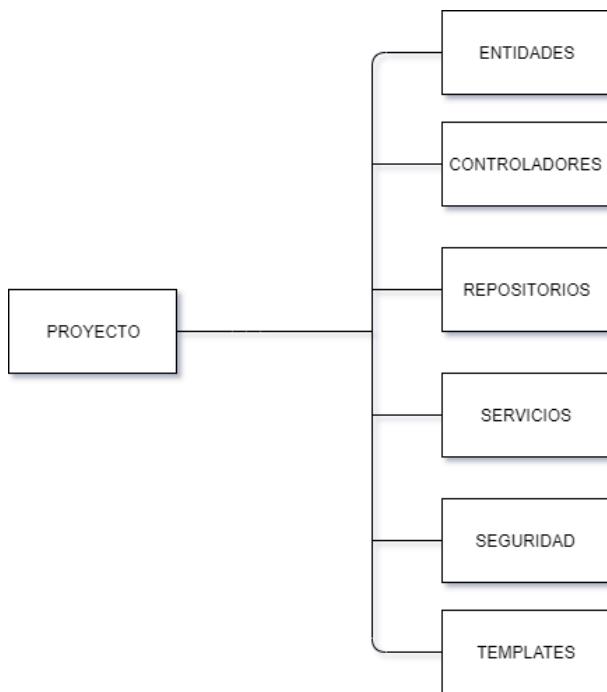
## REQUISITOS NO FUNCIONALES

- **Diseño intuitivo de la aplicación web:** Se puede acceder a las diferentes funcionalidades desde la barra de navegación siempre visible.
- **Configuración de seguridad:** Acceso restringido de usuarios según su rol. Por defecto, los usuarios se crean con rol "registrado", pudiendo acceder a funcionalidades limitadas.
- **Validación de formularios:** Validación de campos en los formularios empleados en la aplicación (creación de usuarios/ejercicios/vídeos).

# DISEÑO

## ESTRUCTURA DEL PROYECTO

**En el esquema mostrado a continuación se observan las capas que componen la estructura del proyecto en el lado del servidor, en lo referente al desarrollo del back end:**



**Entidades:** Representan los objetos que se utilizarán en el proyecto y serán almacenados en la BBDD en forma de tablas

- ✓  com.fatdown.spring.entidades
  - >  Consejo.java
  - >  Ejercicio.java
  - >  Gif.java
  - >  Imagen.java
  - >  Multimedia.java
  - >  Rol.java
  - >  Rutina.java
  - >  Usuario.java
  - >  Video.java
- ✓  com.fatdown.spring.enums
  - >  Categoria.java

**Repositories:** Suponen la capa de abstracción que permite separar el acceso a datos a la BBDD de la implementación de dicho acceso para cada entidad.

- ✓  com.fatdown.spring.repositories
  - >  ConsejoRepository.java
  - >  EjercicioPageableRepository.java
  - >  EjercicioRepository.java
  - >  GifRepository.java
  - >  ImagenRepository.java
  - >  MultimediaRepository.java
  - >  RolRepository.java
  - >  RutinaRepository.java
  - >  UsuarioRepository.java
  - >  VideoPageableRepository.java
  - >  VideoRepository.java

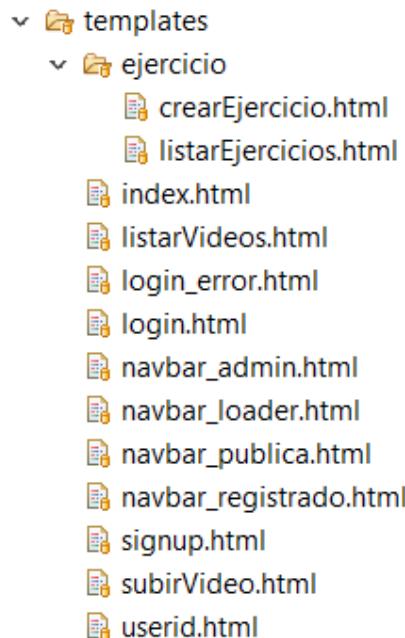
**Servicios:** La función de los servicios es implementar la lógica de negocios del modelo de dominio, realizando llamadas a los repositories.

- ✓  com.fatdown.spring.servicios
  - >  ConsejoServicio.java
  - >  ConsejoServicioImpl.java
  - >  CustomUserDetailsService.java
  - >  EjercicioServicio.java
  - >  EjercicioServicioImpl.java
  - >  GifServicio.java
  - >  GifServicioImpl.java
  - >  ImagenServicio.java
  - >  ImagenServicioImpl.java
  - >  RutinaServicio.java
  - >  RutinaServicioImpl.java
  - >  UsuarioServicio.java
  - >  UsuarioServicioImpl.java
  - >  VideoServicio.java
  - >  VideoServicioImpl.java

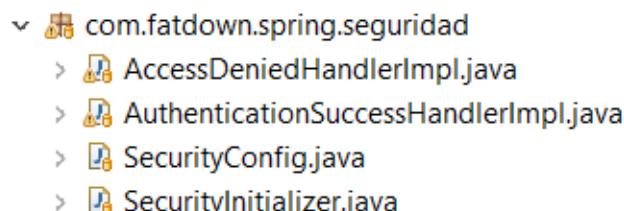
**Controladores:** Componen la capa que gestiona los datos de las entidades realizando llamadas a sus respectivos servicios, direccionando a los métodos y templates a los que se quiere acceder.

- ✓  com.fatdown.spring.controladores
  - >  EjercicioControlador.java
  - >  IndexControlador.java
  - >  MultimediaControlador.java
  - >  RutinaControlador.java
  - >  UsuarioControlador.java

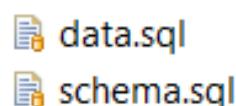
**Templates:** Conforman la capa en la que se volcarán los datos gestionados en los controladores, que se mostrarán por pantalla al usuario, de modo que pueda interaccionar con las distintas funcionalidades del proyecto.



**Seguridad:** En esta parte se contienen las clases que permiten al proyecto validar y restringir los distintos niveles de acceso a los endpoints del proyecto en base a roles, así como también regula la creación de usuarios.



**BBDD:** También se han introducido ficheros .sql para generar las tablas necesarias en la base de datos de nuestro proyecto (schema.sql) así como incluir algunos registros previos de usuarios y vídeos que servirán como ejemplo (data.sql).



**A continuación, se puede observar el recorrido a través de las clases y vistas del lado cliente, así como las llamadas que se realizan los diferentes métodos entre sí en el proceso de creación de un usuario en la aplicación web, hasta que es persistido en la base de datos (línea 54 - UsuarioServiceImpl.java).**

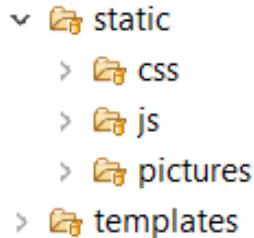
The screenshot shows a Java IDE with three open files:

- UsuarioControlador.java**: Contains methods for handling "/signup" and "/login". The "/signup" method calls `usuarioServicio.crearUsuario(usuario, request)`.
- signup.html**: An HTML form with a POST action of `@{/usuario/signup}`. It includes fields for Nombre y apellido, Email, and Password.
- UsuarioServiceImpl.java**: An implementation of the `UsuarioService` interface. The `crearUsuario` method concatenates date parameters from the request into a single String, sets the concatenated date to the user object, encodes the password using `bCryptPasswordEncoder`, finds a role by ID, adds it to the user, and finally saves the user to the repository.

**Ejemplo de creación de usuario con el método del controlador, el servicio y la llamada desde el formulario**

## FRONT END

**En la imagen mostrada a continuación se observan los componentes de la estructura del proyecto en el lado del cliente, en lo referente al desarrollo del front end:**



**Con el fin de mejorar la experiencia del usuario a la hora de navegar por las distintas vistas y páginas de la aplicación web, se han integrado entre las carpetas del proyecto hojas de estilo CSS así como JS para implementar el diseño de Bootstrap elegido al cual haremos referencia en el apartado reservado a la guía de estilos.**

**También se han introducido otros elementos estáticos, como imágenes que han sido utilizadas en las diferentes vistas y páginas de la aplicación web, tales como el logotipo de la marca o las divertidas imágenes que acompañan a los consejos saludables, así como las infografías motivacionales.**

**A continuación, se muestra como ejemplo la página index.html, la página de inicio. En este overview se puede apreciar el uso de estilos CSS incrustados en la cabecera, así como la utilización de Bootstrap (columna izquierda). En la columna de la derecha, por su parte, puede observarse el desarrollo del carrusel de consejos utilizando Thymeleaf para realizar las llamadas a los métodos del controlador correspondiente.**

```

index.html
20⑩ <script
21   src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
22⑩ <style th:inline="text">
23   .Logo {
24     display: block;
25     margin-left: auto;
26     margin-right: auto;
27     margin-top: 20px;
28     width: 100%;
29   }
30
31   .boton {
32     display: block;
33     left: 50%;
34     right: auto;
35     text-decoration: none;
36   }
37
38   h1 {
39     color: #ff9900;
40     text-align: center;
41   }
42
43   h2 {
44     text-shadow: 2px 2px 2px #000000F;
45   }
46
47   #myCarousel img {
48     max-height: 380px;
49   }
50
51   #myCarousel {
52     margin-top: 2em;
53     width: 65%;
54     display: block;
55     margin-left: auto;
56     margin-right: auto;
57   }
58
59   .grid-container {
60     display: grid;
61     grid-template-columns: 50% 50%;
62     padding: 10px;
63   }
64
65   .descarga {
66     margin-right: auto;
67     margin-left: auto;
68     text-align: center;
69   }
70
71   .descarga h2 {
72     font-weight: bold;
73     text-shadow: 1.5px 1px black;
74     color: #3dadff;
75     margin: 2em;
76   }
77 </style>

```

```

79⑩ <body>
80   <nav th:replace="navbar_loader"></nav>
81⑩   <header>
82     <!-- Cabecera con logo en imágenes que presentan la página-->
83⑩     <figure>
84       
85     </figure>
86
87⑩     <a name="arriba" href="#abajo" class="boton"><button type="button"
88       class="btn btn-warning boton btn-sm">Ir abajo</button></a><br />
89
90
91     <!-- CAROUSEL -->
92
93⑩     <div class="container">
94       <h1>Consejo del día</h1>
95       <div id="myCarousel" class="carousel slide" data-ride="carousel">
96         <ol class="carousel-indicators">
97           <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
98           <li data-target="#myCarousel" data-slide-to="1"></li>
99           <li data-target="#myCarousel" data-slide-to="2"></li>
100          <li data-target="#myCarousel" data-slide-to="3"></li>
101          <li data-target="#myCarousel" data-slide-to="4"></li>
102          <li data-target="#myCarousel" data-slide-to="5"></li>
103          <li data-target="#myCarousel" data-slide-to="6"></li>
104        </ol>
105
106        <!-- Contenedor para slides -->
107        <div class="carousel-inner">
108          <div class="item active">
109            
111            <div class="carousel-caption" th:each="consejo : ${consejos}">
112              <div th:if="${consejo.nombreConsejo == 'h_pesa'}">
113                <h2 th:text="${consejo.descripcionConsejo}"></h2>
114              </div>
115            </div>
116          </div>
117        </div>
118
119        <div class="item">
120          
122          <div class="carousel-caption" th:each="consejo : ${consejos}">
123            <div th:if="${consejo.nombreConsejo == 'h_mirror'}">
124              <h2 th:text="${consejo.descripcionConsejo}"></h2>
125            </div>
126          </div>
127
128        <div class="item">
129          
131          <div class="carousel-caption" th:each="consejo : ${consejos}">
132            <div th:if="${consejo.nombreConsejo == 'h_pecho'}">
133              <h2 th:text="${consejo.descripcionConsejo}"></h2>
134            </div>
135          </div>
136        </div>

```

**Ejemplo de template del índice de la aplicación. A la izquierda, se observan los estilos utilizados. A la derecha, elementos del body con llamadas al controlador.**

## ARQUITECTURA Y DIAGRAMA DE CLASES

**El diseño del diagrama de clases que se ha implementado para este proyecto (imagen disponible en la sección de "Anexos" al final de este documento referenciado como [Anexo 3](#)) se realizó en una fase previa de diseño previa a la fase de desarrollo y codificación de la aplicación web Fat Down, con el propósito de dotar de sentido al proyecto de software así como de una estructura básica en todo caso con potencial de crecimiento.**

**Esto se aprecia en la ya mencionada estructura de capas abstractas siguiendo el patrón de arquitectura Modelo-Vista-Controlador (MVC), lo cual ha permitido desarrollar la aplicación rápidamente, de forma modular, sostenible e incremental.**

**Por otro lado, también se puede destacar del diagrama el uso de las clases abstractas y la herencia en Java en clases como "Multimedia", "Video" o "Gif" con la intención de proporcionar a las diversas funcionalidades y entidades del proyecto de flexibilidad y adaptabilidad que permita, en un futuro, un desarrollo más avanzado en base a nuevas necesidades.**

**Entre las relaciones utilizadas entre las entidades se puede observar una variedad de tipos (OneToOne, OneToMany, ManyToMany, de agregación, de composición, etc) que ha afectado directamente a la forma de mapear dichas entidades y sus respectivas tablas en la BBDD, utilizando Hibernate como herramienta de mapeo objeto-relacional.**

En la imagen siguiente, se muestra el esquema de la BBDD donde se aprecian las tablas referentes a las entidades así como las tablas intermedias (JoinTables) generadas a partir de la relación N:M existente entre dos entidades.



Para finalizar esta sección, se muestran algunos ejemplos de las columnas existentes en algunas de las tablas, así como las tablas intermedias generadas entre algunas de ellas en base a sus relaciones lógicas en la aplicación.

#### Table: usuario

##### Columns:

<u>id_usuario</u>	bigint AI PK
apellidos_usuario	varchar(255)
codseg_usuario	varchar(255)
direcfact_usuario	varchar(255)
<u>email_usuario</u>	varchar(255)
fechanac_usuario	varchar(255)
nombre_usuario	varchar(255)
numtarjeta_usuario	varchar(255)
password_usuario	varchar(255)
titular_usuario	varchar(255)

#### Table: rol

##### Columns:

<u>id_rol</u>	int AI PK
nombre_rol	varchar(40)

#### Table: usuario\_rol

##### Columns:

<u>id_usuario</u>	bigint PK
<u>id_rol</u>	int PK

#### Table: ejercicio

##### Columns:

<u>ID_EJERCICIO</u>	bigint AI PK
nombre_ejercicio	varchar(50)
descripcion_ejercicio	varchar(2000)
categoria_ejercicio	varchar(50)
tipo_ejercicio	varchar(50)
repeticion_ejercicio	int
duracion_ejercicio	int

#### Table: rutina

##### Columns:

<u>id_rutina</u>	bigint AI PK
nombre_rutina	varchar(50)
<u>id_usuario</u>	bigint

#### Table: rutina\_ejercicio

##### Columns:

<u>id_rutina</u>	bigint PK
<u>ID_EJERCICIO</u>	bigint PK

# LÓGICA DE NEGOCIO

**La lógica de negocio es la parte de un sistema que se encarga de codificar las reglas de negocio del mundo real que determinan cómo la información puede ser creada, almacenada y cambiada.**

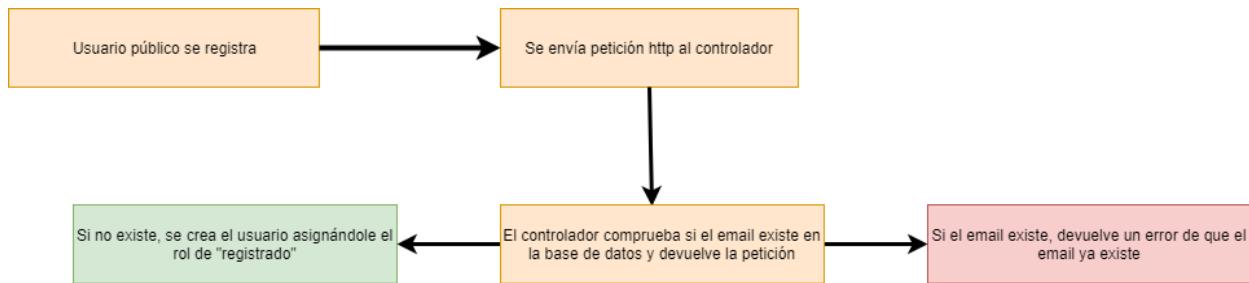
**En este proyecto, un ejemplo sería la implementación de ejercicios, los cuales pueden ser creados y borrados por el administrador, pero por parte del usuario registrado solo pueden ser guardados en rutinas. El borrado de registros en las tablas de ejercicio y rutina se realiza en cascada, por lo que también son eliminadas sus referencias en el resto de tablas que los incluyesen, a fin de mantener la integridad referencial y no tener registros huérfanos en otras tablas**

**Otro caso sería la creación de nuevos usuarios, momento en el que se realiza la asignación de roles a dichos usuarios como "registrados", de modo que una persona tenga restringido el acceso a funcionalidades no deseadas como el borrado de cuentas ajenas u otros elementos.**

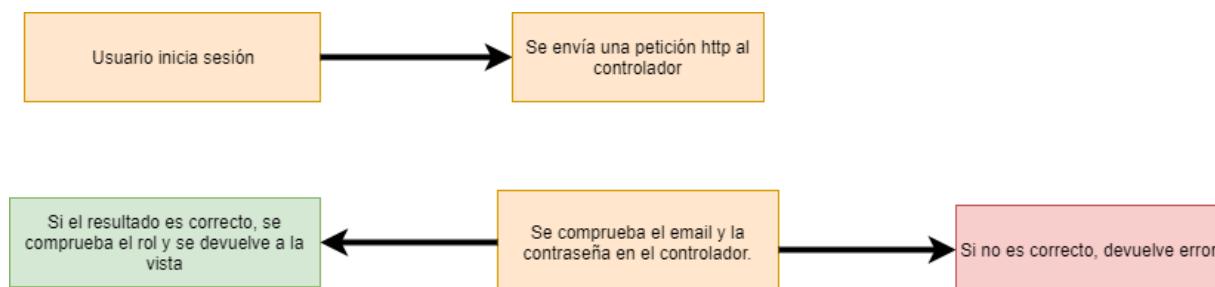
# LÓGICA DE LA APLICACIÓN

En los esquemas que se adjuntan a continuación, podemos seguir la lógica que siguen los controladores

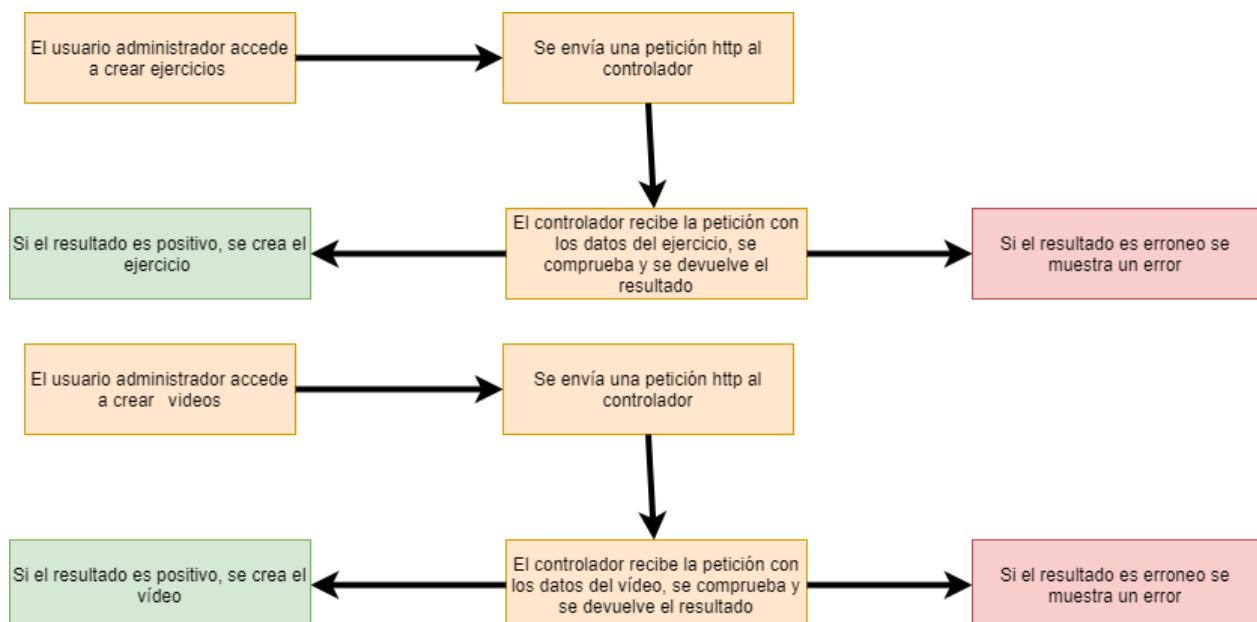
## Registro de un nuevo usuario en la aplicación



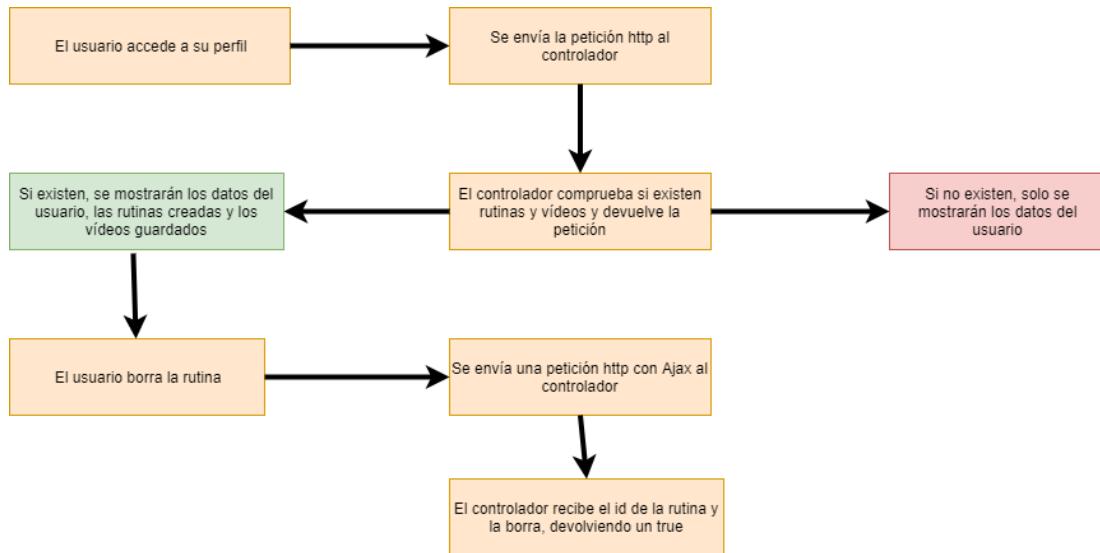
## Inicio de sesión de un usuario registrado en la aplicación



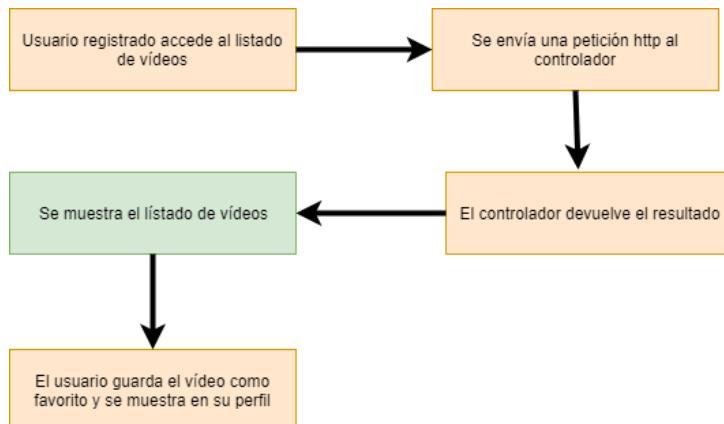
## Creación de nuevos ejercicios y videos por el usuario administrador



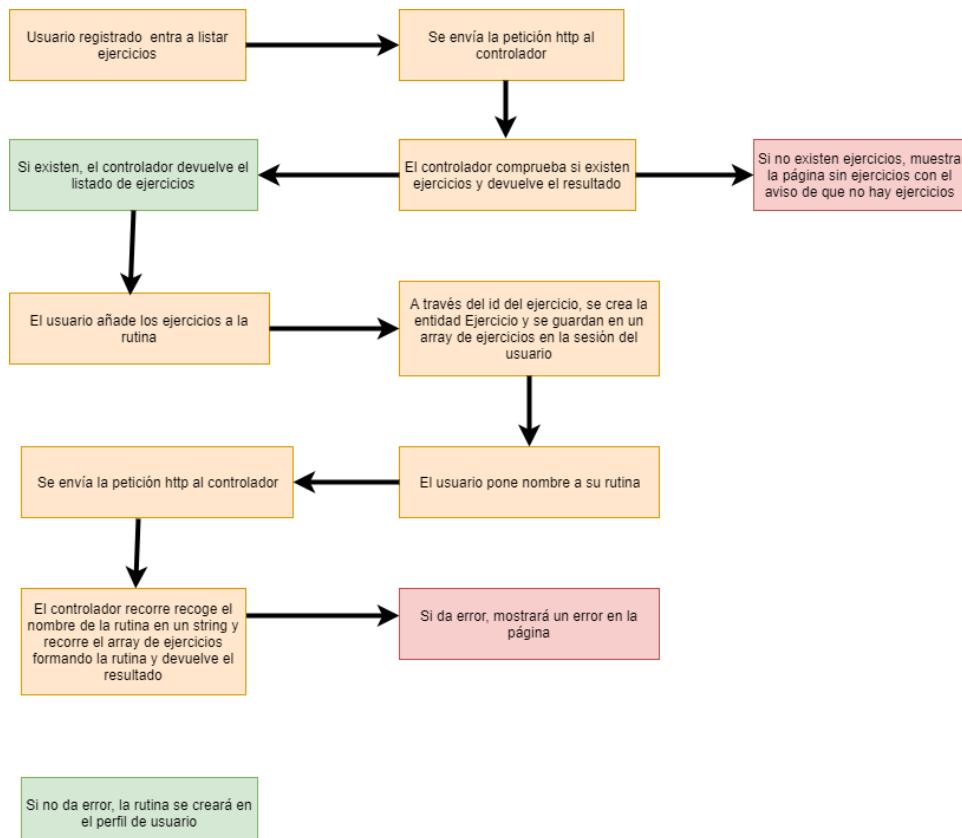
## Acceso de usuario registrado al perfil personal



## Guardado de vídeo favorito por usuario registrado en listado de vídeos



## Creación de rutinas por usuario registrado en listado de ejercicios



# TECNOLOGÍAS

Las tecnologías utilizadas para este proyecto han sido seleccionadas en base a nuestros conocimientos previos, y son principalmente:

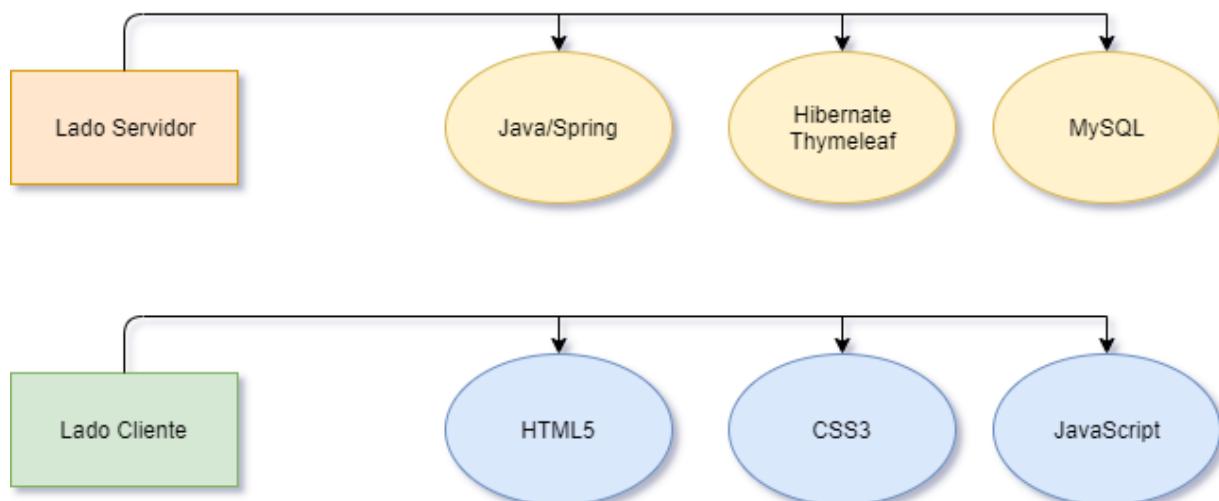
## BACK END

- **Java como lenguaje de programación en el lado servidor**
- **Spring boot como framework de desarrollo**
- **Hibernate como herramienta de mapeo objeto-relacional**
- **Thymeleaf como motor de plantillas (templates)**
- **Maven como herramienta de inyección de dependencias y gestión de librerías**
- **Git como herramienta de control de versiones**
- **Base de datos MySQL, uno de los SGBD Open Source más populares y utilizados en la actualidad, lo que permite una gran portabilidad y soporte del desarrollo de software realizado en este proyecto**

## FRONT END

- **HTML5 como lenguaje de marcas**
- **CSS3 para hojas de estilo**
- **Bootstrap como framework CSS que permite diseñar páginas HTML responsive**
- **JavaScript vanilla como lenguaje de programación en el lado cliente**
- **JQuery**
- **JavaScript Asíncrono con XML (AJAX)**

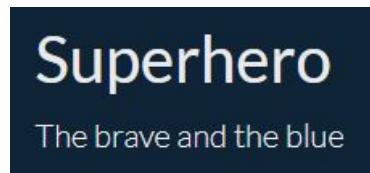
**En el siguiente esquema, se puede apreciar más simplificadamente las tecnologías empleadas en el desarrollo de la aplicación web Fat Down.**



# GUÍA DE ESTILOS

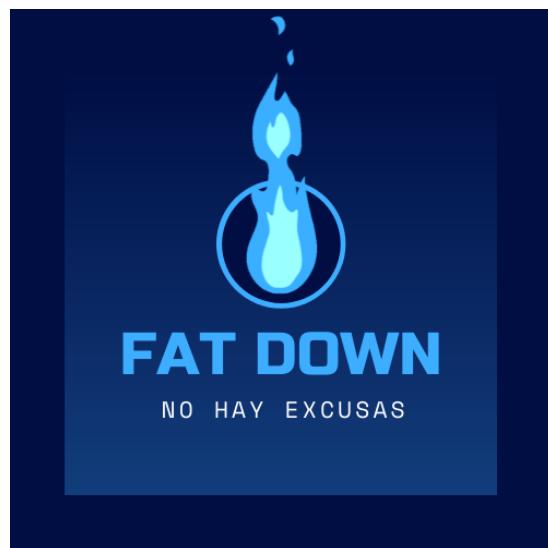
## BOOTSTRAP

Los estilos de este proyecto se han basado en el tema de Bootstrap/Bootswatch "Superhero", que se puede acceder desde el siguiente enlace: [aquí](https://bootswatch.com/superhero/) (<https://bootswatch.com/superhero/>).



La decisión se decantó por este tema debido a la naturaleza del proyecto, una web sobre fitness, tratando de valorar la experiencia del usuario y buscando llamar la atención del mismo, empleando tonos que suelen encontrarse en otras webs o aplicaciones de fitness similares.

## LOGO



**El logo creado, además de perseguir la homogeneidad con el estilo seleccionado anteriormente, contiene una llama azul que simboliza la energía que infusionamos en la actividad física, en el que se incluye el nombre de la marca y aplicación web homónima "Fat Down", junto al lema "No hay excusas" que acompaña al conjunto, en un intento de animar al usuario a sacar fuerzas de flaqueza y conseguir sus objetivos fitness.**

## COLORES

#0000e44

≈ Stratos

#000080

Navy Blue

#3dadff

≈ Dodger Blue

#Ff9900

≈ Orange Peel

#Ffcc99

Peach Orange

**Los colores empleados han sido cuidadosamente seleccionados siguiendo el patrón empleado en el tema "Superhero" de Bootstrap, tratando de mantener coherencia y homogeneidad en todas las vistas y páginas de la aplicación web.**

**Se han utilizado tanto colores análogos (azules) para crear transiciones suaves como complementarios para crear contraste y resaltar información (naranjas).**

# IMPLEMENTACIÓN

Para llevar a cabo la implementación del proyecto se ha seguido una metodología de trabajo ágil que ha permitido al equipo realizar entregas funcionales e incrementales espaciadas en el tiempo en forma de iteraciones.

En las primeras iteraciones, el equipo se centró en desarrollar las funcionalidades relativas a los usuarios:

- **Creación de roles de usuario**
- **Diseño y creación del formulario de alta**
- **Creación de usuario administrador por defecto**
- **Creación de usuarios registrados por defecto**
- **Seguridad: control de alta y acceso de usuarios mediante encriptado de datos y asignación de roles**

En sucesivas iteraciones el desarrollo se centró en la implementación del resto de entidades y sus correspondientes funcionalidades:

- **Creación de vídeos, gifs e imágenes que heredan de "multimedia"**
- **Paginación del listado de vídeos ordenado por categoría**
- **Creación de ejercicios**
- **Filtrado de ejercicios por categoría**
- **Creación de consejos**
- **Mapeo de las relaciones entre las entidades y generación de tablas intermedias**

En la última iteración el objetivo pasó a ser ultimar el diseño de la aplicación web e implementar algunas de las funcionalidades más útiles para el usuario:

- **Creación de rutinas de ejercicios personalizadas**
- **Creación de lista de vídeos favoritos**
- **Homogeneización de vistas y páginas, acorde a la guía de estilos**

**A continuación, se muestra a modo de ejemplo, la configuración de seguridad implementada y utilizada durante el proceso de autenticación de usuarios.**

**Como puede observarse en la última imagen, los registros de los diferentes usuarios aparecen persistidos en la BBDD con el campo contraseña cifrado tras la codificación que tiene lugar al aplicar el bean de codificación BCryptPasswordEncoder.**

**Igualmente, se puede apreciar en el código de la clase SecurityConfig.java cómo son controlados los endpoints de la aplicación web en función del rol asignado a cada usuario, así como los direccionamientos a los formularios de login y logout.**

```

SecurityConfig.java
1 package com.fatdown.spring.seguridad;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4
5 @Configuration
6 @EnableGlobalMethodSecurity(securedEnabled = true)
7 @EnableWebSecurity
8 public class SecurityConfig extends WebSecurityConfigurerAdapter {
9
10     @Autowired
11     private CustomUserDetailsService userDetailsService;
12
13     @Override
14     protected void configure(HttpSecurity http) throws Exception {
15         http.authorizeRequests()
16             .antMatchers("/", "/index", "/js/**", "/css/**", "/images/**", "/pictures/**", "/usuario/**", "/multimedia/listarVideos/**", "/multimedia/listarImagenes/**", "/multimedia/listarIds/**", "/multimedia/recuperarId/**", "/permisos/**").permitAll()
17             .antMatchers("/listarEjercicios", "/ejercicio/**", "/multimedia/**").hasAuthority("admin")
18             .anyRequest().authenticated()
19         .and()
20             .formLogin()
21                 .loginPage("/usuario/login")
22                 .loginProcessingUrl("/usuario/login")
23                 .successHandler(myAuthenticationSuccessHandler())
24                 .permitAll()
25                 .failureUrl("/usuario/login_error")
26         .and()
27             .logout()
28                 .invalidateSession(true)
29                 .deleteCookies("JSESSIONID")
30                 .clearAuthentication(true)
31                 .logoutUrl("/logout")
32                 .logoutRequestMatcher(new AntPathRequestMatcher("/logout"))
33                 .logoutSuccessUrl("/login/logout")
34                 .permitAll()
35         http.csrf().ignoringAntMatchers("/pictures/**");
36     }
37
38     @Override
39     public void configureGlobal(AuthenticationManagerBuilder auth) throws Exception {
40         auth.userDetailsService(userDetailsService).passwordEncoder(passwordEncoder());
41     }
42
43     @Bean
44     public BCryptPasswordEncoder passwordEncoder() {
45         BCryptPasswordEncoder bCryptPasswordEncoder = new BCryptPasswordEncoder();
46         return bCryptPasswordEncoder;
47     }
48
49     @Bean
50     public AuthenticationSuccessHandler myAuthenticationSuccessHandler() {
51         return new AuthenticationSuccessHandlerImpl();
52     }
53
54 }

```

```

AuthenticationSuccessHandlerImpl.java
1 package com.fatdown.spring.seguridad;
2
3 import java.io.IOException;
4
5 @Configuration
6 @EnableWebSecurity
7 @EnableGlobalMethodSecurity(securedEnabled = true)
8 public class AuthenticationSuccessHandlerImpl implements AuthenticationSuccessHandler {
9
10     @Autowired
11     private UsuarioServicio usuarioServicio;
12
13     private RedirectStrategy redirectStrategy = new DefaultRedirectStrategy();
14
15     @Override
16     public void onAuthenticationSuccess(HttpServletRequest request, HttpServletResponse response,
17                                         Authentication authentication) throws IOException {
18
19         UserDetails userDetails = (UserDetails) SecurityContextHolder.getContext().getAuthentication().getPrincipal();
20
21         HttpSession session = request.getSession();
22         Usuario authUser = usuarioServicio.buscarPorEmailUsuario(userDetails.getUsername());
23         Set<GrantedAuthority> grantedAuthorities = new HashSet<GrantedAuthority>(authentication.getAuthorities());
24         session.setAttribute("emailUsuario", authUser.getEmailUsuario());
25         session.setAttribute("idusuario", authUser.getIdUsuario());
26         session.setAttribute("ultimoIn", crearRutina());
27
28         boolean esRegistrado = false;
29         boolean esAdmin = false;
30         final Collection<GrantedAuthority> authorities = authentication.getAuthorities();
31         for (GrantedAuthority grantedAuthority : authorities) {
32             if (grantedAuthority.getAuthority().equals("registrado")) {
33                 esRegistrado = true;
34             } else if (grantedAuthority.getAuthority().equals("admin")) {
35                 esAdmin = true;
36             }
37         }
38
39         String targetUrl;
40         if (!esRegistrado) {
41             targetUrl = "/usuario/userid/" + authUser.getIdUsuario();
42         } else if (esAdmin) {
43             targetUrl = "/index";
44         } else {
45             throw new IllegalStateException();
46         }
47
48         Cookie cookie = new Cookie("idusuario", authUser.getIdUsuario().toString());
49         response.addCookie(cookie);
50
51         redirectStrategy.sendRedirect(request, response, targetUrl);
52     }
53
54     public void setRedirectStrategy(FinalRedirectStrategy redirectStrategy) {
55         this.redirectStrategy = redirectStrategy;
56     }
57
58     protected RedirectStrategy getRedirectStrategy() {
59         return redirectStrategy;
60     }
61
62 }

```

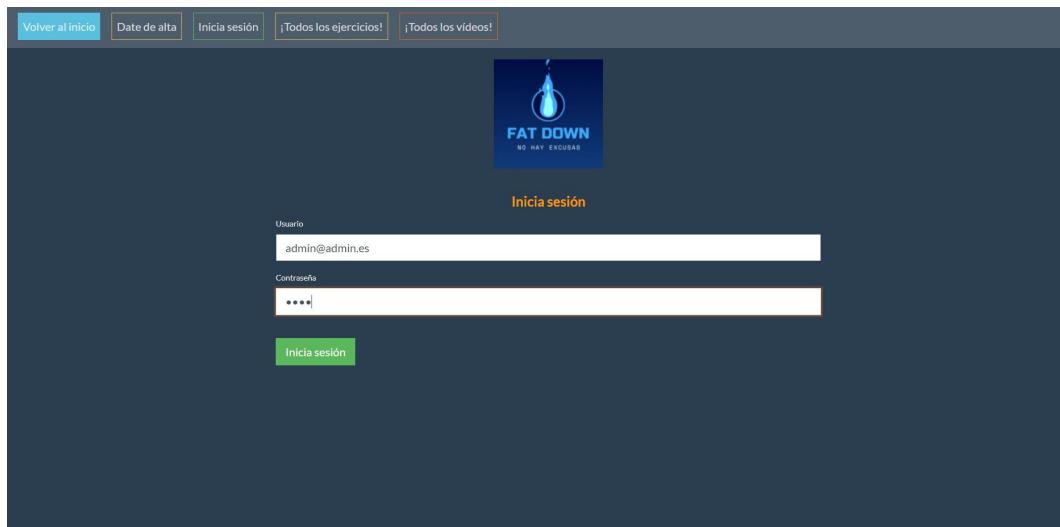
Result Grid   Filter Rows:   Export/Import:   Wrap Cell Content:										
	id_usuario	apellidos_usuario	codseg_usuario	direcfact_usuario	email_usuario	fechanac_usuario	nombre_usuario	numtarjeta_usuario	password_usuario	titular_usuario
▶	1	admin	666	deepweb	admin@admin.es	1/enero/2020	admin	666	\$2a\$10\$w3gWFIRpd9tBjWqSubk5POL0nbLbNg...	administrator
▶	2	Master	619	Andalasia	meme@meme.es	16/noviembre/1994	Danielus	000	\$2a\$10\$PPrD3ybQj4JY0fMqJfROOOiT.v5CrZ... XXX	
▶	3	Chachi	666	Calle Desengaño 21	mecago@entoh.es	19/mayo/1992	Amanduski	888	\$2a\$10\$lnF8jqyBas9JX7SoVctONAbHzgZh7N...	Amanduski
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

usuario 1 < Action Output # Time Action Message  
1 12:46:38 SELECT \* FROM fatdown.usuario LIMIT 0, 1000 3 row(s) returned

# PRUEBAS

**Al finalizar el desarrollo se realizaron pruebas de funcionamiento de la aplicación, para la cual se adjuntan en los anexos al final de este documento dos videos en los que se puede visionar a modo de demo el funcionamiento de la aplicación web.**

**En este apartado también se incluyen como pruebas imágenes de las páginas y vistas que pueden encontrarse en la aplicación:**



The screenshot shows a dark-themed login page. At the top, there are four buttons: "Volver al inicio" (blue background), "Date de alta" (white background), "Inicia sesión" (white background), and "¡Todos los ejercicios!" and "¡Todos los videos!" (orange background). Below these is a logo for "FAT DOWN" featuring a stylized flame icon and the text "NO HAY EXCUSAS". A yellow "Inicia sesión" button is centered below the logo. The main form fields are "Usuario" (with value "admin@admin.es") and "Contraseña" (with value "\*\*\*\*"). A green "Inicia sesión" button is at the bottom of the form.

Vista de formulario de login



The screenshot shows a registration form titled "¡Forma parte de la familia Fat Down!". It has a light blue header bar. The first section is "Háblanos de ti" with fields for "Nombre y apellidos" (placeholder "Usuario"), "Email" (placeholder "ejemplo@example.es"), and "Contraseña" (placeholder "\*\*\*\*"). Below this is a note: "¡Nunca compartimos tu email con nadie!". The next section is "Información para tus pedidos" with fields for "Número de tarjeta de crédito/débito" (placeholder "4545452121"), "Titular de la tarjeta" (placeholder "Usuario Ejemplo"), "Código de seguridad de la tarjeta" (placeholder "123"), and "Dirección de facturación" (placeholder "Calle del Aprobado, 5"). Below this is a note: "¡Nunca compartimos tus datos de pago con nadie!". A green "¡Dame de alta!" button is at the bottom.

Vista de formulario de alta de usuario

**Perfil del usuario**

<b>Datos personales</b>	<b>Datos de pago y facturación</b>
ID: 1 Nombre: admin Apellido: admin Email: admin@admin.es Fecha de nacimiento: 1/enero/2020	Número de tarjeta de crédito/débito: Titular de la tarjeta de crédito/débito: Código de seguridad de la tarjeta de crédito/débito: Dirección de facturación: 666 administrador 666 desweb

**Aquí están tus rutinas personalizadas**

<b>Tu rutina RUTINA 1</b>  Nombre del ejercicio: Sentadillas Nombre del ejercicio: Superman	Borrar rutina
Categoría: PERNAS Duración del ejercicio (en segundos): 30	Repeticiones del ejercicio: 20

**Aquí están tus videos favoritos**

Parte interna del muslo   Ejercicios para piernas en casa  Abdominales de pie   20 minutos	PIERNAS
PIERNAS	AERÓBICOS
Abdominales de ABDOMINALES	AERÓBICOS

### Vista de perfil de usuario con rutinas creada y vídeos favoritos

**Listado de ejercicios**

Filtrar los ejercicios por su tipo:

-- Selección una categoría de ejercicios --

<b>Sentadillas</b>  PIERNAS Número de repeticiones: 20 Consiste en colocarse de pie, con los brazos delante, las piernas extendidas en línea con los hombros y los dedos de los pies ligeramente hacia afuera. Luego dobla las rodillas como si fueras a sentarte, empujando la pelvis hacia atrás y sin que las rodillas traspasen la punta del pie. Despues levántate y répite.  Eliminar ejercicio	Añadir a rutina
<b>Superman</b>  ESPALDA Segundos de duración: 30 Túmbate boca abajo. Estira los brazos hacia adelante, con las palmas hacia abajo. Levanta la parte superior del cuerpo y luego las piernas, formando un arco. Asegúrate de que las rodillas y el pecho no tocan el suelo. Mantén la cabeza y el cuello neutrales. No estires demasiado el cuello y mantén la barbilla clavada.  Eliminar ejercicio	Añadir a rutina

**¡Crea tu rutina!**

Escribe el nombre de tu rutina

**Crear rutina**

### Vista de listado de ejercicios

Volver al inicio | Ir al perfil de admin | Cierra sesión | Crear ejercicios | Crear videos | ¡Todos los ejercicios! | ¡Todos los videos!



**Listado de videos**

Ejercicios para abdomen   12 minutos	ABDOMINALES	Eliminar video Guardar video
Rutina intensa de abdomen para un vientre plano   8 minutos	ABDOMINALES	Eliminar video Guardar video
Ejercicios abdomen fuerte   10 minutos	ABDOMINALES	Eliminar video Guardar video

### Vista de listado de videos

# CONCLUSIONES

**Tal y como se anticipaba en la introducción, el proyecto Fat Down nació con la intención de convertirse en el compañero de entrenamiento ideal, buscando ser una aplicación con un espíritu divertido a la par que motivador.**

**El desarrollo de las distintas funcionalidades basadas en la estructura del patrón de arquitectura MVC, así como el diseño de las diferentes vistas y páginas a través de las cuales interactuará el usuario, han conseguido imbuir al proyecto del sentido que se buscaba a la vez que se han cumplido los requisitos funcionales y no funcionales previstos en el apartado de análisis.**

**En cuanto a las limitaciones del proyecto, se puede destacar que no se han logrado incluir en el plazo previsto alguna de las funcionalidades planteadas inicialmente, debido a la limitación de los recursos con los que se han contado, como el tiempo.**

**El desarrollo ha sido posible en gran parte gracias al autoaprendizaje, necesario para llevar al proyecto a un nivel superior: investigación sobre herencia, clases abstractas y su mapeo objeto-relacional utilizando Hibernate, diseño adicional de infografías motivacionales descargables, inclusión de carrusel de consejos en la página de inicio y utilización de la librería Java Lombok para ahorrar código redundante.**

**Todo lo referido anteriormente ha sido posible gracias a la dedicación, trabajo en equipo, planificación y las innegables ganas de aprender del equipo.**

# **ANEXOS**

- **ANEXO 1**

## **REPOSITORIO GITHUB**

[HTTPS://GITHUB.COM/ALEXRET/FATDOWN.GIT](https://github.com/Alexret/Fatdown.git)

- **ANEXO 2**

## **ENLACES A VÍDEOS DE DEMOSTRACIÓN**

### **I. DEMO USUARIO ADMINISTRADOR**

AQUÍ

[HTTPS://YOUTU.BE/CVSJGX1GJZW](https://youtu.be/cvsjgx1gjzw)

### **II. DEMO USUARIO REGISTRADO**

AQUÍ

[HTTPS://YOUTU.BE/YGOZ8NRIABU](https://youtu.be/ygoz8nriabu)

- **ANEXO 3**

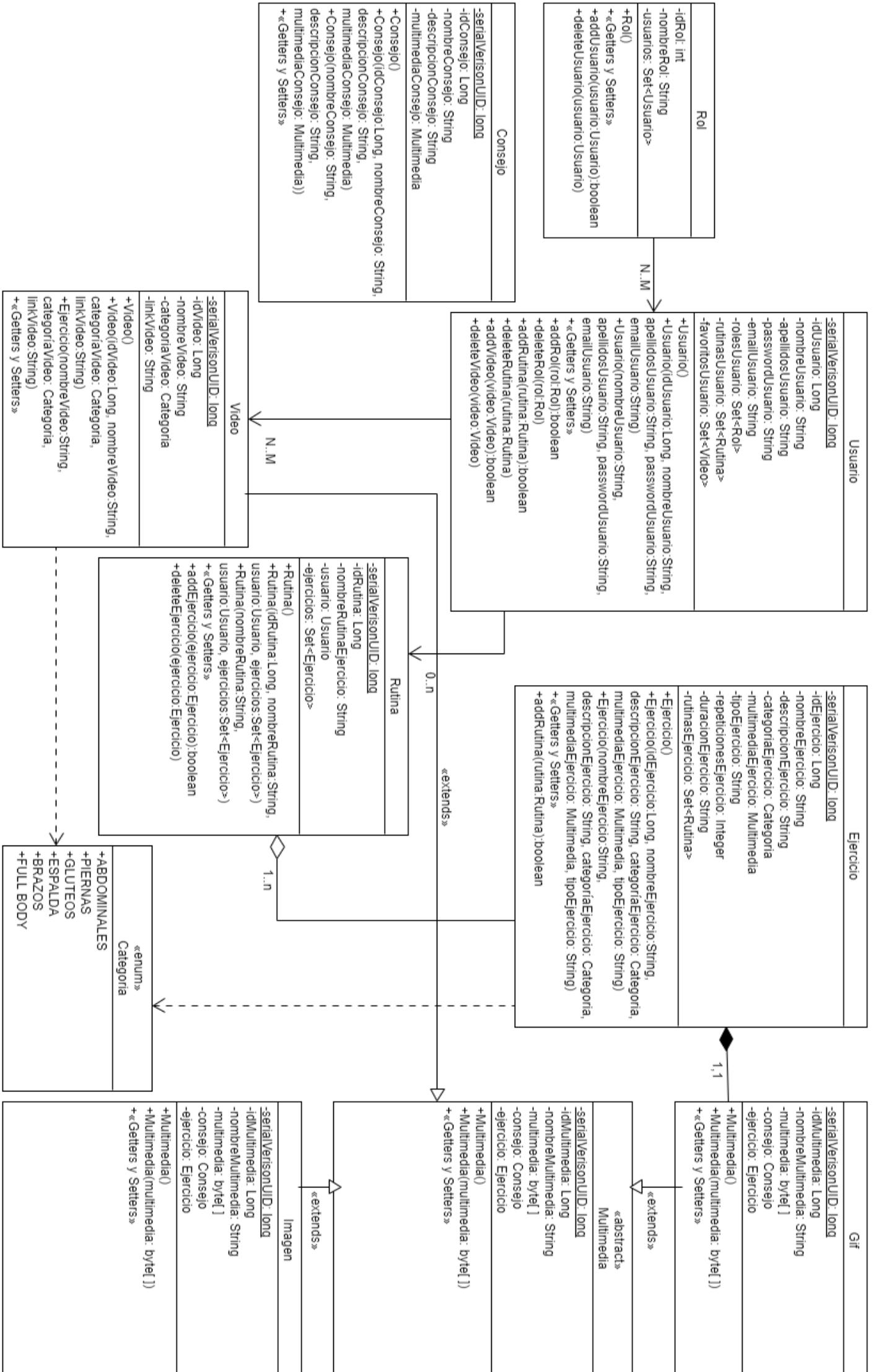
## **DIAGRAMA DE CLASES**

- **ANEXO 4**

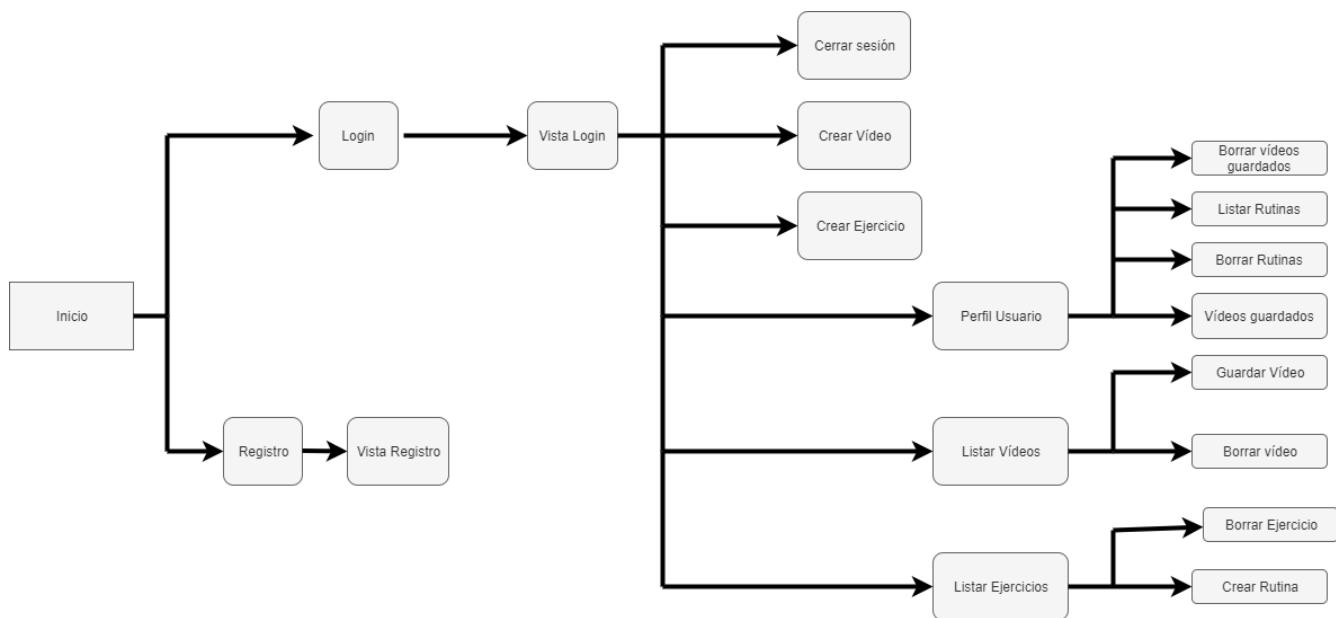
## **MAPA DE NAVEGACIÓN**

- **ANEXO 5**

## **INFOGRAFÍAS MOTIVACIONALES**



# ANEXO 4



# ANEXO 5





**FAT DOWN**  
NO HAY EXCUSAS

