

## Bolsa de Horas

**Sergio Fernanz Esteban**  
Grado Ingeniería Informática  
TFG Java EE

**Consultor: Vicenç Font Sagrista**  
**Nombre Profesor/a responsable de la asignatura : Santi Caballe Llobet**

09/01/2019



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	<i>Bolsa de Horas</i>
<b>Nombre del autor:</b>	<i>Sergio Fernanz Esteban</i>
<b>Nombre del consultor/a:</b>	<i>Vicenç Font Sagrista</i>
<b>Nombre del PRA:</b>	<i>Santi Caballe Llobet</i>
<b>Fecha de entrega (mm/aaaa):</b>	01/2019
<b>Titulación:</b>	<i>Grado de Ingeniería Informática</i>
<b>Área del Trabajo Final:</b>	<i>Java EE</i>
<b>Idioma del trabajo:</b>	<i>Castellano</i>
<b>Palabras clave</b>	<i>Spring Boot, docker y aplicación colaborativa.</i>

**Resumen del Trabajo (máximo 250 palabras):** *Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.*

La finalidad del proyecto es el aumento y ejecución de los conocimientos adquiridos durante el grado aplicando técnicas de ingeniería para realizar un proyecto en las tecnologías Java EE desde cero. La aplicación es una web en la que se han cubierto todas las fases de diseño y desarrollo que deberían llevar a cabo los proyectos de desarrollo de software.

La economía colaborativa está en pleno auge y hay numerosas iniciativas para el intercambio de “bienes o servicios” en tiempo. Ninguna de ellas para mi cumple lo básico para un uso correcto, de ahí que me surgiera la necesidad de realizar una aplicación en la que los usuarios pudieran realizar servicios a otros usuarios utilizando como pago un recurso muy valioso, el tiempo.

En Bolsa de Horas los usuarios comienzan con dos horas y recibirán más horas al prestar servicios y las podrán gastar recibiendo servicios de otros usuarios, es una plataforma de intercambio de servicios por tiempo.

El proyecto ha vivido varias fases, una primera de planificación en la que se establecieron los compromisos temporales a llevar a cabo, una segunda de casos de uso, diseño conceptual, elección de tecnologías y prototipos junto a una fase final de desarrollo de la aplicación.

Este desarrollado entorno al ecosistema de Spring con Spring Boot por su facilidad de despliegue, inyección de dependencias e inversión de control. Del lado de la parte visual con Thymeleaf como motor de plantillas, para la capa de persistencia utiliza la Java Persistence Api. Destacar que está basada en contenedores docker para un fácil despliegue y escalado.

Las conclusiones del trabajo son abundantes, la importancia e impacto de un buen diseño en una aplicación final. Un gran aprendizaje a nivel tecnológico ante

los retos encontrados y el cómo adaptarse al cambio según se va ejecutando un proyecto.

**Abstract (in English, 250 words or less):**

The purpose of the project is the increase and execution of the knowledge acquired during the degree by applying engineering techniques to carry out a project in Java EE technologies from scratch. The application is a website in which all the design and development phases that should be carried out by software development projects have been covered.

The collaborative economy is booming and there are numerous initiatives for the exchange of "goods or services" payed with time. None of them meets the basics for a correct use, hence the need to create an application in which users could offer services to other users paying with a valuable resource, the time.

In Bolsa de Horas users start with two hours and will receive more hours when they provide services and they can spend them receiving services from other users, it is a service exchange platform for time.

The conclusions of the work are abundant, the importance and impact of a good design in a final application and a great learning of different technologies

# Índice

1. Introducción .....	7
1.1 Contexto y justificación del Trabajo .....	7
1.2 Objetivos del Trabajo .....	8
1.3 Enfoque y método seguido .....	9
1.4 Planificación del Trabajo .....	10
Diagrama de Gantt.....	11
1.5 Breve resumen de productos obtenidos .....	12
1.6 Breve descripción de los otros capítulos de la memoria .....	12
2. Análisis y Diseño.....	13
2.1 Análisis de requisitos.....	13
2.2 Requerimientos funcionales .....	13
2.3 Requerimientos no funcionales .....	14
2.4 Casos de uso .....	14
2.5 Diagrama de clases.....	24
2.6 Diagrama de componentes .....	25
2.7 Prototipos .....	26
2.8 Decisiones tecnológicas .....	30
2.8.1 Maven.....	30
2.8.2 Spring Boot.....	30
2.8.2.1 Spring Security .....	31
2.8.2.2 Inversión de control .....	31
2.8.2.3 Inyección de dependencias.....	31
2.8.2.4 Spring Data.....	31
2.8.3 Thymeleaf.....	31
2.8.4 Docker .....	31
3. Implementación.....	33
3.1 Estructura del proyecto .....	33
3.1.1 Carpeta src .....	33
3.1.2 Ficheros docker .....	34
3.1.3 Fichero de Maven .....	34
3.2 Ejemplo de flujo de la aplicación .....	35
3.3 Resultado gráfico de la aplicación .....	36
4. Conclusiones .....	41
5. Glosario .....	43
6. Bibliografía .....	44
7. Anexos.....	45
7.1 Despliegue de la aplicación con docker .....	45
7.2 Generación del artefacto de la aplicación.....	47

## Lista de figuras

Ilustración 1 - Diagrama de Gantt inicio del proyecto	11
Ilustración 2 - Diagrama de Gantt final del proyecto	11
Ilustración 3 – Prototipo de acceso al sistema	26
Ilustración 4 - Prototipo de registro en el sistema	26
Ilustración 5 - Prototipo de alta de anuncio	27
Ilustración 6 - Prototipo de detalle de anuncio	27
Ilustración 7 - Prototipo de búsqueda de anuncio	28
Ilustración 8 - Prototipo de eliminación/modificación de anuncio	28
Ilustración 9 - Prototipo de ver solicitudes	29
Ilustración 10 - Prototipo de enviar/responder mensaje	29
Ilustración 11 - Logo Maven	30
Ilustración 12 - Logo Spring	30
Ilustración 13 - Logo Thymeleaf	31
Ilustración 14 - Logo Docker	32
Ilustración 15 - Estructura de paquetes del proyecto	33
Ilustración 16 - Paquetes de la carpeta src	33
Ilustración 17 - Página de Inicio Aplicación	36
Ilustración 18 - Pantalla de registro	36
Ilustración 19 - Pantalla de creación de anuncio	37
Ilustración 20 - Pantalla de búsqueda de anuncio	37
Ilustración 21 - Pantalla de detalle de anuncio	38
Ilustración 22 - Pantalla de solicitud de anuncio	38
Ilustración 23 - Pantalla de solicitud de servicio	39
Ilustración 24 - Pantalla de servicio completo	39
Ilustración 25 - Pantalla de puntuación de usuario	39
Ilustración 26 - Pantalla de perfil de usuario	40
Ilustración 27 - Pantalla de envío de mensajes	40
Ilustración 28 - Pantalla de mensajes recibidos	40

# 1. Introducción

## 1.1 Contexto y justificación del Trabajo

La economía colaborativa está en auge, es una realidad que cada día más personas están dispuestas a llevar a cabo acciones mucho más respetuosas con el medio ambiente, plataformas de alojamiento como couch surfing, Airbnb o BlaBlaCar son prueba de ello.

Ofrecer un servicio del que eres experto o solvente a un usuario a cambio de otro servicio es una especie de trueque y se lleva realizando miles de años, sin embargo, tiene una carencia, si la persona que quiere un servicio tuyo no tiene o puede ofrecerte nada de tu interés no se podrá realizar la transacción, de ahí nace la idea de Bolsa de Horas y de los bancos del tiempo en general.

Una plataforma confiable en la que los usuarios puedan ofrecer y requerir servicios cambio de horas. Prestarás servicios a personas que no tienen por qué prestarte algún servicio de tu interés, ya que con esas horas podrás buscar en la plataforma aquellos servicios que sean de tus intereses. Los usuarios comenzarán con un número de horas determinado y tendrán en su mano todo lo necesario para ejecutar los servicios con calidad, un sistema de mensajería, un sistema de reputación, un sistema de publicación y solicitud de servicios y búsqueda de anuncios.

Esta idea surge en la etapa final de mi Grado en Ingeniería Informática y me decido a utilizarla para este proyecto. Estamos ante un proyecto web en el que se aplicarán todas las etapas de diseño y ejecución para que la aplicación cumpla unos estándares de calidad y sea adecuada al mercado.

La aplicación se desarrolla en base a los últimos estándares tanto de trabajo como tecnológicos del mercado. Un modelo iterativo que se podría asemejar a scrum y una aplicación basada en servicios montada sobre un contenedor docker que facilitará su despliegue, escalabilidad y sencillez de desarrollo.

Ofrecer una plataforma sencilla y eficaz para la prestación de servicios entre usuarios sin implicación monetaria que permita cubrir ese hueco que existe en el mercado al no existir una aplicación predominante en este tipo de plataformas.

## 1.2 Objetivos del Trabajo

El objetivo principal es construir una plataforma fácil, confiable y de calidad que permita a usuarios intercambiar bienes y servicios utilizando las horas como moneda en las transacciones.

La aplicación web debe ser intuitiva y sencilla para que cualquier tipo de usuario pueda utilizar la plataforma.

Sin lugar a duda también existen objetivos académicos. En primer lugar, poner en práctica todo lo que he aprendido a lo largo de los años en diferentes asignaturas de programación a objetos, modelado UML, diseño de base de datos, patrones de diseño, etc. De forma paralela continuar aprendiendo nuevas tecnologías, aunque este sea un proyecto final el aprendizaje comienza aquí, un desarrollador nunca termina de aprender y renovarse, salir de la zona de confort del stack de java más tradicional y añadir nuevos actores con los que aprender y mejorar. Para ejemplificar esto es pasar de Spring algo más tradicionalista a Spring Boot o utilizar Docker para el despliegue del proyecto y la creación de la base de datos.

En resumen, los objetivos van del lado de realizar una aplicación de calidad junto a la ejecución de los conocimientos adquiridos y el aprendizaje de conocimientos nuevos.



### 1.3 Enfoque y método seguido

En primer lugar, realicé búsquedas por internet para tratar de descubrir servicios similares, me llamó la atención que este tipo de aplicaciones de intercambio son muy populares, pero sin presencia online ya que suelen ser offline. En la Comunidad de Madrid hay 17 centros que ejercen cómo bancos del tiempo, pero tienes que ir a apuntarte de forma física y todas las gestiones se realizan in situ, lo cual cierra el mercado y añade complejidad. Hay un índice de bancos del tiempo de España y américa latina, pero es un simple índice con enlaces a otras webs.

Al observar que no hay ninguna página similar las posibilidades de adaptar un producto existente se redujeron, no hay nada similar con componentes similares. La búsqueda entonces se amplió a productos que se utilizan para otros menesteres, pero se pueden adaptar a este, comencé la búsqueda mirando CMS que tuvieran sistema de anuncios o mensajería, pero todos estaban ampliamente vinculados al ecommerce y no potenciaban la descentralización, es decir, que los usuarios lleguen a tratos entre ellos en base a las horas.

En este escenario me decidí por realizar un producto nuevo adaptado a las necesidades de un sector olvidado. Realizaremos el proyecto dividido en tres etapas diferenciadas, una de análisis, otra de diseño y la última de implementación.

Cómo hemos visto en gestión de proyectos estamos ante un proyecto que se enmarca en el PMI aplicando el estándar PMBOK que es un referente de las prácticas más comunes a utilizar en el sector informático generando una documentación completa del proyecto con su memoria, manual de uso y presentación.

Este tipo de proyectos también son llamados waterfall ya que es un modelo en cascada o secuencial de las tareas tal cómo veremos en la planificación del trabajo. Este tipo de proyectos no suelen encajar muy bien con la metodología agile al establecer horas y entregas concretas, aunque dentro de estas subdivisiones o entregas si se puede utilizar un modelo iterativo. Decir que he utilizado metodología scrum para el proyecto no sería cierto, por lo que afirmaré que se ha realizado un modelo iterativo en el que en cada iteración se ha priorizado la entrega de valor a la aplicación.

## 1.4 Planificación del Trabajo

En el siguiente cuadro se muestran las fechas de las actividades a entregar durante el desarrollo del TFG.

Actividad	Descripción	Fecha de Inicio	Fecha de Fin
PEC 1	Plan de trabajo	19/09/2018	01/10/2018
PEC 2	Requerimientos, análisis y diseño	02/10/2018	05/11/2018
PEC 3	Implementación	06/11/2018	20/12/2018
Entrega final	Memoria, presentación y entrega	21/12/2018	09/01/2019

### Planificación detallada con subetapas

<b>Plan de Trabajo</b>	<b>19/09/2018</b>	<b>01/10/2018</b>	<b>11 días</b>
Lectura y comprensión del TFG	19/09/2018	23/09/2018	4 días
Redacción del plan de trabajo	23/09/2018	30/09/2018	7 días
Entrega PEC 1 – Plan de trabajo	01/10/2018	01/10/2018	1 día

<b>Requerimientos, análisis y diseño</b>	<b>02/10/2018</b>	<b>05/11/2018</b>	<b>34 días</b>
Refinamiento de los requerimientos	02/10/2018	09/10/2018	7 días
Diseño del modelo de clases	09/10/2018	15/10/2018	6 días
Diseño de casos de uso	15/10/2018	21/10/2018	6 días
Diseño del modelo de datos	21/10/2018	27/10/2018	6 días
Generación del documento de entrega	27/10/2018	04/11/2018	8 días
Entrega PEC 2 – Plan de trabajo	05/11/2018	05/11/2018	1 día

<b>Implementación</b>	<b>06/11/2018</b>	<b>20/12/2018</b>	<b>44 días</b>
Desarrollo del código	06/11/2018	06/12/2018	30 días
Desarrollo de las pruebas	06/12/2018	16/12/2018	10 días
Generación del documento de entrega	16/12/2018	19/12/2018	3 días
Entrega PEC 3 – Implementación	20/12/2018	20/12/2018	1 día

<b>Memoria, presentación y entrega</b>	<b>21/12/2018</b>	<b>09/01/2019</b>	<b>19 días</b>
Verificación del software	21/12/2018	26/12/2018	5 días
Generación de la memoria	26/12/2018	02/01/2019	7 días
Generación de la presentación	02/12/2018	08/01/2019	6 días
Entrega proyecto (memoria, presentación y código)	09/01/2019	09/01/2019	1 día

# Diagrama de Gantt

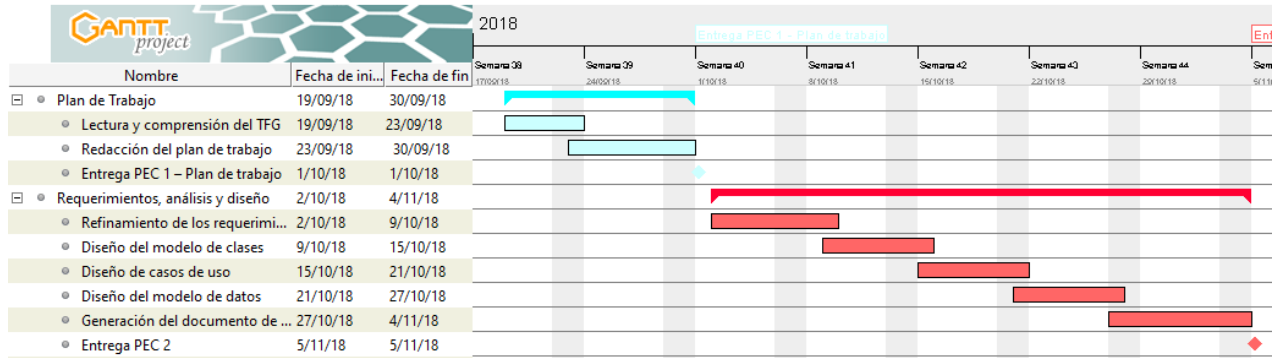


Ilustración 1 - Diagrama de Gantt inicio del proyecto

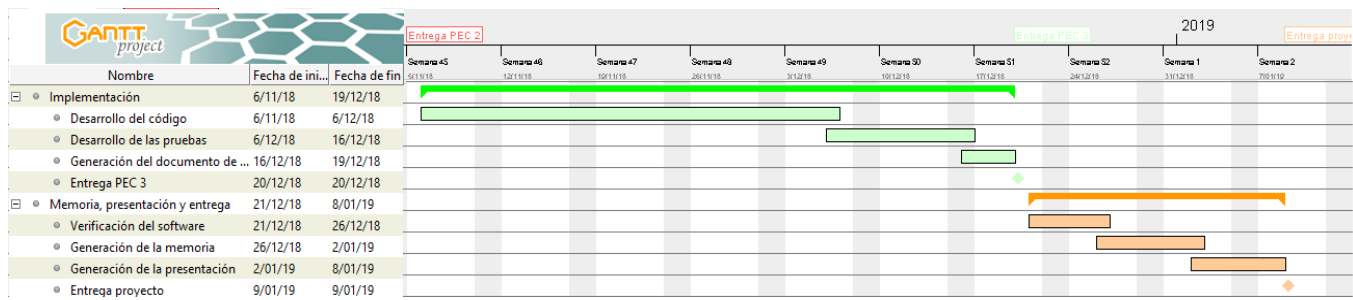


Ilustración 2 - Diagrama de Gantt final del proyecto

## 1.5 Breve resumen de productos obtenidos

Los productos obtenidos durante el desarrollo de la aplicación son los siguientes:

- Documento de análisis y diseño
- Memoria del proyecto
- Presentación virtual
- Código fuente del proyecto
- Jar desplegable automáticamente con docker

## 1.6 Breve descripción de los otros capítulos de la memoria

La memoria consta de dos capítulos principales “Análisis y diseño” e “Implementación”.

En el primero el objetivo es detallar todo el trabajo previo que se ha realizado antes de comenzar la implementación. En primer lugar, del lado del usuario realizando una fase de análisis tomando los requisitos que desea encontrar el usuario en la aplicación, detallando cuales de ellos son funcionales y no funcionales. Posteriormente se comienza el diseño de la aplicación realizando el diagrama de clases ya que la aplicación se va a desarrollar en Java que es un lenguaje orientado a objetos, también se realiza el diagrama de componentes y se analizan las diferentes tecnologías que se van a aplicar en el proyecto durante su implementación.

En el capítulo de implementación se muestra cual es la estructura del proyecto, dónde se ha localizado cada componente dentro de él y también se detallan para que vale cada fichero y cuál es su objetivo a grandes rasgos. Se muestra también el flujo de cómo sería una petición en código desde la plantilla hasta el resultado y algunas capturas de los flujos principales de la aplicación.

## 2. Análisis y Diseño

### 2.1 Análisis de requisitos

La aplicación se dividirá en varios subsistemas en el que cada uno debería cumplir su rol. Primero vamos a definir los sistemas y que queremos que realice cada uno de ellos para posteriormente ir definiendo a más bajo nivel los casos de uso, podemos decir que vamos a pasar de una foto global a su definición por etapas.

### 2.2 Requerimientos funcionales

**Sistema de autenticación de usuarios:** La aplicación permitirá el registro, login y log out de usuarios. Una vez autenticado un usuario tendrá acceso al resto de funcionalidades del sistema.

- Registro: Un usuario deberá rellenar los campos requeridos cómo usuario, contraseña, email o fecha de nacimiento para poder darse de alta en el sistema
- Log in: Un usuario previamente registrado podrá acceder al sistema mediante su usuario y contraseña.
- Log out: Un usuario que previamente ha accedido al sistema podrá cerrar su sesión actual en el sistema.

**Sistema de anuncios:** La aplicación permitirá publicación, búsqueda y modificación de anuncios.

- Búsqueda de anuncios: Un usuario previamente autenticado podrá buscar en el sistema los diferentes anuncios observando sus detalles cómo localización, descripción, franjas horarias etc. Se podrán ordenar por diversidad de criterios cómo coste asociado (en horas), puntuación de los usuarios, localización, etc.
- Solicitud de servicio: Un usuario previamente autenticado podrá solicitar un servicio a un usuario.
- Aceptación de servicio: Un usuario previamente autenticado con una solicitud de servicio en curso podrá aceptar la solicitud para poder prestar el servicio requerido.
- Rechazo de servicio: Un usuario previamente autenticado con una solicitud de servicio en curso podrá rechazar la solicitud para poder prestar el servicio requerido.

**Sistema de mensajería:** La aplicación permitirá enviar mensajes entre usuarios que van a realizar una transacción.

- Enviar mensaje: Un usuario autenticado podrá enviar mensajes a las personas con las que va a realizar la prestación de un servicio.
- Responder mensaje: Un usuario autenticado podrá responder mensajes a las personas con las que va a realizar la prestación de un servicio.

**Sistema de puntuación:** La aplicación permitirá puntuar el servicio una vez finalizado, se puntuará de 1 a 5 estrellas en base a varios criterios cómo facilidad de comunicación, calidad del servicio, puntuación global.

- Puntuar usuario: Un usuario autenticado que ha recibido un servicio con éxito podrá puntuar la transacción.

## 2.3 Requerimientos no funcionales

- NF-1: Se debe permitir el uso de la aplicación por centenas de usuarios mediante escalabilidad sencilla.
- NF-2: La interfaz de usuario debe ser sencilla e intuitiva.
- NF-3: La aplicación tiene que guardar la contraseña de los usuarios de forma segura.
- NF-4: El rendimiento de la aplicación tiene que ser óptimo en tiempos de carga.

## 2.4 Casos de uso

En los diagramas de casos de uso disponemos de tres actores:

- Usuario Autenticado (UA): Es el usuario que dispone de un perfil creado en el aplicativo y una vez que introduce sus credenciales de usuario (correo y contraseña) puede acceder al sistema. Lo abreviaremos como UA en los diagramas.
- Usuario No Autenticado (UNA): Es el usuario que no dispone de un perfil creado en el aplicativo o bien no se ha autenticado en él, debe o bien autenticarse o darse de alta en el sistema para poder acceder a las funciones del sistema. Lo abreviaremos como UNA.
- Usuario Administrador (UADM): Es el usuario que dispone de un perfil creado en el aplicativo y tiene permisos de administrador. Estos usuarios tienen algunas funciones especiales respecto a los usuarios normales con el objetivo de controlar el buen uso de la aplicación.

En primer lugar, detallamos los casos de uso objetivo de la aplicación:

Casos de uso de un Usuario No Autenticado:

- CU\_1: Acceso al sistema
- CU\_2: Registro en el sistema

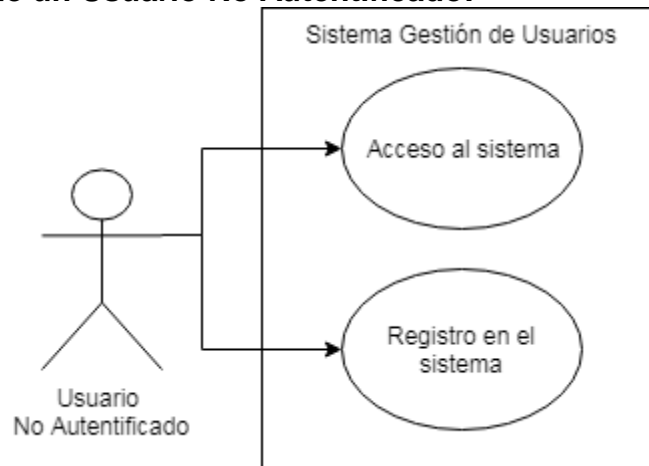
Casos de uso de un Usuario Autenticado:

- CU\_3: Salir del sistema
- CU\_4: Modificación datos de usuario
- CU\_5: Publicación de anuncio
- CU\_6: Eliminación de anuncio
- CU\_7: Modificación de anuncio
- CU\_8: Visualización de anuncio
- CU\_9: Búsqueda de anuncio
- CU\_10: Solicitud de servicio
- CU\_11: Aceptación de servicio
- CU\_12: Rechazo de servicio
- CU\_13: Visualización solicitudes
- CU\_14: Enviar mensaje
- CU\_15: Responder mensaje
- CU\_16: Puntuar usuario.
- CU\_17: Visualizar puntuaciones usuario.

Casos de uso de un Usuario Administrador:

- CU\_18: Bloqueo de usuario
- CU\_19: Bloqueo de anuncio

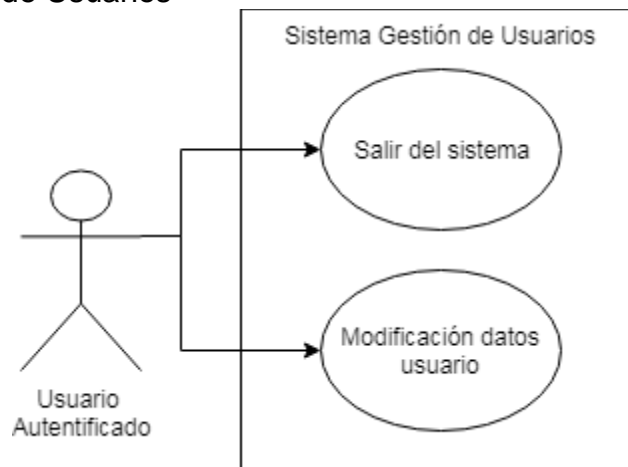
## Casos de uso de un Usuario No Autenticado:



CU_1	Acceso al sistema
Actor principal	Usuario / Administrador
Objetivos	Dar acceso al sistema al usuario con su rol correspondiente.
Precondiciones	No estar autenticado previamente
Postcondiciones	El usuario queda autenticado con su rol de usuario
Escenario principal	<ol style="list-style-type: none"> <li>1. El sistema muestra un formulario en el que se solicita usuario y contraseña</li> <li>2. El usuario introduce su usuario y contraseña y pulsa el botón "Autenticar"</li> <li>3. Se valida el usuario y contraseña contra la base de datos de usuarios</li> <li>4. Se accede a la pantalla principal de BolsaDeHoras</li> </ol>
Escenario alternativo	3a) El usuario o contraseña son incorrectos 4a) Se muestra un mensaje de error en la autenticación

CU_2	Registro en el sistema
Actor principal	Usuario
Objetivos	Dar de alta una cuenta en el sistema
Precondiciones	No estar dado de alta previamente
Postcondiciones	El usuario queda registrado en el sistema con los datos facilitados
Escenario principal	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón "Registrar en el sistema"</li> <li>2. El sistema muestra un formulario a rellenar con los campos necesarios para el alta.</li> <li>3. El usuario rellena la información y pulsa el botón "Dar de Alta"</li> <li>4. El usuario queda registrado en el sistema</li> </ol>

**Casos de uso de un Usuario Autenticado:**  
Sistema Gestión de Usuarios

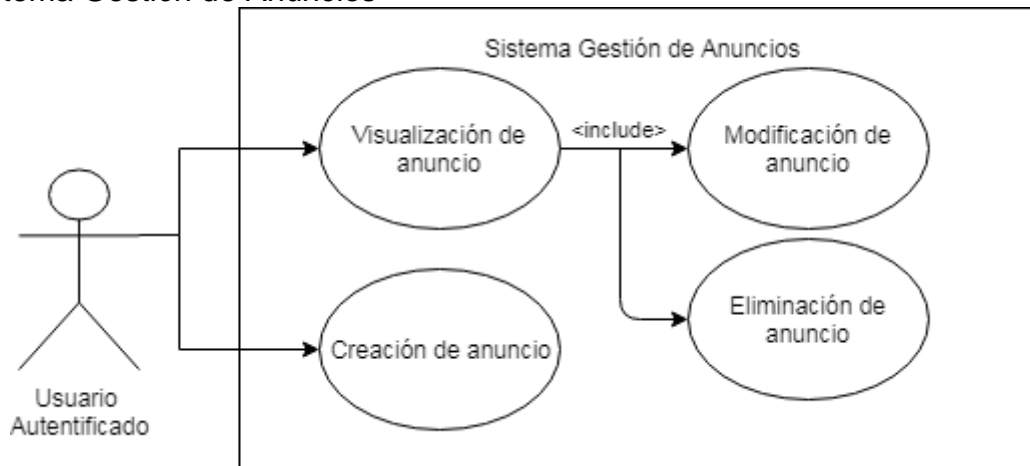


<b>CU_3</b>	<b>Salir del sistema</b>
<b>Actor principal</b>	Usuario/Administrador
<b>Objetivos</b>	Salir del sistema
<b>Precondiciones</b>	Estar autenticado en el sistema.
<b>Postcondiciones</b>	El usuario dejará de estar autenticado en el sistema
<b>Escenario principal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona "Salir del Sistema"</li> <li>2. El sistema se cierra</li> </ol>
<b>Escenario alternativo</b>	

<b>CU_4</b>	<b>Modificación de datos de usuario</b>
<b>Actor principal</b>	Usuario o Administrador registrado
<b>Objetivos</b>	Cambiar los datos de un usuario
<b>Precondiciones</b>	Un usuario previamente registrado y autenticado
<b>Postcondiciones</b>	Los datos del usuario se han modificado con éxito
<b>Escenario principal</b>	<ol style="list-style-type: none"> <li>1. Un usuario accede a la pantalla de modificación de datos.</li> <li>2. Se muestran los campos que se pueden editar.</li> <li>3. Modifica los campos que el usuario desea y están permitidos por el sistema</li> <li>4. El sistema muestra los datos del usuario</li> </ol>
<b>Escenario alternativo</b>	3a ) El usuario no rellena algún campo obligatorio o falla alguna validación 4a ) Se muestra un mensaje de error con los campos fallidos



## Sistema Gestión de Anuncios



CU_5	Publicación de anuncio
Actor principal	Usuario
Objetivos	Publicar un anuncio de una actividad desarrollada por el usuario en la plataforma
Precondiciones	Un usuario previamente registrado y autenticado
Postcondiciones	El anuncio se crea con éxito y permite al resto de usuarios su solicitud
Escenario principal	<ol style="list-style-type: none"> <li>1. Un usuario accede a la pantalla de creación de anuncios.</li> <li>2. Se muestran los campos que se pueden rellenar como Nombre, Coste, horarios, descripción o fotografías.</li> <li>3. El usuario rellena los datos de su actividad a desarrollar.</li> <li>4. El sistema muestra el anuncio creado al usuario.</li> </ol>
Escenario alternativo	3a ) El usuario no rellena algún campo obligatorio o falla alguna validación 4a ) Se muestra un mensaje de error con los campos fallidos

CU_6	Eliminación de anuncio
Actor principal	Usuario
Objetivos	Eliminar un anuncio de una actividad desarrollada por el usuario en la plataforma
Precondiciones	Un usuario previamente registrado y autenticado. Con anuncios creados
Postcondiciones	El anuncio se elimina con éxito
Escenario principal	<ol style="list-style-type: none"> <li>1. Un usuario accede a la pantalla de sus anuncios.</li> <li>2. Se muestran los anuncios creados por el usuario</li> </ol>

	<ol style="list-style-type: none"> <li>3. El usuario selecciona el anuncio a borrar</li> <li>4. El sistema elimina el anuncio</li> </ol>
<b>Escenario alternativo</b>	

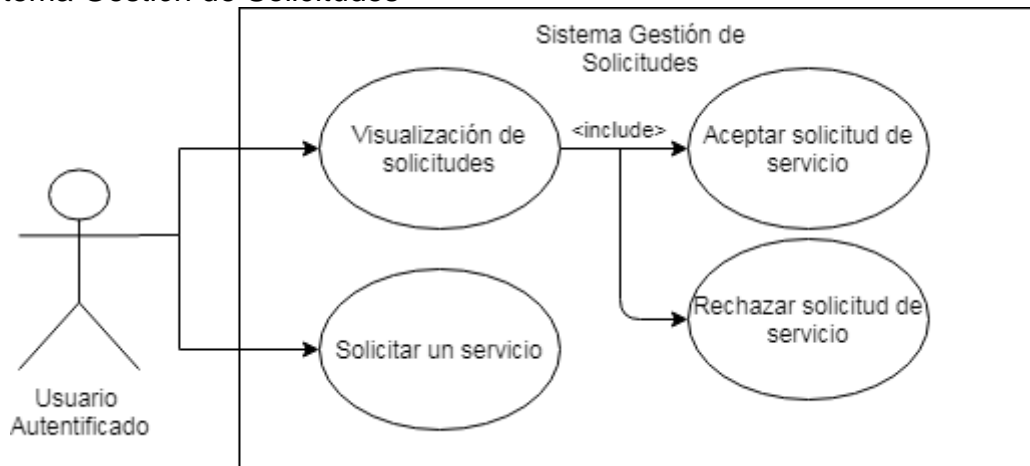
<b>CU_7</b>	<b>Modificación de anuncio</b>
<b>Actor principal</b>	Usuario
<b>Objetivos</b>	Modificar un anuncio de una actividad desarrollada por el usuario en la plataforma
<b>Precondiciones</b>	Un usuario previamente registrado y autenticado. Con anuncios creados previamente
<b>Postcondiciones</b>	El anuncio se crea con éxito y permite al resto de usuarios su solicitud
<b>Escenario principal</b>	<ol style="list-style-type: none"> <li>1. Un usuario accede a la pantalla de sus anuncios.</li> <li>2. El sistema muestra todos los anuncios creados por el usuario</li> <li>3. El usuario elige el anuncio a modificar</li> <li>4. El sistema muestra todos los campos del anuncio elegido</li> <li>5. El usuario modifica los datos de los anuncios deseados</li> <li>6. El sistema muestra el anuncio modificado al usuario.</li> </ol>
<b>Escenario alternativo</b>	<p>3a ) El usuario no rellena algún campo obligatorio o falla alguna validación</p> <p>4a ) Se muestra un mensaje de error con los campos fallidos</p>

<b>CU_8</b>	<b>Visualización de anuncio</b>
<b>Actor principal</b>	Usuario
<b>Objetivos</b>	Visualizar el detalle de un anuncio
<b>Precondiciones</b>	Un usuario previamente registrado y autenticado.
<b>Postcondiciones</b>	El anuncio se visualiza en detalle con éxito
<b>Escenario principal</b>	<ol style="list-style-type: none"> <li>1. Un usuario accede a un anuncio de la aplicación</li> <li>2. El sistema muestra todos los detalles del anuncio</li> </ol>
<b>Escenario alternativo</b>	

<b>CU_9</b>	<b>Búsqueda/Filtrado de anuncios</b>
<b>Actor principal</b>	Usuario
<b>Objetivos</b>	Buscar o filtrar los anuncios disponibles.
<b>Precondiciones</b>	Un usuario previamente registrado y autenticado
<b>Postcondiciones</b>	El conjunto de anuncios se ha filtrado en un subconjunto en base a los criterios del usuario

<b>Escenario principal</b>	<ol style="list-style-type: none"> <li>1. Un usuario elige un filtrado o ordenación en base a cualquiera de los criterios ofrecidos.</li> <li>2. El sistema devuelve los anuncios filtrados o ordenados en base a la selección del usuario.</li> </ol>
<b>Escenario alternativo</b>	

### Sistema Gestión de Solicitudes



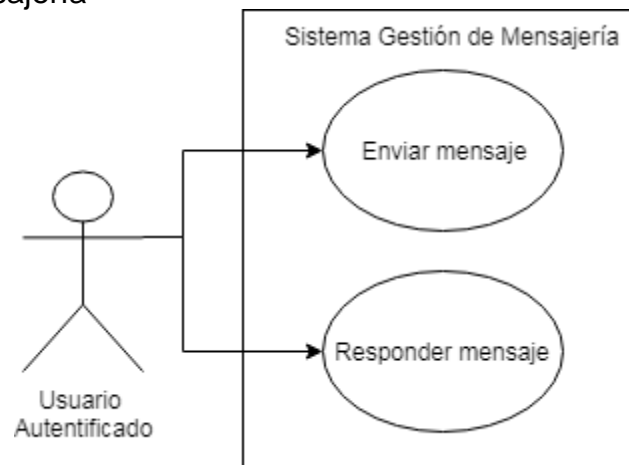
CU_10	Solicitud de servicio
<b>Actor principal</b>	Usuario
<b>Objetivos</b>	Solicitar un servicio de un anuncio
<b>Precondiciones</b>	Usuario previamente registrado. Anuncio previamente creado.
<b>Postcondiciones</b>	Se envía la solicitud al usuario del anuncio con la petición de servicio
<b>Escenario principal</b>	<ol style="list-style-type: none"> <li>1. Un usuario selecciona un servicio del que le gustaría disfrutar</li> <li>2. El sistema muestra la pantalla de solicitud de servicio</li> <li>3. El usuario elige el horario y envía un mensaje con sus detalles</li> <li>4. El sistema envía la solicitud al prestatario del servicio</li> </ol>
<b>Escenario alternativo</b>	<p>3a) El usuario no tiene horas suficientes para la prestación del servicio.</p> <p>4a) El sistema muestra un error por falta de fondos para la transacción</p>

CU_11	Aceptar solicitud de servicio
Actor principal	Usuario
Objetivos	Aceptar la solicitud de un servicio y recibir las horas realizadas.
Precondiciones	Usuario previamente registrado y autenticado. Con un anuncio creado y una solicitud de servicio de otro usuario.
Postcondiciones	El usuario recibe las horas del usuario que solicitó el servicio y el servicio queda como prestado.
Escenario principal	<ol style="list-style-type: none"> <li>1. El usuario acepta la prestación del servicio</li> <li>2. El sistema resta las horas del servicio del usuario que lo solicitó y se las suma al usuario que prestó el servicio.</li> </ol>
Escenario alternativo	

CU_12	Rechazar solicitud de servicio
Actor principal	Usuario
Objetivos	Rechazar la solicitud de un servicio.
Precondiciones	Usuario previamente registrado y autenticado. Con un anuncio creado y una solicitud de servicio de otro usuario.
Postcondiciones	El usuario rechaza el servicio para que este no se preste
Escenario principal	<ol style="list-style-type: none"> <li>1. El usuario rechaza la prestación del servicio</li> <li>2. El sistema notifica al usuario de origen que el servicio no se prestará.</li> </ol>
Escenario alternativo	

CU_13	Visualización de solicitudes
Actor principal	Usuario
Objetivos	Visualizar las solicitudes pendientes de responder
Precondiciones	Un usuario previamente registrado y autenticado. Con anuncios creados y solicitudes sobre estos anuncios
Postcondiciones	Las solicitudes recibidas se muestran.
Escenario principal	<ol style="list-style-type: none"> <li>1. Un usuario accede a la pantalla de solicitudes</li> <li>2. El sistema muestra todas las solicitudes recibidas por un usuario.</li> </ol>

## Sistema de mensajería

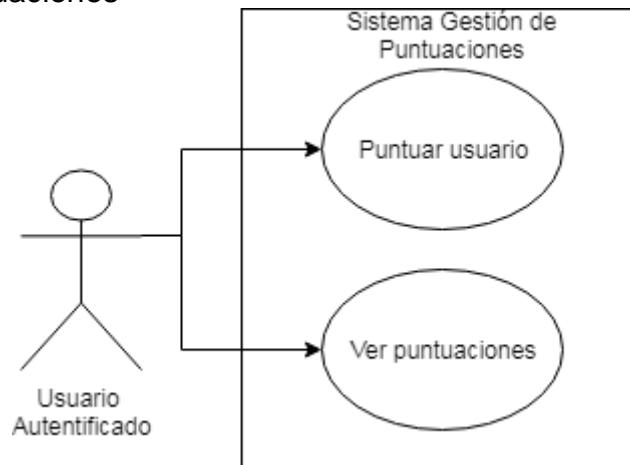


CU_14	Enviar mensaje
Actor principal	Usuario
Objetivos	Enviar un mensaje a un usuario de la aplicación
Precondiciones	Usuario previamente registrado y autenticado.
Postcondiciones	Se envía un mensaje al usuario elegido
Escenario principal	<ol style="list-style-type: none"> <li>3. Un usuario da a “Enviar mensaje” sobre otro usuario registrado en la aplicación</li> <li>4. El sistema muestra al usuario la pantalla de envío de mensaje con asunto y descripción</li> <li>5. El usuario rellena los campos y envía el mensaje.</li> <li>6. El sistema envía y notifica el mensaje al usuario elegido.</li> </ol>
Escenario alternativo	

CU_15	Responder mensaje
Actor principal	Usuario
Objetivos	Responder un mensaje recibido por otro usuario de la aplicación.
Precondiciones	Usuario previamente registrado y con mensajes recibidos.
Postcondiciones	El usuario responde el mensaje correctamente
Escenario principal	<ol style="list-style-type: none"> <li>1. El usuario da a responder mensaje ante un mensaje previamente recibido</li> <li>2. La aplicación muestra la pantalla de respuesta de mensaje con asunto y descripción</li> </ol>

	<ol style="list-style-type: none"> <li>3. El usuario rellena los campos requeridos y selecciona responder mensaje.</li> <li>4. La aplicación envía y notifica la respuesta de mensaje al usuario de origen.</li> </ol>
<b>Escenario alternativo</b>	

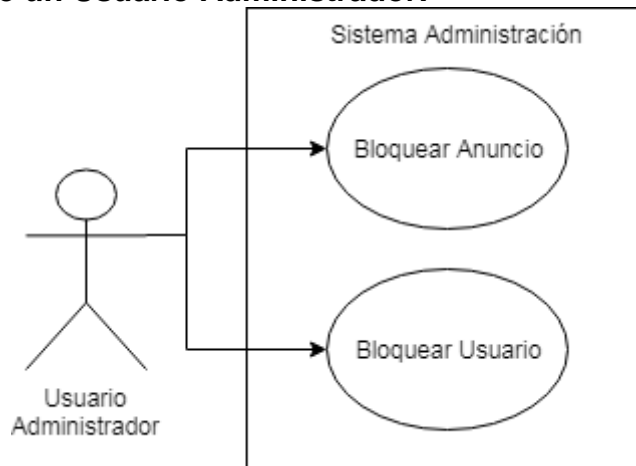
### Sistema de puntuaciones



<b>CU_16</b>	<b>Puntuar usuario</b>
<b>Actor principal</b>	Usuario
<b>Objetivos</b>	Puntuar la transacción de servicio con un usuario
<b>Precondiciones</b>	Un usuario registrado con un anuncio publicado y una solicitud de servicio aceptada.
<b>Postcondiciones</b>	El usuario recibe la puntuación en su perfil
<b>Escenario principal</b>	<ol style="list-style-type: none"> <li>1. Un usuario desea puntuar una solicitud de servicio cumplida</li> <li>2. El sistema ofrece las estrellas y la descripción de la puntuación a dar.</li> <li>3. El usuario rellena los campos requeridos</li> <li>4. El sistema añade la puntuación al perfil del usuario.</li> </ol>
<b>Escenario alternativo</b>	

<b>CU_17</b>	<b>Visualizar puntuaciones usuario</b>
<b>Actor principal</b>	Usuario
<b>Objetivos</b>	Puntuar la transacción de servicio con un usuario
<b>Precondiciones</b>	Un usuario registrado.
<b>Postcondiciones</b>	El usuario observa sus puntuaciones recibidas
<b>Escenario principal</b>	<ol style="list-style-type: none"> <li>1. Un usuario accede a la visualización de puntuaciones de usuarios</li> <li>2. El sistema ofrece las estrellas y la descripción de las puntuaciones recibidas</li> </ol>
<b>Escenario alternativo</b>	

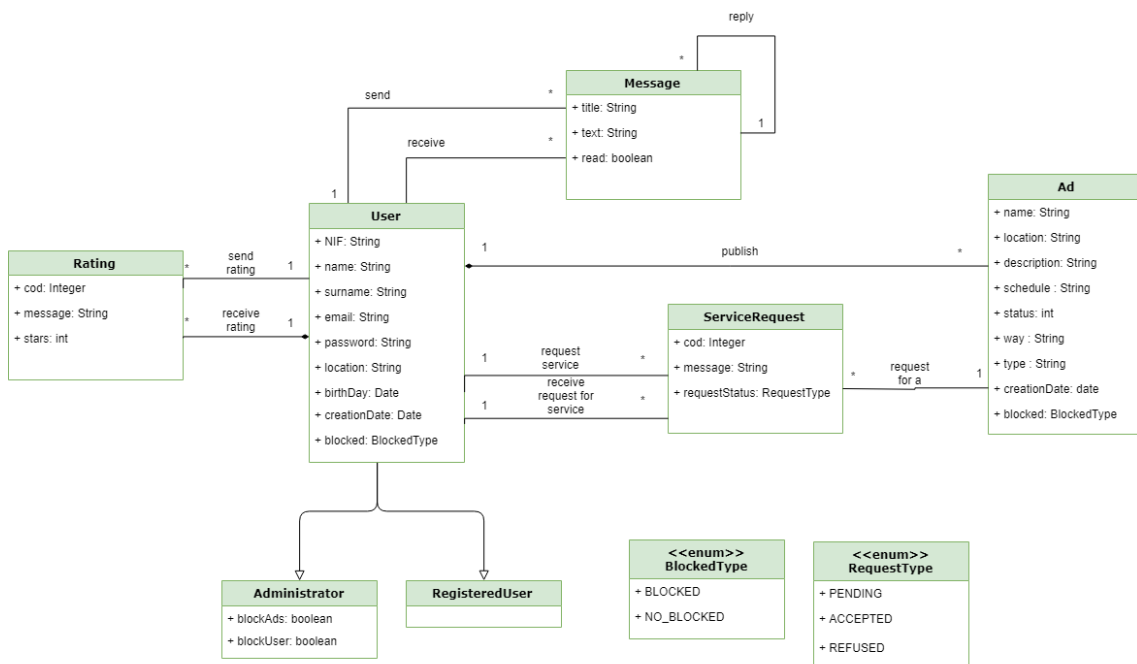
**Casos de uso de un Usuario Administrador:**



<b>CU_18</b>	<b>Bloqueo de anuncio</b>
<b>Actor principal</b>	Administrador
<b>Objetivos</b>	Bloquear el acceso a un anuncio de un usuario
<b>Precondiciones</b>	Un usuario autenticado como administrador.
<b>Postcondiciones</b>	El anuncio se bloquea en la plataforma.
<b>Escenario principal</b>	<ol style="list-style-type: none"> <li>1. Un usuario da en bloquear anuncio</li> <li>2. El sistema bloquea el acceso al anuncio de forma permanente.</li> </ol>
<b>Escenario alternativo</b>	

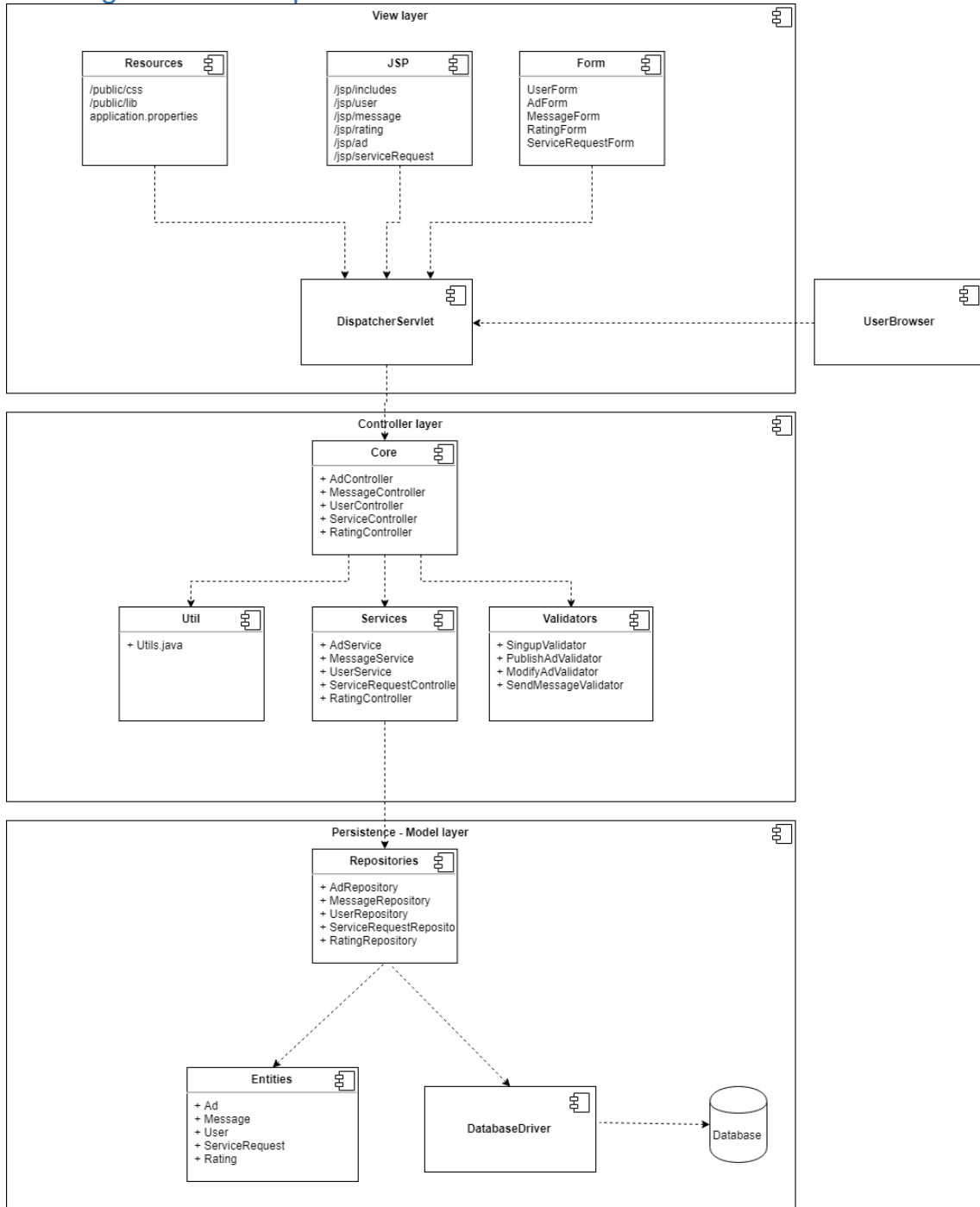
<b>CU_19</b>	<b>Bloqueo de usuario</b>
<b>Actor principal</b>	Administrador
<b>Objetivos</b>	Bloquear un usuario
<b>Precondiciones</b>	Estar autenticado como administrador en el sistema.
<b>Postcondiciones</b>	El usuario quedará bloqueado y no se podrá volver a acceder con ese nombre de usuario
<b>Escenario principal</b>	<ol style="list-style-type: none"> <li>1. El administrador selecciona "Bloquear usuario" sobre un usuario registrado</li> <li>2. El sistema bloquea el acceso al usuario.</li> </ol>
<b>Escenario alternativo</b>	

## 2.5 Diagrama de clases



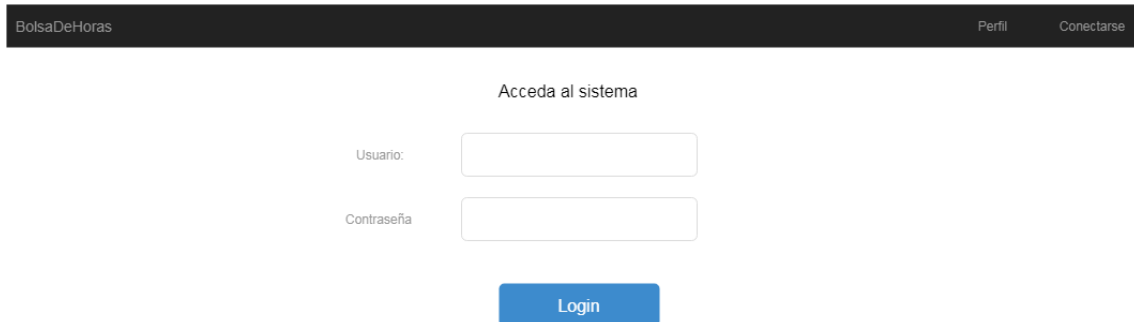


## 2.6 Diagrama de componentes



## 2.7 Prototipos

Se adjuntan modelos de pantallas provisionales en los que se ve la estructura de la aplicación. Estos modelos pueden sufrir variaciones durante la ejecución de la parte front del aplicativo.



BolsaDeHoras Perfil Conectarse

Acceda al sistema

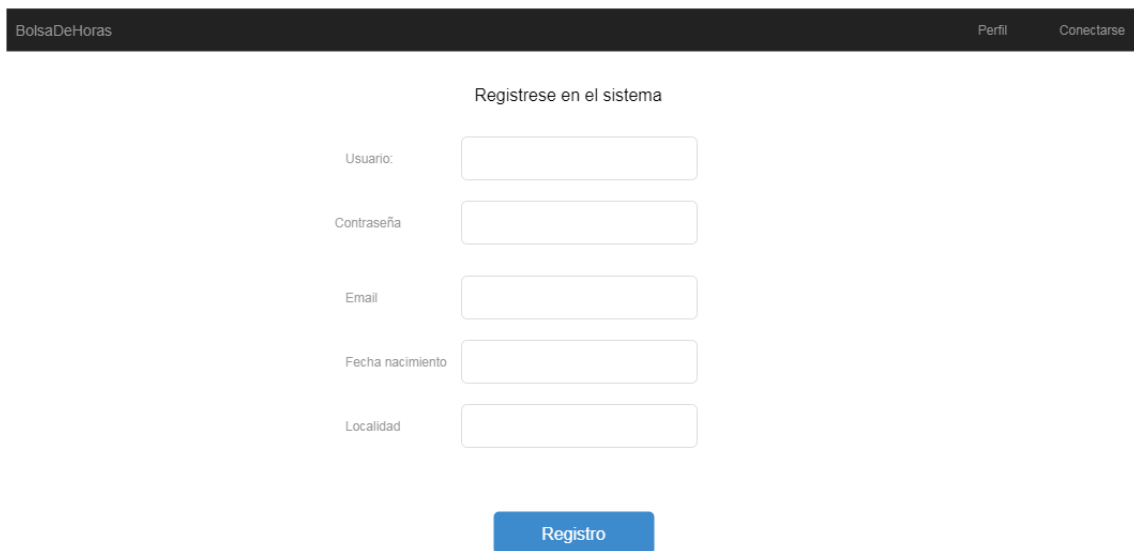
Usuario:

Contraseña:

Login

Detailed description: This is a login form prototype. It features a dark header bar with the text 'BolsaDeHoras' on the left and 'Perfil' and 'Conectarse' on the right. Below the header, the title 'Acceda al sistema' is centered. There are two input fields: one for 'Usuario' and one for 'Contraseña'. A blue button labeled 'Login' is positioned below the input fields.

Ilustración 3 – Prototipo de acceso al sistema



BolsaDeHoras Perfil Conectarse

Regístrese en el sistema

Usuario:

Contraseña:

Email:

Fecha nacimiento:

Localidad:

Registro

Detailed description: This is a registration form prototype. It features a dark header bar with the text 'BolsaDeHoras' on the left and 'Perfil' and 'Conectarse' on the right. Below the header, the title 'Regístrese en el sistema' is centered. There are five input fields: 'Usuario', 'Contraseña', 'Email', 'Fecha nacimiento', and 'Localidad'. A blue button labeled 'Registro' is positioned below the input fields.

Ilustración 4 - Prototipo de registro en el sistema

BolsaDeHoras **Anuncios 84** Solicitudes **42** Administración Perfil Desconectar

Nombre del Servicio  
Clases de Matemáticas

Localización  
Madrid

Descripción  
Soy profesor de matematicas con 10 años de experiencia  
At vero eos et accusamus et iusto odio dignissimos ducimus, qui blanditiis praesentium voluptatum deleniti atque corrupti, quosdolores et quas molestias excepturi sint, obcaecati cupiditate non provident, similique sunt in culpa, qui officia deserunt mollitiaanimi, id est laborum et dolorum fuga. Et harum quidem rerum facilis est et expedita distinctio. Nam libero tempore, cum solutanobis est eligendi optio, cumque nihil impedit, quo minus id, quod maxime placeat, facere possimus, omnis voluptas assumendaest, omnis dolor repellendus.

Forma de ejecución  
Presencial

Horario  
Semanal

Disponibilidad  
 Lunes  
 Martes  
 Miércoles  
 Jueves  
 Viernes  
 Sábado  
 Domingo

Tipo  
 Clases  
 Hogar  
 Informática  
 Internet  
 Otros

Alta Anuncio Cancelar

Ilustración 5 - Prototipo de alta de anuncio

BolsaDeHoras **Anuncios 84** Solicitudes **42** Administración

Nombre del Servicio  
Clases de Matemáticas

Localización  
Madrid

Descripción  
Soy profesor de matematicas con 10 años de experiencia  
At vero eos et accusamus et iusto odio dignissimos ducimus, qui blanditiis praesentium voluptatum deleniti atque corrupti, quosdolores et quas molestias excepturi sint, obcaecati cupiditate non provident, similique sunt in culpa, qui officia deserunt mollitiaanimi, id est laborum et dolorum fuga. Et harum quidem rerum facilis est et expedita distinctio. Nam libero tempore, cum solutanobis est eligendi optio, cumque nihil impedit, quo minus id, quod maxime placeat, facere possimus, omnis voluptas assumendaest, omnis dolor repellendus.

Forma de ejecución  
Presencial

Horario  
Semanal

Disponibilidad  
 Lunes  
 Miércoles

Tipo  
 Hogar

Anuncio de :  
JuanP2

Solicitar servicio  
Enviar mensaje

Ilustración 6 - Prototipo de detalle de anuncio

Búsqueda por palabras   Ordenación Ciudad

<p>320x200</p> <p><b>Anuncio 1</b></p> <p>Cras justo odio, dapibus ac facilisis in, egestas eget quam. Donec id elit non mi porta gravida at eget metus. Nullam id dolor id nibh ultricies vehicula ut id elit.</p> <p><input type="button" value="Ver detalle"/></p>	<p>320x200</p> <p><b>Anuncio 2</b></p> <p>Cras justo odio, dapibus ac facilisis in, egestas eget quam. Donec id elit non mi porta gravida at eget metus. Nullam id dolor id nibh ultricies vehicula ut id elit.</p> <p><input type="button" value="Ver detalle"/></p>	<p>320x200</p> <p><b>Anuncio 3</b></p> <p>Cras justo odio, dapibus ac facilisis in, egestas eget quam. Donec id elit non mi porta gravida at eget metus. Nullam id dolor id nibh ultricies vehicula ut id elit.</p> <p><input type="button" value="Ver detalle"/></p>
<p>320x200</p> <p><b>Anuncio 4</b></p> <p>Cras justo odio, dapibus ac facilisis in, egestas eget quam. Donec id elit non mi porta gravida at eget metus. Nullam id dolor id nibh ultricies vehicula ut id elit.</p> <p><input type="button" value="Ver detalle"/></p>	<p>320x200</p> <p><b>Anuncio 5</b></p> <p>Cras justo odio, dapibus ac facilisis in, egestas eget quam. Donec id elit non mi porta gravida at eget metus. Nullam id dolor id nibh ultricies vehicula ut id elit.</p> <p><input type="button" value="Ver detalle"/></p>	<p>320x200</p> <p><b>Anuncio 6</b></p> <p>Cras justo odio, dapibus ac facilisis in, egestas eget quam. Donec id elit non mi porta gravida at eget metus. Nullam id dolor id nibh ultricies vehicula ut id elit.</p> <p><input type="button" value="Ver detalle"/></p>

Ilustración 7 - Prototipo de búsqueda de anuncio

Mis anuncios

<p>320x200</p> <p><b>Anuncio 1</b></p> <p>Cras justo odio, dapibus ac facilisis in, egestas eget quam. Donec id elit non mi porta gravida at eget metus. Nullam id dolor id nibh ultricies vehicula ut id elit.</p> <p><input type="button" value="Modificar"/> <input type="button" value="Eliminar"/></p>	<p>320x200</p> <p><b>Anuncio 2</b></p> <p>Cras justo odio, dapibus ac facilisis in, egestas eget quam. Donec id elit non mi porta gravida at eget metus. Nullam id dolor id nibh ultricies vehicula ut id elit.</p> <p><input type="button" value="Modificar"/> <input type="button" value="Eliminar"/></p>
---	---

Ilustración 8 - Prototipo de eliminación/modificación de anuncio

Listado de solicitudes

Anuncio	Solicitante	Mensaje	Aceptar	Rechazar
Clases matematicas	PedroMartin	Quiero asistir a tus clases lunes de 18 a 20	<input type="button" value="Aceptar"/>	<input type="button" value="Rechazar"/>
Clases matematicas	PabloGutierrez	Quiero asistir a tus clases online	<input type="button" value="Aceptar"/>	<input type="button" value="Rechazar"/>

Ilustración 9 - Prototipo de ver solicitudes

Mensaje para

JuanP2

Título

Duda respecto a servicio

Mensaje

Tengo una duda con tus clases  
 At vero eos et accusamus et iusto odio dignissimos ducimus, qui blanditiis praesentium voluptatum deleniti atque corrupti, quosdolores et quas molestias excepturi sint, obcaecati cupiditate non provident, similique sunt in culpa, qui officia deserunt mollitiaanimi, id est laborum et dolorum fuga. Et harum quidem rerum facilis est et expedita distinctio. Nam libero tempore, cum solutanobis est eligendi optio, cumque nihil impedit, quo minus id, quod maxime placeat, facere possimus, omnis voluptas assumendaest, omnis dolor repellendus.

Ilustración 10 - Prototipo de enviar/responder mensaje

## 2.8 Decisiones tecnológicas

Para la aplicación se ha utilizado un modelo de capas (MVC). Aunque ahora entraremos en detalle en cada tecnología aplicada y las razones de su uso a grandes rasgos para el despliegue se utiliza Docker. Para la gestión de las dependencias del proyecto se utiliza Maven. Del lado de la vista se utiliza Thymeleaf como motor de plantillas para ofrecer la parte frontal de la aplicación. Los controladores están hechos con Spring Boot.

### 2.8.1 Maven

Maven es una herramienta de software para la gestión de proyectos java. Es similar a Ant o Gradle, pero tiene un modelo de configuración simple basado en XML. El objetivo es gestionar las dependencias y algunas de las configuraciones de nuestra aplicación, mediante la creación de un fichero POM ( Project Object Model) podremos establecer todas las dependencias que requiere nuestro proyecto y de dónde las debería bajar. Con esto conseguimos que cualquier persona pueda ejecutar el proyecto con facilidad, sería tan sólo tener Maven instalado y ejecutar un Maven install para que generara los artefactos del proyecto y descargara las dependencias.



Ilustración 11 - Logo Maven

### 2.8.2 Spring Boot

Spring boot tiene múltiples utilidades el objetivo principal es acortar el tiempo de desarrollo de las aplicaciones, tiene una gran cantidad de módulos que implican multitud de configuraciones. Llegado a este punto nos preguntaremos si no es costoso el descubrimiento y aplicación de estas configuraciones como con Spring Framework, la respuesta es no, ya que aplica el concepto CoC que es un paradigma que minimiza las decisiones a tomar por el desarrollador, simplificando las tareas. Con esto conseguimos ciertos valores y configuraciones por defecto que no necesitan nuestra intervención, sin embargo, si queremos podemos configurarlo de forma más amplia. Tiene un servidor integrado auto arrancable y diferentes módulos llamados starter que nos ayudarán a la ejecución de nuestra aplicación, de forma paralela reduce y simplifica las configuraciones que debemos realizar con el Spring Framework. En nuestro caso lo vamos a utilizar con múltiples objetivos, en primer lugar, como inyector de dependencias, para inversión de control, con su módulo Spring Data para el acceso a la base de datos y la creación de los repositorios y con su modulo Spring Security con el objetivo de crear una autenticación fiable.



Ilustración 12 - Logo Spring

### 2.8.2.1 Spring Security

El objetivo principal de Spring Security es la securización de aplicaciones Spring. Nos permite la creación sencilla de roles distinguiendo entre tipos de usuarios según sus credenciales. Su configuración se realiza mediante la creación de clases de usuario sin la necesidad de ficheros .xml.

### 2.8.2.2 Inversión de control

La inversión de control nos ofrece la posibilidad de controlar el flujo de ejecución de la aplicación de forma automática y transparente, es decir, nosotros no controlamos el flujo de la aplicación lo delegamos al framework de Spring.

### 2.8.2.3 Inyección de dependencias

Estamos ante un patrón de diseño que nos sirve para inyectar componentes, beans, services, etc.. en otras clases. Esto puede sonar común pero la principal diferencia radica en que nosotros no instanciamos esas clases, hay un gestor de dependencias en este caso de Spring, que se encarga de suministrarnos las dependencias especificadas en tiempo de ejecución.

### 2.8.2.4 Spring Data

Este starter nos ofrece la posibilidad de utilizar JPA/Hibernate para la gestión de nuestros repositorios con la base de datos. En el caso que nos acontece utilizaremos JPA con los correspondientes JPAREpository. Estos repositorios nos ofrecen múltiples utilidades por defecto, por ejemplo, la posibilidad de hacer queries sencillas sin implementarlas.

## 2.8.3 Thymeleaf

Thymeleaf es el sustituto natural de las tradicionales jsp. Estamos ante un moderno motor de plantillas basado en HTML5 que nos permite ofrecer plantillas elegantes y de fácil desarrollo. Tiene integración nativa con Spring Boot y para su uso simplemente se requiere de la utilización del starter apropiado y la definición de las plantillas HTML.



Ilustración 13 - Logo Thymeleaf

## 2.8.4 Docker

Docker es una plataforma para desarrollar, desplegar y correr aplicaciones con contenedores. El objetivo de correr aplicaciones en contenedores es simplificar el proceso de despliegue, hacer aplicaciones flexibles y ligeras, interoperabilidad entre diferentes sistemas operativos y escalabilidad de las aplicaciones ante su uso.

Sin el uso de docker para poder correr la aplicación que nos acontece, necesitaríamos la instalación de Mysql, creación del esquema, ejecución de los scripts sql, etc. De forma paralela necesitaríamos configurarnos Maven para el uso del proyecto, descargar las dependencias y generar un build. También requeriríamos tener una versión concreta de java, en el ejemplo es 1.8 y tener configuradas las variables de entorno.

Con docker lo único que necesitamos es tenerlo instalado y ejecutar el comando de arranque de la imagen. En docker indicamos que imagen de la base de datos mysql queremos, con que esquema, usuario y contraseña. También indicamos que versión de Linux queremos con que versión de Java concreta y cómo generar el artefacto para su despliegue. Cualquier persona con sólo tener Docker en su ordenador puede ejecutar el proyecto en cuestión de segundos y sin conocimientos técnicos.

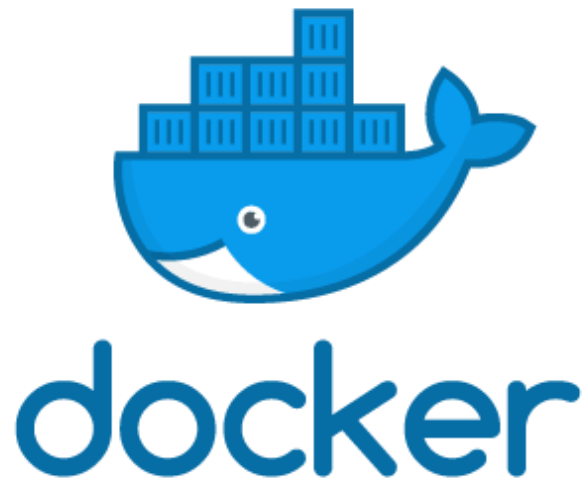


Ilustración 14 - Logo Docker



## 3. Implementación

### 3.1 Estructura del proyecto

Se muestran la estructura de paquetes del proyecto para definir la utilidad de cada componente.

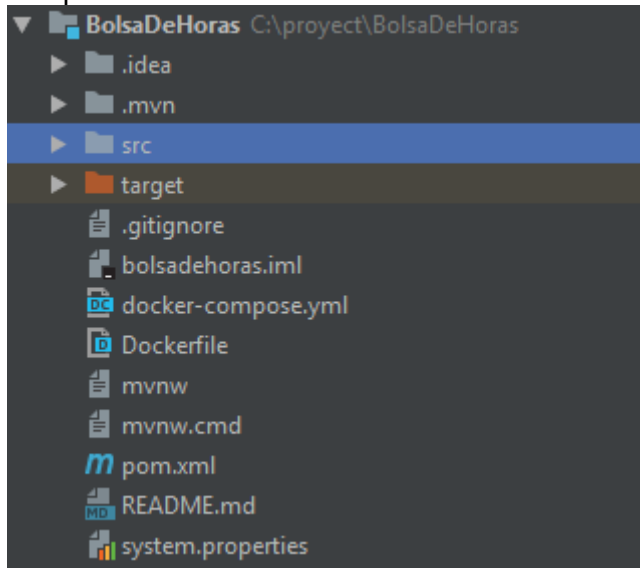


Ilustración 15 - Estructura de paquetes del proyecto

Empezando a analizar por la raíz observamos los siguientes ficheros relevantes:

#### 3.1.1 Carpeta src

Carpeta donde se localiza el código del proyecto tanto la parte frontal como la parte backend.

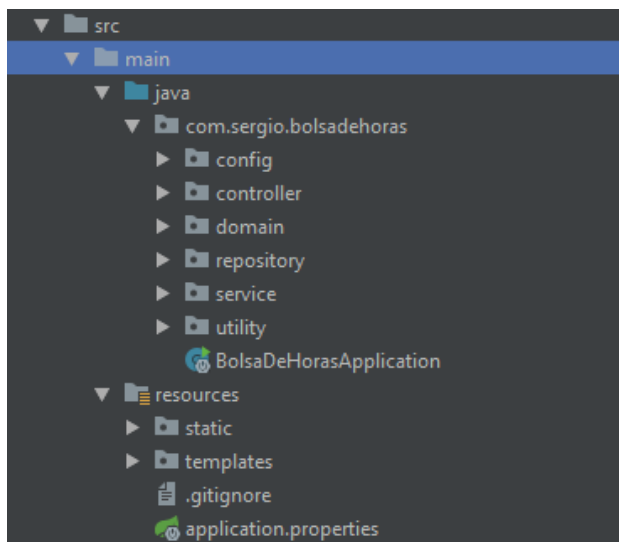


Ilustración 16 - Paquetes de la carpeta src

- Config : Archivos de configuración respecto a la autenticación y los roles de usuario. Se realiza con Spring Security
- Controller : Los controladores que reciben las peticiones frontales y resuelven las plantillas

- Domain : Las entidades de dominio con las anotaciones de JPA. Son los objetos que hacen nuestro dominio (Ad,Message,Rating,ServiceRequest y User).
- Repository : Los repositorios de JPA, nos facilitan el guardado, actualización y borrado de entidades ofreciendo métodos sencillos por defecto, en ellos también podemos ejecutar queries.
- Service : Los servicios, contiene la lógica de negocio y sirve unión entre los controladores y los repositorios, podríamos decir que son los casos de uso, ejecutan en secuencia las acciones.
- Utility : Clase de utilidades con cosas comunes cómo el encriptador de contraseñas de la aplicación

### 3.1.2 Ficheros docker

En la aplicación hay dos ficheros de docker, docker-compose y dockerfile.

Docker-compose.yml : Es el fichero que define las imágenes de docker necesarias para ejecutar la aplicación, en que puertos se ejecutan, que usuarios deben crear, etc.

```
docker-mysql:
  container_name: docker-mysql
  image: mysql:5.7
  volumes:
    - db_data:/var/lib/mysql
  restart: always
  environment:
    MYSQL_ROOT_PASSWORD: bolsadehoras
    MYSQL_DATABASE: bolsadehoras
    MYSQL_USER: bolsadehoras
    MYSQL_PASSWORD: bolsadehoras
  ports:
    - 3306:3306
```

En el ejemplo observamos que queremos crear un contenedor llamado docker-mysql, con la imagen del dockerhub mysql:5.7. Los volúmenes en los que queremos almacenar los datos, el usuario y contraseña de nuestra base de datos con el esquema utilizado y por último los puertos a usar.

Docker file: Contiene configuraciones respecto a la versión de las aplicaciones utilizadas.

```
FROM java:8
LABEL maintainer=""
VOLUME /tmp
EXPOSE 8080
ADD target/bolsadehoras-1.jar bolsadehoras-1.jar
```

En el ejemplo observamos que queremos utilizar una imagen de java 8, de dónde tiene que coger el artefacto construido y en que puerto queremos que se ejecute la aplicación.

### 3.1.3 Fichero de Maven

El fichero pom.xml define las dependencias del proyecto, es el fichero que le indica a Maven que dependencias tiene el proyecto y en que versión concreta debe descargarlas.

```

<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>2.0.0.RELEASE</version>
  <relativePath/>
</parent>

```

En el fragmento podemos observar la dependencia de Spring Boot en su versión 2.0

### 3.2 Ejemplo de flujo de la aplicación

Vamos a observar el flujo que se lleva a cabo en la aplicación ante la entrada a la página de registro

En primer lugar, llega la petición al controlador, que es el que esta escuchando peticiones POST en la api /user/newUser

```

@RequestMapping(value = "/newUser", method = RequestMethod.POST)
public String newUserPost(HttpServletRequest request,
    @ModelAttribute("email") String email,
    @ModelAttribute("username") String username,
    @ModelAttribute("password") String password,
    @ModelAttribute("name") String name,
    @ModelAttribute("surname") String surname,
    @ModelAttribute("location") String location,
    Model model) throws Exception {
    ...
    User createdUser = userService.createUser(user);
}

```

Este controlador llamará al service para el guardado del nuevo usuario después de realizar ciertas validaciones:

```

@Override
@Transactional
public User createUser(User user) throws
Exception {
    ...
    localUser = userRepository.save(user);
}

```

El service en este caso llamará al repositorio para el guardado, el repositorio no tiene implementación concreta del método ya que al implementar CrudRepository ya nos ofrece algunos métodos por defecto como el save.

```

public interface UserRepository extends CrudRepository<User, Long> {

```

Finalmente será el controlador el que devuelva la plantilla de la siguiente pantalla una vez completado el registro

```
return templatePath + "myAccount";
```

### 3.3 Resultado gráfico de la aplicación

BolsaDeHoras Anuncios Solicitudes Mensajes Identificarse Registrarse

## Bienvenido a Bolsa de Horas!

En Bolsa de Horas la moneda es el tiempo. Podrás crear anuncios de los servicios que tú puedes dar y ver los servicios que ofrecen otros usuarios. Una plataforma colaborativa en la que el intercambio es la base de la comunidad.

[Regístrate ahora »](#)

**Horas**  
Cuando te registras comienzas con dos horas. Al prestar servicios a otros usuarios recibirás las horas que hayas indicado en el coste del anuncio y al recibir servicios gastarás las horas que el dueño del anuncio ha indicado cómo coste. Cuando solicitas un servicio tus horas se restan de tu cuenta pero si el usuario cancela la solicitud recuperarás tus horas.

**Mensajería**  
¿Tienes dudas? ¿Quieres contactar con un usuario? Con el sistema de mensajería podrás enviar mensajes a cualquier usuario de la aplicación. Podrás ver tus mensajes enviados y recibidos una vez registrado bajo el enlace de "Mis mensajes". Para enviar un mensaje basta con seleccionar "Enviar Mensaje" en cualquier perfil de usuario.

**Anuncios**  
Podrás rellenar los anuncios con todo tipo de detalles, poner una foto, descripción, localización, mejor franja horaria para su prestación, etc.. En todo momento podrás modificar o borrar tus anuncios. Puedes filtrar los anuncios por categoría o por nombre.

Copyright © 2018 Sergio Fernanz Esteban. Todos los derechos reservados.

Ilustración 17 - Página de Inicio Aplicación

Vamos a crear dos usuarios para ejemplificar los flujos de la aplicación

**Crea una nueva cuenta/Conectate**

[Crea una nueva cuenta](#) [Conectarse](#)

**\* Usuario**  
  
Introduce tu nombre de usuario

**\* Email**  
  
Introduce tu dirección de mail

**\* Contraseña**  
  
Introduce tu contraseña

**Nombre**  
  
Introduce tu nombre

**Apellido**  
  
Introduce tu apellido

**Localizacion**  
  
Introduce tu localizacion

[Crea tu cuenta](#)

Ilustración 18 - Pantalla de registro

BolsaDeHoras Anuncios Solicitudes Mensajes Mi cuenta Desconectate

[Buscar anuncios](#)  
[Crear anuncios](#)  
[Mis anuncios](#)

## Auncio

\* Nombre   
Nombre del anuncio

\* Localizacion   
Localizacion del anuncio

\* Coste en horas por cada servicio.   
Introduce el coste en horas por cada anuncio/servicio. Serán las horas sumadas tras la realización del servicio

Descripcion   
Descripcion del anuncio

\* Forma de ejecución

\* Horario   
Horario recomendado

localhost:8080/ad/add


Ilustración 19 - Pantalla de creación de anuncio

BolsaDeHoras Anuncios Solicitudes Mensajes Mi cuenta Desconectate

Categorías

- Todas
- Clases
- Hogar
- Informatica
- Internet
- Otros

Show 5 entries Search:



**Clases de Inglés**  
 Nombre usuario : Sergio  
 Fecha publicación : 2019-01-03T14:59:48  
 Localizacion : Madrid  
 Horario : Lunes a Viernes de 8 a 14  
 Categoría : clases

Showing 1 to 1 of 1 entries

Previous 1 Next

Ilustración 20 - Pantalla de búsqueda de anuncio

### Detalle de anuncio

[Volver](#)



**Clases de Inglés**  
Anunciado por : [Sergio](#)  
Localización : Madrid  
Fecha de publicación: 2019-01-03T14:59:48  
Horario : Lunes a Viernes de 8 a 14  
Categoría : classes  
Forma de ejecución : remote  
Coste en horas : 2 horas

Tus horas disponibles son: 2 horas  
El coste en horas del anuncio es : 2 horas

[Solicitar anuncio](#)

Format -  
hola sergio,  
estoy muy interesado en tu anuncio te envio solicitud!


Mensaje para el anunciante

Descripción del anuncio :  
Ofrezco clases de ingles en horario de tarde. Tengo un nivel C2 certificado por Cambridge.

Ilustración 21 - Pantalla de detalle de anuncio

### Detalle de anuncio

[Volver](#)



✓La solicitud ha sido realizada con éxito

**Clases de Inglés**  
Anunciado por : [Sergio](#)  
Localización : Madrid  
Fecha de publicación: 2019-01-03T14:59:48  
Horario : Lunes a Viernes de 8 a 14  
Categoría : classes  
Forma de ejecución : remote  
Coste en horas : 2 horas

Descripción del anuncio :  
Ofrezco clases de ingles en horario de tarde. Tengo un nivel C2 certificado por Cambridge.

Ilustración 22 - Pantalla de solicitud de anuncio

Solicitudes recibidas				
Anuncio	Solicitante	Mensaje	Estado Actual	Acciones
Clases de Inglés	Maria	hola sergio, estoy muy interesado en tu anuncio te envio solicitud!	PENDIENTE	<input type="button" value="Aceptar"/> <input type="button" value="Rechazar"/>

Solicitudes enviadas				
Anuncio	Usuario Destino	Mensaje	Estado Actual	Puntua usuario
Clases de Inglés	Sergio	hola sergio, estoy muy interesado en tu anuncio te envio solicitud!	ACEPTADA	<input type="button" value="Puntuar usuario"/>

Ilustración 23 - Pantalla de solicitud de servicio

Solicitudes recibidas				
Anuncio	Solicitante	Mensaje	Estado Actual	Acciones
Clases de Inglés	Sergio	hola sergio, estoy muy interesado en tu anuncio te envio solicitud!	ACEPTADA	<input type="button" value="Puntuar usuario"/>

Solicitudes enviadas				
Anuncio	Usuario Destino	Mensaje	Estado Actual	Puntua usuario
Clases de Inglés	Sergio	hola sergio, estoy muy interesado en tu anuncio te envio solicitud!	ACEPTADA	<input type="button" value="Puntuar usuario"/>

Ilustración 24 - Pantalla de servicio completo

<

Puntuar usuario	
* Destinatario puntuación	<input type="text" value="Sergio"/> Destinatario puntuación
* Puntuacion	<input type="text" value="Puntuacion 9"/> Localizacion del anuncio
Mensaje	<div style="border: 1px solid #ccc; padding: 5px;">                         File Edit View Format                          Formats B I     </div> <div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;">                         Las clases fueron genial, sergio es un gran profesional, muchas gracias!!                     </div> <div style="border: 1px solid #ccc; padding: 5px; font-size: small;">                         P <span style="float: right;">POWERED BY TINY</span> </div> Description de la puntuación
<input type="button" value="Puntuar"/>	

Ilustración 25 - Pantalla de puntuación de usuario

### Perfil de Sergio

Envía un mensaje al usuario

**Puntuaciones del usuario**

Enviada por :	Puntuación	Mensaje
Maria	9	Las clases fueron genial, sergio es un gran profesional, muchas gracias!!

Ilustración 26 - Pantalla de perfil de usuario

### Enviar mensaje

\* Destinatario del mensaje: Sergio  
Destinatario del mensaje

\* Título del mensaje: Duda  
Titulo del mensaje

Mensaje

File Edit View Format

Format B I

Tengo una duda respecto a las clases, podrían ser los domingos por la mañana?

POWERED BY TINY

Enviar mensaje

Ilustración 27 - Pantalla de envío de mensajes

### Mensajes recibidos

Remitente	Titulo	Mensaje	Fecha de envío	Responder
Maria	Duda	Tengo una duda respecto a las clases, podrían ser los domingos por la mañana?	2019-01-03T15:02:16	Responder

### Mensajes enviados

Receptor	Titulo	Mensaje	Fecha de envío
----------	--------	---------	----------------

Ilustración 28 - Pantalla de mensajes recibidos



## 4. Conclusiones

La realización del Trabajo Final de Grado supone la finalización de una serie de años de mucho esfuerzo. Estudiar a distancia no es fácil y es un cambio muy grande respecto a la metodología tradicional que experimentas desde niño, este proyecto ha sido una vez más aprender a remar, pelearte con documentación, manuales, foros, lidiar con decenas de problemas que drenan tu tiempo, pero al final todo llega a buen puerto. Me atrevería a decir que es el culmen del aprendizaje a buscarse la vida y luchar por lo tuyo.

He aprendido a organizarme, ser metódico y consistente. Seguir las planificaciones y analizar los motivos cuando no se han podido seguir. Optimizar el tiempo al máximo, ir al grano en cada cosa y focalizarme al producto. En un proyecto informático se deben cumplir unos estándares que están más que detallados en el proyecto, pero jamás podemos perder el foco de que entregamos producto. Podemos reinventar la rueda tanto en el análisis cómo en el diseño, pero si el usuario no ve esa funcionalidad, no encuentra esa sección, o no tiene ningún beneficio por esa optimización de uno o dos segundos, no estamos entregando nada.

La metodología agile con SCRUM a la cabeza predica justo con eso, entregar producto des priorizando una excesiva documentación o manuales de usuario. Entregar, entregar y entregar producto en cada sprint de dos o tres semanas y qué este tenga la máxima calidad, puedo concluir que he aprendido a diferenciar de lo importante a lo superfluo, donde si interesa invertir tiempo y hacer un código optimizado, de baja complejidad, poco acoplado y altamente cohesionado y dónde se puede pasar por alto ciertos puntos.

De la parte tecnológica también ha sido un gran aprendizaje, hacer el proyecto en contenedores con Docker me ha enseñado otra forma de trabajar. Ya no necesitas instalar decenas de programas para hacer funcionar una aplicación, instala el JDK, la base de datos, Maven, configúralo todo, etc. Basta con definir lo que quieres y cómo lo quieres y ejecutar un simple comando cómo es “docker-compose up” para que se encargue de realizar todas estas gestiones por ti. Esto facilita enormemente el despliegue y la adhesión de nuevos miembros a los equipos, se acabaron las máquinas virtuales o los largos manuales para entrar a un proyecto, las aplicaciones basadas en contenedores han llegado para quedarse.

Con Spring Boot volvemos a priorizar la entrega de producto, con el tradicional Spring Framework la aplicación se nos llenaba de diferentes ficheros de configuración costosos de entender y de crear. Spring Boot nos da facilidad, configura sólo lo que necesites y utiliza todo lo que quieras de forma sencilla. Totalmente integrado con gestión de base de datos, motor de plantillas o plataformas de autenticación es una forma de olvidarnos de todo lo que tenemos alrededor y nos distrae y centrarnos en lo que realmente aporta al producto.

Creo firmemente haber logrado todos los objetivos, he puesto en práctica lo aprendido durante el grado, he aprendido a nivel tecnológico y he entregado un producto final que llevaba tiempo con ganas de realizar. Me parece muy interesante el mercado de economía colaborativa y en concreto el intercambio de servicios por horas está algo abandonado sin ningún referente que permita buscar y realizar transacciones con garantías.

He seguido la planificación con muy pocas variaciones, en cualquier caso, creo que la metodología seguida no ha sido la adecuada, cuando defines como es el caso que nos acontece todo con tanto detalle y esperas a tener todo tan detallado para comenzar te encuentras ciertas lagunas cuando comienzas la implementación. Hay cosas que hasta que no las ves realizadas o no la estás realizando no te percatas de ellas. Debe haber un análisis previo, pero invertir tanto tiempo para realizarlo con tanto detalle me parece un error, las cosas evolucionan y aún más en un mercado iterativo como es el de los proyectos TIC, por lo que analizar en menor detalle, hacer un producto, y mejorarlo en siguientes iteraciones me parece más correcto que analizarlo con tanto detalle desde un comienzo.

No se han introducido cambios para garantizar el éxito del trabajo, pero si en cierto momento he arrastrado ideas y decisiones desde el comienzo que hubieran sido mejorables, por ejemplo, la base de datos utilizada Mysql hubiera sido más positivo utilizar Postgresql, pero en cierto momento ya el cambio estaba fuera del scope y del tiempo del proyecto.

Ha quedado pendiente mejorar el aspecto frontal. La aplicación a nivel de back-end está realizada con una calidad bastante alta, sin embargo, las plantillas visualmente son mejorables. Al final la parte visual es cómo el escaparate de una tienda, si no te llama la atención o no te sientes cómodo no vas a entrar a ella. Creo que se podrían realizar mejoras sustanciales en el aspecto visual lo que mejoraría enormemente la navegabilidad y el uso de la aplicación.

Por otra parte la acabaría de hacer una aplicación distribuida, ahora mismo en el mismo contenedor y en la misma instancia de Spring Boot se arrancan todos los componentes, tanto frontales cómo de backend, dando una vuelta más y tratando de reducir el acoplamiento independizaría totalmente ambas partes siendo contenedores independientes de forma que si lo que tiene más demanda de recursos fueran las plantillas éstas pudieran escalar y mantener el número de instancias de la aplicación intactas o viceversa.

A nivel de funcionalidad están cumplidos, como usuario, los principales requisitos para llevar a cabo transacciones con garantía, quizás en la era de lo instantáneo que es en la que vivimos un sistema de mensajería tradicional cómo el que está realizado es menos inmediato y lo sustituiría en futuras versiones por un chat instantáneo entre usuarios.

## 5. Glosario

Framework: Es una abstracción de una serie de funcionalidades genérica que pueden ser sobrescritas o especializadas por nuestro código. Nos ofrece funcionalidades ya realizadas que facilitan el código del programador.

Bolsa de Horas: Nombre de la aplicación del presente TFG. Una implementación web de lo que tradicionalmente se denomina banco del tiempo.

JPA: Java Persistence Api, es una API de persistencia desarrollada para Java Enterprise Edition.

Spring Boot: Un framework desarrollado por Pivotal que simplifican las funcionalidades de Spring evitando la creación de código de configuración innecesario.

CoC : Convención sobre configuración es un patrón que se aplica en Spring Boot que permite al desarrollador enfocarse en el desarrollo de la aplicación eliminando la necesidad de preocuparse sobre más aspectos del ciclo de vida de la aplicación como el despliegue o la gestión.

Scrum: Marco de desarrollo ágil que tiene como objetivo realizar los proyectos de forma incremental en equipos auto organizados y de forma solapada en vez de incremental.

## 6. Bibliografía

<https://docs.docker.com/> (Consultada desde septiembre 2018 hasta enero 2019)

<https://www.thymeleaf.org/documentation.html> (Consultada desde octubre a noviembre 2018)

<https://docs.oracle.com/javase/8/docs/api/> (Consultada desde septiembre 2018 hasta enero 2019)

<https://dev.mysql.com/doc/> (Consultada desde octubre a diciembre 2018)

<https://maven.apache.org/guides/index.html> (Consultada desde octubre a diciembre 2018)

<https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/> (Consultada desde septiembre 2018 a enero 2019)

<https://docs.spring.io/spring-data/jpa/docs/current/reference/html/> (Consultada desde septiembre 2018 a enero 2019)

<https://javiermartinalonso.github.io/spring/2017/01/02/spring-boot.html>  
(Consultada en enero 2019)

# 7. Anexos

## 7.1 Despliegue de la aplicación con docker

El fichero que se encargará de la ejecución de la aplicación es el docker-compose.yml, en este fichero vienen indicados dos servicios que requiere la aplicación para arrancar docker-mysql y bolsadehoras.

El primero se ejecutará para arrancar la base de datos mysql del proyecto, no necesitaremos realizar ninguna instalación de mysql server, el mismo cogerá la imagen de mysql y arrancará la imagen con los parámetros establecidos. La base de datos será accesible en el puerto 3036.

El segundo servicio hará que se despliegue nuestra aplicación bolsa de horas y empezaremos a ver la consola y está será accesible desde el puerto 8080 de nuestra máquina.

El único requisito para la ejecución del proyecto es instalar Docker.

Instrucciones para Windows:

<https://docs.docker.com/docker-for-windows/install/#what-to-know-before-you-install>

En ese enlace detalla que hay que descargar “Docker For Windows” desde el siguiente enlace:

1. Double-click **Docker for Windows Installer.exe** to run the installer.

If you haven't already downloaded the installer ( **Docker for Windows Installer.exe** ), you can get it from [download.docker.com](https://download.docker.com). It typically downloads to your **Downloads folder** , or you can run it from the recent downloads bar at the bottom of your web browser.

2. Follow the install wizard to accept the license, authorize the installer, and proceed with the install.

You are asked to authorize **Docker.app** with your system password during the install process. Privileged access is needed to install networking components, links to the Docker apps, and manage the Hyper-V VMs.

3. Click **Finish** on the setup complete dialog to launch Docker.

Durante la instalación no realizar ningún cambio, finalizar la instalación y se nos reiniciará el ordenador.

Si se fuera a ejecutar en Mac las instrucciones serían las siguientes:

<https://docs.docker.com/docker-for-mac/install/#install-and-run-docker-for-mac>

Debería aparecernos un icono cómo que Docker está ejecutándose:



En caso contrario debemos buscar la aplicación y ejecutarla.

Navegamos a la ruta del proyecto en mi caso C:/projects/bolsadeehoras y en la raiz ejecutamos

```
docker-compose up
```

```
Microsoft Windows [Versión 10.0.17763.195]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Sergio-PC>cd ..

C:\Users>cd ..

C:\>cd proyect

C:\proyect>cd BolsaDeHoras

C:\proyect\BolsaDeHoras>docker-compose up
```

Detallando que irá pasando al ejecutar el docker-compose up del proyecto pasará por varias etapas, primero descargando la aplicación de mysql en la versión 5.7

```
Pulling docker-mysql (mysql:5.7)...
```

En segundo lugar, descargará la imagen de java y cogerá de nuestra carpeta de build (/target/bolsadehoras-1.jar) el jar de nuestra aplicación y lo ejecutará

```
Status: Downloaded newer image for java:8
---> d23bdf5b1b1b
Step 2/6 : LABEL maintainer="serferban@gmail.com"
---> Running in 8ff7f3743d03
Removing intermediate container 8ff7f3743d03
---> a0c83b7f1cd4
Step 3/6 : VOLUME /tmp
---> Running in 5275e0909e5e
Removing intermediate container 5275e0909e5e
---> 196ed0b2351b
Step 4/6 : EXPOSE 8080
---> Running in 8ffa4548abef
Removing intermediate container 8ffa4548abef
---> d316bc7f7ca4
Step 5/6 : ADD target/bolsadehoras-1.jar bolsadehoras-1.jar
---> 0b2927ee3326
Step 6/6 : ENTRYPOINT ["java","-jar","bolsadehoras-1.jar"]
---> Running in 2969f3f3f684
Removing intermediate container 2969f3f3f684
---> 24f2a67d4ec0
Successfully built 24f2a67d4ec0
Successfully tagged bolsadehoras-1:latest
```

Esto nos desplegará la imagen de mysql y la imagen de Java, empezaremos a ver los logs de ambas cosas en la consola abierta. El mensaje final será:

```
bolsadehoras_1 | 2018-12-20 18:39:50.015 INFO 1 --- [ main] c.s.b.BolsaDeHorasApplication : Started BolsaDeHorasApplication in 5.808 seconds (JVM running for 6.211
```

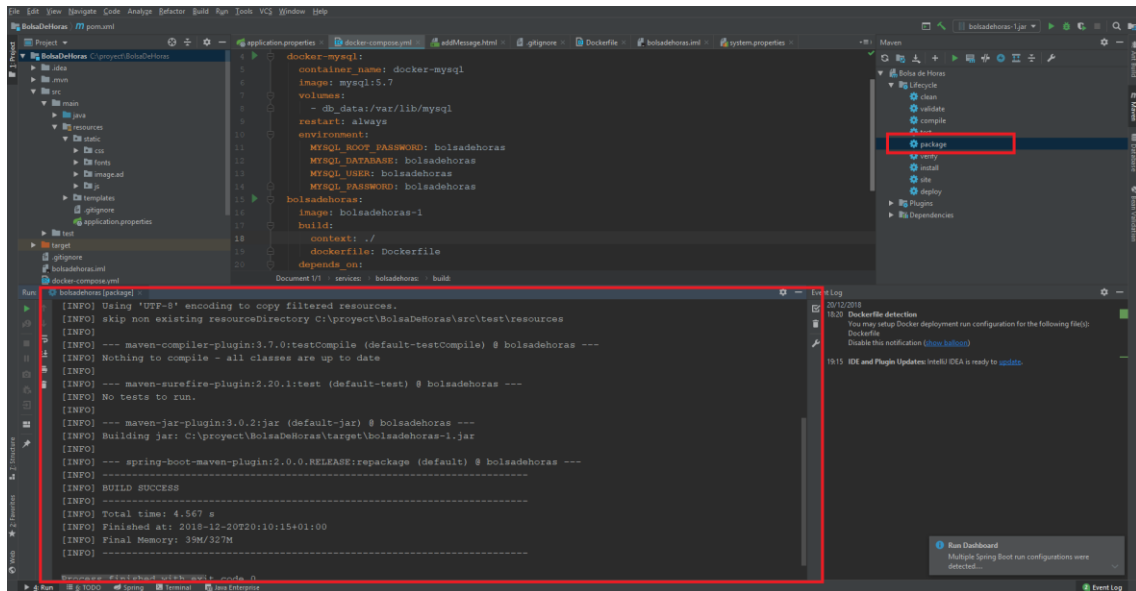
(Started BolsaDeHorasApplication in 5.808 seconds)

Ahora tendremos la aplicación arrancada con su base de datos en el puerto 8080. Para ver la aplicación navegaremos a

localhost:8080







Automáticamente si ahora ejecutamos un docker-compose up desplegaremos el artefacto actualizado.