## HERRAMIENTAS DE DESARROLLO

Para la implementación del proyecto se ha hecho uso de las siguientes herramientas. Estas se han integrado mediante el entorno de programación detallado en el punto 2.

### LENGUAGES DE PROGRAMACIÓN

#### Java



Figura 1 - Java: <https://www.java.com/en/>

El principal lenguaje de programación ha sido Java, con el que se han implementado íntegramente todas las funcionalidades que se ejecutan en el servidor, como la lógica de negocio de la aplicación, las validaciones y acceso a datos, entre otros. Las motivaciones para elegir Java frente a otros lenguajes de programación se sintetizan de la siguiente manera:

* Familiaridad por parte del desarrollador: Se trata del lenguaje más utilizado en el Grado, por lo que el aprendizaje de nuevas herramientas es rápido e intuitivo. Además, no ha sido necesario invertir tiempo en el aprendizaje de un lenguaje nuevo.
* Programación orientada a objetos: Durante la preproducción del proyecto, y en la fase de análisis y diseño, se han identificado un número considerable de entidades, que se encuentran fuertemente relacionadas entre sí. Por ello, ha resultado más intuitivo emplear un lenguaje orientado a objetos, que transforma estas entidades en clases, además de las otras ventajas que acarrea la programación orientada a objetos, como modularidad y escalabilidad.
* Popularidad: Al tratarse de uno de los lenguajes más populares, no es difícil encontrar material de ayuda y aprendizaje.
* Formación: El índice TIOBE[[1]](#footnote-1), entre otros sistemas de medición de popularidad, declaran a Java como el lenguaje más popular de la industria, por lo que su dominio se considera una fortaleza fundamental para un egresa del Grado para obtener un prometedor futuro profesional.

#### JavaScript

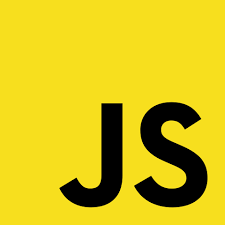


Figura 2 – JavaScript

El segundo lenguaje de programación más usado ha sido JavaScript. Se ha empleado principalmente para dotar de dinamismo a las páginas de la aplicación, así como para validar entradas.

Además, también se ha empleado para desarrollar la herramienta que calcula el balance de fuerzas en los enfrentamientos que ha sido la única parte de la lógica de negocio que se ha desarrollado en el cliente. El objetivo de esta decisión es que el Game Master disponga de un panel de mando donde pueda conocer el balance de fuerzas y los efectos de los cambios en este en tiempo real y dinámicamente.

Para completar el desarrollo en el cliente, se han empleado las siguientes librerías de JavaScript:

* jQuery[[2]](#footnote-2): Librería empleada extensivamente en toda la aplicación para facilitar y agilizar el scripting de las funcionalidades JavaScript. Entre otras utilidades, se ha empleado para cambiar dinámicamente el aspecto de la aplicación y realizar llamadas asíncronas mediante AJAX al servidor y manipular los datos recibidos en el cliente.
* jQueryUI[[3]](#footnote-3): Librería que contiene un conjunto de herramientas construidas sobre jQuery que mejoran la experiencia de usuario. Se ha empleado espontáneamente para incluir animaciones y arrastre de elementos.
* jQuery Validation[[4]](#footnote-4): Plugin del jQuery empleado en todos los formularios para validar sus entradas instantáneamente en navegador del cliente, lo que ahorra uso del servidor.

### LENGUAGES DE MARCADO

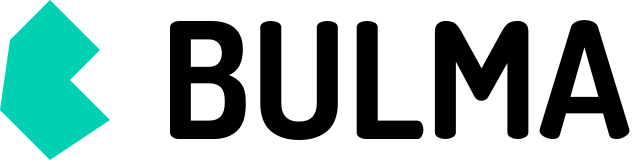


Figura 3 - Bulma: <https://bulma.io/>

Para el marcado y la estructuración de la aplicación, se han empleado HTML5 y CSS3, como es habitual en las aplicaciones que cuentan con páginas web. Para agilizar el proceso de aplicar estilo a la aplicación, se ha empleado el framework Bulma, un sustituto de Bootstrap que permite aplicar un estilo *responsive* y homogéneo a las páginas mediante un solo fichero CSS. Se ha optado por este framework frente a otros por la sencillez y modularidad que ofrece, además de contar con una documentación completa y ejemplos ya preparados. No obstante, ya que el estilo no era una prioridad en el desarrollo, se ha optado por emplear las configuraciones por defecto de Bulma.

### SPRING

## ENTORNO DE PROGRAMACIÓN

Para desarrollar el Proyecto mediante las utilidades y las tecnologías mencionadas anteriormente, se han empleado las siguientes herramientas:

### ECLIPSE



Figura 4 - Eclipse: <https://www.eclipse.org/>

La principal herramienta de desarrollo para el proyecto ha sido el IDE Eclipse. En concreto, se ha empleado Eclipse Java EE para Desarrollo Web, versión 4.7 Oxygen. Se ha optado por esta versión en concreto ya que es la más completa, ofreciendo, por un lado, integración con otros útiles como Maven y Git, y por otro, multitud de herramientas para el desarrollo de aplicaciones web, como el servidor Apache Tomcat, o editores de HTML y JavaScript, entre otras.

Las justificaciones a la hora de elegir Eclipse frente a otros IDE son las siguientes:

* Al tratarse de uno de los IDE más populares, existe una amplia gama de material consultivo, tutoriales y ayudas para poner en marcha cualquier tupo de proyecto, y resolver dudas durante su desarrollo.
* Buena coordinación con el Spring Framework.
* Software libre y de código abierto, de descarga y uso gratuito.
* Se trata de una de las herramientas más usadas en la industria, por lo que su conocimiento y familiarización son fortalezas clave para el futuro. Esta formación se complementa con el conocimiento adquirido durante el Grado, donde NetBeans es el IDE más utilizado.

### XAMPP

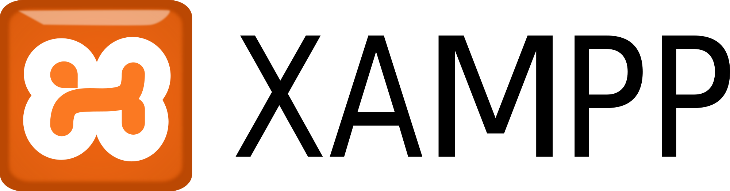


Figura 5 - XAMPP: <https://www.apachefriends.org/index.html>

La principal herramienta para la persistencia y el acceso a datos ha sido XAMPP, que permite desplegar rápidamente una base de datos de tipo MySQL para el entorno de desarrollo desde un sistema Windows, y manipularla mediante la GUI proporcionada por PhpMyAdmin. Además, esta manipulación también se puede realizar con la misma facilidad sobre una base de datos remota en el entorno de producción.

### OVH VIRTUAL PRIVATE SERVER



Figura 6 - OVH: <https://www.ovh.com/world/vps/>

Contratar un hosting para una aplicación web Java no se ha considerado económicamente viable para el alcance de este proyecto. En su lugar, se ha contratado un servidor virtual privado (referido como VPS, del ingés, *Virtual Private Server*). Por una décima parte de lo que cuesta un hosting para una aplicación Java, se obtiene una máquina virtual remota Ubuntu 16.04, con una dirección IPv4 accesible desde el exterior, donde se ha procedido a instalar MariaDB y a lanzar el fichero JAR generado por Spring Boot (que ya viene con Tomcat embebido). Además, se nos proporciona un panel de control donde se puede monitorizar el tráfico, uso de disco y memoria, entre otros. Todo ello alojado en el centro de datos de la compañía en Estrasburgo, Francia.

### FREENOM



Figura 7 - Freenom: <https://www.freenom.com/en/index.html>

Freenom ofrece dominios gratuitos durante doce meses. Se ha registrado un dominio *.tk* (perteneciente al territorio de Tokelau, administrado por Nueva Zelanda), y se ha redirigido a la IP proporcionada para la VPS de OVH mediante la herramienta de gestión de DNS proporcionada por Freenom.

### HERRAMIENTAS DE SSH

Ya que Windows no cuenta con una herramienta de SSH nativa, se ha optado por emplear dos herramientas de software libre y código abierto para gestionar la VPS remotamente:

#### PuTTY



Figura 8 PuTTY: <https://www.putty.org/>

Permite conectarse a través de SSH al usuario de Ubuntu desde Windows y acceder a la terminal para ejecutar los comandos. Además, permite copiar y pegar entre el host y la terminal.

#### WinSCP



Figura 9 – WinSCP: <https://winscp.net/eng/index.php>

Permite conectarse desde un host Windows de manera segura a la VPS y gestionar, mediante una GUI, el sistema de archivos, como si se tratase del mismo explorador de archivos de Windows. Además, permite copiar ficheros desde el host al la VPS remota y viceversa.

1. <https://www.tiobe.com/tiobe-index/> [↑](#footnote-ref-1)
2. <https://jquery.com/> [↑](#footnote-ref-2)
3. <https://jqueryui.com/> [↑](#footnote-ref-3)
4. <https://jqueryvalidation.org/> [↑](#footnote-ref-4)