AppPhyFire

Memoria

Proyecto de Fin de Carrera

INGENIERÍA INFORMÁTICA



**Septiembre de 2018**

Autor

**Alejandro Hernández González**

Tutores

**Sara Rodríguez González**

**M.ª Isabel Asensio Sevilla**

**David Álvarez León**

Dña. Sara Rodríguez González, Dña. M.ª Isabel Asensio Sevilla y D. David Álvarez León profesoras e investigador de la Universidad de Salamanca respectivamente.

CERTIFICAN:

Que el trabajo titulado “AppPhyFire” ha sido realizado por D. Alejandro Hernández González, con DNI 70902005-C y constituye la memoria del trabajo realizado para la superación de la asignatura Proyecto de Fin de Carrera de la Titulación Ingeniería Informática de esta Universidad.

Y para que así conste a todos los efectos oportunos.

En Salamanca, a 28 de junio de 2018

Dña. Sara Rodríguez González Dña. M.ª Isabel Asensio Sevilla D. David Álvarez León

Dpto. de Informática y Automática Dpto. de Matemática Aplicada Dpto. de Matemática Aplicada

Universidad de Salamanca Universidad de Salamanca Universidad de Salamanca

# Resumen

Según el Sistema Europeo de Información sobre Incendios Forestales (EFFIS)[1], los incendios forestales suponen una amenaza actual y futura para los ecosistemas de los países europeos y otras regiones mediterráneas. Los datos de EFFIS indican que en 2017 se han producido el triple de incendios que entre 2008 y 2016. Estas cifras alarmantes apuntan al cambio climático como el principal responsable, en cuanto a temperaturas más elevadas durante períodos más largos y sequías más severas, haciendo que el riesgo de incendios sea mayor, produciéndose un mayor número de incendios, y de mayor tamaño.

Por mencionar algunos de los incendios recientes que han causado mayor impacto, recordemos que en 2015 los bosques boreales de Alaska se vieron afectados por un inusual número de incendios. En 2016, 590.000 ha se vieron afectadas por el gran incendio de Alberta, Canadá, y en California casi 230.000 ha fueron devastadas por un gran número de incendios debidos a las extremas condiciones climáticas. Durante el verano de 2017, entre el 17 y el 24 de junio de 2017, tuvo lugar el mayor incendio de Portugal que afectó especialmente a la zona de Pedrógão Grande y que acabó con 50.000 hectáreas de bosque, matando a 64 personas. Ese mismo verano, en España, se quemaron más de 100.000 hectáreas. Uno de los incidentes más significativo ha sido incendio que afectó al Parque Nacional de Doñana, la mayor reserva ecológica de Europa, Patrimonio de la Humanidad desde 1994, con 8.500 ha afectadas. En octubre del mismo año, una ola de incendios sacudió Galicia y arrasó 49.000 ha, causando la muerte de 4 personas. En estas mismas fechas la tragedia se repitió en el norte de Portugal, 38 fallecidos y miles de hectáreas afectadas.

Dada la importancia e impacto medioambiental, social y económico de los incendios forestales, los esfuerzos para luchar contra sus devastadoras consecuencias deben reforzarse a todos los niveles, desde todas las administraciones, autonómicas, nacionales y europeas, pero también desde todos los sectores que puedan aportar una mejora en el conocimiento de los incendios forestales y, por lo tanto, en la lucha para atajar este problema. En este sentido, el sector científico, con la participación del sector tecnológico, debe tener un papel relevante.

La aplicación desarrollada en este proyecto de fin de carrera se enmarca en el entorno del proyecto de investigación de la Junta de Castilla y León SA020U16, que implementa los modelos semi-físicos (PhFFS y HDWM) desarrollados por el grupo de investigación en Simulación Numérica y Cálculo Científico (SINUMCC) de la Universidad de Salamanca en un sistema de información geográfica (ArcGIS) y su acceso a través de una plataforma web. Estas herramientas denominadas PhyFire y HDWind son a las que da acceso la App desarrollada en este trabajo [2].

Así pues, AppPhyFire nace como una aplicación móvil que pretende dar acceso a través de plataformas móviles a una herramienta de calidad a los profesionales y administraciones dedicados a la extinción y prevención de incendios forestales, para que pueda ser usada desde su propio entorno de trabajo. La aplicación móvil permite visualizar los resultados de las simulaciones creadas en el sistema PhyFire desde cualquier punto en el que exista una conexión a internet. Esto otorga ventajas frente a los dispositivos tradicionales, como la posibilidad de ver los parámetros de la simulación desde la propia zona de actuación de los equipos de extinción.

**Palabras clave:** Incendio, aplicación, Android, viento, simulación, Java, REST, MYSQL, GIS.

# Summary

According to the European Forest Fire Information System (EFFIS) [1], forest fires pose a current and future threat to the ecosystems of European countries and other Mediterranean regions. EFFIS data indicate that in 2017 there have been three times as many fires as between 2008 and 2016. These alarming figures point to climate change as the main cause, in terms of higher temperatures during longer periods and more severe droughts, causing the risk of fires is greater, producing a greater number of fires, and larger.

To mention some of the recent fires that have caused the greatest impact, let us remember that in 2015 the boreal forests of Alaska were affected by an unusual number of fires. In 2016, 590,000 ha were affected by the great fire in Alberta, Canada, and in California almost 230,000 ha were devastated by many fires due to extreme weather conditions. During the summer of 2017, between June 17 and 24, 2017, the largest fire in Portugal took place, which especially affected the Pedrógão Grande area and ended with 50,000 hectares of forest, killing 64 people. That same summer, in Spain, more than 100,000 hectares were burned. One of the most significant incidents has been fire that affected the Doñana National Park, the largest ecological reserve in Europe, a World Heritage Site since 1994, with 8,500 affected. In October of the same year, a wave of fires rocked Galicia and razed 49,000 hectares, causing the death of 4 people. On these same dates the tragedy was repeated in the north of Portugal, 38 dead and thousands of hectares affected.

Given the importance and environmental, social and economic impact of forest fires, efforts to combat their devastating consequences must be reinforced at all levels, from all administrations, regional, national and European, but also from all sectors that can contribute an improvement in the knowledge of forest fires and, therefore, in the fight to tackle this problem. In this sense, the scientific-technological sector, with the participation of the technological sector, must have a relevant role.

The application developed in this final year project is part of the research project of the Junta de Castilla y León SA020U16, which implements the semi-physical models (PhFFS and HDWM) developed by the research group in Numerical Simulation and Scientific Calculation (SINUMCC) of the University of Salamanca in a geographic information system (ArcGIS) and its access through a web platform. These tools called PhyFire and HDWind are accessed by the App developed in this work [2].

Thus, AppPhyFire uses a mobile application that aims to give access through mobile platforms to a quality tool for professionals and administrations dedicated to the extinction and prevention of forest fires, so that it can be used from their own work environment. The mobile application allows the results of simulations created in the PhyFire system from any point where there is an internet connection. This gives advantages over traditional devices, such as the possibility to see the parameters of the simulation of the zone of action of the extinguishing equipment.

**Keywords:** Fire, application, Android, wind, simulation, Java, REST, MYSQL, GIS.

# Contenido

[1. Introducción y antecedentes 1](#_Toc518457871)

[1.1. Organización de la memoria 7](#_Toc518457872)

[2. Objetivos del proyecto 8](#_Toc518457873)

[2.1. Objetivos software 8](#_Toc518457874)

[2.2. Objetivos técnicos 9](#_Toc518457875)

[2.3. Objetivos personales 10](#_Toc518457876)

[3. Técnicas y herramientas 11](#_Toc518457877)

[3.1. Plataformas/Herramientas de Desarrollo 11](#_Toc518457878)

[4. Aspectos relevantes del desarrollo del proyecto 17](#_Toc518457879)

[4.1. Comunicación 17](#_Toc518457880)

[4.2. Metodología 18](#_Toc518457881)

[5. Producto final 33](#_Toc518457882)

[5.1.1. Acceso a la aplicación 33](#_Toc518457883)

[5.1.2. Registro 33](#_Toc518457884)

[5.1.3. Inicio de sesión 34](#_Toc518457885)

[5.1.4. Menú principal 35](#_Toc518457886)

[5.1.5. Demos 35](#_Toc518457887)

[5.1.6. Simulaciones 36](#_Toc518457888)

[5.1.7. Visualización de la simulación 37](#_Toc518457889)

[5.1.7.1. Eliminación de la simulación 38](#_Toc518457890)

[5.1.8. Cerrar sesión 39](#_Toc518457891)

[5.1.9. Sección de ayuda 39](#_Toc518457892)

[5.1.10. Cambio de idioma 39](#_Toc518457893)

[6. Conclusiones y líneas de trabajo futuras 41](#_Toc518457894)

[6.1. Análisis crítico de mejora 41](#_Toc518457895)

[6.2. Líneas de trabajo futuras 42](#_Toc518457896)

[7. Bibliografía 43](#_Toc518457897)

# Índice de figuras

[Figura 1 - Datos de entrada y salida de los modelos 4](#_Toc518457922)

[Figura 2 - preproceso 5](#_Toc518457923)

[Figura 3 - proceso 6](#_Toc518457924)

[Figura 4 - postproceso y visualización 6](#_Toc518457925)

[Figura 5 - cuota de mercado S.O. móviles en españa (%) 11](#_Toc518457926)

[Figura 6 - logos de android y android studio 11](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457927)

[Figura 7 - logo de arcgis 12](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457928)

[Figura 8 - logo de java 12](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457929)

[Figura 9 - logo de spring 12](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457930)

[Figura 10 - logo de tomcat 13](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457931)

[Figura 11 - logo de mysql 13](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457932)

[Figura 12 - logo de phpmyadmin 13](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457933)

[Figura 13 - logo de uml 14](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457934)

[Figura 14 - logo de visual paradigm 15](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457935)

[Figura 15 - logo de ms project 15](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457936)

[Figura 16 - logo de github 15](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457937)

[Figura 17 - Logo de REM 15](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457938)

[Figura 18 - logo de ms word 16](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457939)

[Figura 19 - logo de postman 16](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457940)

[*Figura 20 - módulo de gestion de simulaciones* 21](#_Toc518457941)

[*Figura 21 - modelo de dominio* 21](#_Toc518457942)

[*Figura 22 - propuesta de arquitectura* 22](#_Toc518457943)

[*Figura 23 - [UC-0001] inicio de sesión* 23](#_Toc518457944)

[*Figura 24 - [UC-0004]: listar simulaciones del usuario* 23](#_Toc518457945)

[*Figura 25 - patron mvc* 24](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457946)

[*Figura 26 - El flujo de procesamiento de solicitudes en Spring Web MVC (alto nivel)* 25](#_Toc518457947)

[*Figura 27 - arquitectura del servidor* 25](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457948)

[*Figura 28 – diseño arquitectónico de la aplicación móvil* 26](#_Toc518457949)

[*Figura 29 - realizacion de caso de uso [UC-0003] listar simulaciones de usuario* 27](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457950)

[*Figura 30 - realizacion de caso de uso [UC-0001] Iniciar sesión* 28](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457951)

[*Figura 31 - diagrama de despliegue* 29](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457952)

[*Figura 32 - desarrollo en Android studio con el emulador* 30](#_Toc518457953)

[*Figura 33 - Diferentes logos de la aplicación* 30](#_Toc518457954)

[*Figura 34 - Petición a la API y respuesta del servidor* 31](#_Toc518457955)

[*Figura 35 - Demostración del modelo* 32](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457956)

[*Figura 36 - Inicio de sesión* 33](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457957)

[*Figura 37 - Registro* 33](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457958)

[*Figura 38 - Login incorrecto (a)* 34](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457959)

[*Figura 39 - Login incorrecto (b)* 34](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457960)

[*Figura 40 - Menú principal de la App* 35](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457961)

[*Figura 41 - Lista de demostraciones* 35](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457962)

[*Figura 42 - Demostración de Osoño* 35](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457963)

[*Figura 43 - Demostración mostrada en manual* 36](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457964)

[*Figura 44 - Demostración mostrada en automático* 36](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457965)

[*Figura 45 - lista de simulaciones* 37](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457966)

[*Figura 46 - Lista de simulaciones con mensaje* 37](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457967)

[*Figura 47 - Visualización de la simulación* 37](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457968)

[Figura 48 - Menú de selección de capa 38](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457969)

[Figura 49 - Menú de selección de paso de tiempo 38](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457970)

[Figura 50 - Eliminar simulación 38](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457971)

[Figura 51 - Mensaje de cierre de sesión 39](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457972)

[Figura 52 - Menú principal en inglés 40](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457973)

[Figura 53 - Menú de selección de idioma 40](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457974)

[Figura 54 - Aplicación pública en Google Play 41](file:///G:\Unidades%20de%20equipo\TFG-%20Fire%20simulation-Alejandro\Memoria\AppPhyFire(memoria).docx#_Toc518457975)

Índice de tablas

[Tabla 1 - iteraciones del proyecto 19](#_Toc518457976)

[Tabla 2 - objetivo principal de la aplicación 19](#_Toc518457977)

[Tabla 3 - UC-0003: mostrar simulacion 20](#_Toc518457978)

# Introducción y antecedentes

La simulación de la propagación de incendios forestales tiene aplicaciones directas en prevención (mapas de riesgos, políticas de reforestación y diseño de cortafuegos), en la lucha contra incendios (predecir el avance de un incendio ayuda a movilizar y optimizar recursos, mejorar seguridad de los bomberos durante las obras de extinción, emitir advertencias y planificar la evacuación). En la actualidad existen numerosos modelos de incendios forestales, que se pueden clasificar de acuerdo con la naturaleza de su construcción: físicos, semi-físicos o empíricos. Para que un modelo se clasifique como físico debe atender tanto a la física como a la química de la propagación del fuego; modelos semi-físicos están definidos como aquellos que buscan representar solo la física del problema; y modelos empíricos son aquellos basados ​​en una descripción fenomenológica o regresión estadística del comportamiento de fuego [3] [4]. Estos últimos son los más usados y, aun siendo muy rápidos, tienen muchas limitaciones ya que dependen de una tasa de propagación del frente de fuego que generalmente se calcula empíricamente, por lo que solo son aplicables a situaciones en las que las condiciones son idénticos a los utilizados en su formulación y ajuste. La mayoría de este tipo de modelos se basa en la aplicación del Modelo de Rothermel [5].

Por otro lado, la razón principal de que los modelos físicos o semi-físicos no sean utilizados o se usen tan solo en investigación se debe a su complejidad y a su coste computacional.

La introducción de técnicas de programación para el aprovechamiento de los sistemas multiprocesador ha permitido un aumento significativo en la capacidad de computación de los sistemas. Estos avances han permitido que los modelos físicos y semi-físicos reduzcan el tiempo necesario para realizar la simulación. El proyecto de investigación del que esta aplicación forma parte (PhyFire) utiliza modelos físicos simplificados para una simulación más rápida que en tiempo real aprovechando estas técnicas, en concreto la API OpenMP [6].

PhyFire es una herramienta desarrollada en la Universidad de Salamanca por parte del grupo de investigación de Simulación Numérica y Cálculo Científico SINUMCC. PhyFire simula la propagación del avance de un incendio forestal integrada en un sistema de información geográfica de modo que automatiza la entrada de datos al modelo y representa los resultados de la simulación en un sistema de información geográfica (SIG), además de hacer todo ello accesible a través de una plataforma web (http://sinumcc.usal.es). La herramienta SIG en la que está integrada es ArcMap 10.4 de ArcGIS de Esri Desktop Suite. Los modelos de los que hace uso la herramienta PhyFire también has sido desarrollados por el grupo de investigación SINUMCC y son: el modelo físico simplificado PhFFS de simulación de la propagación de un frente de fuego forestal, y el modelo HDWF de simulación de campos de viento de alta definición. Además, PhyFire es capaz de realizar simulaciones en un tiempo siempre mucho menor que el tiempo real de propagación del incendio.

* ***Physical Forest Fire Spread (PhFFS):***

El modelo PhFFS es un modelo físico simplificado de propagación de incendios forestales basado en las leyes de conservación de la energía y de la masa, y resuelto con técnicas numéricas eficientes que permiten proporcionar los resultados de la simulación en tiempos muy inferiores al tiempo real de simulación. El modelo tiene en cuenta, como principales mecanismos de transmisión de calor, la convección y la radiación. Depende principalmente de la temperatura, humedad, intensidad y dirección del viento, tipo y cantidad de combustible, y pendiente del terreno. La influencia del contenido de humedad del combustible y la absorción de calor por pirólisis se tienen en cuenta a través de un operador que representa la entalpía [7]. La radiación no local de las llamas sobre la capa de combustible permite tener en cuenta el efecto que el viento y la pendiente tienen sobre la inclinación de la llama [8]. Este modelo se ha aplicado con éxito en casos reales [9] y experimentales [10].

El modelo PhFFS depende de tres parámetros relacionados con la radiación, la convección y la pérdida de calor en la vertical. Además de los datos meteorológicos, pendiente del terreno y tipo y distribución del combustible inicial, así como la temperatura inicial que nos indica la posición del foco o frente inicial, el modelo depende de usa serie de variables de entrada con valores en función del tipo de combustible, y que se describen en el trabajo de Prieto et al. 2017 [2]. El modelo proporciona en cada instante de tiempo el estado de las variables adimensionales de las ecuaciones, a saber, temperatura adimensional (u), y el combustible sólido adimensional (c).

El modelo PhFFS está implementado en C++, utilizando API OpenMP [6], para explotar las plataformas multiprocesador de hoy en día para reducir el tiempo de computación. Esta implementación se registra en el Registro de Propiedad Intelectual de España el 16 de julio 2015 con número de asiento registral 00/2015/4720.

* ***High Definition Windfield Model (HDWF):***

El modelo HDWF se basa en una adaptación de las ideas de los modelos de aguas poco profundas, mediante una aproximación asintótica de las ecuaciones de Navier-Stokes, teniendo en cuenta que las dimensiones horizontales son mucho más grandes que las verticales. El objetivo es proporcionar un campo de velocidad del viento en 3D en la capa de aire sobre la superficie de estudio, resolviendo sólo ecuaciones lineales en 2D para que pueda acoplarse con modelo de propagación de incendios 2D.

HDWF proporciona un campo de vientos 3D de alta resolución en una capa de aire sobre la superficie de estudio, por encima de la cual la temperatura superficial y la topografía no tienen efecto sobre el viento, resolviendo solo ecuaciones 2D que dependen de la temperatura en la superficie del suelo, la topografía y rugosidad del terreno, y la componente horizontal del viento meteorológico en la frontera del dominio de simulación.

Los detalles de cómo este modelo se deriva de una aproximación asintótica de las ecuaciones de Navier-Stokes se puede encontrar en el trabajo de Asensio et al. (2005) [11]. La implementación del HDWF se ingresa en el Registro Español de Propiedad Intelectual el 16 de julio de 2015 en la entrada de registro 00/2015/4721.

El uso conjunto de los modelos HDWF y PhFFS revela la influencia de los efectos térmicos en el viento, tales como las altas temperaturas que ocurren durante un incendio, pero también el efecto del viento (y también de la pendiente) en la inclinación de la llama al calcular el efecto de la radiación en la propagación del fuego.

* ***Integración en ArcGIS Desktop***

La integración de los modelos PhFFS y HDWM en ArcGIS se ha desarrollado como un complemento de Python para ArcMap. Un complemento de Python es una personalización, una colección de herramientas en una barra de herramientas que se conecta a una aplicación de ArcGIS for Desktop (ArcMap en este caso) para proporcionar funcionalidad suplementaria para realizar tareas personalizadas. El complemento desarrollado incluye un menú y una barra de herramientas que contiene una colección de herramientas personalizadas diseñadas para facilitar el uso de los modelos PhFFS y HDWM. La funcionalidad de cada herramienta se implementa como una secuencia de comandos utilizando el lenguaje de programación de Python y la biblioteca de geoprocesamiento de ArcPy [2].

La integración de los modelos PhFFS y HDWM en una herramienta GIS logra varios objetivos. En primer lugar, proporciona una herramienta simple, intuitiva y fácil de usar que es más accesible para un público más amplio que podría no estar familiarizado con los modelos PhFFS y HDWF. Además, se consigue la automatización de la adquisición y el procesamiento de datos de toda la información geográfica requerida, simplifica y reduciendo así tanto el proceso de simulación como el tiempo requerido para efectuarla. También se previenen errores de datos de entrada, lo que puede causar fallos de software o salidas incorrectas, y asegura datos completos y confiables, así como la estandarización de los datos de entrada georreferenciados. Por lo tanto, el desarrollo de este sistema mejora su usabilidad reduciendo el tiempo de simulación y el conocimiento necesario del usuario.

Como evolución de este producto y para hacerlo accesible a un mayor número de usuarios se ha desarrollado su integración en ArcGIS Server

* ***Integración en ArcGIS Server***

Los modelos PhFFS y HDWF se han integrado en ArcGIS Server mediante el desarrollo de una Extensión de Objeto de Servidor (SOE). Esto permite que el modelo se pueda ejecutar de forma concurrente y aislada. De este modo, se consigue que varios usuarios puedan hacer sus simulaciones al mismo tiempo utilizando la plataforma web, cosa que no era posible en la versión de PhyFire para escritorio.

Para entender cómo se integran los modelos PhFFS y HDWF en la herramienta SIG PhyFire en ArcGIS Server, es clave entender los datos de entrada y salida de ambos modelos que se resumen en la Figura 1, así como los pasos que realiza la herramienta en cada simulación y que se describen a continuación.

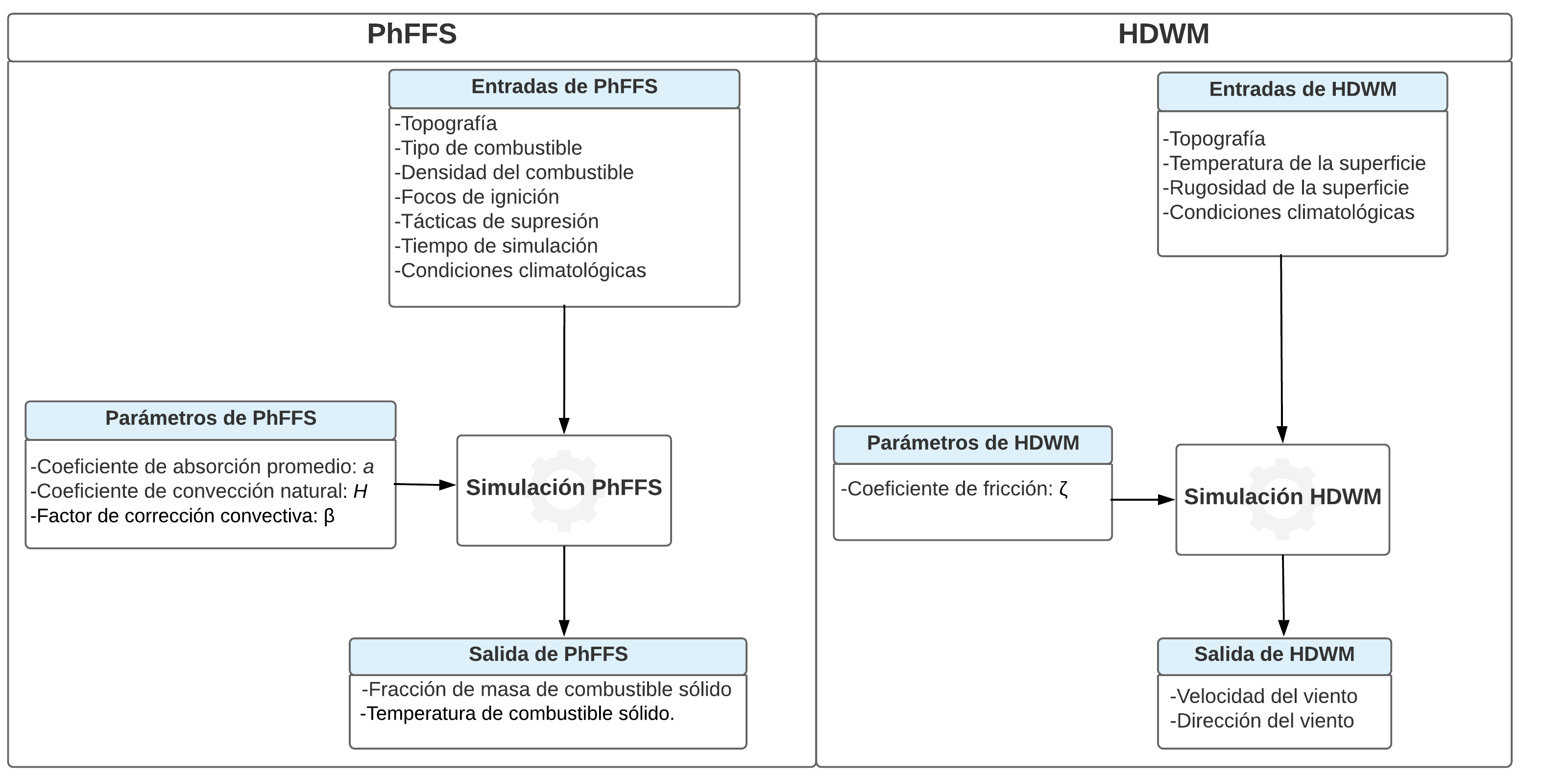


Figura 1 - Datos de entrada y salida de los modelos

* **Introducción de datos**

La herramienta PhyFire presenta al usuario un mapa base, en el que selecciona el área en la que se realizará la simulación. Además, el usuario debe introducir otros datos como el foco de incendio, líneas de defensa (opcionalmente), temperatura ambiente, humedad y viento (velocidad y dirección). Se tiene la posibilidad de que el viento sea constante para todo el dominio o introducir datos puntuales de velocidad y dirección de viento en varios puntos del dominio. Si se establece viento constante sólo se ejecuta el modelo PhFFS, mientras que cuando se introducen datos puntuales de viento, el modelo HDWF calcula un campo de viento sobre el área de simulación que mejor se ajusta a dichos datos puntuales. Este campo de viento será utilizado como datos de viento en el modelo PhFFS. El usuario puede establecer también el tiempo de simulación.

* **Pre-proceso**

Una vez que se han introducido los datos indicados, la herramienta PhyFire hace uso de tres archivos correspondientes a la topografía, tipo de combustible y carga de combustible del dominio de estudio, para obtener los datos georreferenciados necesarios para la simulación [2]

Los datos de altimetría de la zona de estudio se recortan del mapa topográfico del que dispone PhyFire, elaborado a partir del modelo digital de elevación (DEM) del Instituto Geográfico Nacional (IGN).

Los datos de tipo de combustible se extraen de un segundo mapa elaborado ad-hoc para la herramienta PhyFire a partir del Mapa Forestal Español 1: 25,000 (MFE25) combinado con el Cuarto Inventario Forestal Nacional de España (IFN4), o el Mapa Forestal Español 1: 50,000 (MFE50) con la información del Tercer Inventario Nacional Forestal Español (IFN3), dependiendo de la zona de estudio. Este mapa reúne toda la información relacionada con el tipo de combustible según la clasificación de Rothermel [12].

Los datos de carga de combustible se extraen de un tercer mapa elaborado para la herramienta PhyFire a partir del Sistema Español de Información de Cobertura de Tierras (SIOSE) seleccionando todas las superficies donde no puede propagarse un incendio (tierras estériles, masas de agua, infraestructuras de transporte, etc.). De esta forma, se extrae la información relevante para determinar todas aquellas zonas que pueden actuar como cortafuegos artificiales o naturales y que afectan la propagación del fuego.

Todos los datos previamente mencionados se almacenan en la base de datos MySQL en formato JSON y se le asigna un número de identificación (id) como podemos observar en la Figura 2.

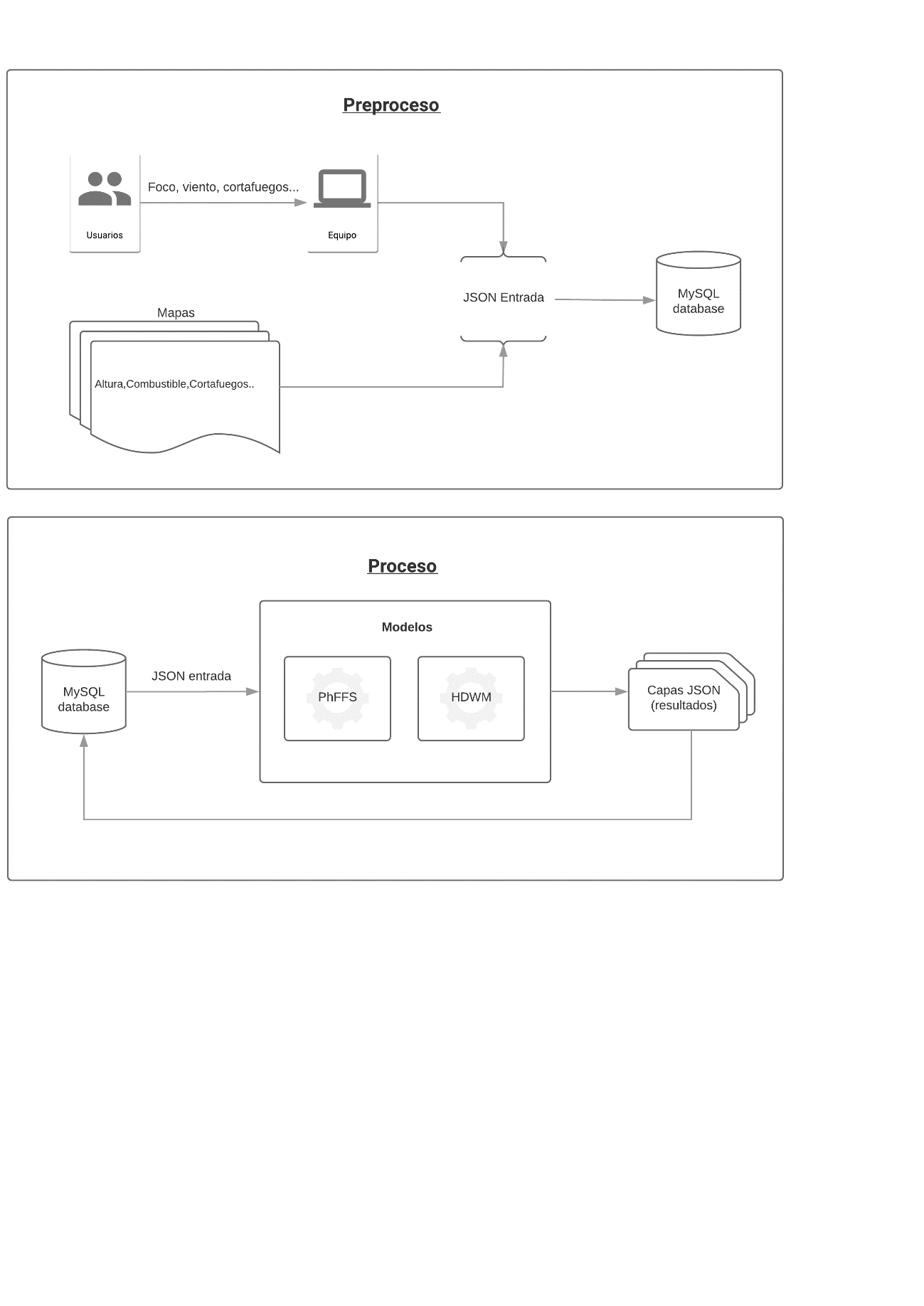


Figura 2 - preproceso

* **Proceso**

En esta etapa se realiza la simulación del avance del incendio según los datos introducidos por el usuario y los datos obtenidos durante el pre-proceso, que se pasan a los modelos PhFFS y HDWF como datos de entrada en ficheros ASCII.

El modelo PhFFS (y eventualmente HDWF) resuelve las correspondientes ecuaciones utilizando principalmente el Método de Elementos Finitos, junto con otras técnicas numéricas, sobre un mallado del dominio de simulación.

Los datos de salida que proporciona el modelo PhFFS, temperatura adimensional del combustible sólido (u), carga de combustible adimensional (c), se almacenan en ficheros ASCII que recogen los raster de estas variables sobre cada punto del mallado del área de simulación y en cada instante de tiempo en el que se requiera hacer una representación gráfica.

En la Figura 3 podemos ver el funcionamiento de este proceso, junto las variables de entrada y salida anteriormente mencionadas.

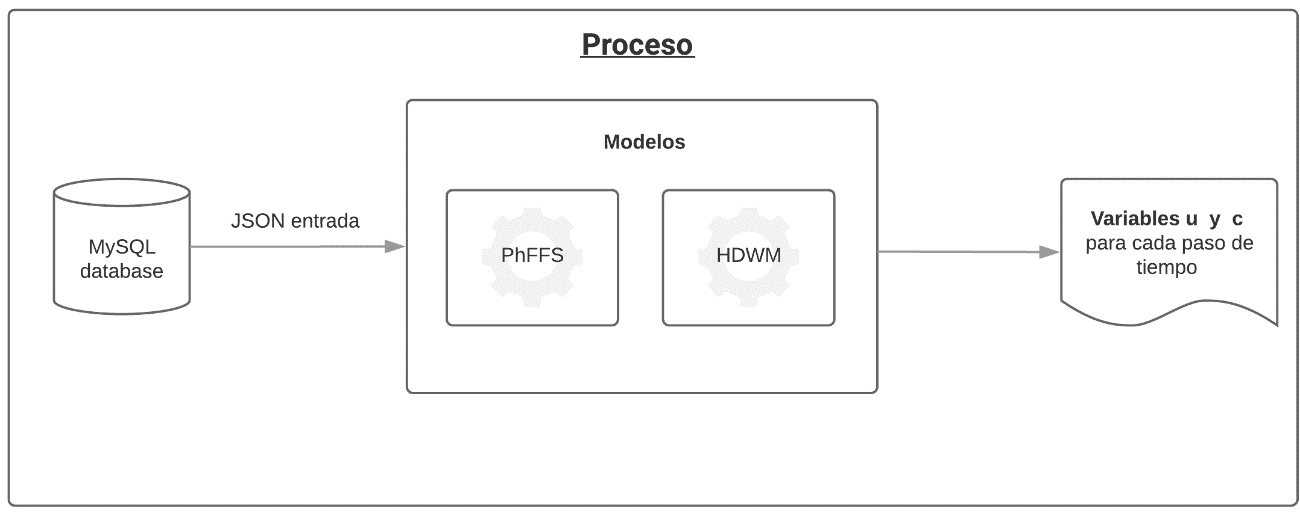


Figura 3 - proceso

* **Postproceso y visualización**

Los ficheros ASCII con los datos de salida de la simulación requieren de una transformación para poder ser visualizados en su correspondiente cartografía. Por ello, se transforman los datos ASCII generados por los modelos mediante operaciones SIG para obtener finalmente un fichero JSON que contiene una representación de los resultados.

Los datos se almacenan en la base de datos como resultado de la simulación usando el identificador previamente indicado. Estos datos almacenados en JSON serán los que se utilicen para realizar la visualización de los resultados sobre el mapa.

A continuación, vemos en la Figura 4 el proceso de postproceso y visualización simplificado:

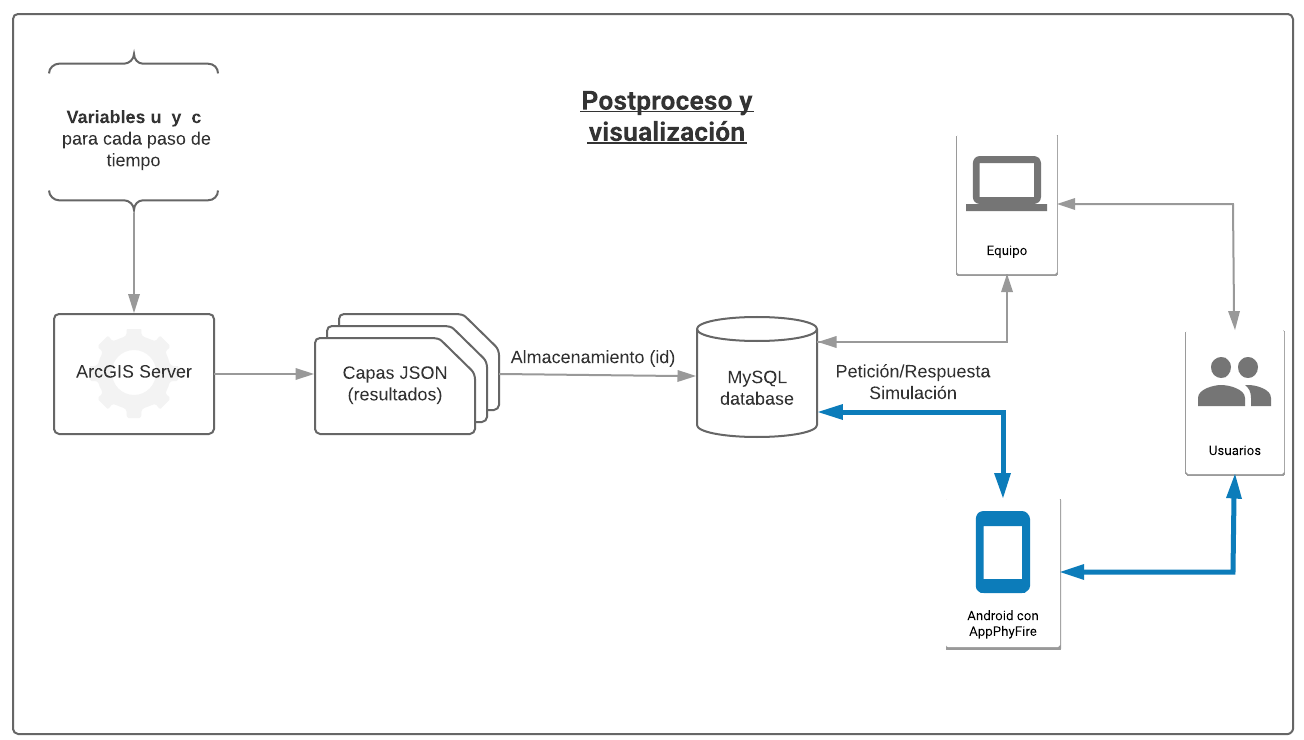


Figura 4 - postproceso y visualización

Debido a la capacidad de ArcGIS Server de generar las capas de la simulación en formato JSON, podemos utilizar su API de Android para recogerlos y mostrarlos a los usuarios en sus terminales con la aplicación instalada (como se aprecia en la zona destacada en azul de la Figura 4), saliéndonos así del marco de los dispositivos tradicionales. AppPhyFire se integra de este modo con el resto del sistema PhyFire y consigue mostrar a los usuarios cualquier simulación que hayan realizado, otorgando numerosas ventajas adicionales como la posibilidad de visualizarlo desde la propia zona de actuación y así mejorar la captura de datos o validación de los existentes.

# Organización de la memoria

Este documento se corresponde con la memoria del proyecto y consta de las siguientes secciones:

1. **Introducción y antecedentes**. Explicación preliminar del trabajo, con una síntesis del estado actual del problema planteado (sección actual).

2. **Objetivos del proyecto**. Las motivaciones para la realización de este trabajo a nivel técnico y personal.

3. **Técnicas y herramientas**. Material utilizado tanto para la construcción del sistema como para la creación de la documentación.

4. **Aspectos relevantes de desarrollo**. Decisiones y características importantes del desarrollo del proyecto.

5. **Producto final.** Descripción detallada de toda la funcionalidad implementada.

6. **Conclusiones y líneas de trabajo futuras**. Reflexión sobre los objetivos alcanzados con el proyecto y los aspectos a mejorar o a añadir si se decidiera continuar con el trabajo en el futuro.

7. **Bibliografía**. Referencias utilizadas a lo largo de esta memoria.

La memoria está compuesta por este documento (descripción del proyecto) y la descripción técnica formada por 5 anexos:

* **Anexo I Plan de proyecto software.** Planificación temporal del proyecto.
* **Anexo II Especificación de requisitos.** Recoge todos los requisitos que requiere la aplicación para su correcto funcionamiento.
* **Anexo III Especificación de Diseño.** Contiene el diseño del sistema basándonos en los requisitos y que es la base para la implementación del sistema.
* **Anexo IV Documentación técnica de programación.** Instrucciones necesarias para la correcta configuración tanto del entorno de trabajo como del despliegue final del sistema.
* **Anexo V Manual de usuario.** Guía explicativa del uso del sistema.

# Objetivos del proyecto

La informática es un campo cuya reciente expansión a través de los dispositivos móviles ha incrementado su importancia en el día a día de las personas. Gracias a los smartphones, las personas no solo pueden comunicarse mediante nuevos métodos, sino también acceder a información desde cualquier lugar. El hecho de interactuar con la tecnología de esta manera ha cambiado profundamente la vida de las personas y la forma de relacionarse durante la última década. El sector ha visto incrementada su expansión con la aparición de los dispositivos móviles, los cuales tienen actualmente un gran impacto en el día a día de las personas. Gracias a ellos, los usuarios disponen de nuevas formas de conexión y de acceso a la información desde cualquier punto, este es un hecho que ha cambiado profundamente tanto la vida de las personas como nuestras formas de interrelación.

En este nuevo paradigma, han surgido multitud de herramientas cuya finalidad es mejorar la vida cotidiana y profesional de los usuarios en forma de aplicación móvil, como por ejemplo aplicaciones que te permiten realizar tus compras desde tu terminal o consultar el tiempo en cualquier parte del mundo.

Con esta perspectiva que tiene asumida el usuario de su terminal como herramienta útil, PhyFire puede ayudar a los profesionales en su ardua tarea con la lacra que suponen los incendios forestales y optimizar su trabajo y recursos para la administración al ayudar de forma eficiente en la toma de decisiones sobre las actuaciones sobre el terreno.

Así pues, AppPhyFire nace como una aplicación que complementa al proyecto PhyFire, aportando más flexibilidad y usabilidad, ya que el objetivo principal es permitir a los profesionales la visualización de simulaciones desde su propio entorno de trabajo, pero también desde la misma zona de actuación, permitiéndole recabar datos para su posterior uso en la aplicación y de esta forma obtener simulaciones más precisas y, por lo tanto, más útiles.

A continuación, en los apartados **2.1 ,2.2** y **2.3** se presentan tanto los objetivos técnicos y de software de la aplicación como los objetivos con carácter personal:

# Objetivos software

Los objetivos descritos a continuación tratan de descomponer el objetivo principal en otros más específicos, de manera que pueda ser posible su comprobación y seguimiento a lo largo del proceso de desarrollo:

* **Visualización de simulaciones.** Los usuarios registrados deben poder seleccionar una simulación que hayan realizado con anterioridad y visualizarla en el terminal. Además, si esa simulación dispone de diferentes pasos de tiempo, deberá poder seleccionar uno específico o varios, tanto de la posición del frente como de la zona quemada.
* **Visualización de demostraciones.** Los usuarios registrados deben poder acceder a las demostraciones de funcionamiento de los modelos, estas demostraciones aumentarán la confianza de los profesionales en la fiabilidad y precisión de los modelos.
* **Interfaz sencilla.** La aplicación se presentará intuitiva y fácil de usar por cualquier persona. Se dotará de las pantallas suficientes para el manejo total de la funcionalidad.
* **Gestión de usuarios/Control de acceso.** El sistema contará con un control de acceso de manera que se permita su uso solamente a usuarios registrados. Debe permitir iniciar sesión y acceder a las simulaciones realizadas.
* **Gestión de las simulaciones.** El usuario debe ser capaz de ver todos los datos técnicos de la simulación además de poder eliminarla.
* **Integración con el sistema PhyFire.** La aplicación deberá ser compatible con los sistemas del proyecto PhyFire e integrarse en el mismo sin que el usuario perciba ninguna disparidad.
* **Multilenguaje.** La aplicación deberá poder configurarse para su visualización en distintos idiomas.
* **Facilitar el futuro mantenimiento y ampliación de la funcionalidad.** El producto final debe ser mantenible y ampliable para que pueda ser continuado por algún otro desarrollador y añadir funcionalidad como notificaciones o la posibilidad de lanzar simulaciones.

# Objetivos técnicos

Para el desarrollo del proyecto se deben de llevar a cabo un conjunto de objetivos técnicos que

permitan el correcto entendimiento y manejo de ciertas técnicas y herramientas que se van a

usar a lo largo de éste:

* **Desarrollo de una aplicación móvil (Android).** El objetivo es que cada usuario pueda hacer uso de la aplicación independientemente del dispositivo móvil Android que tenga, tratando de no restar funcionalidad.
* **Utilización de REST y JSON**. La interacción entre la aplicación y el servidor tendrá lugar

a través de peticiones REST y respuestas en formato JSON, por tanto, un objetivo será

conocer en profundidad el funcionamiento de ambos y cómo utilizarlos desde la

aplicación y desde el propio servidor. Esta comunicación hará posible descentralizar la

funcionalidad básica de la aplicación, situándola en el servidor, por lo que se facilitará

el desarrollo de la aplicación para otras plataformas en el futuro. Dicho servidor mantiene la información de los usuarios y sus simulaciones asociadas en una base de datos y proporciona diferentes servicios funcionales a través de la API REST. De este modo se desacopla cierta funcionalidad y procesamiento del dispositivo móvil y, por tanto, se agiliza la aplicación de los usuarios.

Del mismo modo, este desacoplamiento permite que se puedan crear nuevas versiones de la aplicación para otras plataformas, sin necesidad de duplicar código de funcionalidades básicas, facilitando también el mantenimiento de las aplicaciones.

* **Utilización de un patrón DAO** para las conexiones con las bases de datos. De este modo, se separa la implementación del sistema del gestor de bases de datos, haciendo posible cambiarlo en el futuro si se deseara o fuera necesario.
* **Aplicación de la arquitectura en capas MVC (Modelo Vista Controlador).** Para desvincular la representación del sistema que recibe el usuario de la forma en que se

implementa la funcionalidad y, la funcionalidad, a su vez, del modelo de datos.

* **Conocimiento del sistema de información geográfica ArcGIS.** Se trata de un sistema que agrupa varias aplicaciones para la captura, edición, análisis, tratamiento, diseño, publicación e impresión de información geográfica. Se han desarrollado conocimientos sobre el producto *ArcGIS Runtime SDK* para Android.

Es ampliamente utilizado en el mercado por lo que su conocimiento puede permitir el acceso a un perfil profesional demandado en el sector.

# Objetivos personales

Además de las motivaciones técnicas del proyecto, también hay otra serie de objetivos a nivel

personal que han llevado a la realización de este proyecto y que permiten que el reto de enfrentarse a un desarrollo de esta envergadura sea posible. Dichas motivaciones son:

* **Poner en práctica los conocimientos adquiridos** durante los años de titulación aplicándolos a un proyecto real en el que se combinen distintas áreas.
* **Utilización de nuevas herramientas y lenguajes** que permitan ampliar los conocimientos y afrontar proyectos futuros con una mayor preparación.
* **Aprendizaje acerca del desarrollo de aplicaciones móviles**. Debido a un interés personal y dada la creciente importancia de este tipo de software, es importante ampliar conocimientos en este campo.
* **Utilización del proyecto como carta de presentación de cara a la futura salida al mercado laboral**. Este aspecto se ve incentivado por el hecho de que este proyecto ha sido premiado con una beca dentro del programa Transferencia de Conocimiento Universidad Empresa (TCUE-2017/2018), organizado por la Fundación General de la Universidad de Salamanca y también por la Fundación General de Universidades de Castilla y León. Se trata de un programa cuya finalidad principal es acercar el ámbito de la empresa a los estudiantes, y así incentivar a estos a que emprendan con sus propias ideas. Para ello, un comité elige entre la multitud de propuestas, las que consideran que tienen más posibilidades de ser comercializadas en el mercado. Ser beneficiario de esta beca hace que seas capaz de desarrollar la idea de una manera más formal, con un plan de negocio más claro y definido. Facilitando la tarea de generar un prototipo robusto y que vaya cogiendo fuerza según se mejora y se analiza la idea y el proyecto.
* **Crear una herramienta capaz de ayudar a otras personas**. Los incendios forestales tienen un extenso y evidente impacto allí donde ocurren, tanto social como económico y medioambiental. Crear una herramienta que ayude a los profesionales en su tarea de prevención, es un gran aliciente para la realización del proyecto y que el resultado final sea satisfactorio para que puedan desempeñar esta ardua tarea con la mayor precisión y eficiencia posible.

# Técnicas y herramientas

A continuación, se detalla una descripción exhaustiva de las herramientas y técnicas empleadas para la realización de este trabajo y los motivos de la elección de dichas herramientas.

# Plataformas/Herramientas de Desarrollo

* **Android.**

Es un sistema operativo basado en el núcleo Linux. Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tabletas y también para relojes inteligentes, televisores y automóviles. Se ha elegido esta plataforma móvil principalmente por mantener una cuota de mercado significativamente mayor que la competencia, esta razón sumada al menor coste medio de los terminales, no asegura una difusión más sencilla entre las administraciones o usuarios particulares [13].



Figura 5 - cuota de mercado S.O. móviles en españa (%)

La versión de Android mínima que soporta la aplicación es 5.0 (API versión 21).

La decisión de que *Android* fuera el sistema operativo destino del proyecto llevó también a

elegir *Android Studio* como el IDE más adecuado para programar la aplicación.

* **Android Studio**

Figura 6 - logos de android y android studio

Es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android y se basa en IntelliJ IDEA y ha sido publicado de forma gratuita a través de la Licencia Apache 2.0 [14].

* **ArcGIS**

Figura 7 - logo de arcgis

Se trata del sistema de información geográfica (GIS) elegido por el proyecto PhyFire. Proporciona herramientas para su integración en las distintas plataformas, en concreto se han utilizado sus API para Java y para Android en el servidor y cliente respectivamente [15].

* 1. Lenguajes de Programación
* **Java**

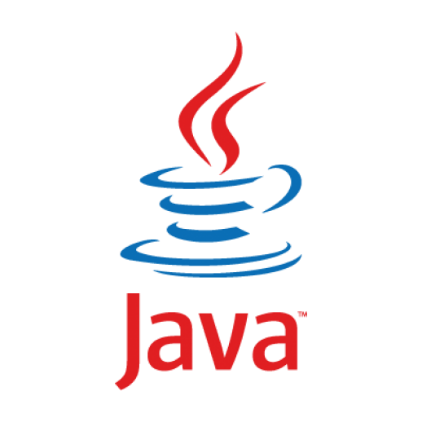
***Java*** es un lenguaje de programación de propósito general, concurrente y orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo, lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Es el principal lenguaje utilizado para la programación de aplicaciones en *Android* (siendo posible utilizar otros lenguajes como *C++* o *Kotlin*), aunque en este caso presenta algunas diferencias en cuanto a la compilación habitual de los programas en *Java* ya que no hay una máquina virtual *Java* en la plataforma. El *bytecode Java* no es ejecutado, sino que primero se compila en un ejecutable *Dalvik* y se ejecuta en la Máquina Virtual *Dalvik*. Ésta es una máquina virtual especializada, diseñada específicamente para *Android* y optimizada para dispositivos móviles que funcionan con batería y que tienen memoria y procesadores limitados. A partir de la versión 5.0 (versión mínima que soporta esta aplicación), se utiliza el *Android Runtime (ART)* en lugar de *Dalvik*.

Figura 8 - logo de java

Este lenguaje de programación ha sido empleado tanto en la parte del servidor como en la aplicación móvil.

* **Spring Framework**

Pivotal Java Spring Logo.pngSpring es un Framework para el desarrollo de aplicaciones y contenedor de inversión de control, de código abierto para la plataforma Java. El núcleo del Framework puede usarse por cualquier aplicación Java. Spring Framework proporciona un modelo de programación y configuración integral para aplicaciones empresariales modernas basadas en Java, en cualquier tipo de plataforma de implementación. Un elemento clave de Spring es el soporte de infraestructura a nivel de aplicación: Spring se enfoca en la "fontanería" de aplicaciones empresariales para que los equipos puedan enfocarse en la lógica empresarial a nivel de aplicación, sin lazos innecesarios con entornos de implementación específicos. Es de código abierto [16].

Figura 9 - logo de spring

En el proyecto se ha utilizado Spring Boot, el cual es el punto de partida para construir todas las aplicaciones basadas en Spring. Spring Boot está diseñado para ponerlo en funcionamiento lo más rápido posible, con una configuración inicial mínima de Spring. De esta herramienta se ha utilizado su facilidad para el desarrollo de una API REST, su Soporte de tiempo de ejecución incorporado (Tomcat) y su soporte para SQL.

* 1. Herramientas en el servidor

Seguidamente se describen las herramientas que forman parte de la administración del

Servidor:

* **Apache Tomcat**

Apache Tomcat (también llamado Jakarta Tomcat o simplemente Tomcat) funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Oracle Corporation [17].

Figura 10 - logo de tomcat

Durante el desarrollo se utiliza una versión de Tomcat contenida en el Framework de Spring, con el producto desarrollado se utilizará una versión de Tomcat en otro servidor.

* **MySQL**

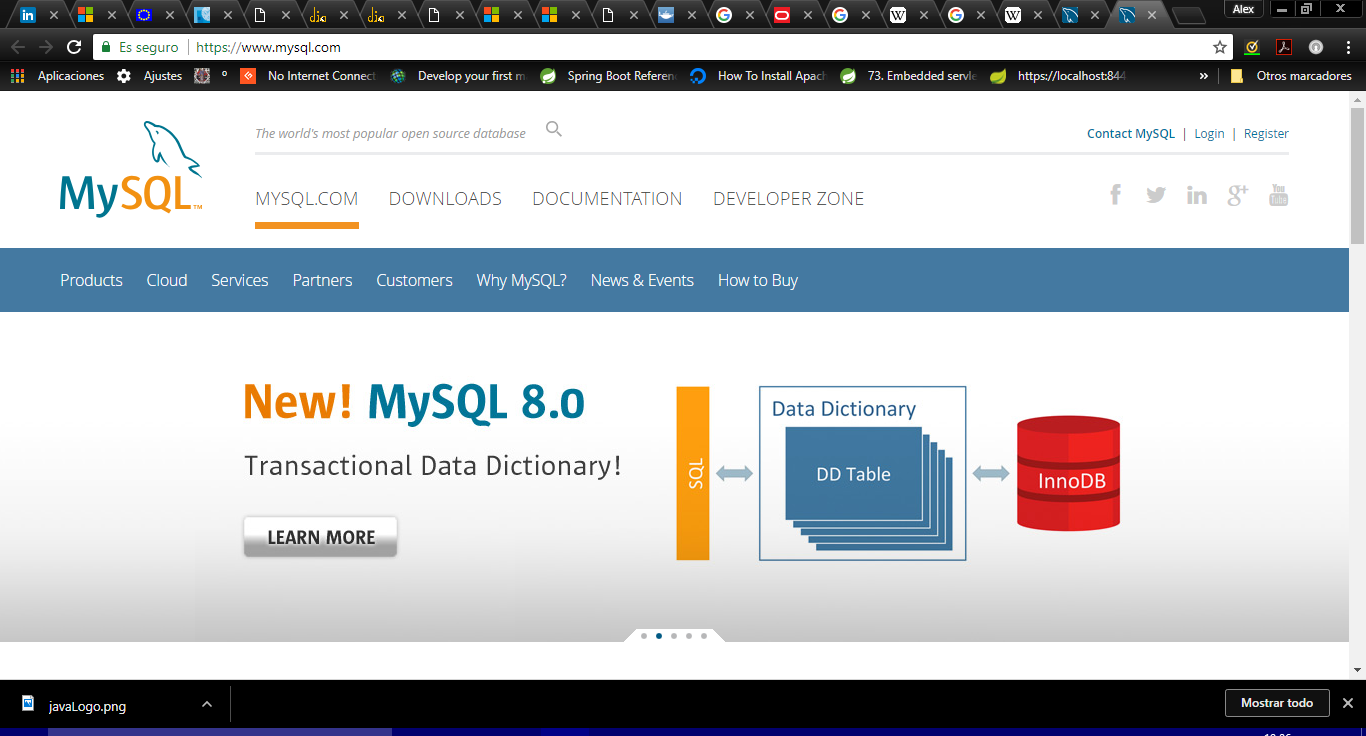
*MySQL* es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base datos de código abierto más popular del mundo y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web [18].

Figura 11 - logo de mysql

* **phpMyAdmin**



Es una herramienta de gestión de bases de datos *MySQL* escrita en *PHP*. La principal característica es que dicha gestión se realiza a través de internet por medio de un navegador web. Esto facilita enormemente el operar con la base de datos ya que no es necesario conectarse por medio de *SSH* o un protocolo similar al servidor que almacena los datos, basta con acceder a la página web e iniciar sesión con las credenciales de la base de datos [19].

Figura 12 - logo de phpmyadmin

* 1. Herramientas CASE

Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadoras), son diversas Aplicaciones informáticas destinadas a aumentar la productividad en el Desarrollo de software reduciendo el coste de estas en términos de tiempo y de dinero. Estas herramientas nos pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el diseño de proyectos, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras. Es un sistema de software que intenta proporcionar ayuda automatizada a las actividades del proceso de desarrollo de software. Los sistemas CASE a menudo se utilizan como apoyo al método [20].

A continuación, se presentan las herramientas CASE utilizadas:

* **UML**

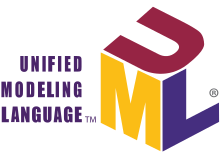
El Lenguaje Unificado de Modelado (UML) constituye el lenguaje de modelado de propósito general por excelencia para el diseño de aplicaciones orientadas a objetos y estandarizado en el campo de la ingeniería informática. Este lenguaje tiene una notación gráfica muy expresiva y completa. Incluye un conjunto de técnicas de notación gráfica para crear modelos abstractos de sistemas específicos, referidos como modelo UML. Ofrece un estándar para

Figura 13 - logo de uml

describir un “plano” del sistema (modelo).

Concretamente, se han utilizado los siguientes tipos de diagramas, dentro todos los proporcionados por UML:

* **Diagramas de casos de uso.** Modelado funcional que refleja las interacciones entre

los actores y el sistema

* **Diagramas de clases.** Colección de elementos de modelado estáticos como clases,

tipos, sus contenidos y relaciones

* **Diagramas de paquetes.** Muestran cómo los elementos de modelado se organizan en

paquetes y las dependencias entre ellos.

* **Diagramas de despliegue.** Muestra la arquitectura de ejecución del sistema en forma

de nodos conectados por caminos de comunicación.

* **Diagrama de actividad.** Representan el comportamiento dinámico del sistema

haciendo hincapié en la secuencia de actividades que se llevan a cabo y las

condiciones que guardan o disparan tales actividades.

* **Diagramas de secuencia.** Representa una interacción centrándose en la secuencia

de intercambio de mensajes entre objetos.

UML es el lenguaje gráfico de modelado de sistemas de software más conocido y utilizado en la actualidad.

* **Visual Paradigm**

Visual Paradigm (VP-UML) es una herramienta UML CASE que admite UML 2, SysML y Business Process Modeling Notation (BPMN) de Object Management Group (OMG). Además del soporte de modelado de UML, proporciona generación de informes y capacidades de ingeniería de código, incluida la generación de código [21].

Figura 14 - logo de visual paradigm

Respecto al proyecto, esta herramienta se ha utilizado en la creación de todos los diagramas UML anteriormente mencionados, gracias a una licencia que provee la Universidad de Salamanca para el uso de este Software.

* **Microsoft Project**

Microsoft Project (o MSP) es un software de administración de proyectos diseñado, desarrollado y comercializado por Microsoft para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo.[22]

Se ha utilizado en toda la planificación temporal del proyecto (inicial y final) y en la elaboración de diagramas de GANT como se describe en el Anexo I - planificación temporal del proyecto.

Figura 15 - logo de ms project

* **GitHub**

GitHub es una forja (plataforma de desarrollo colaborativo) para alojar proyectos utilizando el sistema de control de versiones Git. [23]

Figura 16 - logo de github

En el proyecto se ha utilizado para controlar las distintas versiones del código servidor y de la aplicación Android, usando distintas ramas (desarrollo y producción) para aumentar la trazabilidad de los cambios.

* **REquirements Management (REM)**

Es una herramienta experimental gratuita de Gestión de Requisitos diseñada para soportar la fase de Ingeniería de Requisitos de un proyecto de desarrollo software de acuerdo con la metodología definida en la Tesis Doctoral “Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información”, presentada por Amador Durán en septiembre de 2000 [24].

Figura 17 - Logo de REM

Se ha utilizado para toda la gestión de requisitos del proyecto.

* 1. Otras herramientas
* **MS Word**

Figura 18 - logo de ms word

Microsoft Word es el programa de ofimática por excelencia. Se trata de un programa procesador de texto muy potente, con multitud de posibilidades. Fue creado por la empresa *Microsoft*, y viene integrado predeterminadamente en el paquete ofimático denominado *Microsoft Office*, del que la Universidad de Salamanca proporciona una licencia de uso.

En el proyecto se ha utilizado para la elaboración del presente documento y de todos los anexos adjuntos.

* **Postman**

Programa de escritorio para probar APIs REST mediante el envío de envío de peticiones HTTP. Permite especificar el método de envío (GET, POST, PUT o DELETE), parámetros, configurar la cabecera, etc. y muestra por pantalla de forma ordenada el resultado devuelto por el servidor.

Figura 19 - logo de postman

En este proyecto se ha utilizado para observar las peticiones POST del cliente (aplicación Android) y de su correspondiente respuesta del servidor.

* **Gradle**

*Gradle* es un sistema de automatización de construcción de código abierto, que se ha utilizado en este proyecto, para la instalación de dependencias en ambas partes (Aplicación y Servidor), con la integración que dispone el IDE y con la aplicación de un plugin que facilita su uso respectivamente. Esta herramienta también facilita la construcción de los ejecutables necesarios para el despliegue del sistema *\*.apk* en el caso de Android y *\*.war* en el caso del servidor.

*Gradle* se encuentra integrado de forma nativa en el IDE Android Studio mencionado anteriormente y se puede incorporar en el IDE Eclipse mediante la instalación de un “*plug-in*” llamado “*Buildship Gradle Integration 2.0*”.

# Aspectos relevantes del desarrollo del proyecto

Este apartado pretende recoger los aspectos más interesantes del desarrollo del sistema software centro del proyecto, comentados por el autor de éste y que pueden servir como base documental para continuar con el desarrollo. Se ha dividido en dos partes, las cuales se han considerado los aspectos más relevantes del desarrollo:

* **Comunicación.** Explicación a grandes rasgos del funcionamiento de la comunicación entre la aplicación y el servidor, su importancia y algunos aspectos técnicos como su formato o seguridad.
* **Metodología**. Marco de desarrollo de software empleado, justificación de su elección y resumen del desarrollo y evolución del producto en el tiempo.

# Comunicación

Uno de los aspectos más importantes de este proyecto, desde la perspectiva de la complejidad del desarrollo, es la comunicación entre los sistemas que lo componen: el cliente (aplicación Android) y el servidor. Esto se debe a que toda la funcionalidad se basa en una correcta comunicación con el sistema PhyFire en concreto, su base de datos que contiene la información referida a los usuarios registrados y sus simulaciones.

El método escogido para realizar dicha comunicación es la creación de una *API REST* a la cual accederá el cliente Android. Definimos *REST* como cualquier interfaz entre sistemas que use *HTTP* para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles como *JSON.* A continuación, se explican las razones de esta decisión:

* **Separación entre el cliente y el servidor**. El protocolo *REST* separa totalmente la interfaz de usuario del servidor y el almacenamiento de datos. Eso tiene algunas ventajas cuando se hacen desarrollos. Por ejemplo, mejora la portabilidad de la interfaz a otro tipo de plataformas, aumenta la escalabilidad de los proyectos y permite que los distintos componentes de los desarrollos se puedan evolucionar de forma independiente.

Este punto es clave porque permite a este proyecto evolucionar de forma conjunta al sistema PhyFire, pero independiente.

* **La API REST siempre es independiente del tipo de plataforma o lenguaje.** La *API REST* siempre se adapta al tipo de sintaxis o plataformas con las que se estén trabajando, lo que ofrece una gran libertad a la hora de cambiar o probar nuevos entornos dentro del desarrollo. Lo único que es indispensable es que las respuestas a las peticiones se hagan siempre en el lenguaje de intercambio de información usado, normalmente *XML* o *JSON*.

PhyFireApp realiza principalmente peticiones *REST* de tipo *POST*, debido a que este método es considerado más seguro debido a que las solicitudes *GET* pueden almacenarse en caché, marcar distribuir y compartir, en definitiva, se pueden “*hackear”* notablemente más sencillo que una petición de tipo *POST*.

También se debe a que *POST* no tienen limitación de tamaño y debemos intercambiar multitud de información de extensión variable.

Como medida adicional de seguridad, se envían las peticiones sobre *HTTPS* que utiliza un cifrado basado en *SSL/TLS* para crear un canal cifrado. De este modo se consigue que la información sensible (usuario y claves de paso normalmente) no pueda ser usada por un atacante que haya conseguido interceptar la transferencia de datos de la conexión, ya que lo único que obtendrá será un flujo de datos cifrados que le resultará imposible de descifrar.

Además, como medida adicional de seguridad, cualquier información sensible enviada se realiza con una encriptación adicional en *BASE64*, nunca en “*texto plano*”.

# Metodología

Uno de los aspectos más relevantes de un proyecto, es su calidad. Con el fin de conseguir el nivel de calidad más alto posible en el producto, en este proyecto se ha seguido la metodología de Proceso Unificado.

El Proceso Unificado se repite a lo largo de una serie de ciclos, llamados iteraciones, cada uno de los cuales consta de las mismas fases: Inicio, Elaboración, Construcción y Transición. Cada una de estas fases, tiene un peso diferente en cada iteración dependiendo del desarrollo del propio proyecto. Así, inicialmente se centrará más en la fase de inicio (con mayor énfasis en los requisitos) y en las últimas fases se terminará la implementación y se harán las pruebas para garantizar que el producto construido cumple las expectativas y para corregir posibles errores.

Este proceso utiliza el lenguaje estándar de modelado *UML* (Unified Modeling Language), lo que permite que se pueda adaptar a gran variedad de proyectos y organizaciones distintas, permitiendo la producción de software de calidad que satisfaga las necesidades del usuario final, dentro de la planificación marcada.

Las principales características de este proceso son tres y se presentan a continuación.

* **Es un proceso dirigido por los casos de uso.** Éstos encapsulan la funcionalidad del sistema y a su vez se utilizan para guiar todo el proceso de desarrollo. Desde las fases iniciales, este proyecto ha contado con unos requisitos sobre la funcionalidad muy estables debido a que fue concebido en paralelo al resto del sistema PhyFire también en desarrollo.
* **Centrado en la arquitectura**. La arquitectura de un sistema software es un extracto del conjunto de modelos del sistema, describe la construcción de las diferentes vistas dándonos una idea de la forma completa del sistema.
* **Tiene una naturaleza iterativa e incremental.** Hace que el Proceso Unificado se repita a lo largo de una serie de ciclos que constituyen la vida del sistema. Este proyecto ha necesitado de 6 iteraciones, debido a su complejidad, las cuales se detallan a continuación en la Tabla 1 y mucho más en profundidad en el Anexo I (Planificación temporal) que se entrega junto con el proyecto y muestra en detalle todas las actividades realizadas, junto con su duración y sus relaciones de dependencia.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Número de Iteración | Fase | Inicio | Finalización | Hito del proyecto completado |
| Iteración 1 | Inicio | 03/10/2017 | 06/11/2017 | Documentación de requisitos inicial |
| Iteración 2 | Elaboración | 07/11/2017 | 26/12/2017 | Modelo de análisis |
| Iteración 3 | Elaboración | 27/12/2017 | 31/01/2018 | Modelo de diseño |
| Iteración 4 | Construcción | 01/02/2018 | 16/03/2018 | Primera versión preliminar de la aplicación |
| Iteración 5 | Construcción | 19/03/2018 | 14/05/2018 | Versión preliminar de la aplicación con toda la funcionalidad |
| Iteración 6 | Transición | 15/05/2018 | 15/05/2018 | Aplicación terminada, validada y entregada junto con su documentación |

Tabla 1 - iteraciones del proyecto

* **Primera iteración.**

**Fecha de inicio:** 03/10/2017

**Fecha de finalización:** 06/11/2017

Esta iteración se encuentra dentro de la fase de Inicio, comenzó con la primera reunión con los tutores de este proyecto, tras la asignación de este. Como corresponde a esta fase del Proceso Unificado, en primer lugar, se identificaron los objetivos principales como por ejemplo el mostrado en la Tabla 2 y se comenzó a elaborar la primera documentación de requisitos del sistema.

Para llevar a cabo esta tarea se utilizó en programa REM y la metodología de Durán y Bernárdez. Se realizaron trabajos también en lo relativo a la planificación temporal creando una planificación inicial.

|  |  |
| --- | --- |
| OBJ-0001 | Objetivo Aplicación Móvil |
| Versión | 1.0 (03/10/2017) |
| Autores | * [Alejandro Hernández González](file:///G:\\Mi%20unidad\\UNIVERSIDAD\\TFG\\Requisitos%20AppPhyFire_REM_Default_Spanish_CRS.html" \l "STK-0002) |
| Fuentes | * [David Álvarez León](file:///G:\Mi%20unidad\UNIVERSIDAD\TFG\Requisitos%20AppPhyFire_REM_Default_Spanish_CRS.html#STK-0004) * [M.ª Isabel Asensio Sevilla](file:///G:\Mi%20unidad\UNIVERSIDAD\TFG\Requisitos%20AppPhyFire_REM_Default_Spanish_CRS.html#STK-0005) * [Sara Rodríguez González](file:///G:\Mi%20unidad\UNIVERSIDAD\TFG\Requisitos%20AppPhyFire_REM_Default_Spanish_CRS.html#STK-0003) |
| Descripción | El sistema deberá consultar las simulaciones del usuario para una visualización en el dispositivo, de forma que pueda usar el dispositivo desde la propia zona de actuación. |
| Subobjetivos | Ninguno |
| Importancia | vital |
| Urgencia | inmediatamente |
| Estado | Pendiente de validación |
| Estabilidad | media |
| Comentarios | Objetivo principal de la aplicación |

Tabla 2 - objetivo principal de la aplicación

|  |  |  |
| --- | --- | --- |
| UC-0004 | Mostrar simulación | |
| Versión | 1.0 (04/11/2018) | |
| Autores | * [Alejandro Hernández González](file:///G:\\Mi%20unidad\\UNIVERSIDAD\\TFG\\Requisitos%20AppPhyFire_REM_Default_Spanish_CRS.html" \l "STK-0002) | |
| Fuentes | * [David Álvarez León](file:///G:\Mi%20unidad\UNIVERSIDAD\TFG\Requisitos%20AppPhyFire_REM_Default_Spanish_CRS.html#STK-0004) | |
| Dependencias |  [[UC-0003] Listar las simulaciones realizadas por el usuario](file:///G:\Mi%20unidad\UNIVERSIDAD\TFG\Requisitos%20AppPhyFire_REM_Default_Spanish_CRS.html#UC-0003)   [[OBJ-0001] Objetivo Aplicación Móvil](file:///G:\Mi%20unidad\UNIVERSIDAD\TFG\Requisitos%20AppPhyFire_REM_Default_Spanish_CRS.html#OBJ-0001) | |
| Descripción | El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario seleccione una de las simulaciones que tiene listadas. | |
| Precondición | El usuario ha iniciado sesión y tiene simulaciones listadas | |
| Secuencia normal | **Paso** | **Acción** |
| 1 | Se realiza el caso de uso [Listar las simulaciones realizadas por el usuario (UC-0003)](file:///G:\\Mi%20unidad\\UNIVERSIDAD\\TFG\\%20%20%20%20%20%20%20%20%20%20%20%20" \l "UC-0003) |
| 2 | El actor [Usuario (ACT-0001)](file:///G:\\Mi%20unidad\\UNIVERSIDAD\\TFG\\Requisitos%20AppPhyFire_REM_Default_Spanish_CRS.html" \l "ACT-0001" \o "Este actor representa al usuario registrado en la aplicación y que dispone de simulaciones que desea visualizar) selecciona una de las simulaciones disponibles |
| 3 | El sistema muestra la primera capa con los elementos esenciales de la simulación |
| Postcondición | Se muestran la primera capa de paso de tiempo de la simulación | |
| Rendimiento | **Paso** | **Tiempo máximo** |
| 3 | 1 minuto(s) |
| Frecuencia esperada | 1 veces por día(s) | |
| Importancia | vital | |
| Urgencia | hay presión | |
| Estado | pendiente de validación | |
| Estabilidad | media | |
| Comentarios | Ninguno | |

Tabla 3 - UC-0003: mostrar simulacion

La documentación de requisitos al completo se encuentra en el CD adjunto, descrita en el Anexo II de esta documentación. Además, se incluye el proyecto con el que ésta fue creada, para abrir con la herramienta REM por si se desea aclarar algún detalle o realizar modificaciones en el caso de que otro desarrollador deseara continuar con este proyecto.

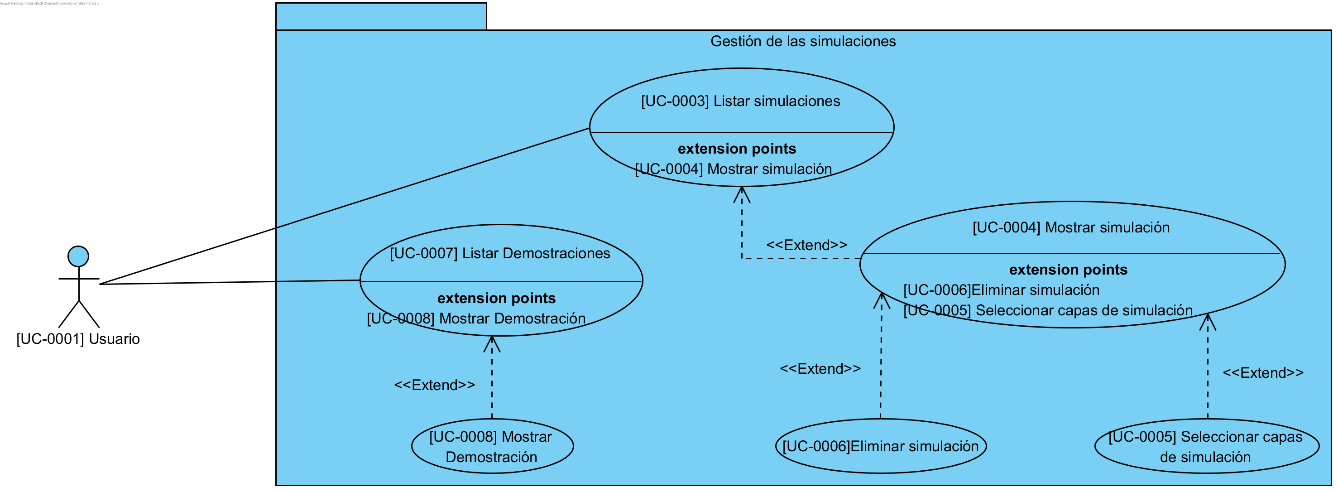
**Hito logrado:** Documentación inicial de requisitos del sistema.

* **Segunda iteración.**

**Fecha de inicio:** 07/11/2017

**Fecha de finalización:** 06/11/2017

El proyecto se encuentra en la fase de Elaboración de Proceso Unificado. En este punto, con una aproximación de los requisitos ya realizada, se tienen que refinar para tener una base sólida para el desarrollo. Con este fin, se produjo una reunión con los tutores en la que, además, se concretó la participación de este proyecto en la convocatoria del programa Transferencia de Conocimiento Universidad Empresa (TCUE-2017/2018), convocatoria de la que este proyecto resultaría finalmente seleccionado y premiado.

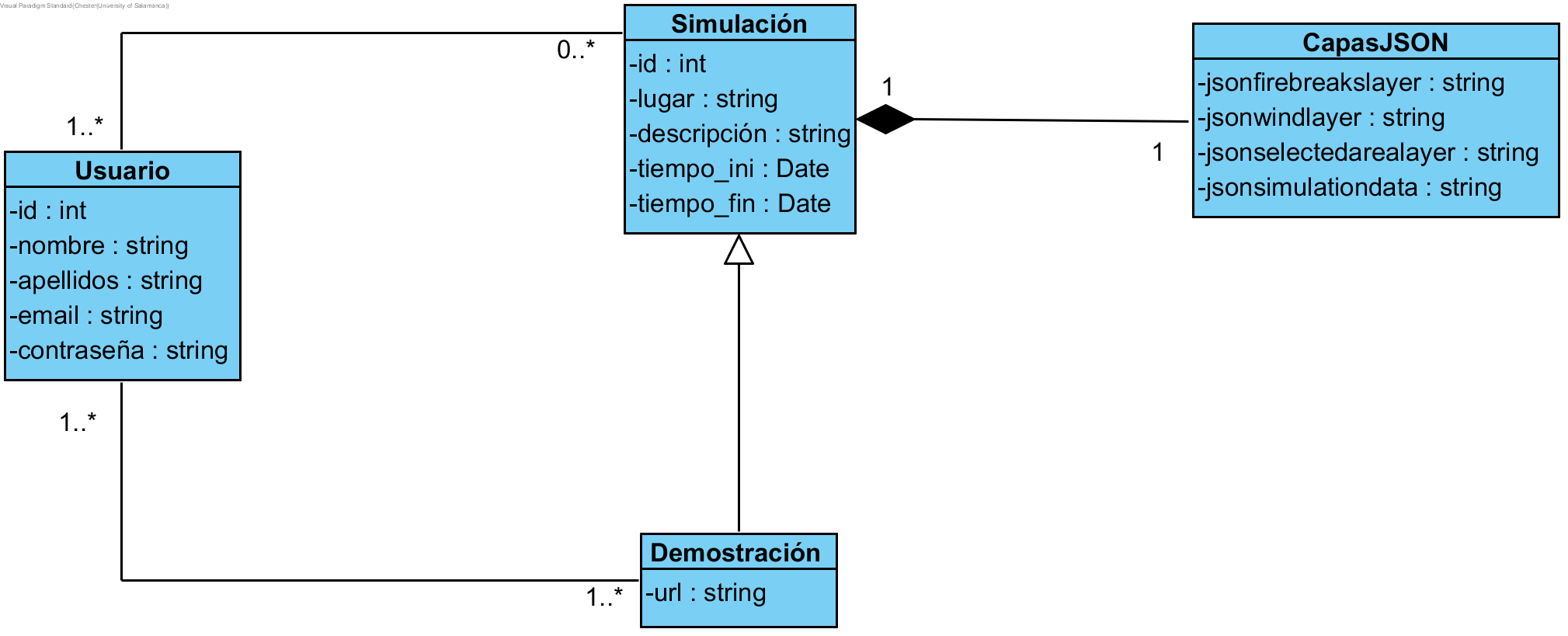
Se refinaron los requisitos existentes y se añadieron algunos nuevos como “*UC-0008: Listar demostraciones” y “UC-0009: Mostrar demostraciones”*. Podemos ver el módulo al que pertenece en la **Figura 20**.

*Figura 20 - módulo de gestion de simulaciones*

Una vez realizado el catálogo de requisitos el siguiente paso en el proceso de desarrollo lo constituye la elaboración de un modelo conceptual del dominio, o modelo de análisis, en el

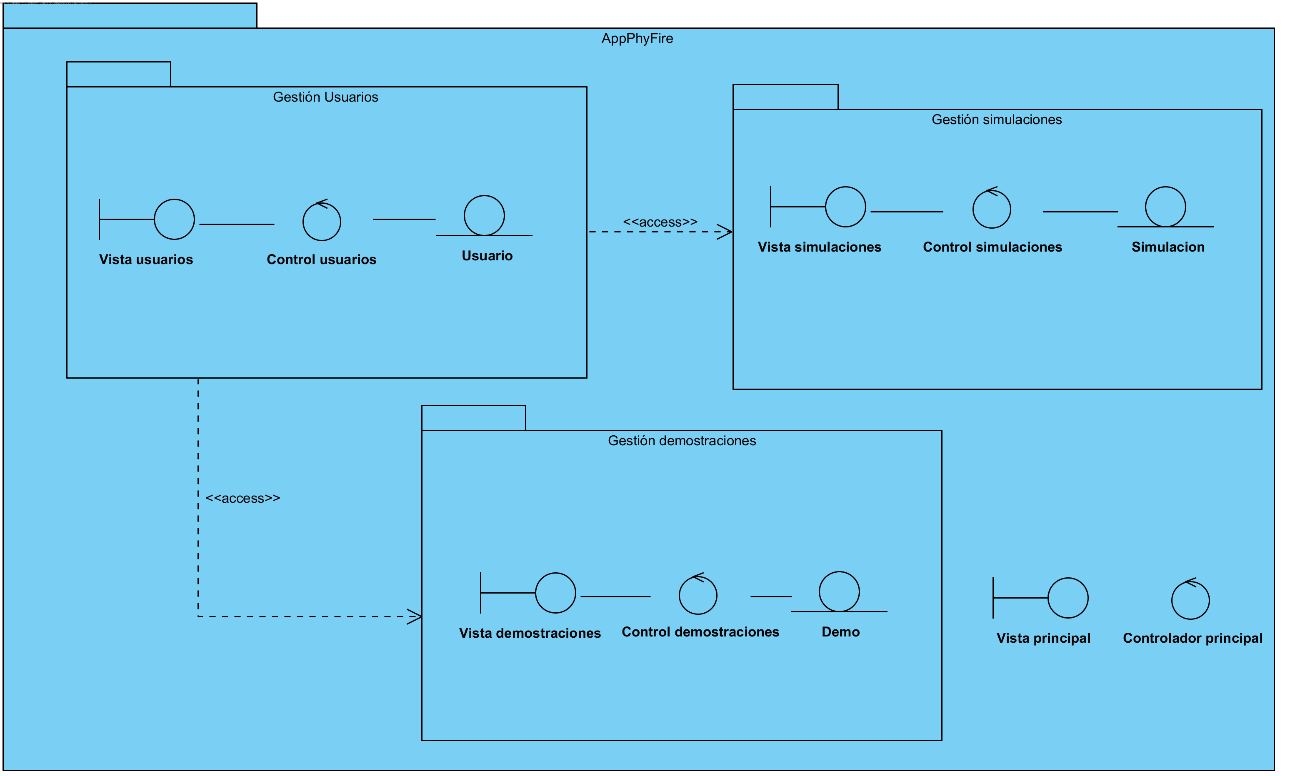
cual se mostrarán las principales clases del dominio de la aplicación descubiertas y las relaciones entre las mismas, lo que posibilitará una primera aproximación a la semántica del

problema. En la **Figura 21** vemos el modelo de dominio del sistema, el cual, con las modificaciones convenientes, será el usado tanto por la aplicación móvil como en la base de datos del servidor.



*Figura 21 - modelo de dominio*

Una vez identificados los paquetes de análisis, podemos situar las diferentes clases de análisis con su vista, controlador y clase entidad/modelo correspondientes sistema que podemos ver en la **Figura 22.**

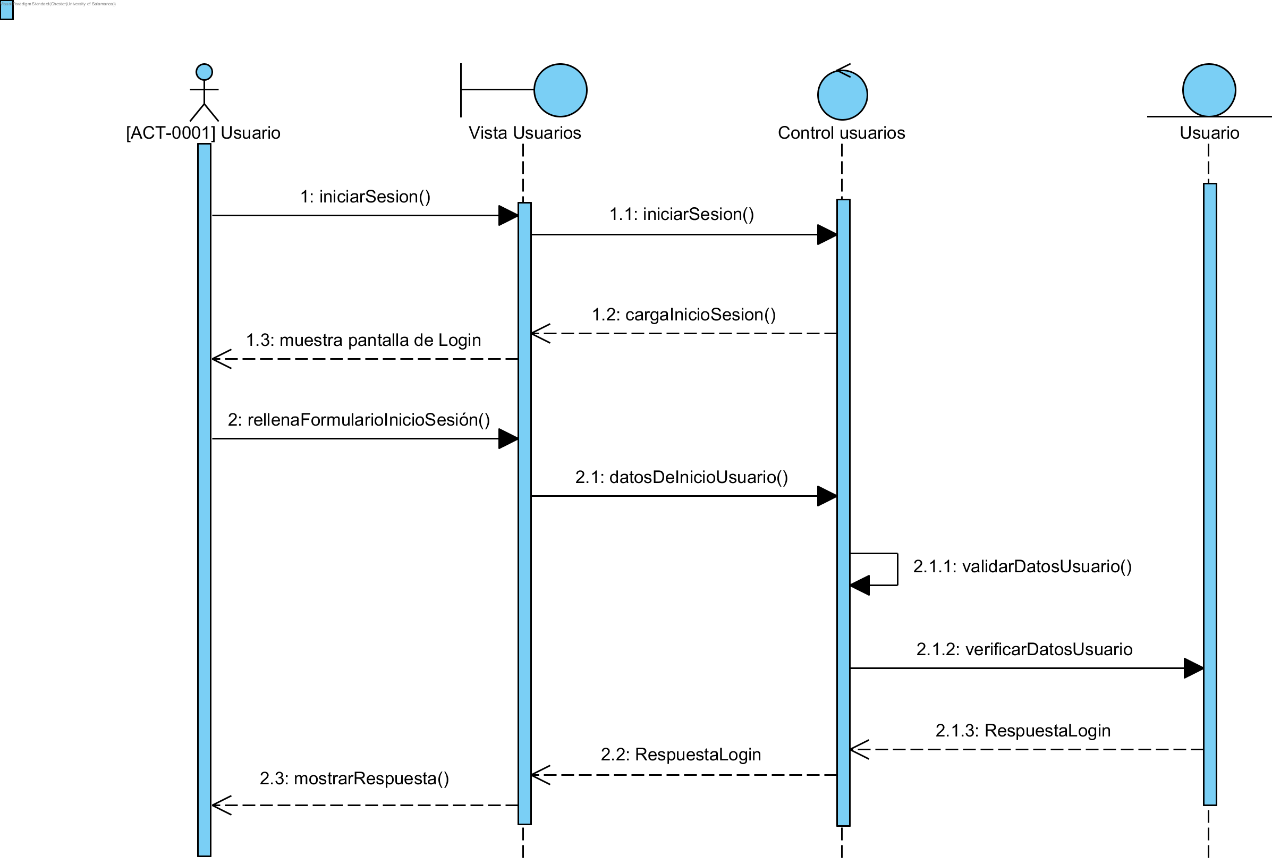


*Figura 22 - propuesta de arquitectura*

La conexión entre los paquetes se denota por <<access>> debido a que esta se utiliza para las importaciones privadas de paquetes. La especificación UML 2.4.1 dice "Una importación de paquete se muestra usando una flecha discontinua con una punta de flecha abierta desde el espacio de nombre de importación al paquete importado. Se muestra una palabra clave cerca de la flecha discontinua para identificar a qué tipo de paquete se destina la importación. las palabras clave son «import» para una importación de paquete público y «access» para una importación de paquete privada [25].

En último lugar, se crean los diagramas de secuencia que representan los casos de uso y derivan de las clases de análisis. Estos diagramas no están basados en un diseño definitivo si no de la propuesta de arquitectura anterior, sin embargo, son capaces de mostrar bastante información sobre el funcionamiento general de sistema para el posterior modelado de su comportamiento en las siguientes fases del proceso.

En la ***Figura 23*** podemos ver el diagrama de secuencia del caso de uso *UC-0001* correspondiente al inicio de sesión del usuario y en la ***Figura 24*** el propio del caso de uso *UC-0004* listar simulaciones del usuario, la cual incluye controladores de Usuario y de Simulación debido a que las simulaciones se muestran al producirse el inicio de sesión del usuario.



*Figura 23 - [UC-0001] inicio de sesión*



*Figura 24 - [UC-0004]: listar simulaciones del usuario*

El resto de los diagramas realizados se pueden consultar con detalle en el archivo de Visual Paradigm (\*.vpp) adjunto en el CD del proyecto.

**Hito logrado:** Refinamiento de requisitos y modelo de análisis.

* **Tercera iteración.**

**Fecha de inicio:** 27/12/2017

**Fecha de finalización:** 31/01/2018

El objetivo de esta iteración, la cual se aún se enmarca en la fase de elaboración como la fase anterior, es la definición del modelo de Dominio del sistema, teniendo en cuenta las particularidades de cada sistema (o subsistemas) a desarrollar.

En este punto del proceso, es necesario modelar la arquitectura del sistema. Se ha optado una arquitectura Modelo-Vista-Controlador debido a que se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo, además de adaptarse a la perfección a las características de este proyecto.

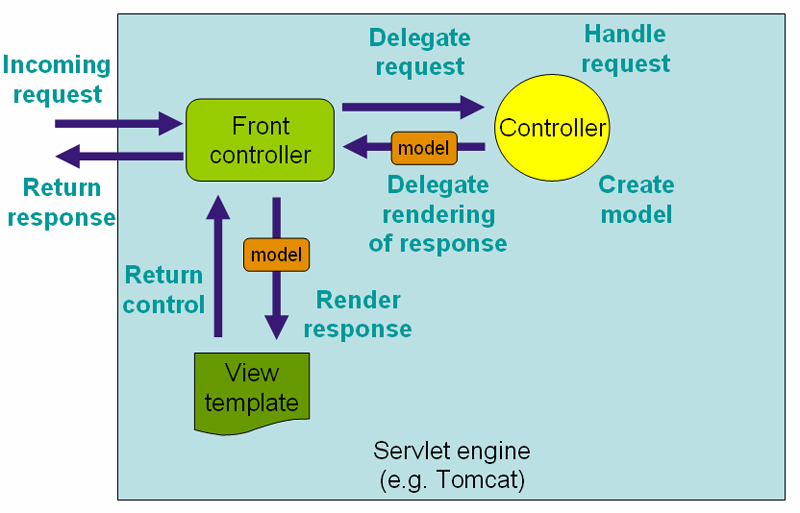
***Modelo Vista Controlador*** es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos (***Figura 25***).

*Figura 25 - patron mvc*

* El **Modelo** que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.
* La **Vista**, o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos interacción con éste.
* El **Controlador**, que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno [26].

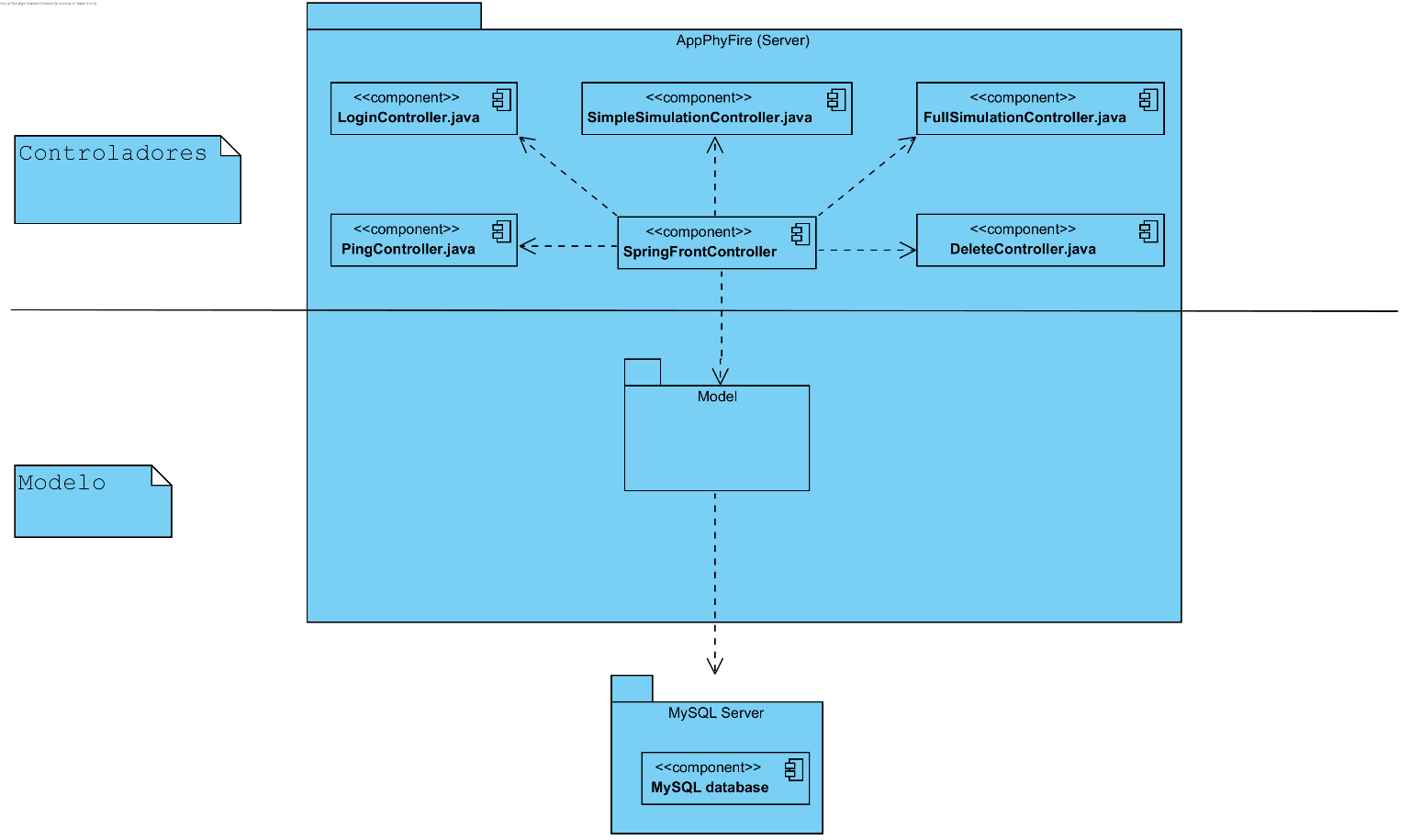
Como se ha explicado anteriormente, para el desarrollo del servidor se ha utilizado el Framework de Spring, en concreto Spring MVC para Java y Spring Boot para una configuración muchísimo más sencilla del mismo.

La arquitectura del Framework de Spring está basada en el patrón MVC y se ha seguido para la creación de nuestra API REST que servirá de servidor para nuestra aplicación Android. En la **Figura 26** se aprecia como gestiona el procesamiento de una petición Spring con este patrón [27].



*Figura 26 - El flujo de procesamiento de solicitudes en Spring Web MVC (alto nivel)*

Con la ayuda del Framework, no necesitamos gestionar las peticiones ya que este se encarga de delegarlas a nuestros controladores específicos. Este aspecto se ve reflejado en el diseño arquitectónico de nuestro servidor (***Figura 27***).

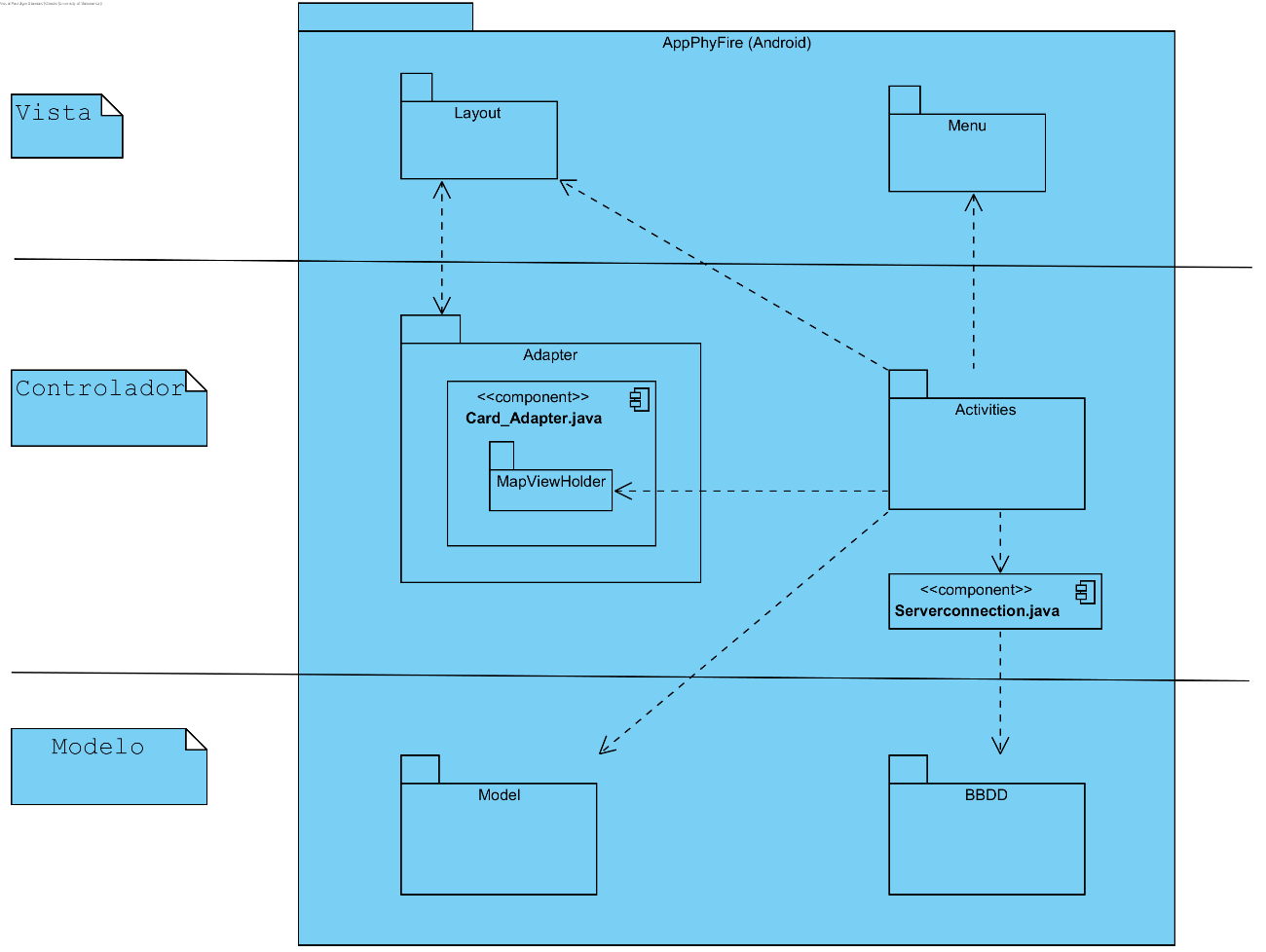


*Figura 27 - arquitectura del servidor*

En esta parte del sistema podemos ver una arquitectura de dos capas Modelo-Controlador, ya que no existe una vista para la interacción con el usuario ya que resulta innecesaria. Esta arquitectura nos permite desacoplar la *API REST* que ofrece (funcionalidad) de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia (modelo) que está formado por las clases del paquete “*Model*” y la BBDD.

Los controladores gestionan las llamadas a la API y al modelo, incluyendo el “controlador” del Framework que trabaja con las peticiones entrantes y con su respuesta del controlador específico que corresponda.

En cuanto a la parte de la aplicación, se muestra su diseño arquitectónico con las particularidades del desarrollo de una aplicación Android (identificando las clases *Activity*, *Frament*, etc), pero siguiendo el patrón MVC.

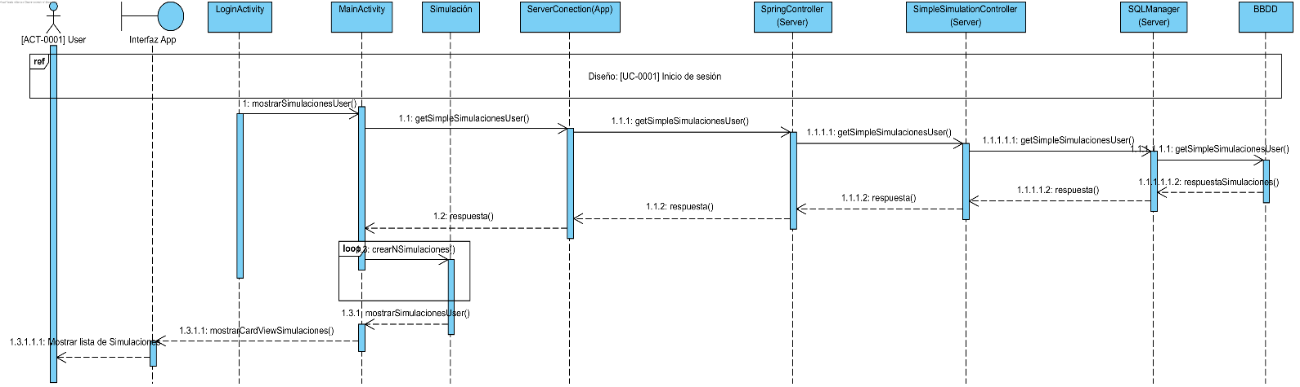


*Figura 28 – diseño arquitectónico de la aplicación móvil*

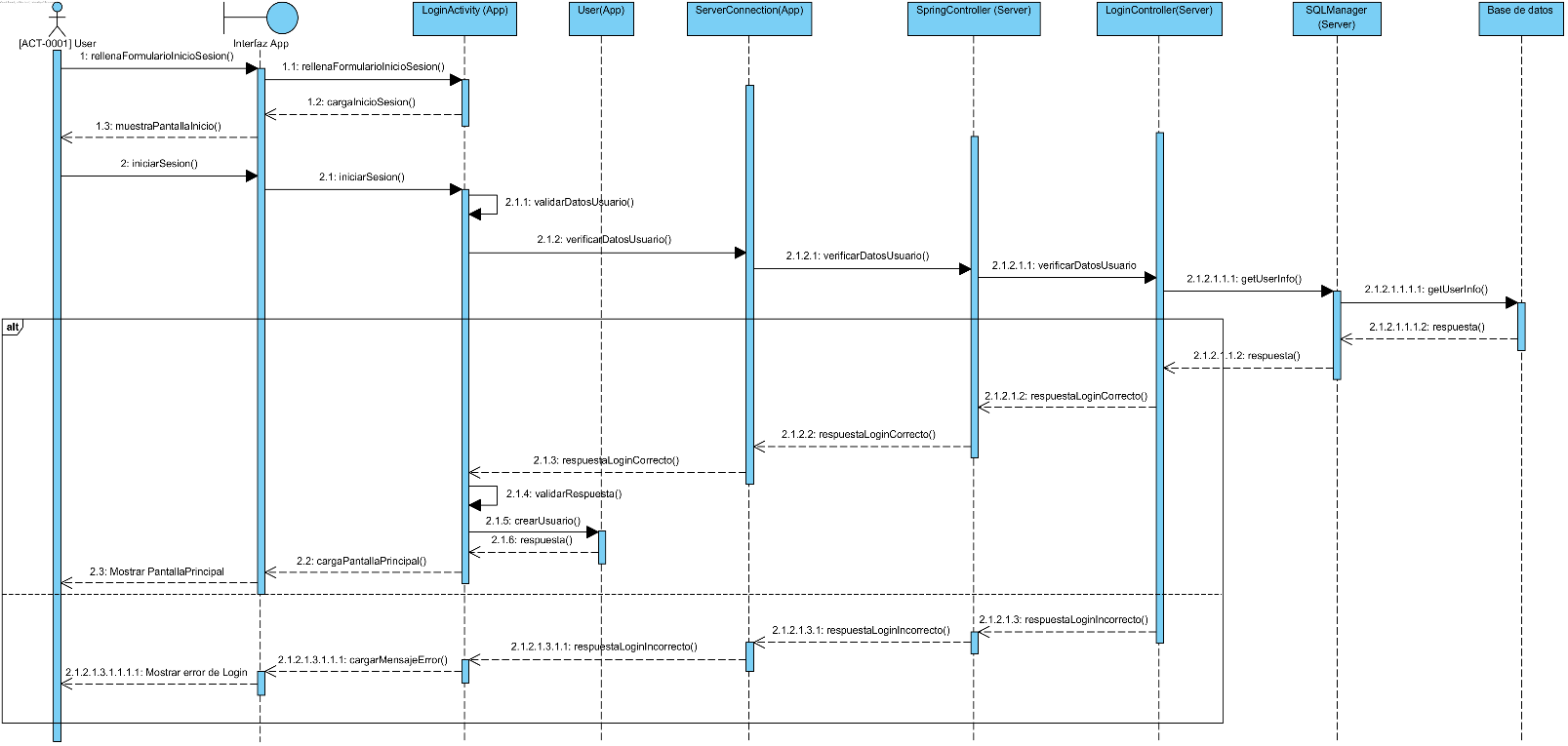
Una vez definida la arquitectura, podemos mostrar las relaciones e intercambio de mensajes de los casos de uso mediante diagramas de secuencia, basándonos en los diagramas de secuencia obtenidos en la fase de análisis y que se denominan realización de los casos de uso.

A continuación, se pueden ver algunos ejemplos, como la realización de caso de uso *[UC-0003] Listar simulaciones de usuario* en la

**Figura *29*** y el caso de uso en [*UC-0001] Inicio de sesión en la* **Figura 30.**

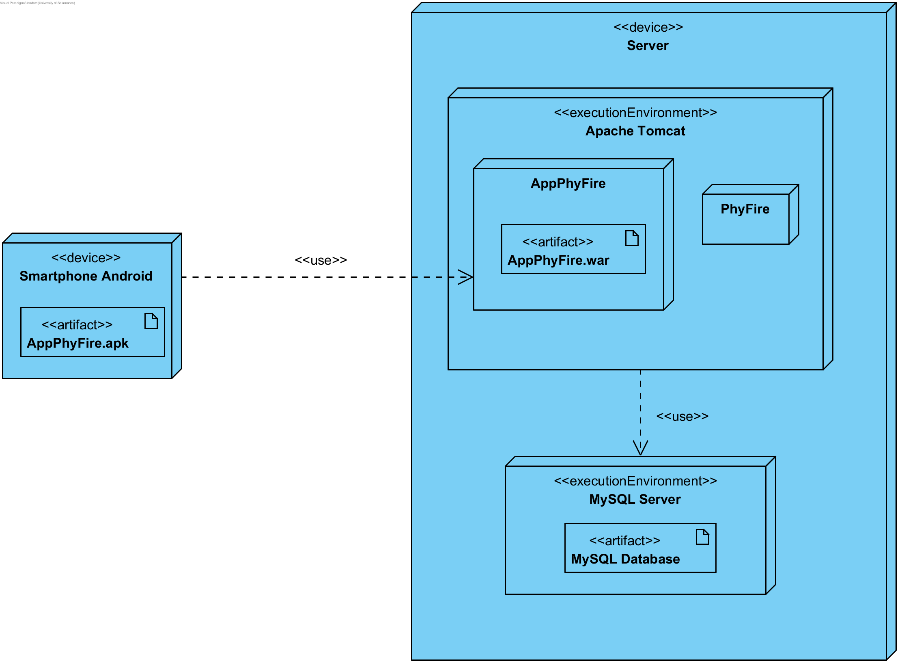


*Figura 29 - realizacion de caso de uso [UC-0003] listar simulaciones de usuario*



*Figura 30 - realizacion de caso de uso [UC-0001] Iniciar sesión*

Por último, en esta fase debemos mostrar muestra los nodos más relevantes de la plataforma y la parte de hardware o física necesaria para la puesta en marcha de la aplicación; así como todas las relaciones existentes entre las mismas, para ello, se crea un diagrama de despliegue que podemos ver en la **Figura 31**.



*Figura 31 - diagrama de despliegue*

El resto de los diagramas correspondientes a esta fase se pueden revisar en profundidad en el Anexo III de este proyecto o en el archivo en el archivo de *Visual Paradigm (\*.vpp)* adjunto en el CD del proyecto.

**Hito logrado:** Refinamiento modelo de análisis y modelo de diseño.

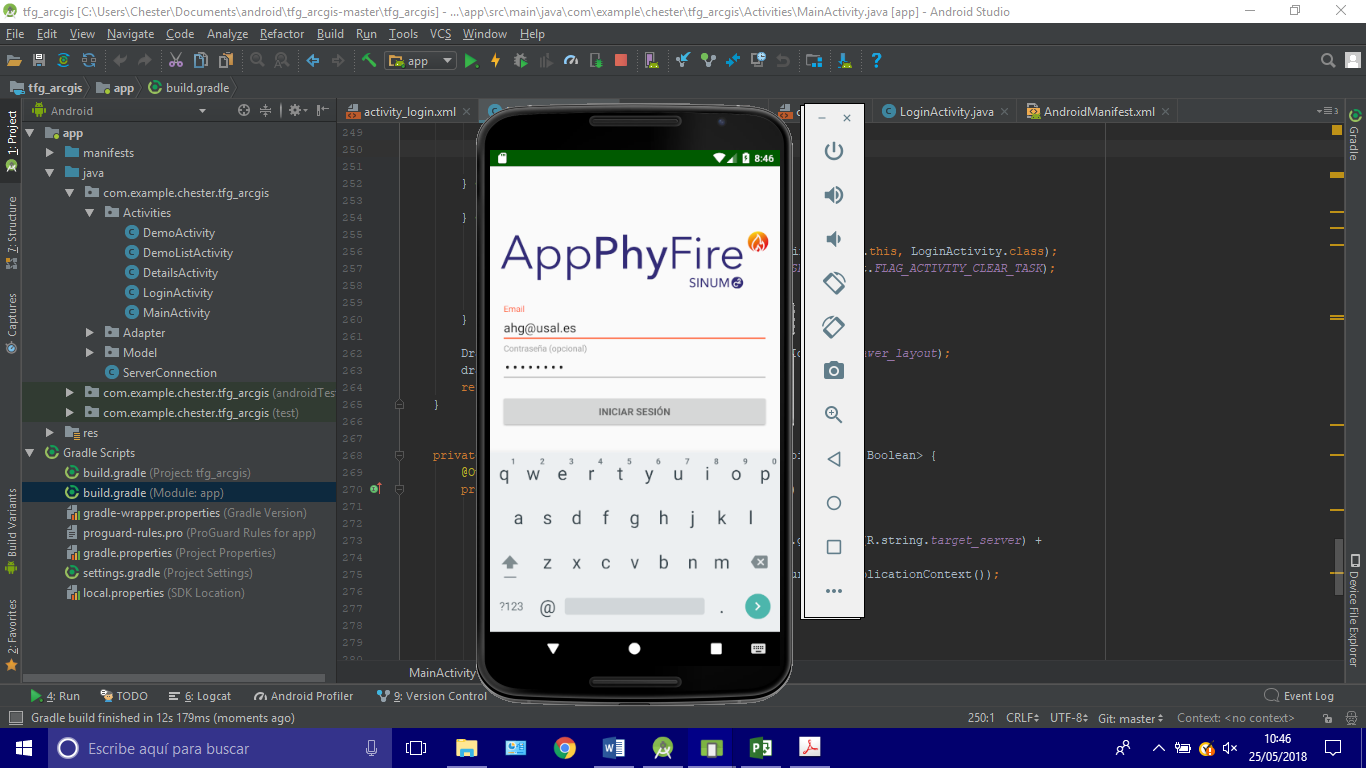
* **Cuarta iteración.**

**Fecha de inicio: 01/02/2018**

**Fecha de finalización: 16/03/2018**

Una vez finalizada la fase de elaboración, entramos en la fase de construcción, en la que se comienza con la implementación del sistema. Esta iteración comienza con una reunión con los tutores cuya finalidad es validar el diseño del sistema y comenzar así el desarrollo con la base más sólida posible.

En primer lugar, hay que establecer el entorno de trabajo. Para ello, escogemos las herramientas más adecuadas para el desarrollo en cada caso. La herramienta elegida para el desarrollo de la aplicación es Android Studio, debido a ser el IDE de Android por excelencia y con más facilidades para el desarrollador (emulador, sistema de control de versiones incorporado, etc.) y para la parte del servidor, se ha escogido *Eclipse*, que permite integración total con *Spring Boot y Gradle*.

Los detalles de cómo configurar el entorno apropiadamente los podemos encontrar en el Anexo IV que se entrega con este proyecto.

*Figura 32 - desarrollo en Android studio con el emulador*

La primera tarea que se realiza es crear las vistas de la aplicación y con el fin de obtener una interacción con el usuario lo más óptima posible, se realizaron algunas pruebas con distintos aspectos de diseño de la interfaz (logotipos, colores, posicionamiento de elementos de interacción, etc.) hasta dar lugar a un primer diseño de la interfaz de usuario.

En la **Figura 33** se ven diversos diseños del logo de aplicación creados en esta fase.

|  |  |
| --- | --- |
|  |  |
|  |  |

*Figura 33 - Diferentes logos de la aplicación*

A continuación, se implementaron los casos de uso correspondientes al módulo de gestión de usuarios *[UC-0001] Inicio de sesión* y [*UC-0002] Cerrar sesión*. Para llevar a cabo estos casos de uso, es necesaria la comunicación con el servidor y de éste con la base de datos. Por lo tanto, debido a la complejidad de este desarrollo, el tiempo dedicado a la implementación de esta parte ha sido notablemente elevado (20 días como podemos ver en la planificación temporal). Se implementa una *API REST* sobre https y con cifrado adicional al protocolo SSL/TTL en BASE64 para asegurarnos de que ningún dato sensible del usuario que se envíe por la red se haga en texto plano ya que, de otra forma, la información que se transmite entre el cliente y el servidor podría verse comprometida.

**Hito logrado:** Primera versión preliminar de la aplicación con la comunicación con el servidor establecida.

* **Quinta iteración.**

**Fecha de inicio:** 19/03/2018

**Fecha de finalización:** 14/05/2018

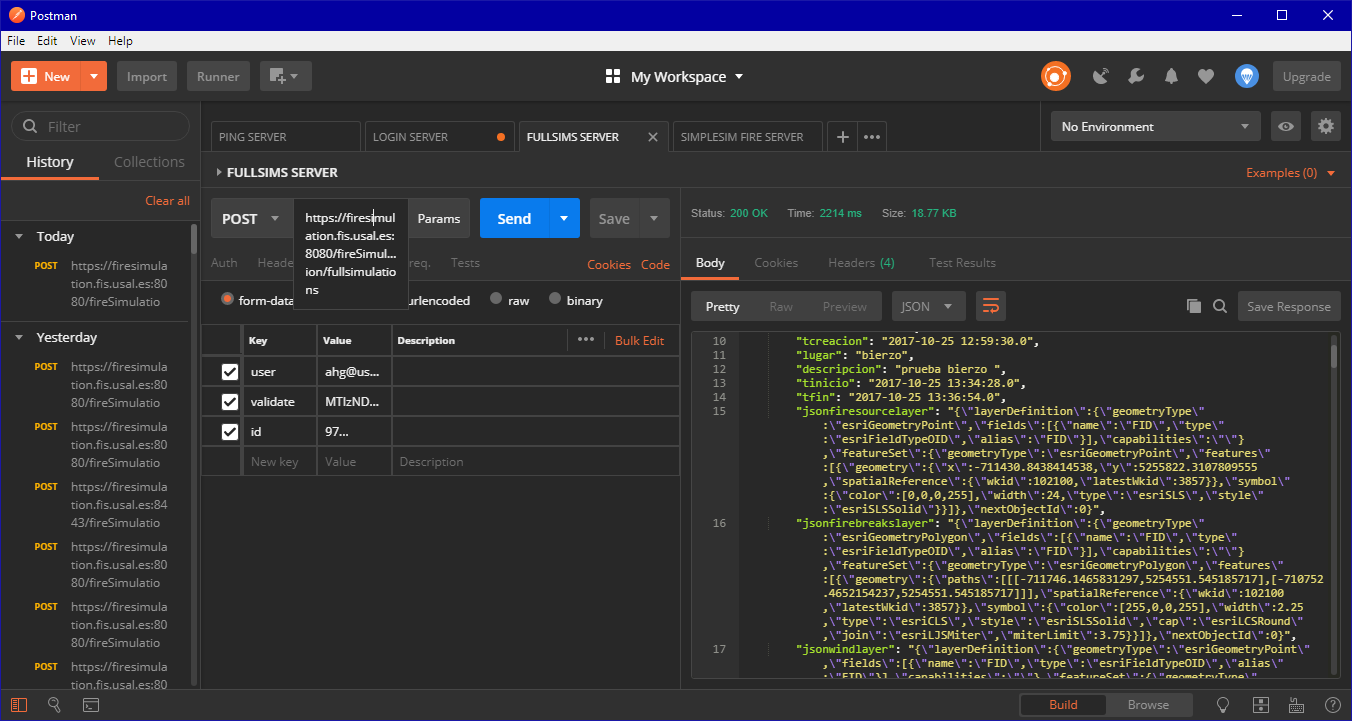
En esta iteración, perteneciente a la fase de construcción, se continúa con la implementación de la funcionalidad de la aplicación en concreto, a los casos de uso pertenecientes al módulo de gestión de las Simulaciones (*[UC-0003] Listar Simulaciones del usuario, [UC-0004] Mostrar simulación, [UC-0005] Seleccionar capas de simulación, [UC-0006] Eliminar simulación, [UC-0007] Listar Demostraciones y [UC-0008] Mostrar demostración*).

Cabe destacar la complejidad técnica de la representación del resultado de una simulación. Cada simulación genera un conjunto de capas que indican la superficie quemada, así como el frente activo de fuego.

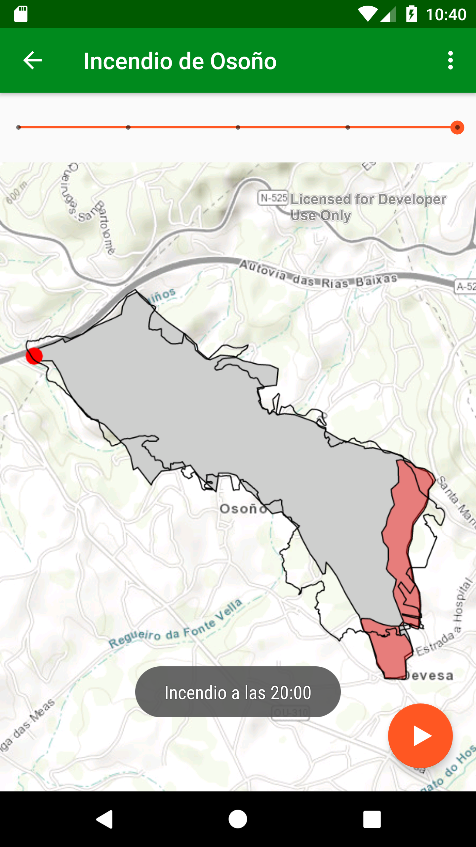
Phyfire está configurado para la creación de una capa de simulación cada media hora. Como resultado, en una simulación de 2 horas, se crearán 8 capas (4 para frentes activos y 4 para superficies quemadas) y se almacenan en un único JSON en la base de datos.

Desde la aplicación (cliente Android) se debe leer este resultado y dibujar cada una de las capas en un mapa añadiendo además la funcionalidad de mostrar y ocultar cada una de las capas a petición del usuario en un tiempo y tamaño razonable.

En la ***Figura 34*** se puede ver una petición de una simulación concreta a la API creada y su correspondiente respuesta del servidor (con la herramienta *Postman*), la cual es equivalente a la petición realizada por el cliente Android.

Se puede apreciar cómo está almacenada una parte de la información de las capas y la complejidad de los datos con los que se está tratando y que hay que tratar de representar en el dispositivo.

*Figura 34 - Petición a la API y respuesta del servidor*

En este punto se realizaron también las demostraciones de funcionamiento de los modelos físicos, poniendo ejemplos del comportamiento incendio forestales reales y de la simulación en ese mismo punto, comprobando la precisión de los modelos y afianzando la confianza del usuario en el sistema. Como se puede ver en la **Figura 35**, el área quemada (gris) y la posición del frente (rojo) coinciden con alto grado de exactitud con el incendio real (línea negra) en ese momento del incendio. Se muestran varios pasos de tiempo (30 min.) que podemos recorrer con el “slider” superior o con una animación que se produce al apretar el botón de abajo a la derecha.

*Figura 35 - Demostración del modelo*

Por último, se realizaron pruebas para detectar posibles errores de implementación surgidos durante la anterior fase y se corrigieron dichos fallos, además de pruebas de estabilidad del sistema en el servidor local y de validación de la funcionalidad.

**Hito logrado:** Versión preliminar de la aplicación con toda la funcionalidad.

* **Sexta iteración.**

**Fecha de inicio:** 15/05/2018

**Fecha de finalización:** 07/06/2018

Nos encontramos ya en la fase final del desarrollo del sistema, la transición, que ha durado una iteración en la que se realizan actividades enfocadas a la entrega del producto, como la corrección de los errores detectados en la anterior fase de pruebas o en nuestro caso, debido a que este producto forma parte del sistema PhyFire, pruebas de integración con el mismo.

En este punto, toda la documentación generada (también el código) debe cumplir con los estándares de calidad para crear un producto final entregable. Con este fin se revisan tanto el presente documento, como los anexos de éste y se corrigen todos los fallos detectados.

Por último, se entrega a los tutores para su validación y corrección, en el que una vez aprobado, se marca la finalización del proceso de desarrollo del proyecto.

**Hito logrado:** Aplicación terminada, validada y entregada junto con su documentación.

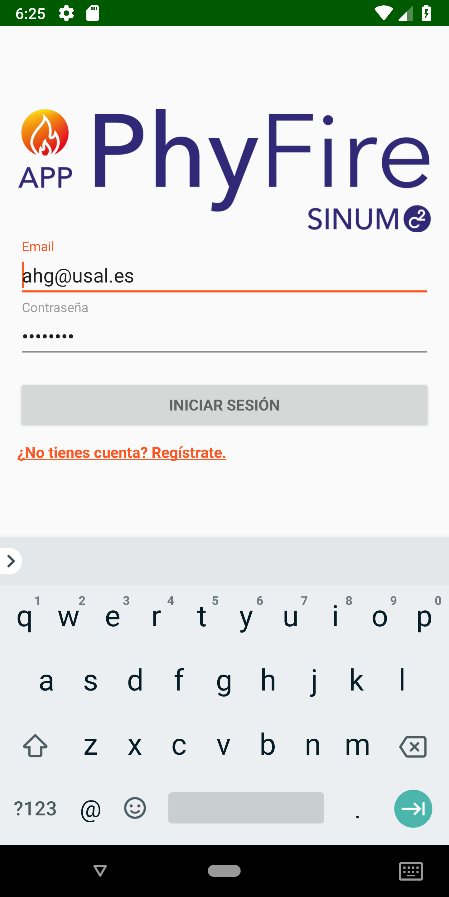
# Producto final

Con el fin analizar las conclusiones a las que se ha llegado con el desarrollo de este trabajo, se resumen las necesidades que se resuelven, pero también el cómo, indicando los pasos a realizar por el usuario.

# Acceso a la aplicación

Tras iniciar la aplicación por primera vez, se muestra la pantalla de inicio de sesión. Desde esta misma pantalla se puede acceder también al formulario de registro como se ve en la **Figura 36.**

Para ello solo es necesario pulsar sobre el texto “¿No tienes cuenta? Regístrate”. Que nos redirige al sitio web del sistema PhyFire que nos permite el registro (**Figura 37**).



*Figura 36 - Inicio de sesión*

*Figura 37 - Registro*

# Registro

Para realizar el registro utilizando una dirección de correo electrónico tan solo se debe acceder a la pantalla de registro, completar el formulario y pulsar el botón “Registrarse”.

Si no se produce ningún fallo (por ejemplo, no hay ningún usuario registrado con el mismo email), se completa el registro. En caso contrario, el usuario es notificado del error.

# Inicio de sesión

Para iniciar sesión en la aplicación tan solo se necesita introducir el email y la contraseña elegidas durante la fase de registro y pulsar el botón “Iniciar sesión”. A continuación, el sistema comprobará los datos introducidos y, si son correctos, se cargará la pantalla principal.

Si, por el contrario, el email introducido no se encuentra registrado en el sistema o la contraseña no se corresponde la de dicho email, se mostrará un aviso al usuario y se cancelará el inicio de sesión.

Imagen que contiene captura de pantalla

Descripción generada con confianza muy altaImagen que contiene captura de pantalla

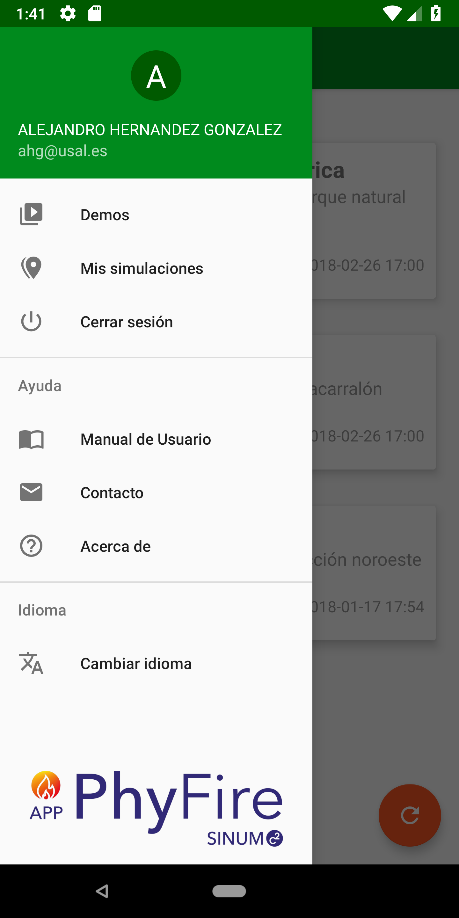
Descripción generada con confianza muy alta

*Figura 38 - Login incorrecto (a)*

*Figura 39 - Login incorrecto (b)*

Si no se ha producido ningún error durante el inicio, se mostrará la pantalla principal de la aplicación con el menú principal y la lista de simulaciones realizadas en el sistema PhyFire por el usuario.

# Menú principal

Una vez identificado en la aplicación, el usuario puede ver el menú principal

que se corresponde con toda la funcionalidad de la aplicación. Pero antes de definir en qué consiste dicha interfaz, es necesario centrarse en su funcionamiento

Se accede a través del botón del menú de la parte superior de la pantalla  o deslizando el dedo hacia la derecha desde el borde izquierdo de la pantalla.

Como se puede observar en la **Figura 40**, el menú principal se despliega en la parte izquierda y

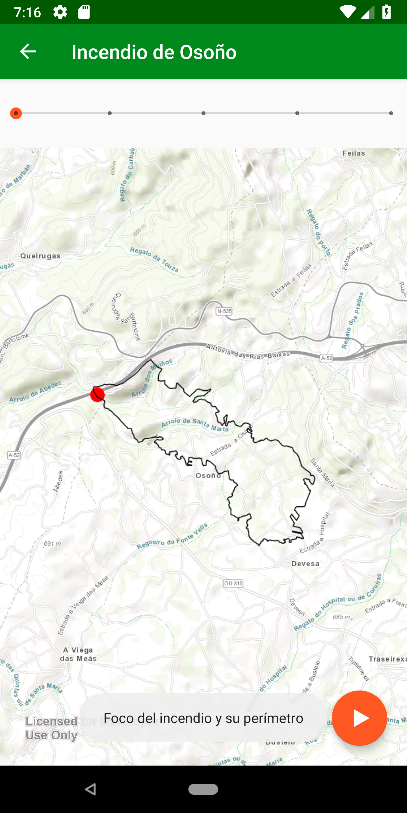
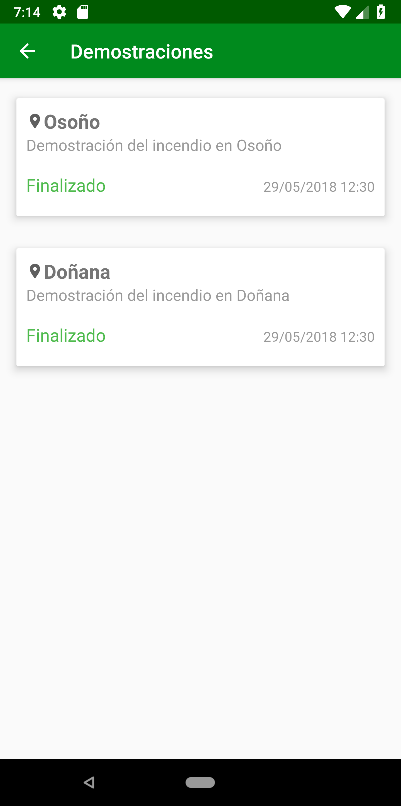
permite el acceso a las cinco pantallas más relevantes del sistema:

* ***Demos***
* ***Mis Simulaciones***
* ***Sección de ayuda***
* ***Cerrar Sesión***
* ***Manual de Usuario***
* ***Contacto***
* ***Acerca de***
* ***Cambiar idioma***

En las siguientes secciones de este manual se describe el funcionamiento de dichas pantallas

*Figura 40 - Menú principal de la App*

# Demos

Se trata de ejemplos de funcionamiento del modelo de simulación de incendios forestales PhyFire por medio de demostraciones que comparan un incendio real con uno simulado, comprobando el grado de precisión del modelo.

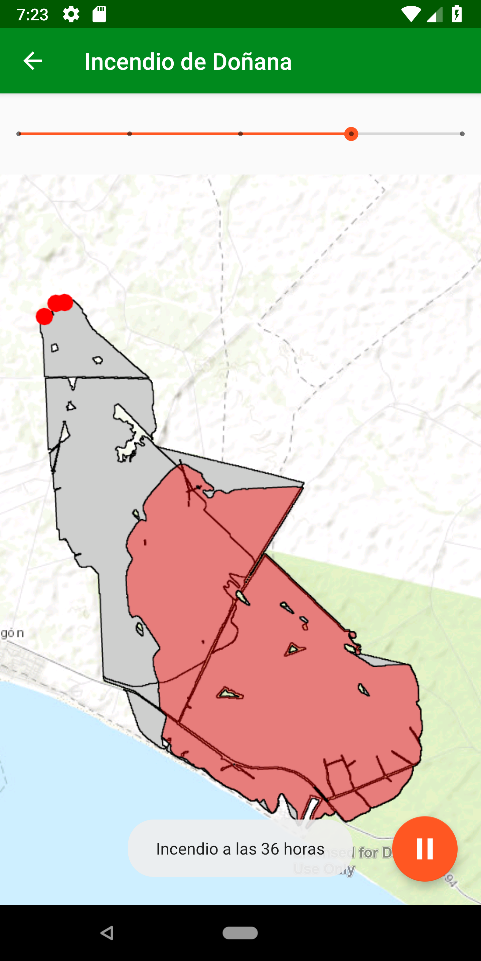
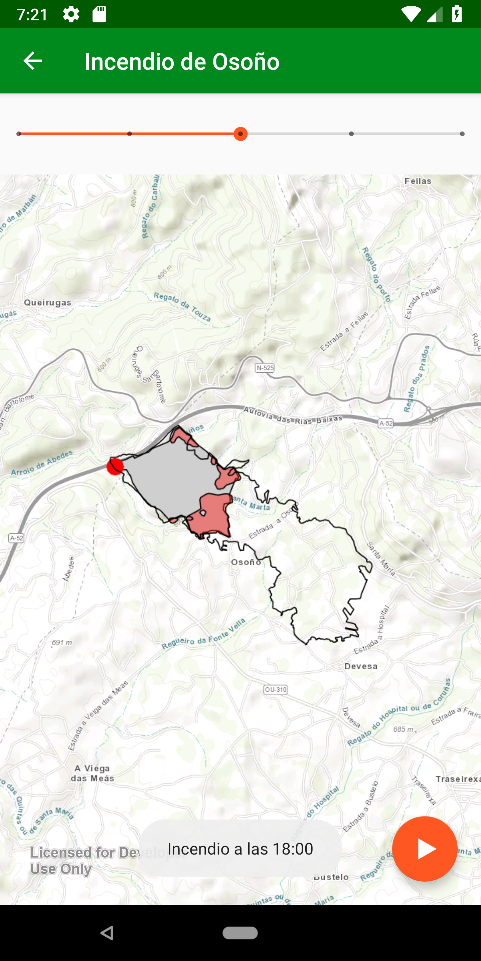
Se accede al pulsar el botón “Demos” del menú principal. A continuación, se muestra una lista con las demos disponibles y seguidamente (**Figura 41**), si se pulsa en una de ellas, nos muestra el contenido de ésta (**Figura 42)**.

*Figura 41 - Lista de demostraciones*

*Figura 42 - Demostración de Osoño*

Una vez se ha accedido a una simulación en particular, tenemos varias opciones para reproducir la demostración, en primer lugar, podemos desplazar el “slider” de la parte superior y ver los distintos pasos de tiempo (**Figura 43**), o pulsar el botón “Play” ubicado en la parte inferior derecha y que se reproduzca automáticamente **Figura 44**.

En todo caso, se muestra un mensaje en la parte inferior mostrando el paso de tiempo que se está visualizando.

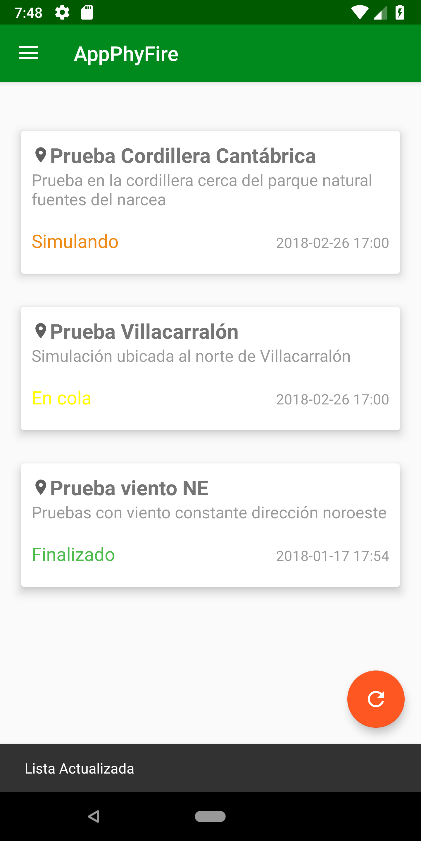
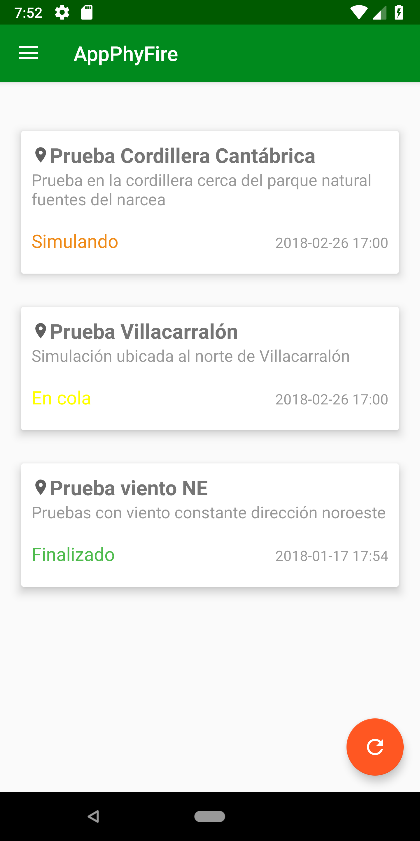


*Figura 43 - Demostración mostrada en manual*

*Figura 44 - Demostración mostrada en automático*

# Simulaciones

Se trata de la funcionalidad principal de la aplicación, muestra las simulaciones que tiene el usuario en el sistema y sus datos (estado, descripción, fecha, etc.) como se aprecia en la **Figura 45**.

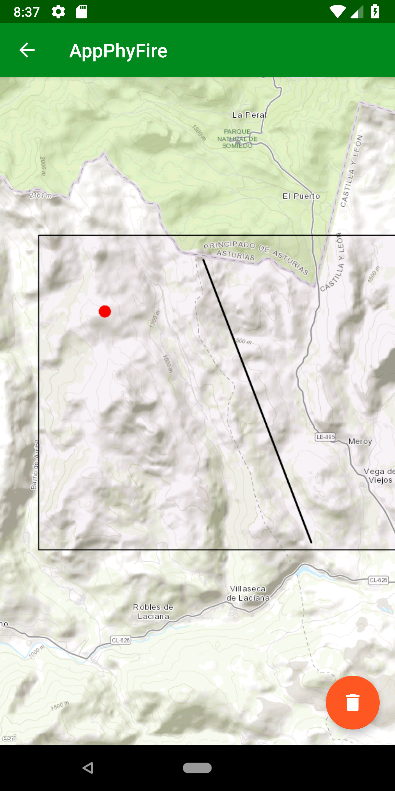
En el caso de que se pulse el botón de la parte inferior derecha, actualizaremos la lista con los cambios si existieran y nos mostrará un mensaje informativo en la parte inferior (**Figura 46**).

*Figura 45 - lista de simulaciones*

*Figura 46 - Lista de simulaciones con mensaje*

# Visualización de la simulación

Muestra al usuario las capas de la simulación, permite producir varias situaciones en función del estado de la simulación.

En el caso de que la simulación no esté completada (se encuentre en cola para ser simulada o en proceso de simulación), se muestran las capas relativas a la información que ha introducido el usuario como los cortafuegos o los focos, se puede ver un ejemplo de este caso en la **Figura 47.**

Si la simulación se ha completado, se podrán visualizar las capas con los resultados, es decir, se muestra un menú en la parte superior que permite seleccionar cualquier paso de tiempo que el usuario haya simulado como se aprecia en laFigura 49**.**

Cada paso de tiempo dispones de dos capas, la zona ya quemada (gris) y la posición del fuego en ese instante (rojo), en laFigura 48se ve cómo se puede seleccionar las distintas capas.

*Figura 47 - Visualización de la simulación*

Imagen que contiene texto, mapa

Descripción generada con confianza muy altaImagen que contiene texto, mapa

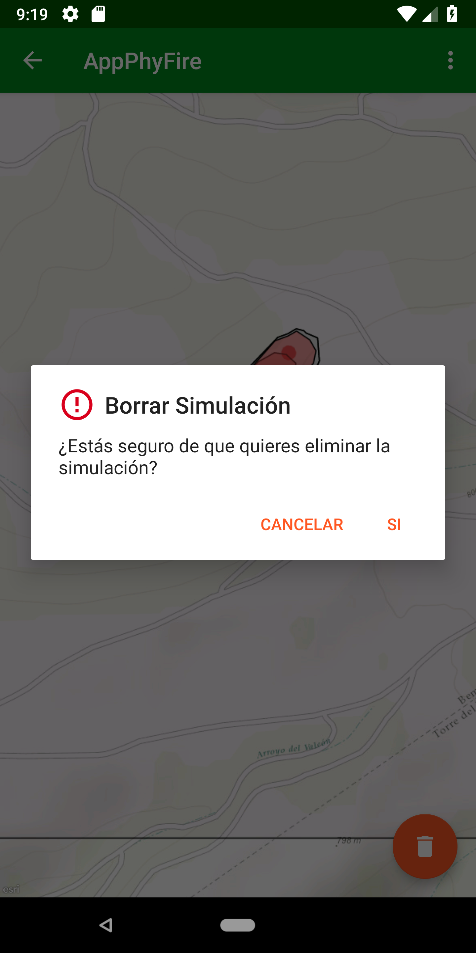
Descripción generada con confianza muy alta

Figura 48 - Menú de selección de capa

Figura 49 - Menú de selección de paso de tiempo

# Eliminación de la simulación

Se trata de un elemento común de gestión. El usuario puede eliminar una de las simulaciones que tiene disponibles si la considera errónea o simplemente ha finalizado el estudio de ésta.

Antes de proceder a la eliminación, se mostrará el mensaje de confirmación mostrado en laFigura 50**,** para asegurarse de que esta acción no se realice de forma accidental.

Si se pulsa “Cancelar”, esta acción quedará sin efecto alguno, en cambio, si se pulsa “Aceptar”, se eliminará la simulación y se mostrará de nuevo la lista de las simulaciones disponibles.

Figura 50 - Eliminar simulación

# Cerrar sesión

Es necesario proporcionar al usuario una forma de cierre ordenado de la aplicación, de forma que pueda cerrar su sesión sin pérdida de información ni errores.

Para ello al pulsar “Cerrar Sesión”, se le mostrará un mensaje en pantalla a modo de confirmación. En caso de pulsar “Cancelar”, esta acción quedará sin efecto, en caso contrario, al pulsar “Aceptar”, se cerrará la sesión y se le devolverá a la pantalla de Login.

En laFigura 51se ve el mensaje de cierre de sesión que se le presenta al usuario.

Figura 51 - Mensaje de cierre de sesión

# Sección de ayuda

La misión principal de esta sección del menú principal es prestar ayuda al usuario en cualquier duda sobre el funcionamiento de la aplicación y consta de varias funciones:

* ***Manual de Usuario.*** Muestra al usuario el manual de usuario (Anexo V que se entrega con este proyecto).
* ***Contacto.*** Ofrece al usuario la posibilidad de contactar por email con los responsables del proyecto.
* ***Acerca de.*** Muestra una ventana emergente con la información de la versión y de los creadores del proyecto.

# Cambio de idioma

Para que la aplicación disponga de mayor difusión, se da la posibilidad al usuario de alternar entre el español y el inglés.

En primer lugar, la aplicación se mostrará en el idioma que tenga configurado en el terminal por defecto, y una vez se ha accedido, se ofrece un menú en el que poder cambiarlo si así desea el usuario, pudiendo volver al estado anterior en cualquier momento.

En la Figura 53se aprecia el menú con de selección de idioma emergente y en laFigura 52el menú principal en inglés que se genera al cambiar de idioma.

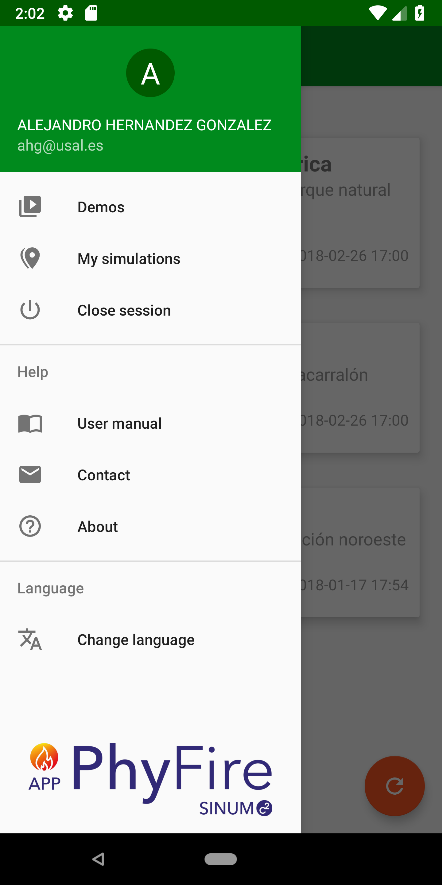
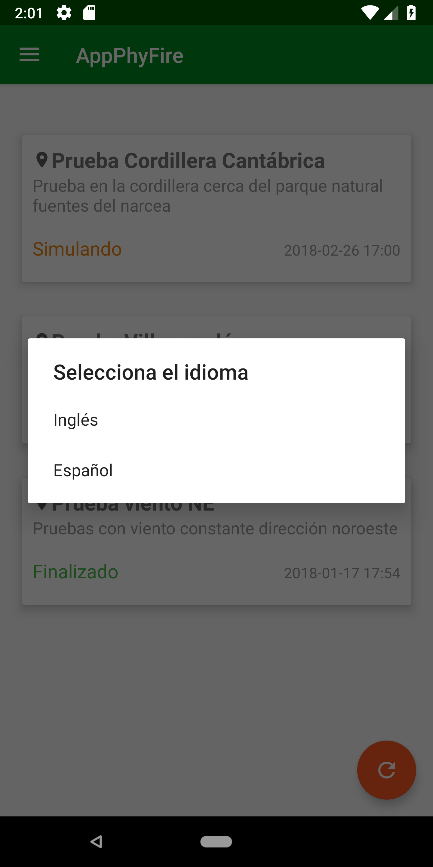
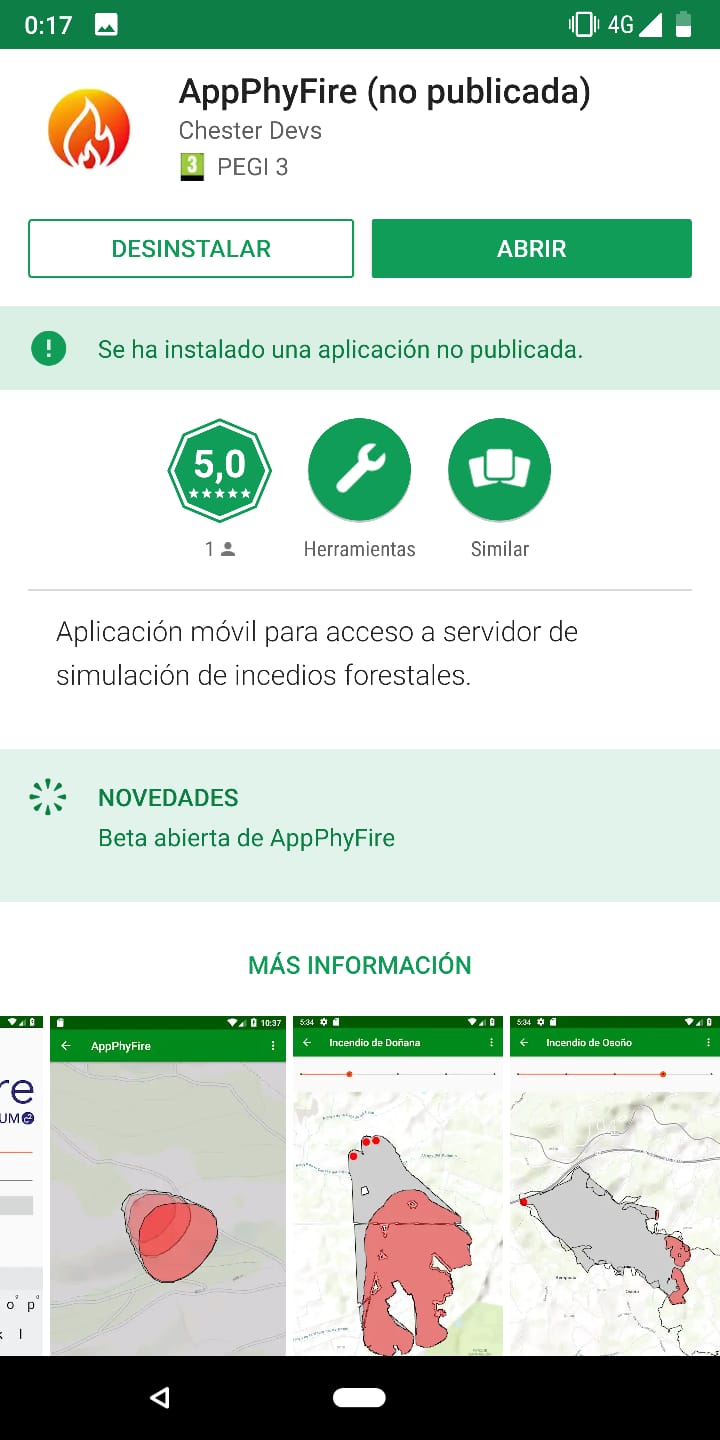


Figura 52 - Menú principal en inglés

Figura 53 - Menú de selección de idioma

# Conclusiones y líneas de trabajo futuras

En este apartado se describen las conclusiones que se derivan del desarrollo del proyecto. Se tratan de un conjunto de conclusiones relacionadas con los resultados del proyecto (producto final) y un conjunto de conclusiones técnicas. Además, se incluye un informe crítico indicando cómo se puede mejorar el proyecto.

Con la finalización de este proyecto se puede concluir que se han cumplido los objetivos propuestos y que, por tanto, se ha obtenido un sistema completamente funcional.

Se han completado además otros requisitos no funcionales, como la mantenibilidad del sistema de cara a una posible continuación del proyecto por otro desarrollador o a las actualizaciones de la aplicación.

La aplicación construida es compatible con la gran mayoría de móviles del mercado (Android 5.0 o API level 21 a Android 8.1 o API level 28 en adelante).

Debido a la gran fragmentación del sistema Android (gran cantidad de versiones y modelos distintos) se ha optado por publicar la aplicación en forma de versión beta en Google Play (versión preliminar técnica) como se aprecia en la Figura 54.

Esta versión permite recoger datos que reporten los usuarios que estén interesados en probar la aplicación, para después analizarlos y actuar en consecuencia (mejora de la funcionalidad, corrección de errores, etc.)

Figura 54 - Aplicación pública en Google Play

# Análisis crítico de mejora

Todo proyecto Software tiene en mayor o menor medida cierto margen de mejora, y este puede venir principalmente por dos vías:

* **Nueva funcionalidad.** Se puede mejorar el producto añadiéndole nueva funcionalidad que complemente a la funcionalidad que ya disponga el producto. Entraremos en profundidad en este aspecto en el apartado “**Líneas de trabajo futuras”.**
* **Mejora/Optimización de la funcionalidad existente.** Consiste en perfeccionar algún aspecto de la aplicación actual (tiempos de espera, mejoras de la comunicación cliente-servidor, experiencia de usuario…) de forma que el producto experimente alguna mejora en su funcionalidad.

Respecto a AppPhyFire, en opinión del desarrollador del proyecto, existe un aspecto en el que puede ser necesario una actualización y se trata de la interfaz de usuario.

La interfaz es un aspecto muy importante, ya que se trata de la capa que interactúa directamente con el usuario y que determinará en gran medida el grado de satisfacción de éste con el sistema.

La interfaz actual de AppPhyFire se ha desarrollado con “*Material Design*”, que se trata de un “lenguaje visual”[28] o guías de diseño (colores, formas, tamaños, tipografías, etc.) creado por Google para sus productos y para las aplicaciones del sistema Android, que proporciona a los desarrolladores multitud de herramientas y directrices para crear sus interfaces.

En la actualidad, se están realizando pruebas por parte de Google con una actualización de estas guías en el llamado “*Material Design 2*” [29], que actualiza este paradigma de diseño para la última versión de su sistema Android.

La razón principal por la que no se ha utilizado este método en AppPhyFire se debe a que *Material Design 2* se encontraba aún en desarrollo y por lo tanto no estaba aún establecido en el sector (ni siquiera Google ha empezado a emplearlo), sin embargo, se considera que el futuro puede ser un aspecto importante para tener en cuenta en la mejora de este aspecto de la aplicación.

# Líneas de trabajo futuras

En esta sección se describen algunos aspectos que podría incorporar AppPhyFire y que le dotarían de nueva funcionalidad que optimizarían aún más el trabajo de los profesionales o mejorarían la experiencia del usuario de la aplicación:

* **Sistema de notificaciones.** Android dispone de un sistema de notificaciones que puede ser aprovechado por la aplicación para advertir al usuario, por ejemplo, de que una de sus simulaciones ya está disponible para su visualización, ya que las simulaciones no se ejecutan de manera instantánea y requieren de tiempo de procesamiento en el servidor.
* **Compartición de simulaciones.** Podría ser útil que los profesionales pudieran compartir fácilmente simulaciones realizadas en el sistema, de cara a mejorarlas o a realizar estudios sobre las mismas.
* **Almacenamiento local de las simulaciones.** Se podría visualizar la simulación incluso sin conexión a la red o cuando esta es inestable, caso frecuente en zonas apartadas o rurales.
* **Lanzamiento de simulaciones desde la aplicación.** Este aspecto puede ser muy importante para el usuario, no sólo podría visualizar las simulaciones desde la propia zona de actuación si no que podría también mejorar las existentes creando nuevas. Desde el punto de vista técnico, esta mejora no es trivial, debido a que en ocasiones para realizar simulaciones hay que introducir un volumen notable de datos y conseguirlo con una experiencia de usuario satisfactoria puede resultar complejo.

# Bibliografía

[1] Joint Research Centre Science for Policy Report, *Forest Fires in Europe, Middle East and North Africa 2016*. 2017.

[2] D. Prieto Herráez, M. I. Asensio Sevilla, L. Ferragut Canals, J. M. Cascón Barbero, and A. Morillo Rodríguez, “A GIS-based fire spread simulator integrating a simplified physical wildland fire model and a wind field model,” *Int. J. Geogr. Inf. Sci.*, vol. 31, no. 11, pp. 2142–2163, 2017.

[3] A. L. Sullivan, “Wildland surface fire spread modelling, 19902007. 1: Physical and quasi-physical models,” *Int. J. Wildl. Fire*, vol. 18, no. 4, pp. 349–368, 2009.

[4] A. L. Sullivan, “Wildland surface fire spread modelling, 19902007. 2: Empirical and quasi-empirical models,” *Int. J. Wildl. Fire*, vol. 18, no. 4, pp. 369–386, 2009.

[5] R. C. Rothermel, “A mathematical model for predicting fire spread in wildland fuels,” *USDA For. Serv. Res. Pap. INT USA*, no. INT-115, p. 40, 1972.

[6] G. Barbara Chapman, *Using OpenMP : portable shared memory parallel programming*. .

[7] L. Ferragut, M. I. Asensio, and S. Monedero, “A numerical method for solving convection-reaction-diffusion multivalued equations in fire spread modelling,” *Adv. Eng. Softw.*, vol. 38, no. 6, pp. 366–371, Jun. 2007.

[8] F. L., A. M. I., and M. S., “Modelling radiation and moisture content in fire spread,” *Commun. Numer. Methods Eng.*, vol. 23, no. 9, pp. 819–833, Oct. 2006.

[9] L. Ferragut and M. I. Asensio, *Advances in Differential Equations and Applications*, vol. 4, no. October. 2014.

[10] D. Prieto, M. I. Asensio, L. Ferragut, and J. M. Cascón, “Sensitivity analysis and parameter adjustment in a simplified physical wildland fire model,” *Adv. Eng. Softw.*, vol. 90, pp. 98–106, 2015.

[11] M. I. Asensio, L. Ferragut, and J. Simon, “A convection model for fire spread simulation,” *Appl. Math. Lett.*, vol. 18, no. 6 SPEC. ISS., pp. 673–677, 2005.

[12] R. E. Burgan and R. C. Rothermel, *Behave: Fire Behavior Prediction and Fuel Modeling System - FUEL Subsystem*, no. May 1984. 1984.

[13] L. GUENVEUR, “Kantar Media análisis del mercado de Android en España.” [Online]. Available: https://es.kantar.com/tech/móvil/2017/junio-2017-cuota-de-mercado-de-smartphones-en-españa-2017/#).

[14] Android Academy, “Conoce Android Studio,” *Android Developers*, 2015. [Online]. Available: https://developer.android.com/studio/intro/index.html?hl=es-419.

[15] Esri, “ArcGis.” [Online]. Available: http://www.esri.es/arcgis/.

[16] Pivotal Software, “Spring Framework.” [Online]. Available: http://projects.spring.io/spring-framework/.

[17] Java Community Process, “Tomcat.” [Online]. Available: http://tomcat.apache.org/.

[18] Oracle Corporation, “What’s New in MySQL 8.0.” [Online]. Available: https://www.mysql.com/why-mysql/white-papers/whats-new-mysql-8-0/.

[19] The phpMyAdmin devel team, “Phpmyadmin Documentation.” [Online]. Available: https://docs.phpmyadmin.net/en/latest/index.html.

[20] ECURED, “Herramienta CASE.” [Online]. Available: https://www.ecured.cu/Herramienta\_CASE.

[21] Visual Paradigm, “Comprehensive UML Tools.” [Online]. Available: https://www.visual-paradigm.com/solution/uml/comprehensive-uml-tools/.

[22] Microsoft, “Microsoft Project.” [Online]. Available: https://products.office.com/es-es/project/project-and-portfolio-management-software?tab=tabs-1.

[23] Github, “Hoja de referencia para github git.”

[24] A. Durán, “Introducción a REM 1.2.2 1,” pp. 1–35, 2006.

[25] Omg, “UML 2.4.1 Superstructure Specification,” *October*, vol. 02, no. August, pp. 1–786, 2004.

[26] Universidad de Alicante, “Patron MVC.” [Online]. Available: https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html.

[27] Spring, “Introduction to Spring Web MVC framework.” [Online]. Available: https://docs.spring.io/spring/docs/3.2.x/spring-framework-reference/html/mvc.html.

[28] Google, “Material Design Guidelines.” .

[29] S. Hall, “Material Design 2.” .