

Kpax: Migración a Elgg 2.1.1

Rubén Vinuesa Sánchez

Máster en Software Libre

Administración de web y comercio electrónico

Francisco Javier Noguera Otero

Daniel Riera Terrén

19-06-2016

Copyright © 2016 Ruben Vinuesa Sanchez

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled (annexed) "GNU Free Documentation License".

FICHA DEL TRABAJO FINAL

| | |
|--------------------------------|---|
| Título del trabajo: | <i>Kpax: Migración a Elgg 2.1.1</i> |
| Nombre del autor: | <i>Rubén Vinuesa Sánchez</i> |
| Nombre del consultor/a: | <i>Francisco Javier Noguera Otero</i> |
| Nombre del PRA: | <i>Daniel Riera Terrén</i> |
| Fecha de entrega: | 06/2016 |
| Titulación: | <i>Master Software Libre</i> |
| Área del Trabajo Final: | <i>Administración de web y comercio electrónico</i> |
| Idioma del trabajo: | <i>Español - Castellano</i> |
| Palabras clave | <i>Kpax, Elgg, módulo</i> |

Resumen del Trabajo (máximo 250 palabras)

Kpax es una red social, que implementa un sitio web donde poder encontrar juegos, los cuales tienen en su definición algún propósito educativo o que permiten mejorar ciertas habilidades personales. Está construida mediante módulos con la aplicación de licencia libre *Elgg*.

Con este proyecto se pretende implantar *Kpax* en la última versión de la aplicación *Elgg*, que corresponde a la 2.1.1 en el momento de escribir este documento. Las versiones con las que se inició el trabajo original, hace ya algunos años, están discontinuadas. Cualquier mejora implica un gran esfuerzo, muchas horas de trabajo y tiene una curva de aprendizaje elevada. El objetivo final de este trabajo es que todos los módulos relacionados con *Kpax* se puedan instalar y estar operativos, en *Elgg* 2.1.1 haciendo las modificaciones necesarias para ello.

Para llegar a ese objetivo se plantea crear un sistema, que partiendo desde cero, instale todo lo necesario mediante un script, configurando lo mínimo posible de forma manual. Una vez preparado, pasar a montar los diferentes módulos de *Kpax* (modificados), para que finalmente esté totalmente operativa la aplicación y preparada para futuros proyectos.

Siguiendo el procedimiento documentado en este trabajo se llega al resultado planteado y se finaliza con un sistema tal y como se había diseñado.

Concluir comentando que *Kpax* es una aplicación que ofrece numerosas posibilidades para el desarrollo de futuros proyectos y que ha sido muy interesante el poder aplicar lo aprendido en el master.

Abstract (in English, 250 words or less):

Kpax is a social network that implements a web site where you can find games, which have in defining an educational purpose or that improve certain personal skills. It is built using modules with the free license application *Elgg*.

This project is intended to implement *Kpax* in the latest version of the application *Elgg*, corresponding to 2.1.1 at the time of this writing. Versions with the original work began, some years ago, they are discontinued. Any improvement involves a great effort, long working hours and has a high learning curve. The ultimate goal of this work is that all *Kpax* related modules can be installed and be operational in *Elgg* 2.1.1 making the necessary changes to it.

To reach this goal it is proposed to create a system which, starting from zero, install everything needed by a script, setting the minimum possible manually. Once prepared, pass to ride the different modules *Kpax* (modified), so that the application is finally fully operational and ready for future projects.

Following the procedure documented in this paper reaches the proposed outcome and ends with a system as it was designed.

Conclude by saying that *Kpax* is an application that offers numerous possibilities for the development of future projects and has been very interesting to apply what I learned in the master.

Resumen del Proyecto.

Este proyecto nace con la necesidad de actualizar la red social *Kpax* [1], en cuyo sitio web se dan de alta juegos que tienen un propósito educativo y que ha sido desarrollada por personal de la UOC, tanto docente como alumnado.

El primer objetivo del proyecto será crear un entorno en el que se utilicen todas las últimas versiones del software base, siendo este, el sistema operativo (Linux), el servidor web Apache [2], el gestor de la base de datos MySQL [3] o lenguajes como PHP [4].

El siguiente objetivo es el implementar el núcleo sobre el que se asienta *Kpax*, el cual aporta la funcionalidad principal. Se utiliza *Elgg* [5] que es una aplicación de licencia libre que permite crear redes sociales. Actualmente la versión de *Elgg* utilizada en la aplicación *Kpax* es la versión 1.6, siendo bastante antigua, por lo que se quiere pasar a la última disponible, que en el momento de desarrollar este trabajo es la 2.1.1.

La aplicación *Elgg* más los módulos¹ [6] incluidos en el producto y los propios de *Kpax*, hacen posible la red social *Kpax*.

El objetivo principal y final de este proyecto es migrar la red social *Kpax* a *Elgg* 2.1.1, con una adaptación inicial de los 5 módulos principales de la versión 1.0 de *Kpax* [7].

- *Kpax*; módulo principal de la aplicación desarrollado por la comunidad universitaria.
- *Theme_Kpax*; módulo que permite cambiar el tema utilizado por defecto en *Elgg* a uno propio para *Kpax*. Desarrollado por la comunidad universitaria para *Kpax*.
- *Logirequired*; módulo que permite deshabilitar ciertas opciones de la página principal, antes de hacer el inicio de sesión. Está desarrollado por la comunidad *Elgg*.
- *Apiadmin*; módulo que permite gestionar las claves “API” utilizadas por la aplicación. Desarrollado por la comunidad *Elgg*.
- *HTML5*; módulo que permite utilizar la funcionalidad html 5. Desarrollado por la comunidad *Elgg*.

¹ Módulo; en toda la documentación de *elgg* se utiliza el concepto “plugin”.

Índice de contenido

| | |
|---|----|
| Resumen del Proyecto..... | 1 |
| 1. Introducción..... | 5 |
| 2. Estudio de viabilidad..... | 9 |
| 2.2. Estudio de la situación actual..... | 10 |
| 2.3. Definición de los requisitos del sistema..... | 11 |
| 2.4. Estudio, valoración y selección de la solución..... | 11 |
| 3. Análisis del sistema..... | 13 |
| 3.1. Definición del sistema..... | 13 |
| 3.2. Establecimiento de requisitos..... | 15 |
| 3.3. Definición de interfaces de usuario..... | 15 |
| 3.4. Especificación del plan de pruebas..... | 16 |
| 4. Diseño del sistema..... | 17 |
| 4.1. Arquitectura..... | 17 |
| 4.2. Identificación de subsistemas..... | 18 |
| 4.3. Elección de las licencias más adecuadas..... | 19 |
| 4.4. Requisitos de implantación..... | 20 |
| 5. Desarrollo / Implantación del sistema..... | 21 |
| 5.1. Planificación de las actividades de integración del sistema..... | 21 |
| 5.2. Desarrollo..... | 24 |
| Conclusiones..... | 40 |
| Bibliografía..... | 41 |

Índice de Figuras

| | |
|---|----|
| Figura 1 Arquitectura <i>Kpax</i> | 5 |
| Figura 2 Estructura directorio módulo <i>Kpax</i> | 7 |
| Figura 3 Estructura directorio módulo <i>groups</i> | 7 |
| Figura 4 Arquitectura lógica <i>Kpax</i> | 17 |
| Figura 5 Organización en subsistemas..... | 18 |
| Figura 6 Diagrama de Gantt del proyecto..... | 21 |
| Figura 7 Trello..... | 22 |
| Figura 8 Página de trabajo de Trello..... | 22 |
| Figura 9 Tarjeta Trello..... | 23 |
| Figura 10 Github..... | 23 |
| Figura 11 Creación máquina virtual..... | 24 |
| Figura 12 Instalación Ubuntu Linux..... | 24 |
| Figura 13 Instalación <i>VBoxAdditions</i> | 25 |
| Figura 14 Ejecución script de instalación..... | 25 |
| Figura 15 Finalización script de instalación..... | 29 |
| Figura 16 Configuración <i>Elgg</i> | 29 |
| Figura 17 Configuración cuenta administrador <i>Elgg</i> | 29 |
| Figura 18 Página inicial <i>Kpax 2</i> | 30 |
| Figura 19 Panel de administración <i>Kpax</i> | 30 |
| Figura 20 Módulo <i>Elgg Developers Tools</i> | 32 |
| Figura 21 Configuración <i>Elgg Developers Tools</i> | 32 |
| Figura 22 Error <i>Apiadmin</i> | 33 |
| Figura 23 Módulo <i>web services</i> | 33 |
| Figura 24 Página principal módulo <i>Apiadmin</i> | 33 |
| Figura 25 <i>Kpax</i> sin <i>Loginrequired</i> | 34 |
| Figura 26 <i>Kpax</i> con <i>Loginrequired</i> | 34 |
| Figura 27 <i>manifest.xml</i> del módulo <i>Html5</i> | 35 |
| Figura 28 Módulo <i>Loginrequired</i> | 35 |
| Figura 29 Error instalación módulo <i>Kpax</i> | 36 |
| Figura 30 Modificación <i>manifest.xml Kpax</i> | 36 |
| Figura 31 Generación API para <i>Kpax</i> | 37 |
| Figura 32 Modificación <i>kpaxSrv.php</i> | 37 |
| Figura 33 Error en <i>start.php</i> de módulo <i>Kpax</i> | 37 |
| Figura 34 Fichero <i>start.php</i> de <i>Kpax</i> modificado..... | 38 |
| Figura 35 Módulo <i>Kpax 2</i> | 38 |
| Figura 36 <i>Kpax 2</i> en ejecución..... | 38 |
| Figura 37 Modificación <i>manifest.xml</i> de <i>Theme_Kpax</i> | 39 |
| Figura 38 Página de <i>Kpax</i> con su propio tema..... | 39 |

Índice de Tablas

| | |
|-------------------------------------|----|
| Tabla 1 Software a utilizar | 19 |
| Tabla 2 Módulos a implementar | 19 |

1. Introducció.

La Universitat Oberta de Catalunya (en adelante UOC), cuenta con una aplicaci3n web de prop3sito educativo llamada *Kpax* 1.0. Este proyecto fue iniciado en 2011 con el objetivo de crear una red social para el aprendizaje, donde se diesen de alta juegos con la característica com3n de que estos tuviesen alguna funci3n educativa en su desarrollo o lo que es lo mismo que entre su jugabilidad contasen con ciertos aspectos que obligasen al jugador a mejorar, potenciar o entrenar cualidades propias.

Kpax 1.0 fue desarrollado sobre la aplicaci3n *Elgg*, m3s los correspondientes m3dulos, creados principalmente en PHP y J2EE (Java con web services).

En su configuraci3n inicial *Kpax* se compone de lo mostrado en la siguiente figura.

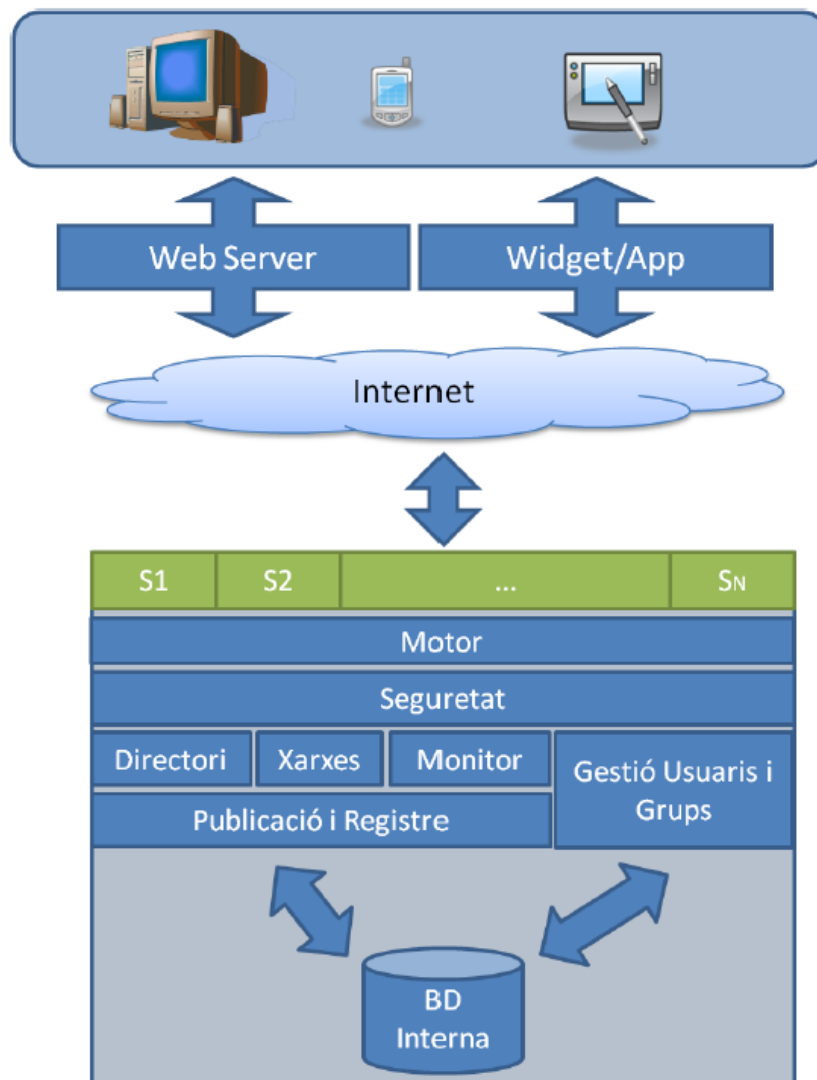


Figura 1 Arquitectura *Kpax*

Elgg es el motor de la red social que en su versión 1.6, dota de toda su funcionalidad "núcleo" a la aplicación web *Kpax* 1.0. La citada versión de *Elgg*, es de aproximadamente de agosto del 2009 y sobre esta se han ido desarrollando los diferentes módulos de *Kpax*. Estos son propios del personal de la universidad, tanto docente como alumnado. Se considera que actualmente tienen una integración muy compleja y que cada módulo implica muchos cambios, como para seguir utilizando o ampliando esa infraestructura. Hay que considerar también que la curva de aprendizaje se vuelve cada vez más compleja en relación directa con los dos conceptos mencionados.

Por una parte la versión de *Elgg* 1.6 es muy antigua y ya se encuentra discontinuada por lo que no se publican mejoras ni para ella ni para los módulos. Por otra parte se han hecho muchas modificaciones a los diferentes componentes de la aplicación *Kpax* que han supuesto que el proyecto se ha vuelto inmanejable y muy complejo. Para solucionarlo se quiere implementar una nueva versión de *Kpax*, con la correspondiente adaptación de toda la funcionalidad existente. Se pretende también que la siguiente versión de *Kpax* siga haciendo uso de llamadas a servicios remotos utilizando para ello *Api's REST* [8] y devolviendo ficheros *JSON* [9]. Ofrecer también otras posibilidades, como por ejemplo que el servidor esté conectado a una nueva base datos en *MongoDB* [10], conteniendo toda la información de la aplicación *Kpax* 2.0.

Así y por lo comentado, el objetivo final es implementar una nueva red social, denominada "*Kpax* 2.0", en la que todos sus componentes y módulos estén actualizados, se implementen otras tecnologías y pueda servir de base para futuros proyectos. La idea general de construir la base de la aplicación, desarrollando los diferentes módulos que la componen independientes unos de otros en la medida de lo posible.

Es importante detallar que el objetivo principal de este proyecto es adaptar una serie de módulos a la nueva versión del motor de red social. A continuación se explica, de forma genérica qué es un módulo y de que se compone ya que es un concepto ampliamente utilizado a lo largo del documento.

Los módulos son componentes software que pueden modificar el comportamiento de *Elgg* y añadir nuevas funcionalidades. Como ejemplo se puede destacar el propio *Kpax*, que gracias a un determinado número de módulos se crea un sitio web para facilitar un lugar de encuentro donde ubicar juegos con un contenido educativo.

Los módulos se pueden obtener principalmente en tres sitios, en la propia comunidad *Elgg* [6], en *Github* [11] o en sitios de terceros. Evidentemente también se pueden desarrollar a medida.

Los módulos se pueden clasificar en función de su cometido, como los que añaden alguna funcionalidad nueva, los que modifican el entorno gráfico (temas), los que modifican o añaden paquetes de idioma, etc...

La estructura de un módulo [12] consiste en una carpeta con el nombre de este y un mínimo de dos ficheros. El fichero *manifest.xml*, donde se encuentra información del propio módulo y el fichero *start.php* con el código mínimo necesario para poder ejecutarlo.

A continuación se puede ver la estructura de directorios raíz de 2 módulos.

Index of file:///var/www/html/elgg-2.1.1/mod/kpax/

[^ Up to higher level directory](#)

Name








-  actions
-  languages
-  lib
-  manifest.xml
-  pages
-  start.php
-  views

Figura 2 Estructura directorio módulo *Kpax*

Index of file:///var/www/html/elgg-2.1.1/mod/groups/

[^ Up to higher level directory](#)

Name













-  actions
-  composer.json
-  graphics
-  icon.php
-  languages
-  lib
-  manifest.xml
-  start.php
-  tests
-  upgrades
-  views
-  views.php

Figura 3 Estructura directorio módulo *groups*

La instalación de un módulo, en su forma básica, pasa por descargarlo de donde corresponda, copiarlo en la carpeta */mod* de la instalación de *Elgg*, verificar que se considera correcto en la página de administración y si es así, activarlo.

Todo el desarrollo del proyecto se detalla en este documento, que se estructura de la siguiente forma:

- Capítulo 1; esta introducción al proyecto.
- Capítulo 2; Estudio de la viabilidad, donde se consideran el conjunto de necesidades planteadas para establecer primero qué problema se pretende resolver y si el proyecto se puede llevar a cabo, teniendo en cuenta las diferentes soluciones existentes y los recursos de los que se dispone.
- Capítulo 3; Análisis del sistema, para estudiar las necesidades que se quieren satisfacer en el proyecto, con el fin de describir detalladamente la solución tecnológica establecida en la fase anterior. También se especificarán las interfaces de usuario que permitirán a estos interactuar con el sistema.
- Capítulo 4; Diseño del sistema, para realizar el diseño tecnológico de la solución, proponiendo una arquitectura global de esta y estudiando cada uno de los casos de uso existentes.
- Capítulo 5; Desarrollo, se construye el sistema, a partir de sus subsistemas, que compone el proyecto de una forma ordenada, teniendo en cuenta temas como el entorno de desarrollo utilizado, documentación generada, etc...
Se aplica la metodología adecuada en función de necesidades.
Se considera que este apartado también cubre la fase de implantación del sistema ya que por el tipo de proyecto que es, son equivalentes.

2. Estudio de viabilidad

Se considera el conjunto de necesidades planteadas para establecer primero qué problema se pretende resolver y si el proyecto se puede llevar a cabo, teniendo en cuenta la solución existente y los recursos de los que se dispone.

2.1. Establecimiento del alcance del sistema

La Universitat Oberta de Catalunya (UOC) es una universidad a distancia que ofrece numerosos estudios en diferentes ámbitos. Dentro del área de estudios de informática y multimedia hay diferentes proyectos desarrollados por personal docente, externo o alumnos.

Uno de esos proyectos es la red social “Kpax”, la cual se creó con el objetivo principal de ofrecer una herramienta que reuniese las potencialidades que ofrecen los juegos (serios) [13] como herramientas para el aprendizaje informal y la gamificación como herramienta para la motivación de usuarios (según definición de Daniel R.T., uno de los creadores de la aplicación). También tiene el propósito de que el personal de la universidad o relacionado con ella, pudiese desarrollar o poner en práctica lo visto en sus estudios.

Esta aplicación y con el paso del tiempo se ha ido ampliando de diferentes formas, llegando a un estado en el que ya prácticamente resulta inviable hacer alguna modificación sin que se tenga que invertir mucho tiempo y modificar muchos componentes.

Con el fin de que la UOC, pueda seguir haciendo uso de esta aplicación para sus fines educativos u otros que considere, ha decidido renovar la misma a partir de los siguientes objetivos:

- Todo el software utilizado tiene que seguir teniendo licencia libre.
- Dentro de la medida de lo posible se tiene que utilizar las últimas versiones de los productos.
- Se tiene que migrar a la nueva versión, todo el código de los módulos principales de la red *Kpax*.

A nivel económico no se requiere gasto alguno en principio ya que se quiere gestionar como un proyecto de software libre.

A nivel técnico se tienen las necesidades totalmente cubiertas ya que todo el personal implicado tiene los conocimientos necesarios para afrontar la migración sin problemas.

A nivel legal no se encuentran problemas ya que todo está bajo normativa de uso de licencias libres.

A nivel operativo la única necesidad es dejar operativo el entorno en las últimas versiones de cada uno de los productos.

Este proyecto afecta a otros como:

- Migración de servicios en Java.
- Migración de base de datos a *MongoDB*.

2.2. Estudio de la situación actual

Dentro de este apartado se estudia el estado del sistema actual y se hace un primer diagnóstico del mismo, para ello se solicita toda la información necesaria a los responsables y consultores de la aplicación.

La información facilitada se puede dividir en tres grandes partes, la primera de ellas hace referencia a la plataforma utilizada con el software base correspondiente, la segunda al núcleo sobre la que se asienta la aplicación y la tercera a la propia aplicación.

La plataforma base utiliza software libre, haciendo referencia a sistema operativo, servidor web, gestor de base de datos o aplicaciones de desarrollo.

El núcleo sobre la que se asienta la aplicación *Kpax*, corresponde a un motor de red social creado por la comunidad de código abierto, permite ampliar su funcionalidad mediante módulos o extensiones. Se desarrolla principalmente en PHP aunque permite otros como Java.

La aplicación *Kpax* se compone también de diferentes módulos creados en PHP, partiendo de un principal, más otros complementarios. Algunos de ellos han sido desarrollados por la comunidad universitaria y otros por la comunidad *Elgg*.

Una vez analizada la información obtenida en la descripción del sistema se llega a las siguientes conclusiones que se comentan a continuación.

- Todo el software base se encuentra obsoleto o desactualizados en referencia a las versiones existentes.
- El motor de la red social es una versión muy antigua que ya no permite más actualizaciones ni parches.
- En la revisión del código de los módulos principales se encuentra que ciertas partes puedan estar obsoletas en función de implementarlos en nuevas versiones del producto.
- Realizar cualquier posible cambio en el entorno actual implica numerosas modificaciones en diferentes componentes.
- Se exige también un conocimiento muy elevado de cómo se ha desarrollado en su conjunto por lo que limita su ampliación.

2.3. Definición de los requisitos del sistema

Después de hacer un estudio y análisis a partir de la información facilitada por los responsables de la aplicación, así como la revisión realizada al sistema, se establecen los siguientes requisitos diferenciados por su funcionalidad.

A partir de los requisitos técnicos se establece una serie de mejoras que se deben cumplir en el nuevo sistema. Por un lado todos los componentes software necesarios se tendrán que actualizar a la última versión posible, teniendo que ser estos compatibles con el motor de la red social, utilizando la última versión existente. En consecuencia de lo comentado se tendrá que hacer una adaptación inicial de todos los módulos de la aplicación *Kpax* al nuevo sistema. En el caso de que sea necesario de harán las correcciones necesarias en el código.

En función de los requisitos operativos, se considera como principal que se tiene que mantener las mismas características y estructura (inicial) que existe en el sitio original.

Haciendo referencia a los requisitos legales, hay que establecer que la licencia de todos los productos utilizados se tiene que enmarcar dentro del marco de GNU de la "*Free Software Foundation*" [14] o equivalente.

Este proyecto no tiene requisitos económicos o por lo menos en un principio.

2.4. Estudio, valoración y selección de la solución

Al realizarse una migración de versión de producto no se contempla alternativas a la solución pedida ni valoraciones por cada una de ellas, por lo que se desarrollan estos apartados de forma conjunta, siendo la solución elegida la que se comenta a continuación. Se utiliza para ello las últimas versiones posibles, todas ellas cumpliendo el requerimiento de ser compatible en su totalidad con la aplicación del motor de la red social.

Por cada uno de los componentes hay que detallar lo siguiente.

- Se mantendrá el sistema operativo con una determinada distribución de Linux en una versión compatible con todos los componentes utilizados.
- El servidor web a utilizar será nativo de Linux, cumpliendo las especificaciones y configuraciones detalladas en el manual de instalación de la aplicación.
- Se mantiene el mismo gestor de base de datos en su versión más actual, considerando que se utilizará para el motor de la red social y no para la aplicación *Kpax*, ya que se quiere llevar esta a un servidor independiente en *MongoDB* que será utilizado mediante llamadas *REST*.

- Se utilizarán los módulos de lenguajes necesarios, según los requerimientos del motor de la red social.
- Se implementará la última versión, en el momento de desarrollar este proyecto, del motor de la red social.
- Se hará una adaptación inicial de todos los módulos necesarios para la aplicación *Kpax*, modificando lo necesario para poder ejecutarlos en el nuevo entorno.

En resumen, tal y como se ha comentado anteriormente, se mantienen los mismos productos pero en sus versiones más actuales. Se mantiene la licencia libre, por lo que su valoración económica es cero y la configuración general de todos los productos implicados es equivalente. En este caso se cumplen los requisitos funcionales y técnicos definidos al respecto.

3. Análisis del sistema

En este capítulo se estudian las necesidades que se quieren satisfacer en el proyecto, con el fin de describir detalladamente la solución tecnológica establecida en la fase anterior. También se especificarán las interfaces de usuario que permitirán a estos interactuar con el sistema.

3.1. Definición del sistema

En este apartado se describen los requisitos exactos que cumplirá el sistema, considerando todos los puntos comentados anteriormente.

- Se requiere el utilizar las últimas versiones del software, con la característica común a todas ellas que sean compatibles con la aplicación del motor de la red social.
- Todos los productos tendrán licencia libre ya que es uno de los objetivos a cumplir en el propio proyecto.
- Es de obligado cumplimiento, como mínimo, el mantener la misma funcionalidad que existe en la versión original.

Una vez establecido lo anterior, se cambia a la parte principal del proyecto, en la que se implementan los diferentes módulos dentro del motor de la red social, lo cual se expone a continuación.

Se descarga de la comunidad *Elgg*, se adaptan y se instalan los siguientes módulos:

- El módulo “Apiadmin” para la gestión de claves “API”.
- El módulo “Loginrequired” para la modificación de la página principal impidiendo accesos no autorizados a partes de la aplicación sin el inicio de sesión adecuado.
- El módulo “html5” para habilitar las características relacionadas con HTML 5.

Se descarga del repositorio Github facilitado por la universidad, se adapta y se instalan los siguientes módulos:

- El módulo *Kpax* con la funcionalidad “Develop” incluida, siendo este el principal de la aplicación.
- El módulo *Theme_Kpax*, utilizado para modificar el tema de *Elgg*, creando uno propio para *Kpax*.

Se considera un entorno tecnológico con unas características como las que se especifican a continuación, cumpliendo los requerimientos anteriormente comentados.

- Se utiliza un sistema operativo Ubuntu Linux en su versión 15.10, la cual ha sido confirmado que es totalmente compatible con *Elgg* y los diferentes componentes.
- Un servidor web Apache 2, según requerimiento de *Elgg*.
- Un servidor de base de datos MySQL 5+, según requerimiento de *Elgg*.
- Un servidor de PHP 5.5 +, según requerimiento de *Elgg*.
- El motor de red social *Elgg* 2.1.1.

Haciendo referencia a las normas y estándares a seguir, se establece que la implementación de todo el sistema se tiene que dar en tres grandes apartados. Cumpliendo estos, se habrá conseguido que la aplicación *Kpax* esté operativa con un mínimo de funcionalidad.

Se comienza por el sistema operativo, descargado en versión “ISO” de la página de Ubuntu y haciendo la instalación inicial sin nada más.

Se utiliza un script para instalar el resto de componentes necesarios, hasta dejar casi finalizada la configuración de *Elgg*.

Una vez establecida la configuración final de *Elgg*, se copian los módulos *Kpax*, *Theme_Kpax*, *Apiadmin*, *Loginrequired* y *html5*, según el procedimiento comentado.

Se identifican los siguientes usuarios para la definición de requisitos y aceptación de la solución final.

- Daniel R.T. como responsable que solicita el proyecto.
- Francisco Javier N.O. como responsable técnico del proyecto.

3.2. Establecimiento de requisitos

Añadiendo la información facilitada por los responsables del proyecto, se establecen los mismos requisitos que los comentados en el apartado anterior. Como el objetivo final del proyecto es que cada módulo esté operativo, se tendrán que hacer las correcciones necesarias para que se pueda copiar en la carpeta de *Elgg* que los contiene, que se configuren adecuadamente, que la aplicación los considere adecuados y que se puedan activar, para proporcionar su funcionalidad. Se tendrán que instalar y configurar por este orden.

- El módulo *Apiadmin*
- El módulo *Loginrequired*.
- El módulo *HTML5*.
- El módulo *Kpax* con la funcionalidad "develop".
- El módulo *Theme_Kpax*.

3.3. Definición de interfaces de usuario

No se considera en este proyecto ningún cambio a la aplicación original por lo que se respeta la funcionalidad. En el caso de los interfaces de usuario se mantienen los existentes que en líneas generales estos son:

- Usuario administrador.
- Desarrollador.
- Usuario de la aplicación.

Ya se ha comentado anteriormente que la aplicación *Kpax*, tiene su propio tema el cual modifica la página web que se habilita por defecto con *Elgg*.

3.4. Especificación del plan de pruebas

Una vez implementado el sistema y migrados los diferentes módulos principales, se hace una serie de pruebas para confirmar el funcionamiento correcto. Todas estas pruebas se harán con el sistema totalmente implantado y en presencia de todo el personal implicado.

Unitarias; por las que por cada módulo se comprueba que carga correctamente en el sistema y *Elgg* lo aprueba, se puede activar, no da errores y funciona como se le espera. La pruebas las hace el encargado de la migración así como por el responsable del proyecto en caso de considerarlo oportuno.

Integración; por las que y en conjunción con el resto de componentes se valida cada componente de la migración. Tanto el encargado de la migración como el responsable del proyecto hacen estas pruebas.

De sistema; por las que y en conjunción con el resto de componentes se valida todo el sistema en su conjunto. Es el responsable del proyecto el encargado de hacer estas pruebas.

De aceptación; por la que los responsables del proyecto validan la migración.

En el anexo se adjunta una hoja de cálculo para hacer el seguimiento de las diferentes pruebas.

4. Diseño del sistema

Se realiza el diseño tecnológico de la solución y se propone una arquitectura global de esta, estudiando cada uno de los casos de uso existentes.

4.1. Arquitectura

Se define la arquitectura del sistema como los diferentes componentes que forman parte de la solución. Tal y como se muestra en la imagen, la solución se puede dividir en dos grandes apartados, por un lado tenemos el cliente, donde se implementará toda la infraestructura de *Elgg / Kpax* y por otro la parte servidora, donde se habilitarán los diferentes servicios remotos a los que se accederá mediante llamadas REST con obtención de resultados en ficheros JSON. Con esta arquitectura se consigue modularizar el sistema y hacer posible la independencia de unos componentes con otros, siendo esto uno de los requisitos a cumplir.

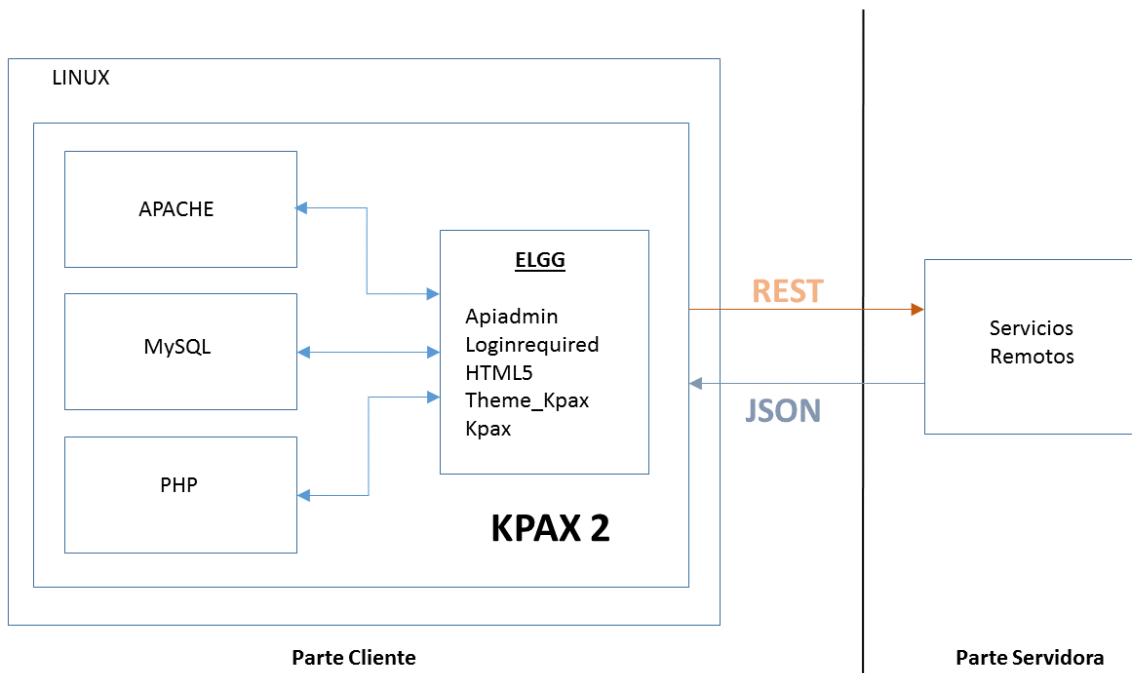


Figura 4 Arquitectura lógica Kpax

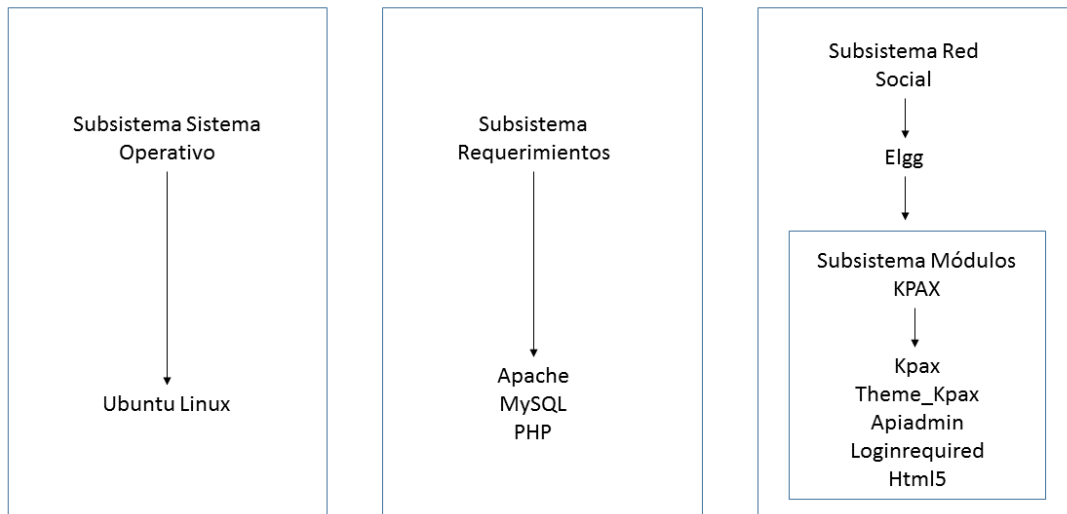
Según se muestra en la figura para hacer posible *KPAX 2*, en su parte cliente, es necesario contar con un servidor Linux (Ubuntu 15.10) al que se le añade y configura Apache 2, MySQL 5+ y PHP 5.5. Con la instalación y disposición de *Elgg 2.1.1* [15] más los diferentes módulos que componen *Kpax*, se logra que esté operativa la aplicación. Se muestra también las conexiones entre los diferentes componentes, indicando si estas se producen a nivel local o no. En el caso especial de la base de datos, se tiene que *Elgg* lo hará a nivel local a MySQL, pero *Kpax* lo hará a nivel remoto por usar MongoDB (pertenece a otro proyecto).

4.2. Identificación de subsistemas

Para facilitar la organización del proyecto y todos los componentes utilizados, se establece una clasificación alternativa en subsistemas que estén directamente relacionados. La imagen que se muestra a continuación lo detalla para la parte cliente, directamente relacionado con la arquitectura del sistema.

Se establecen tres grandes subsistemas, el primero de ellos corresponde al sistema operativo y todas las actualizaciones necesarias para que sea totalmente compatible con el motor de red social. A continuación se establece el subsistema con el software necesario para la operatividad del motor de red social, siendo el servidor web con Apache, la base de datos (local) con MySQL y el intérprete de scripts con PHP. Se incluye también todas las actualizaciones necesarias y configuraciones. Por último se tiene el subsistema de red social con el componente principal que corresponde a *Elgg*. A su vez y dentro de este, se establece el subsistema de *Kpax* con los diferentes módulos que componen la aplicación.

No se comenta nada de la parte servidora ya que no corresponde a este proyecto.



Subsistema cliente

Figura 5 Organización en subsistemas

4.3. Elección de las licencias más adecuadas

A continuación se detallan los diferentes componentes software utilizado, nombre, versión y licencia, cumpliendo los requisitos especificados en capítulos anteriores.

| Componente | Software | Versión | Licencia |
|--------------------|--------------|---------|---|
| Sistema operativo | Ubuntu Linux | 15.10 | GNU Public License |
| Servidor web | Apache | 2.4 | Apache Software License |
| Base de datos | MySQL | 5 + | GNU Public License (uso no comercial) |
| Interprete scripts | PHP | 5.5 | PHP License v3.01 (Open Source License) |
| Motor red social | <i>Elgg</i> | 2.1.1 | GNU Public License / MIT License |

Tabla 1 Software a utilizar

Los diferentes módulos que componen la aplicación *Kpax*, siguen un esquema diferente, aunque el licenciamiento es equivalente. Se especifica en el autor, si corresponde este a un módulo realizado específicamente para la aplicación *Kpax* (UOC) o es un módulo genérico realizado por la comunidad *Elgg* (sin especificar el autor real).

| Módulo | Versión | Licencia | Autor |
|--------------------------------|---------|--------------------|---------------------------------|
| <i>Kpax</i> (modificado) | 0.1.1 | GNU Public License | Universitat Oberta de Catalunya |
| <i>Theme_Kpax</i> (modificado) | 0.1.1 | GNU Public License | Universitat Oberta de Catalunya |
| Apiadmin | 1.0 | GNU Public License | Comunidad <i>Elgg</i> |
| Loginrequired | 1.9.8 | GNU Public License | Comunidad <i>Elgg</i> |
| Html5 | 1.0 | GNU Public License | Comunidad <i>Elgg</i> |

Tabla 2 Módulos a implementar

4.4. Requisitos de implantación

Se establece como requisitos de implantación como la instalación de los diferentes subsistemas, haciendo posible que con la finalización correcta de cualquiera de ellos se pueda dar paso a la instalación del siguiente.

Subsistema sistema operativo.

Dependiendo de la carga de usuarios se establece un servidor físico o máquina virtual con al menos un procesador, 2 GB's de memoria RAM y conectividad de red. La instalación del sistema operativo se da a través de una ISO descargada de la web de Ubuntu y una vez hecho se tendrá que actualizar a las últimas versiones de todos los productos software instalados. Se considera finalizado este subsistema cuando tanto el sistema operativo y las actualizaciones estén ubicadas, corregidos los posibles problemas encontrados y verificada la conectividad de red, principalmente a Internet.

Subsistema requerimientos.

Los requerimientos corresponden a aquellos componentes software que son necesarios según las especificaciones del motor de red social. Estos son el servidor web Apache, el sistema gestor de base de datos en local MySQL y el intérprete de scripts PHP. Su instalación se da a través del comando "apt-get install producto". De esta forma se consigue la última versión posible con las actualizaciones necesarias. Se considera finalizado este subsistema cuando se ha instalado y configurado adecuadamente cada uno de los componentes siguiendo las directrices marcadas en el manual de instalación de *Elgg*.

Subsistema motor de red social.

El componente principal de todos los subsistemas comentados es el motor de red social que corresponde con la aplicación *Elgg*. Se descarga la última versión de la web correspondiente, se descomprime, se configura con las bases de datos en local necesaria y se crea el usuario administrador. La finalización de este subsistema se corresponde con el acceso a la página de inicio de sesión y el acceso mediante el usuario administrador.

Subsistema módulos.

La funcionalidad de la aplicación *Kpax*, se establece a través de la instalación de un módulo principal (*Kpax*) más otros tantos complementarios. Como los que se van a montar corresponden en su mayoría a los utilizados en la versión anterior de *Kpax*, se hace un proceso previo de comprobación y posible corrección de código. Este se corresponde con la descarga del módulo, el copiarlo a la carpeta correspondiente de *Elgg*, ver si el sistema permite activarlo, siendo así se activará para comprobar si funciona correctamente y si no se tendrá que corregir. Se considera finalizado este subsistema cuando todos los módulos relacionados con *Kpax* funcionan adecuadamente.

Ya se ha comentado anteriormente que para facilitar la implementación de los diferentes subsistemas, se creará un script que automatizará todo el proceso, excluyendo la configuración final de *Elgg* y de los diferentes módulos de *Kpax*.

5. Desarrollo / Implantación del sistema

En este apartado se va a construir el sistema, a partir de sus subsistemas, de una forma ordenada, teniendo en cuenta el entorno utilizado, documentación generada, etc...

Se aplicará una metodología considerada adecuada en función de necesidades.

Se considera que este apartado también cubre la fase de implantación del sistema ya que por el tipo de proyecto que es, ambos apartados son equivalentes.

5.1. Planificación de las actividades de integración del sistema

La planificación de este proyecto se refleja en el diagrama de Gantt que se muestra a continuación. Se dan unas fechas aproximadas que pueden cambiar en función de necesidades. Se indican con hitos las fechas importantes en las que se tiene que hacer entregas previamente pactadas.

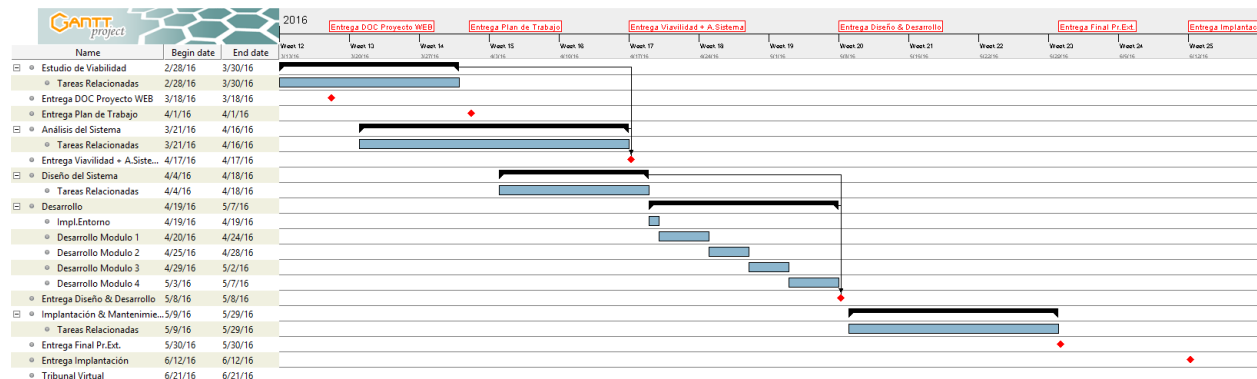


Figura 6 Diagrama de Gantt del proyecto

Por otro lado y para hacer un seguimiento del proyecto de forma adecuada, se pacta por todas las partes seguir una metodología del tipo ágil [16]. A través de esta técnica se establecen reuniones semanales en las que cada individuo del proyecto comenta qué ha hecho en la última semana, con qué está trabajando, posibles problemas que pudiese tener y qué tareas tiene previstas para el futuro. Se consigue de esta forma que el proyecto avance y que no se quede atascado en ningún momento así como que otros integrantes pudiesen ayudar en la resolución de tareas.

Como herramienta para el seguimiento del proyecto se va a utilizar Trello [17]. Esta permite mediante la utilización de fichas explicativas de cada unidad mínima del proyecto, indicar en qué estado está, quien la tiene asignada, comentarios, fechas, etc...

En la siguiente imagen se muestra la web de Trello, después de hacer inicio de sesión y en la que se ve el proyecto *Kpax2*.

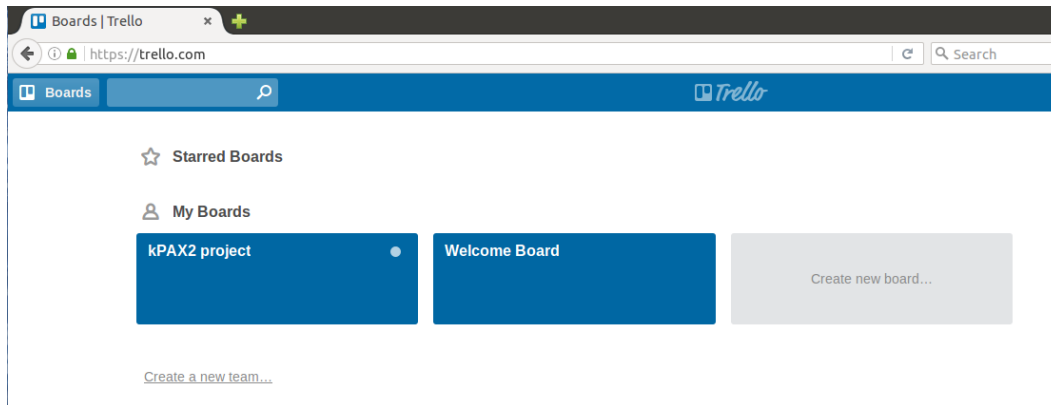


Figura 7 Trello

Después de entrar en el citado proyecto ya aparecen todas las fichas asociadas y el estado de cada una de ellas organizado por columnas o listas.

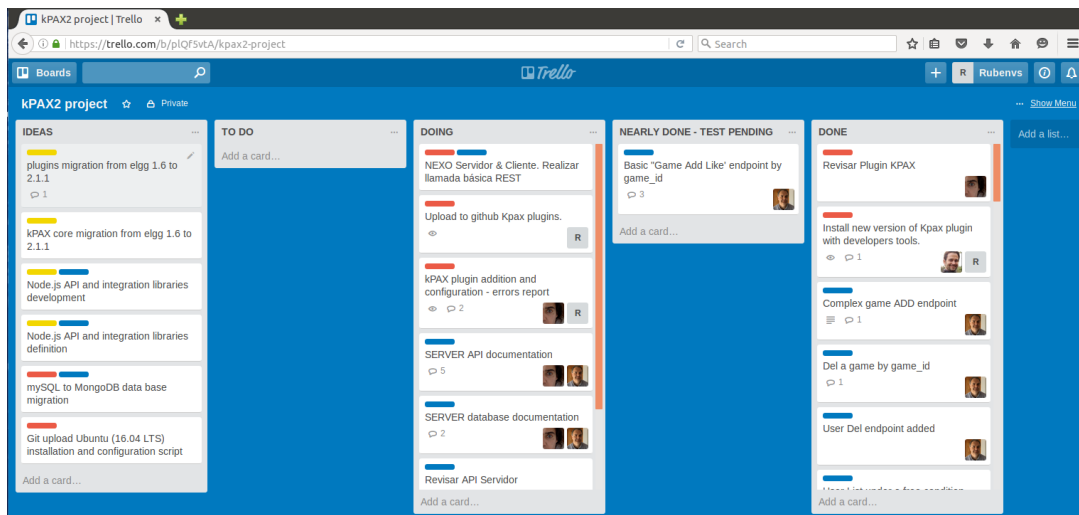


Figura 8 Página de trabajo de Trello

Para este proyecto se ha decidido crear diferentes listas. Se parte de una lista llamada “ideas” en la que se se van creando tantas tarjetas como tareas a realizar que se creen como posibles en el proyecto. Después hay otra columna donde se van indicando las diferentes tareas las cuales ya han sido confirmadas por el grupo para hacer. En la siguiente lista ya se detallan las tareas con las que está trabajando cada componente del grupo. Por último existe la columna realizado donde se van dejando las diferentes tarjetas en las que la tarea asociada ha finalizado.

La estructura de una tarjeta es bastante sencilla, pero a la vez muy potente ya que permite reflejar, con mucha información, el estado de una tarea.

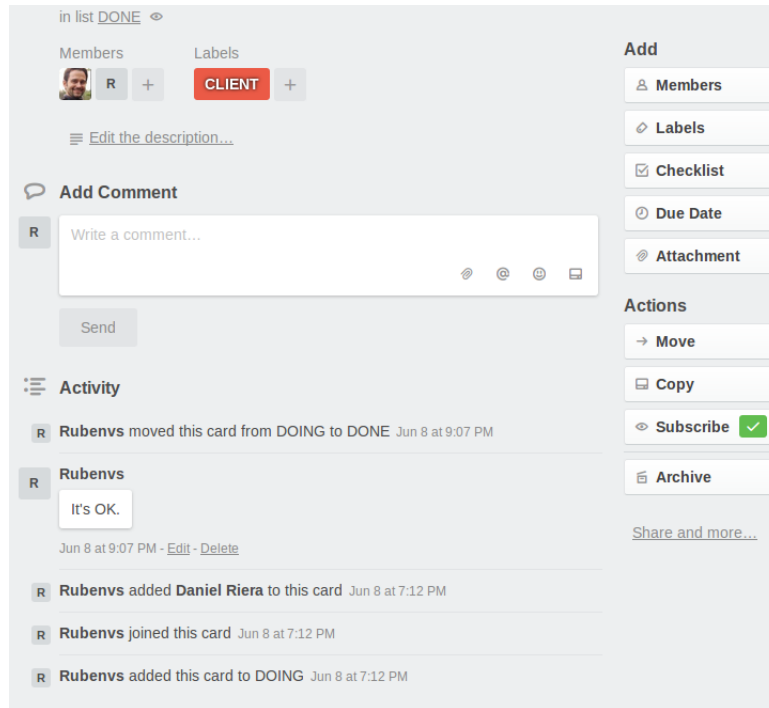


Figura 9 Tarjeta Trello

En este proyecto se van a manejar diferentes versiones de código, por lo que es necesario tener un repositorio con todo ello que permita el trabajo en grupo [18]. Para ello se va a utilizar una herramienta llamada Github. Se permite trabajar en local, para posteriormente subirlo a la web al repositorio del grupo.



Figura 10 Github

5.2.Desarrollo

En este apartado se va a detallar la forma de implementar los diferentes subsistemas hasta llegar al último en el que se dejará operativa la aplicación *Kpax*.

Subsistema “sistema operativo”.

Se crea una máquina virtual en Virtual Box con 1 procesador, 2 GB's de RAM y un disco duro de 25 GB's de almacenamiento variable.

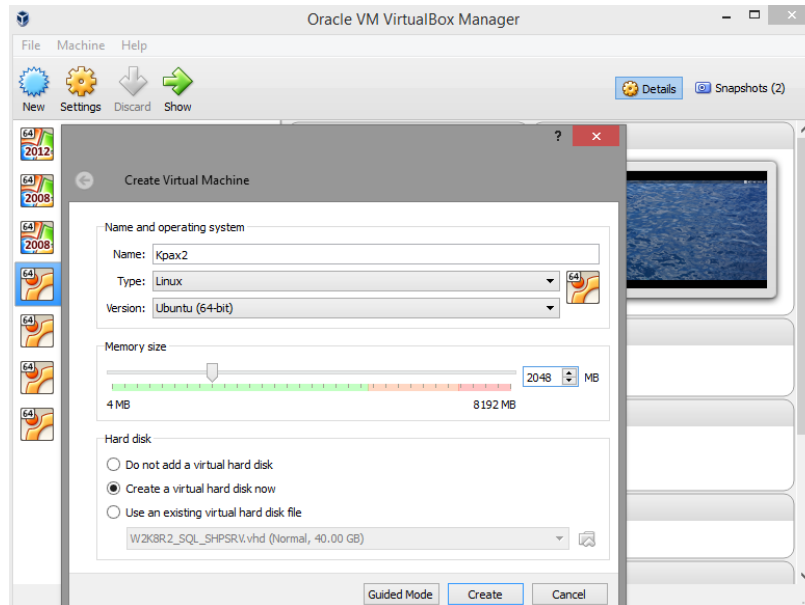


Figura 11 Creación máquina virtual

Una vez creada se indicará la ISO de Ubuntu Linux y se instalará.

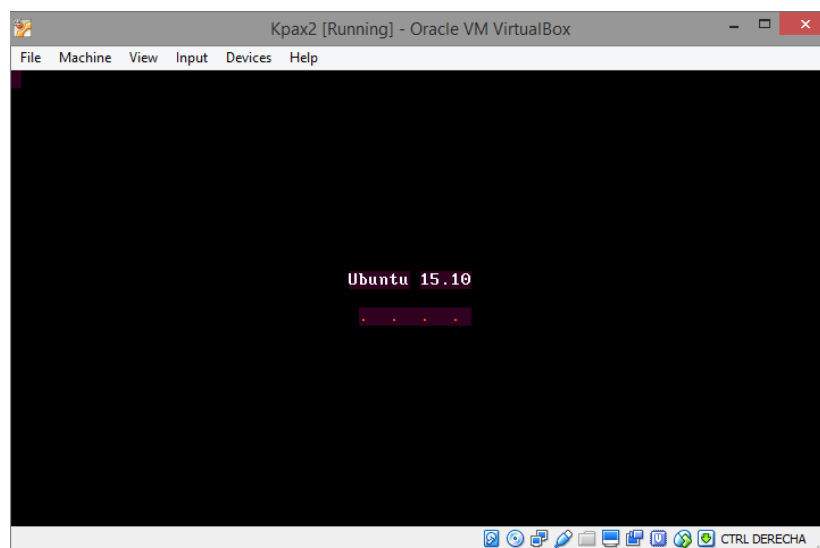


Figura 12 Instalación Ubuntu Linux

Se configurará el sistema en inglés y se contestará a las preguntas que va haciendo. Una vez finalizado y reiniciado es necesario instalar la VBoxAdditions para la correcta comunicación entre la máquina virtual y el host que la aloja.

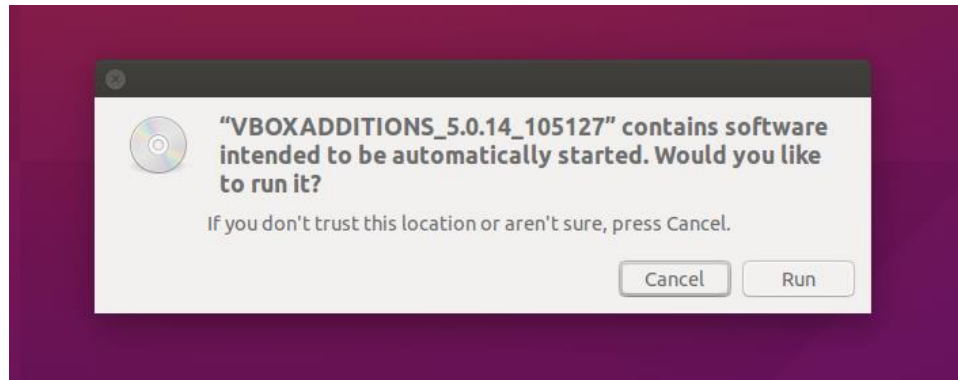


Figura 13 Instalación *VBoxAdditions*

Una vez realizado se reinicia el sistema de nuevo y se comprueba la conectividad a red, sobre todo a Internet ya que es un requerimiento para poder continuar con la actualización del sistema a partir del script y poder pasar al siguiente subsistema.

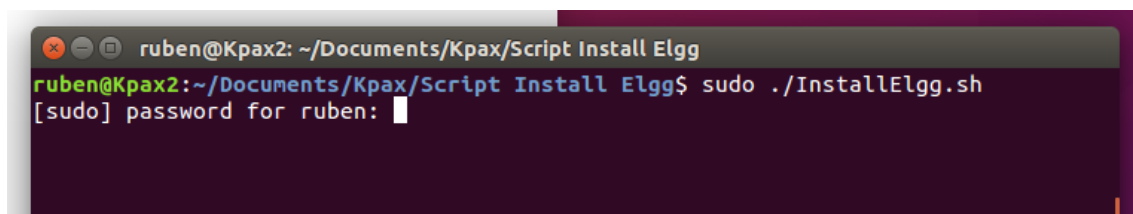


Figura 14 Ejecución script de instalación

A continuación se muestra los comandos utilizados en la actualización del sistema.

```
# Recomendado total system update & upgrade
clear
echo "System Update"
apt-get update
read -rsp $'Press any key to continue...\n' -n1
clear
echo "System Upgrade"
apt-get upgrade
read -rsp $'Press any key to continue...\n' -n1
```

Subsistema “requerimientos”.

Corresponde a la instalación básica del servidor web Apache, el gestor de base de datos en local MySQL y el intérprete de PHP. Hay que indicar que durante la instalación de MySQL pedirá contraseña para el usuario administrador.

```
# Apache
clear
echo "About to Install Apache"
apt-get install apache2
read -rsp $'Press any key to continue...\n' -n1
|
# MySQL
clear
echo "About to Install MySQL"
apt-get install mysql-server
read -rsp $'Press any key to continue...\n' -n1

# PHP
clear
echo "About to Install PHP"
apt-get install php5 libapache2-mod-php5 php5-mysqlnd php5-gd
read -rsp $'Press any key to continue...\n' -n1
```

Para la configuración de Apache, por un lado el permitir la utilización de la disposición indicada en el fichero .htaccess (requerimiento de *EIgg*) y por otro la utilización de URL's “amigables” por parte del módulo rewrite.

```
# Config /etc/apache2/apache2.conf
# 'AllowOverride None' with 'AllowOverride All'
clear
echo "About to configure Apache"
sed -i 's/AllowOverride None/AllowOverride All/g' /etc/apache2/
apache2.conf

# Configure Apache rewrite
a2enmod rewrite
service apache2 restart
read -rsp $'Press any key to continue...\n' -n1
```

A continuación se configurará MySQL y se crearán varias tablas para *Elgg*. Se pedirá la contraseña del usuario administrador por lo que habrá que facilitarla varias veces.

Create MySQL Database & tables for Elgg

```
clear
```

```
echo "About to Create MySQL Database & Tables for Elgg"
```

```
mysql -u root -p -e "CREATE DATABASE elggDB;CREATE USER elgguser  
IDENTIFIED BY 'elggpassword';GRANT ALL ON elggDB.* TO elgguser;"
```

```
mysql -u root -p elggDB -e \
```

```
"CREATE TABLE elggDB_access_collection_membership (user_guid int(11)  
NOT NULL,access_collection_id int(11) NOT NULL, PRIMARY KEY  
(user_guid,access_collection_id)) ENGINE=MyISAM DEFAULT CHARSET=utf8;  
CREATE TABLE elggDB_access_collections (id int(11) NOT NULL  
AUTO_INCREMENT,name text NOT NULL,owner_guid bigint(20) unsigned NOT  
NULL,site_guid bigint(20) unsigned NOT NULL DEFAULT '0' BDTMADV KEY
```

A partir de este momento ya estará finalizado este subsistema. Para poder pasar al siguiente, se tiene que haber confirmado que todo lo anterior se ha instalado y configurado correctamente y no se han producido errores.

Subsistema “motor de red social”.

Se sigue con la instalación a través del script, aunque la parte final se hace manualmente. Se inicia con la descarga, directamente de la web de *Elgg* y la descompresión en un directorio.

```
# Downloading Elgg and unzip
clear
echo "Downloading Elgg and unzip"
cd /var/www/
wget http://elgg.org/getelgg.php?forward=elgg-2.1.1.zip -O elgg.zip > /dev/null
unzip elgg.zip -d /var/www/html > /dev/null
rm elgg.zip
read -rsp $'Press any key to continue...\n' -n1
```

A continuación se configuran sus parámetros básicos, como la creación del directorio de trabajo para *Elgg*, un enlace del directorio a *Kpax2* y el cambio de unos parámetros en el fichero *settings.php*.

```
# Setting data directory and writeable by the webserver (www-data = Apache user).
echo "Setting data directory and writeable by the webserver"
mkdir -p /var/elggdata
chown www-data:www-data /var/elggdata

# Make a link to Kpax2
echo "Make a link to Kpax2"
ln -sf /var/www/html/elgg-2.1.1 /var/www/html/kpax2

# Configure settings.php
echo "About to configure settings.php"
cp /var/www/html/elgg-2.1.1/vendor/elgg/elgg/elgg-config/settings.example.php /var/www/html/kpax2/elgg-config/settings.php
cd /var/www/html/kpax2/elgg-config/
sed -i 's/{{timezone}}/Europe\Amsterdam/g' settings.php
sed -i 's/{{dbuser}}/elgguser/g' settings.php
sed -i 's/{{dbpassword}}/elggpassword/g' settings.php
sed -i 's/{{dbname}}/elggDB/g' settings.php
sed -i 's/{{dbhost}}/localhost/g' settings.php
sed -i 's/{{dbprefix}}/elggDB_/g' settings.php
read -rsp $'Press any key to continue...\n' -n1
```


Si todo ha sido correcto aparecerá una indicación a la página de *Elgg* para seguir con la configuración (manual).

```
ruben@Kpax2: ~/Documents/Kpax/Script Install Elgg
Navigate to http://localhost/kpax2/install.php to install
Follow the installation wizard
Press any key to continue...
```

Figura 15 Finalización script de instalación

Entrando en la página se configurará el nombre del sitio, la dirección de correo electrónico del administrador, el directorio de trabajo y el acceso por defecto.

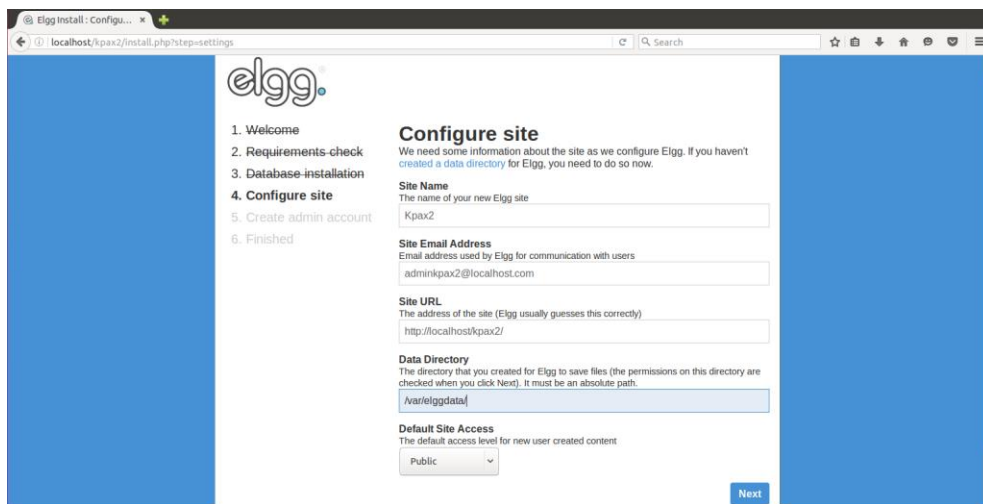


Figura 16 Configuración *Elgg*

A continuación pedirá datos para el administrador.

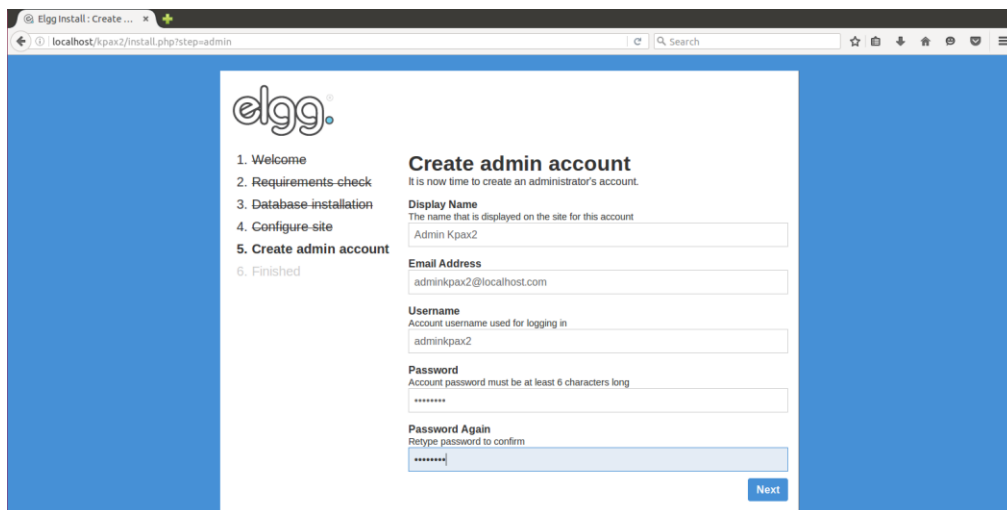


Figura 17 Configuración cuenta administrador *Elgg*

Por lo que sí es todo correcto, se podrá ir al sitio y a la administración del mismo.

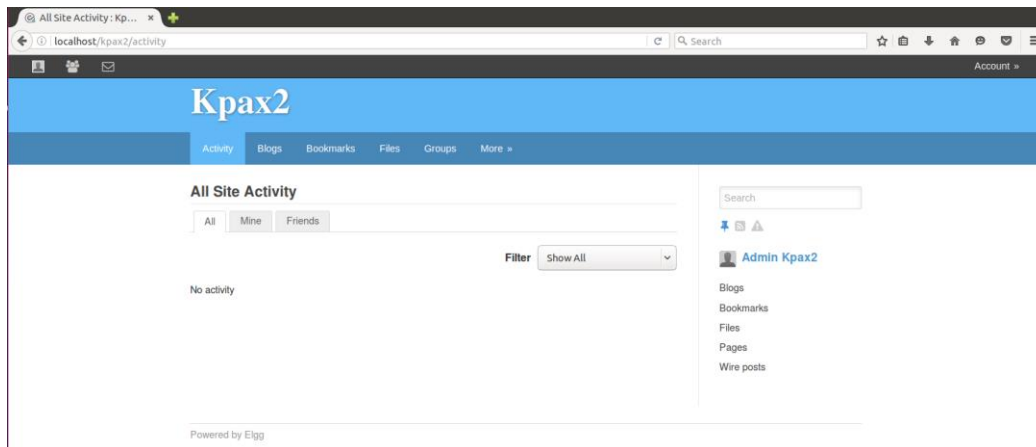


Figura 18 Página inicial Kpax 2

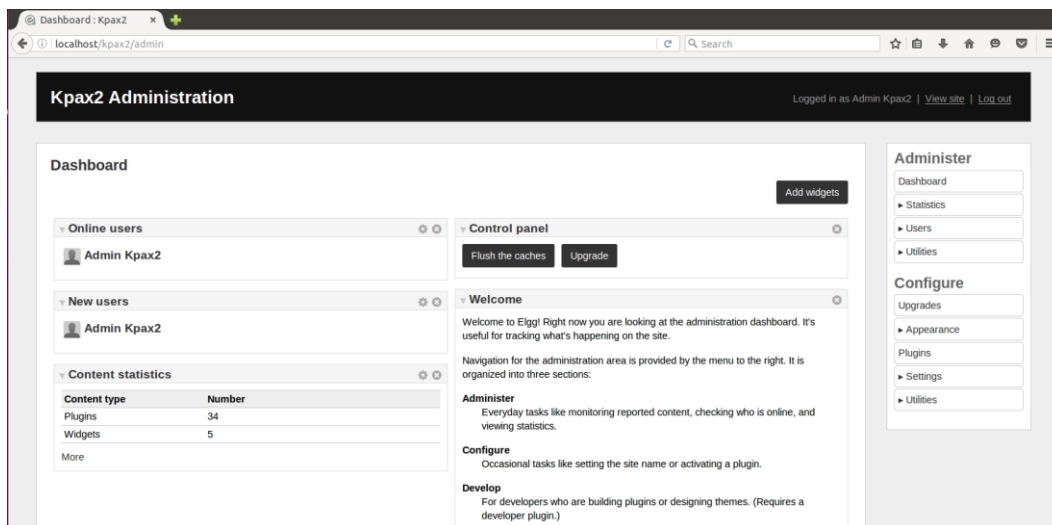


Figura 19 Panel de administración Kpax

En este punto ya se puede dar finalizado la instalación y configuración de este subsistema, tanto con el script como con la configuración manual de *Elgg*. Se puede considerar el entorno preparado para iniciar la implementación de *Kpax 2* a través de sus diferentes módulos en el último subsistema.

Subsistema “módulos”.

A partir de este momento y si todo lo anterior se ha realizado correctamente se puede dar paso a la instalación de la aplicación *Kpax* a través de sus diferentes módulos.

Se establece un proceso, el cual se va seguir en cada uno de ellos, realizándose de la siguiente forma:

- Se tiene que conseguir el módulo a implantar, para ello se descarga de la web correspondiente o se pide a los responsables de la aplicación.
- Una vez conseguido se copia en un directorio y se descomprime si es necesario.
- Se tiene que revisar la estructura de directorios del módulo, para confirmar que cumple con los requerimientos de *Elgg*.
- El fichero *manifest.xml* tiene que tener el formato y estructura adecuadas.
- Tiene que existir en la raíz de la estructura de carpetas del módulo, al menos un fichero llamado *start.php* con las instrucciones básicas a ejecutar.
- A continuación se tiene que copiar todo el directorio del módulo en */var/www/html/kpax2/mod/...*
- El siguiente paso es verificar que en la página de administración de los módulos, aparece el citado elemento, permitiendo que se active, en caso de ser así se hará.
- Llegado este momento queda únicamente confirmar que se ejecuta correctamente y se observa el resultado esperado.
- Por último se documenta el proceso de modificación, configuración e instalación del módulo.

Tal y como se ha comentado en diferentes apartados de este documento los módulos a instalar son los siguientes:

- *Kpax*; módulo principal de la aplicación desarrollado por la comunidad universitaria.
- *Theme_Kpax*; módulo que permite cambiar el tema utilizado por defecto en *Elgg* a uno propio para *Kpax*. Desarrollado por la comunidad universitaria para *Kpax*.
- *Logrequired*; módulo que permite deshabilitar ciertas opciones de la página principal, antes de hacer el inicio de sesión. Está desarrollado por la comunidad *Elgg*.
- *Apiadmin*; módulo que permite gestionar las claves “API” utilizadas por la aplicación. Desarrollado por la comunidad *Elgg*.
- *HTML5*; módulo que permite utilizar la funcionalidad html 5. Desarrollado por la comunidad *Elgg*.

A continuación se describe el proceso llevado por cada uno de los módulos, incluyendo también aquellos que son necesarios o complementarios.

El primero que se recomienda en activar y configurar es el módulo “Elgg Developers Tools. Se utiliza para mostrar por pantalla los problemas que van sucediendo con los diferentes módulos en todo el proceso de desarrollo, cuando ya no sea necesario se puede volver a desactivar. Para tenerlo presente se coloca al principio de la lista.

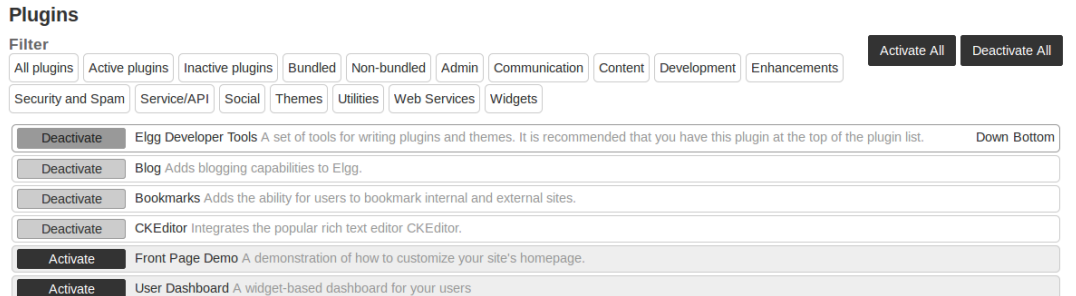


Figura 20 Módulo Elgg Developers Tools

Su configuración es bastante sencilla, para ello ir a la opción “Develop - Settings”, deshabilitar los diferentes caches, que se visualicen errores fatales de PHP y en nivel de traza en advertencia.

Developers : Settings

Control your development and debugging settings below. Some of these settings are also available on other admin pages.

Use simple cache

Turn off this cache when developing. Otherwise, changes to your CSS and JavaScript will be ignored.

Use system cache

Turn this off when developing. Otherwise, changes in your plugins will not be registered.

Display fatal PHP errors

By default, Elgg's .htaccess file suppresses the display of fatal errors.

Trace level

This controls the amount of information logged. See `elgg_log()` for more information.

Log to the screen

This displays `elgg_log()` and `elgg_dump()` output and a DB query count.

Show raw translation strings

This displays the translation strings used by `elgg_cache()`.

Figura 21 Configuración Elgg Developers Tools

Aceptar la configuración y el sistema ya estará preparado para montar módulos y ver los errores en caso de que se produzcan.

Subsistema “módulos” – Módulo Apiadmin.

El citado módulo se descarga de Github en su versión 1.0 [19].

Se verifica que tanto *manifest.xml* como *Start.php* son correctos. Se copia al directorio /Mod/... Desde la página de administración de módulos, aparece como que *Elgg* lo considera correcto por lo que se mueve al principio de la lista y se arranca.

Al probar el módulo, desde la opción que se crea en el menú de administración, se obtiene un error, indicando que se está llamando a una función no definida.



Figura 22 Error Apiadmin

Para resolver la incidencia hay que activar el módulo “*Web Services*” que viene incluido con *Elgg*. En versiones anteriores a la 1.8 de *Elgg*, venía incorporado en el núcleo, a partir de la 1.9, se decidió facilitarlo a parte para no tener activas características como las claves API. Se activa y se coloca al principio de la lista.

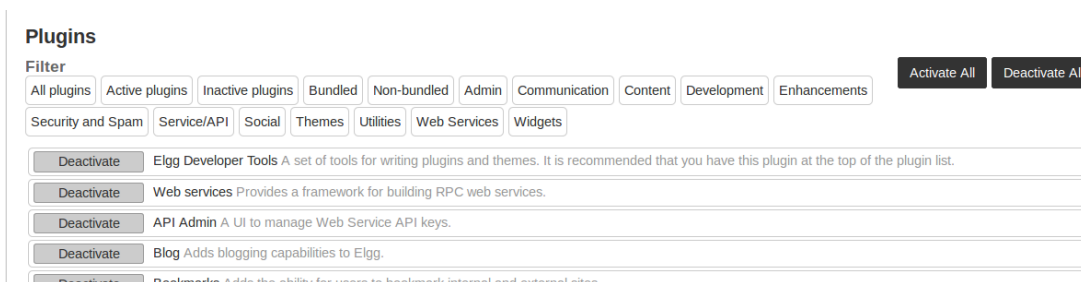


Figura 23 Módulo web services

En este momento ya se pondrá generar claves API tal y como se ve en la imagen.

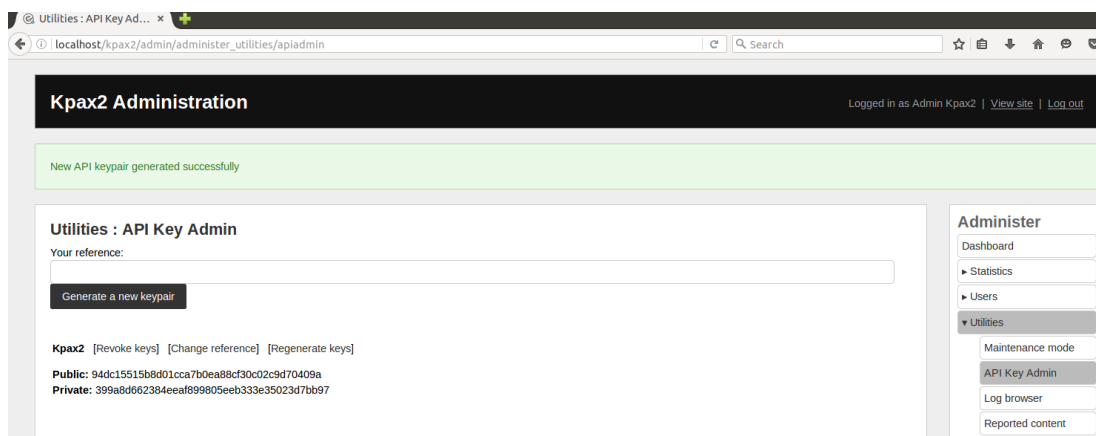


Figura 24 Página principal módulo Apiadmin

Hay que indicar que aunque aparezcan opciones para revocar, regenerar o eliminar claves, estas no vienen implementadas en el módulo.

Subsistema “módulos” – Módulo Loginrequired.

Se descarga de Github en su versión 1.9.8 [20] con una estructura correcta.

Se verifica que tanto *manifest.xml* como *Start.php* son válidos.

Se copia al directorio */mod...*

Desde la página de administración de módulos, aparece como que *Elgg* lo considera correcto por lo que se mueve al principio de la lista, se arranca y se prueba.

Sin la ejecución de módulo.

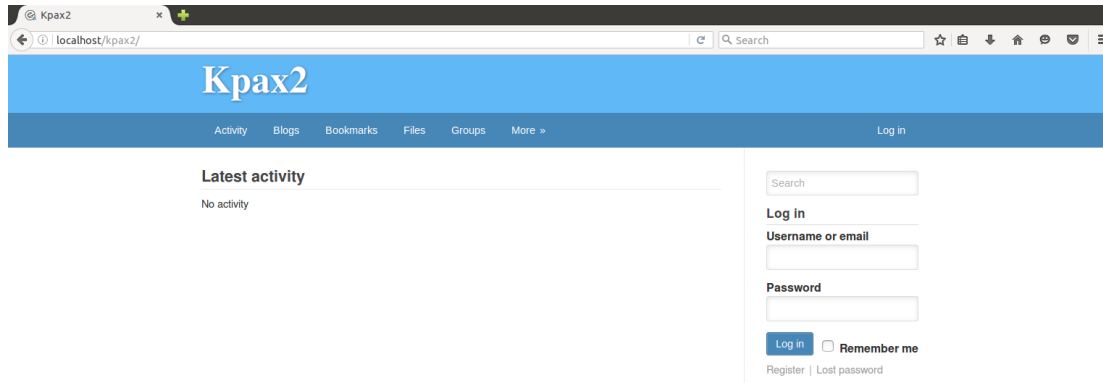


Figura 25 Kpax sin Loginrequired

Con la ejecución del módulo.

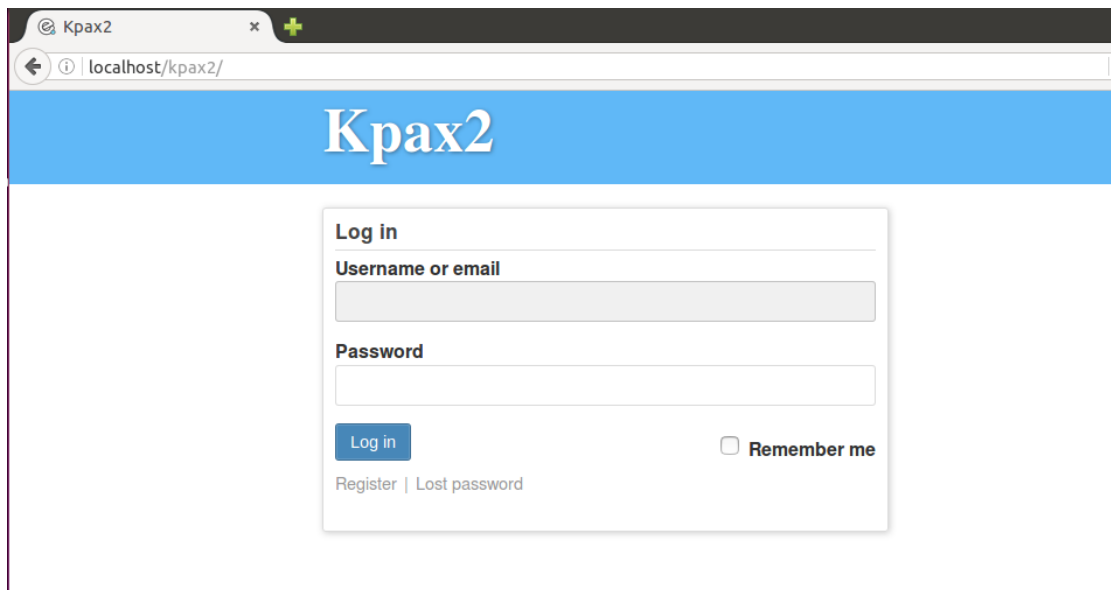


Figura 26 Kpax con Loginrequired

Subsistema “módulos” – Módulo Html5.

Este módulo es facilitado por los responsables del proyecto.

Su estructura a nivel de carpetas es correcta

El fichero *start.php* es correcto, pero se tiene que cambiar en el fichero *manifest.xml* "elgg_version" por "elgg_release" y también indicar una versión válida.



```
<?xml version="1.0" encoding="UTF-8"?>
<plugin_manifest xmlns="http://www.elgg.org/plugin_manifest/1.8">
  <name>HTML5</name>
  <author>Evan Winslow</author>
  <version>1.0</version>
  <description>
    Provides several javascript libraries + views that a
  </description>
  <copyright>(C) Evan Winslow 2010</copyright>
  <website>http://github.com/ewinslow/elgg-html5</website>
  <requires>
    <type>elgg_release</type>
    <version>1.7</version>
  </requires>
</plugin_manifest>
```

Figura 27 *manifest.xml* del módulo *Html5*

Se copia a la carpeta */mod* y se activa; no se observa ningún funcionamiento anómalo. Tampoco se pueden hacer pruebas ya que contiene librerías principalmente a usar por otros programas.

Se deja colocado al principio junto al resto de módulos.

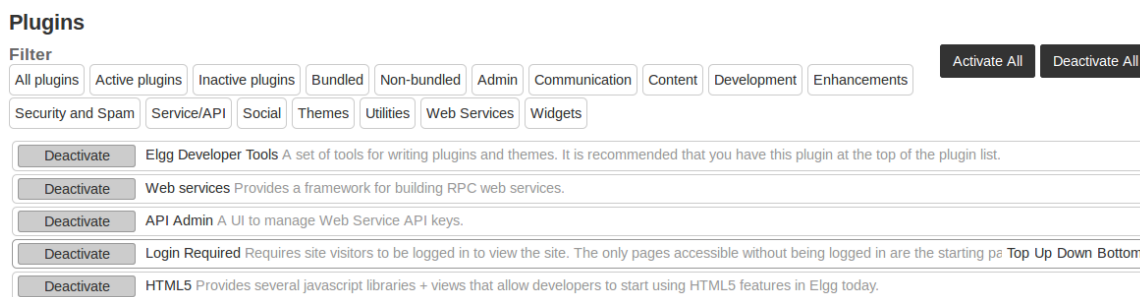


Figura 28 Módulo *Loginrequired*

Subsistema “módulos” – Módulo *Kpax* (ficha *Develop* incluida).

Este módulo es el principal de todo el proyecto ya que es el que contiene la propia aplicación *Kpax*, que con sus modificaciones pasará a ser *Kpax 2*. Se incluye en el mismo, una característica llamada “*Develop*”. Este es facilitado por los responsables del proyecto.

Lo primero que se observa que su estructura de directorios es correcta, pero el fichero *manifest.xml* no, por lo que hay que modificarlo. Si no se hace aparecerá un error como el que se muestra en la siguiente figura.



Figura 29 Error instalación módulo *Kpax*

Para solucionarlo se cambia el fichero tal y como se indica, para cumplir con los requerimientos de *Elgg*.

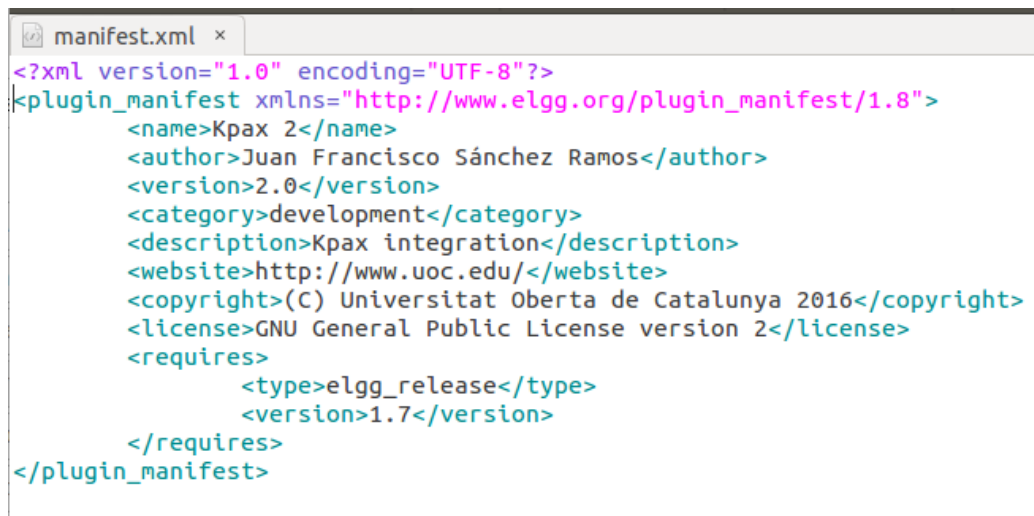


Figura 30 Modificación *manifest.xml* *Kpax*

El fichero *start.php* también es correcto pero habrá que hacer las modificaciones que se comentan un poco más adelante.

En este momento ya se podrá activar, pero antes de hacerlo, habrá que generar una nueva clave API para el servicio por lo que hay que modificar el fichero `/var/www/html/kpax2/mod/kpax/lib/kpaxSrv.php`

Utilities : API Key Admin

Your reference:

Generate a new keypair

Kpax2 [Revoke keys] [Change reference] [Regenerate keys]

Public: 94dc15515b8d01cca7b0ea88cf30c02c9d70409a

Private: 399a8d662384eeaf899805eeb333e35023d7bb97

Figura 31 Generación API para *Kpax*

A continuación se asigna a la variable “\$apikey” el valor con la clave pública previamente generado.

```
kpaxSrv.php x
<?php
/**
 * Class get information to service.
 *
 * @author juanfraser
 */
class kpaxSrv {
    protected $url = "http://localhost:8080/webapps/avrKpax/";
    private $key;
    private $apiKey = "94dc15515b8d01cca7b0ea88cf30c02c9d70409a";
    //private $apiKey = "e4afd792d8730dded98e67ac6e9752bd35e764bc"; // Public API key generated by elgg
    private $oauthKpax = null;
}
```

Figura 32 Modificación *kpaxSrv.php*

En este momento ya se podrá activar el módulo *Kpax*, aunque seguirán apareciendo problemas que corregir.

The web services are now a plugin in Elgg 1.9. You must enable this plugin and update your web services to use `elgg_ws_expose_function()`.

Figura 33 Error en *start.php* de módulo *Kpax*

Para solucionarlo hay que hacer unas modificaciones en el fichero *start.php* que consisten en cambiar

`“expose_function('some', 'parameters'....)”`

por

`“elgg_ws_expose_function('some','parameters',....)”`

```
elgg_ws_expose_function("user.auth", "auth_user", array("username" => array('type' => 'String', 'required' => true), "password" => array('type' => 'String', 'required' => true)), 'Auth user elgg', 'GET', true, false);

function auth_sign($username="") {
    $credentials = array('username' => $username);
    $user = get_user_by_username($credentials['username']);
    if (empty($user)) {
        return "OK";
    } else {
        $user = get_user_by_username("uoc.edu_.". $credentials['username']);
        if(empty ($user))return "OK";
        return "ERROR";
    }
}

elgg_ws_expose_function("auth.sign", "auth_sign", array("username" => array('type' => 'String', 'required' => true)), 'Auth sign elgg', 'GET', true, false);
```

Figura 34 Fichero *start.php* de *Kpax* modificado

Ya se puede activar el módulo y observar el resultado.

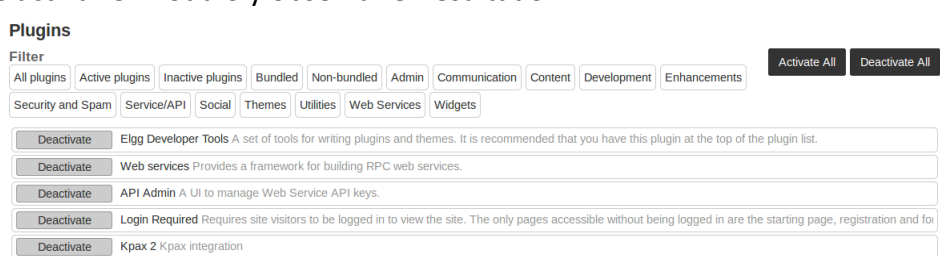


Figura 35 Módulo *Kpax 2*

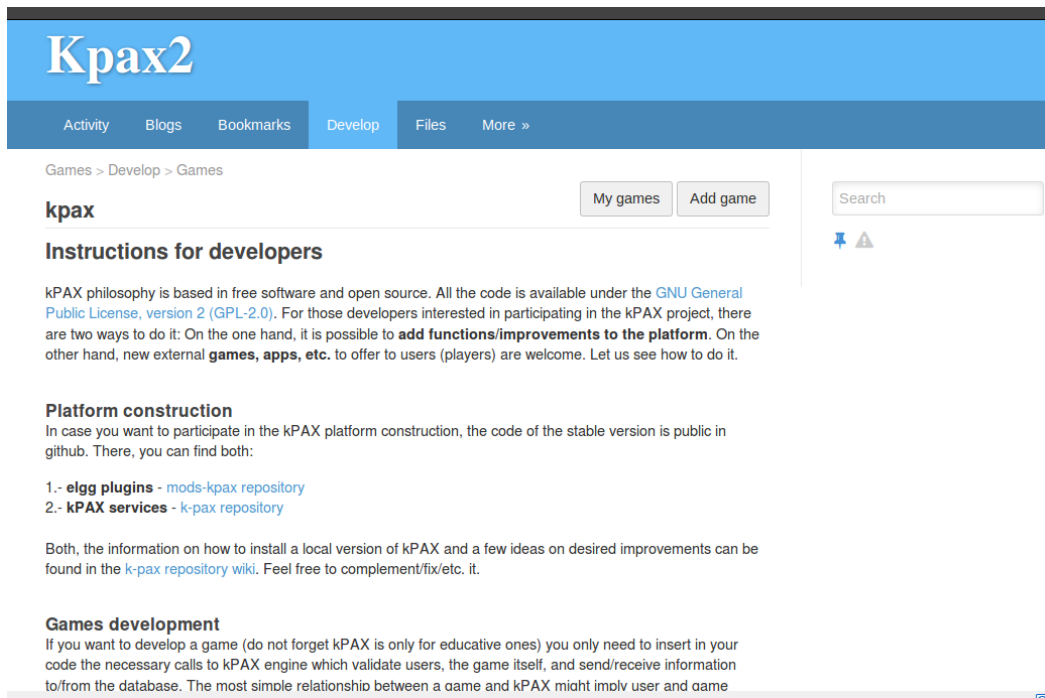


Figura 36 *Kpax 2* en ejecución

Subsistema “módulos” – Módulo *Theme_Kpax*.

Este es el último módulo a implementar en este proyecto, su estructura es correcta, pero se tiene que modificar en *manifest.xml* la sentencia "*elgg_version*" por "*elgg_release*".

A screenshot of a code editor with two tabs labeled 'manifest.xml'. The code in the editor is as follows:

```
This is the kPAX platform first own theme.
</description>
<website>http://kpax.uoc.edu/elgg</website>
<copyright>Copyright 2012 by UOC</copyright>
<license>GNU Public License version 2</license>

<requires>
  <type>elgg_release</type>
  <version>1.8</version>
</requires>
```

Figura 37 Modificación *manifest.xml* de *Theme_Kpax*

Este módulo hay que dejarlo al final de la lista ya que tiene dependencia de otros, no siendo así, no se podrá ejecutar.

Una vez copiado al directorio */mod* se puede comprobar cómo cambia el tema para la aplicación *Kpax*.

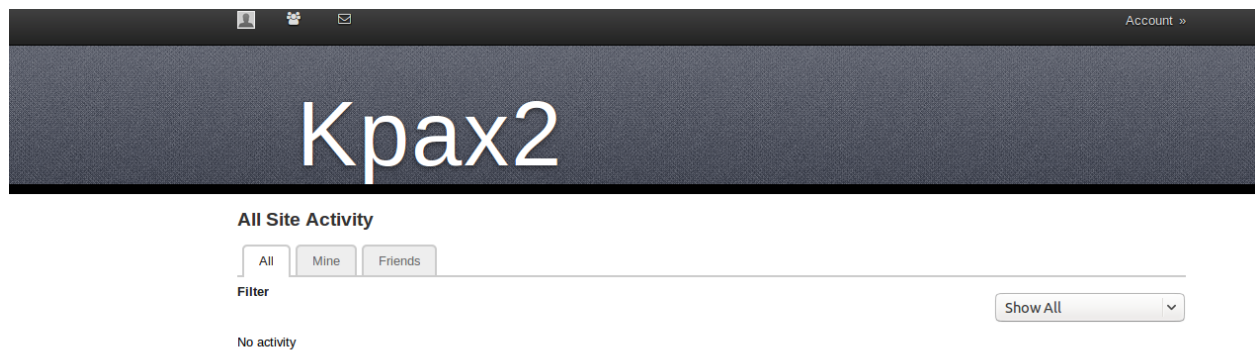


Figura 38 Página de *Kpax* con su propio tema

Conclusiones

En este proyecto se ha podido comenzar con la migración de la red social *Kpax* a *Elgg* 2.1.1, a través de la adaptación básica de su módulo principal más otros necesarios. Se puede considerar que todos sus objetivos se han cumplido y se deja una plataforma preparada para futuros proyectos, unido ello la posibilidad de tener el proceso automatizado de instalación en casi al 100%, lo cual ayudará enormemente en la creación de entornos de prueba de una forma sencilla y rápida.

Para poder llevarlo a cabo se han tenido que repasar diferentes elementos vistos a lo largo del máster y otros complementarios como los indicados a continuación.

- Metodología de tipo ágil, para el desarrollo del proyecto en conjunción con la herramienta web Trello y reuniones semanales con los integrantes del proyecto.
- Herramienta para el versionado del código Github.
- Estudio del motor de red social *Elgg* 2.1.1.
- Estudio de la aplicación *Kpax*, su arquitectura y módulos relacionados.

Una vez montada la plataforma, se pasó a la configuración de los diferentes módulos, siendo ello la parte importante del proyecto. Todo el proceso se hizo ordenado, probando diferentes versiones y modificándolas hasta encontrar la adecuada. Se ha tenido que trabajar en cada uno de ellos, cambiando a nivel de código lo necesario para la correspondiente adaptación. El que ha resultado de especial interés ha sido el propio de *Kpax* ya que por sí sólo reunía todos los requerimientos para la buena marcha del plan. Hay que tener en cuenta que desde la versión 1.6 hasta la 2.1.1 del *Elgg*, ha habido bastantes cambios entre medias por lo que el principal temor era que no se pudiesen modificar sin requerir muchas modificaciones y muchas horas de trabajo.

Finalmente y con todo el sistema implementado, se han podido realizar y confirmar pruebas mediante llamadas REST, obteniendo los resultados esperados.

A título personal ha sido un proyecto de gran utilidad, unido todo ello al propio contenido del máster. La forma de trabajo, dentro del grupo basado en la metodología ágil, me ha resultado totalmente adecuada, además de abrir la posibilidad de utilizarla en mis tareas profesionales.

Tanto la aplicación *Elgg* como la propia *Kpax*, son ejemplos claros de la fuerza que tienen hoy en días las redes sociales, considero que me han aportado un conocimiento mayor de este tipo de herramientas, además de las posibilidades que existen al explotarla en el marco de las licencias libres.

Bibliografía

- [1] D. Riera, kPAX. Plataforma d'aprenentatge en xarxa. Juga seriosament, Barcelona: Universitat Oberta de Catalunya (UOC), 2011.
- [2] «Apache» [En línea]. Disponible en: <https://httpd.apache.org/>.
- [3] «MySQL» [En línea]. Disponible en: <https://www.mysql.com/>.
- [4] «PHP» [En línea]. Disponible en: <http://php.net/>.
- [5] «Elgg» [En línea]. Disponible en: <https://elgg.org/>.
- [6] «Módulos Elgg» [En línea]. Disponible en: <https://elgg.org/plugins>.
- [7] «Módulos Kpax 1.0» [En línea]. Disponible en: <https://github.com/jsanchezramos/mods-kpax>.
- [8] «REST Api» [En línea]. Disponible en: https://es.wikipedia.org/wiki/Representational_State_Transfer.
- [9] «JSON» [En línea]. Disponible en: <https://en.wikipedia.org/wiki/JSON>.
- [10] «MongoDB» [En línea]. Disponible en: <https://www.mongodb.com/>.
- [11] «Github» [En línea]. Disponible en: <http://github.com/>.
- [12] «Estructura Módulo Elgg» [En línea]. Disponible en: <http://learn.elgg.org/es/stable/guides/plugins/plugin-skeleton.html>.
- [13] «Juegos Serios» [En línea]. Disponible en: https://en.wikipedia.org/wiki/Serious_game.
- [14] «Free Software Foundation» [En línea]. Disponible en: <https://www.fsf.org/>.
- [15] «Instalación Elgg» [En línea]. Disponible en: <http://learn.elgg.org/en/2.0/intro/install.html>.
- [16] Martel Antonio, «Gestión Práctica de Proyectos con Scrum. Desarrollo Ágil.» Amazon, 2014.
- [17] «Trello» [En línea]. Disponible en: <http://trello.com>.
- [18] «Kpax 2 en Github» [En línea]. Disponible en: https://github.com/drierat/kPAX2_elgg_frontend_core.
- [19] «Apiadmin en Github» [En línea]. Disponible en: <https://github.com/Elgg/apiadmin>.
- [20] «Loginrequired en Github» [En línea]. Disponible en: <http://github.com/iionly/loginrequired>.