



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería  
Informática**

**UBUMonitor**

**Monitorización de alumnos  
en la plataforma Moodle.**



Presentado por Yi Peng Ji  
en Universidad de Burgos — 30 de junio  
de 2019

Tutor: Dr.D. Raúl Marticorena Sánchez  
y Dr.D. Carlos Pardo Aguilar



---

# Índice general

---

<b>Índice general</b>	<b>I</b>
<b>Índice de figuras</b>	<b>III</b>
<b>Índice de tablas</b>	<b>v</b>
<b>Apéndice A Plan de Proyecto Software</b>	<b>1</b>
A.1. Introducción . . . . .	1
A.2. Planificación temporal . . . . .	1
A.3. Estudio de viabilidad . . . . .	5
<b>Apéndice B Especificación de Requisitos</b>	<b>9</b>
B.1. Introducción . . . . .	9
B.2. Objetivos generales . . . . .	9
B.3. Catalogo de requisitos . . . . .	9
B.4. Especificación de requisitos . . . . .	11
<b>Apéndice C Especificación de diseño</b>	<b>13</b>
C.1. Introducción . . . . .	13
C.2. Diseño de datos . . . . .	13
C.3. Diseño procedimental . . . . .	14
C.4. Diseño arquitectónico . . . . .	18
C.5. Diseño de la interfaz . . . . .	20
<b>Apéndice D Documentación técnica de programación</b>	<b>21</b>
D.1. Introducción . . . . .	21
D.2. Estructura de directorios . . . . .	21

D.3. Manual del programador . . . . .	23
D.4. Compilación, instalación y ejecución del proyecto . . . . .	42
D.5. Pruebas del sistema . . . . .	49
<b>Apéndice E Documentación de usuario</b>	<b>51</b>
E.1. Introducción . . . . .	51
E.2. Requisitos de usuarios . . . . .	51
E.3. Instalación . . . . .	51
E.4. Manual del usuario . . . . .	51
E.5. Visualización de gráficas . . . . .	54
<b>Bibliografía</b>	<b>65</b>

---

# Índice de figuras

---

B.1. Diagrama de casos de uso. . . . .	11
C.1. Simulación de base de datos. . . . .	14
C.2. Diagrama de secuencia del inicio de sesión. . . . .	16
C.3. Diagrama de secuencia de recogida de datos de los servicios web. . . . .	17
C.4. Diagrama de secuencia de descarga de los registros del curso. . . . .	18
C.5. Diagrama de despliegue de la aplicación. . . . .	19
D.1. Elegir la descarga de JDK. . . . .	24
D.2. Descarga . . . . .	24
D.3. Descarga del paquete de Eclipse. . . . .	25
D.4. Pulsar el botón de descarga. . . . .	26
D.5. Menú <b>Help</b> y seleccionar <b>Install New Software...</b> . . . . .	26
D.6. Instalación de la herramienta. . . . .	27
D.7. Aceptar la licencia de Oracle BSD y descargar. . . . .	28
D.8. Habilitar servicios web. . . . .	29
D.9. Habilitar servicios web parte 2. . . . .	29
D.10. Acceso al listado de funciones del servicio web de Moodle. . . . .	30
D.11. Acceder a la Documentación API. . . . .	30
D.12. Listado de funciones del servicio web de Moodle. . . . .	31
D.13. Clasificación de los registros junto con sus delegados. . . . .	32
D.14. Un delegado UserAffectedCmd . . . . .	33
D.15. Extracto de Componentes y eventos.json . . . . .	35
D.16. Menú <b>Cell</b> y opción <b>Run All</b> . . . . .	36
D.17. Selección múltiple de los archivos CSV. . . . .	37
D.18. Seleccionar el archivo JSON. . . . .	37
D.19. Error producido al usar un CSV que no está en inglés. . . . .	38

D.20.Menú <b>Cell</b> y opción <b>Run All</b> . . . . .	39
D.21.Seleccionar el archivo JSON. . . . .	39
D.22.Celda que parará el flujo el ejecución. . . . .	40
D.23.Copiar los componentes que se encuentra en el Resource Bundle español. . . . .	40
D.24.Copiar los eventos que se encuentra en el Resource Bundle español. . . . .	41
D.25.Copiar los tipos de logs. . . . .	42
D.26.Menú File y opción Import... . . . .	43
D.27.Existing Projects into Workspace . . . . .	44
D.28.Selección del archivo ZIP e importar proyecto. . . . .	45
D.29.Exportar jar . . . . .	46
D.30.Seleccionar Runnable JAR file. . . . .	47
D.31.Configuración del Runnable JAR. . . . .	48
D.32.Ejecución de la aplicación. . . . .	49
E.1. Primera pantalla de inicio de sesión. . . . .	52
E.2. Pantalla de selección de curso . . . . .	53
E.3. Ventana emergente pidiendo la contraseña antigua. . . . .	54
E.4. Pantalla inicial. . . . .	55
E.5. Listado de usuario matriculados en el curso. . . . .	56
E.6. Listado de calificaciones. . . . .	57
E.7. Gráfico de líneas de las calificaciones. . . . .	58
E.8. Gráfico radar de las calificaciones. . . . .	59
E.9. Gráfico de BoxPlot general de las calificaciones. . . . .	59
E.10.Gráfico BoxPlot de grupo de las calificaciones. . . . .	60
E.11.Tabla de calificaciones de tres personas y las medias del grupo y general. . . . .	60
E.12.Opciones para las gráfica de registros. . . . .	61
E.13.Listado de componentes ordenados alfabéticamente. . . . .	62
E.14.Listado de componente y evento ordenados alfabéticamente por componente. . . . .	63
E.15.Gráfica de barras apiladas con dos participantes y tres compo- nentes agrupados por día de la semana. . . . .	64

---

# Índice de tablas

---

A.1. Sueldos anuales, mensuales y de 5 meses. . . . .	5
A.2. Dependencias de Java. . . . .	6
A.3. Dependencias de JavaScript. . . . .	7
D.1. Códigos nemotécnicos de los delegados. . . . .	34





## Apéndice A

---

# Plan de Proyecto Software

---

### A.1. Introducción

En este apartado se comentará la planificación temporal de proyecto junto con su viabilidad económica y legal.

### A.2. Planificación temporal

El desarrollo de la aplicación esta planteado usando Scrum como metodología de gestión del proyecto. Aunque cabe destacar que está adaptado para trabajar con una persona a diferencia de los 4-8 usuales y también con reuniones semanales en vez de diarias. A continuación se muestran las características que se han seguido de la filosofía ágil:

- Desarrollo iterativo o incremental mediante interacciones (*sprints*).
- La duración de los *sprints* han sido de uno a tres semanas y cada una de ellas contiene las tareas o *issues* realizadas.
- Reuniones semanales de más de una hora comentando el avance del desarrollo y presentando partes del producto final.
- También se planifica las siguientes tareas a realizar durante las reuniones.

### Sprint 0 (14/01/2019 - 08/02/2019)

Durante la primera semana se buscó reuniones con tres docentes de la Universidad (José Manuel Galán Ordax, Bruno Baruque Zanón y Raúl Marticorena Sánchez) sobre posibles proyectos que tuvieran en mente.

Después de hablar con los tres en esa semana, hubo varios de reflexión sobre la difícil elección del trabajo. Finalmente se elige el proyecto de Raúl Marticorena.

En la siguiente semana se prepara la primera reunión comentando en profundidad los detalles del trabajo y los objetivos finales que se espera.

En este *sprint* también se prepara las instalaciones de las herramientas de necesarias para el desarrollo. Se ha usado el manual del programador aportado por el autor de UBUGrades 2.0 como medida de apoyo.

### Sprint 1 (08/02/2019 - 15/02/2019)

Buscar la manera de guardar los datos extraídos del servicio web en ficheros locales cifrados usando la contraseña de acceso a Moodle como clave. Se decide usar el algoritmo de encriptación **Blowfish** [10] que permite claves de 32 hasta 448 bits. Cabe destacar que se han descartado los algoritmos de cifrado que usan tamaños específicos de claves. Un ejemplo de ellos es el algoritmo *Advanced Encryption Standard* (AES) [8] que necesita claves de cifrado de 128, 192 o 256 bits.

Se acuerda empezar de nuevo el trabajo de UBULogs al no funcionar para la nueva versión de Moodle de la Universidad usando una alternativa diferente.

También se arregla un error del anterior proyecto el cual trataba al nombre de usuario como si fuera un correo.

### Sprint 2 (15/02/2019 - 13/03/2019)

En este *sprint* se busca la forma de iniciar sesión con el usuario y contraseña de Moodle y así poder descargar los registros del curso. Inicialmente se intentó usar solo el paquete **java.net** [3] para realizar las conexiones al no ser una de dependencia externa. Sin embargo los resultados fueron infructuosos y finalmente se empleó Jsoup [4], una librería muy famosa de web scraping.

Además se decide modificar completamente las clases encargadas de recoger información de los servicios web de Moodle.

### **Sprint 3 (13/03/2019 - 21/03/2019)**

Los objetivos fueron arreglar los errores encontrados en el anterior *sprint* y cambiar el paquete model modificando las clases y añadiendo todos los tipos disponibles de actividades y recursos del curso.

### **Sprint 4 (21/03/2019 - 26/03/2019)**

Centrado exclusivamente en recoger los datos de los servicios web y transformarlo en objetos de Java. La información que nos interesa son los usuarios matriculados, los contenidos (actividades y recursos), las calificaciones, los roles y grupos del curso.

Todo esto se extrae de los JSON que devuelve el servicio web de Moodle, se han creado métodos que recogen los campos más útiles del JSON.

### **Sprint 5 (26/03/2019 - 03/04/2019)**

Buscado la forma de encontrar cómo actualizar los datos, tanto de las calificaciones como de los registros del curso.

En cuanto a los registros del curso se descubre una forma de descargar los datos por días a través de un parámetro de la URL. Los tiempos de actualización de datos baja considerablemente con este método al descargar registros parciales en vez de completos.

También se refactoriza códigos de anteriores *sprints* y **Carlos Pardo Aguilar** se incorpora como segundo tutor.

Se acuerda en qué tipo de gráfica debe mostrar los registros del curso, barras apiladas.

### **Sprint 6 (03/04/2019 - 26/04/2019)**

Los objetivos son arreglar los errores de programación encontrados. También aparece un bug de los servicios web con la función **gradereport\_user\_get\_grade\_items** que devuelve un mensaje de error si la **Retroalimentación** está oculta en el calificador.

Después de varias propuestas se llega el consenso de usar la otra función del servicio web **gradereport\_user\_get\_grades\_table** mucho más incómoda de trabajar. Este inconveniente ha supuesto un retraso de una semana en el proyecto.

Se añadió un fichero de configuración para guardar los nombres de usuario y el servidor de Moodle en la pantalla de inicio de sesión.

### **Sprint 7 (26/04/2019 - 11/05/2019)**

Durante este proceso se implementó la agrupación de los registros por usuarios, componente y distintos tipos de tiempo (días, meses, años, etc) y finalmente contándolos. Se calcula también la media de todos los participantes según el número de registros.

Con todos estos datos se crean métodos que transforma en formato específico de Chart.js.

### **Sprint 8 (11/05/2019 - 18/05/2019)**

Durante este *sprint* se ha buscado una manera de añadir fácilmente nuevos tipos de registros según el componente y evento. Finalmente se ha creado varias herramientas en Python que busca los diferentes tipos que hay a partir de los registros en CSV y los guarda en un fichero JSON. Con este fichero se ha creado otra herramienta para generar las enumeraciones Java de los componentes y eventos junto con las traducciones. Todo esto se comenta con más detalles en el manual del programador (D.3).

### **Sprint 9 (18/05/2019 - 23/05/2019)**

Los objetivos fueron empezar la documentación de la memoria y probar si se puede añadir la opción de agrupar por componente y evento en los registros. Para esto se creó una nueva rama con resultados satisfactorios.

También se añadieron las fotografías de los participantes del curso.

### **Sprint 10 (23/05/2019 - 08/06/2019)**

En este *sprint* se añadieron varias mejoras de interacción con el usuario:

- Valores predefinido en el campo *host* de la pantalla de inicio.
- Mostrar la categoría del curso al en la pantalla de selección del curso.
- Añadir una opción de ajustar el valor máximo del eje Y en el gráfico de barras.

**Sprint 11 (08/06/2019 - 22/06/2019)**

Los objetivos fueron completar la memoria, documentación de los métodos de Java, arreglar varios errores y refactorización del código arreglando la gran mayoría de *code smells* detectados por Sonarqube y Codacy.

**Sprint 12 (22/06/2019 - 30/06/2019)**

Completar los apartados restantes del anexo y su revisión final.

**A.3. Estudio de viabilidad****Viabilidad económica**

Desarrollado por una sola persona durante 5 meses. Calculando usando salarios medios brutos de 19.485€ para un programador junior [9]. Empleando la calculadora de sueldos netos de Cinco Días [5] nos da los resultados en costes anuales. Calculamos también el sueldo mensual y el coste en 5 meses:

Concepto	Coste anual	Coste mensual	Coste 5 meses
Sueldo bruto	19.485,00 €	1.623,75 €	8.118,75 €
Retenciones por IRPF	2.221,30 €	185,11 €	925,54 €
Cuotas a la Seg. Social	1.237,30 €	103,11 €	515,54 €
<b>Sueldo neto</b>	<b>16.026,40 €</b>	<b>1.335,53 €</b>	<b>6.677,67 €</b>

Tabla A.1: Sueldos anuales, mensuales y de 5 meses.

En principio el producto no está pensado venderlo, la intención es que sea de código libre y gratuito. Los potenciales clientes serían las Universidades que usen Moodle, por lo que si podríamos cobrar por crear versiones específicas de la aplicación para cada una de las 80 Universidades de España [11]. Al tener la aplicación una versión internacional se podría incluso usarse en Universidades extranjeras.

Vendiendo licencias de la aplicación 50€ a unas 150 Universidades cubriría todos los costos del desarrollo.

Otra forma de ingresos sería a partir de donaciones de la comunidad.

## Viabilidad legal

Se mantiene la licencia de MIT al ser una de las más permisivas para productos de código libre. Permite a la gente hacer casi todo lo que quieran con el proyecto, incluyendo crear y distribuir versiones de código cerrado.

En el desarrollo del proyecto se han empleado varias librerías externas, algunas de ellas nuevas y otras que estaban en el anterior proyecto se han quitado en este desarrollo.

En definitiva, se han usado varias librerías de Java, JavaScript y un paquete de iconos.

## Dependencias de Java

En la siguiente tabla se muestran las dependencias de Java junto con su versión y licencia.

Dependencia	Versión	Licencia
Apache Commons CSV	1.6	Apache-2.0
Apache Commons Math	3.6.1	Apache-2.0
Apache Log4j	1.2.17	Apache-2.0
Gson	2.8.5	Apache-2.0
JSON In Java	20180813	JSON License
JSoup Java HTML Parser	1.11.3	MIT
SLF4J API Module	1.7.26	MIT
SLF4J LOG4J 12 Binding	1.7.26	MIT
ThreeTen Extra	1.5.0	BSD 3-Clause

Tabla A.2: Dependencias de Java.

## Dependencias de JavaScript

En esta otra tabla se muestra las dependencias de JavaScript. Cabe destacar que Google Charts no es de código libre, no permite descargar y usarlo en un fichero local sin conexión [2].

Dependencia	Versión	Licencia
Chart.js	2.8.0	MIT
color-hash	1.0.3	MIT
Google Charts	Desconocida	Freeware
html2canvas	0.4.1	MIT

Tabla A.3: Dependencias de JavaScript.

### Paquete de iconos

Se ha usado el paquete de iconos Font Awesome [1] que también se emplea en Moodle.

Los iconos de Font Awesome están sacados de **fa2png** versión **4.7.0** y licencia **SIL OFL 1.1**.





## *Apéndice B*

---

# Especificación de Requisitos

---

### B.1. Introducción

En este anexo se comentará los objetivos generales del proyecto y detallando sobre los requisitos funcionales y no funcionales establecidos durante el desarrollo del proyecto.

### B.2. Objetivos generales

El proyecto tiene como objetivos la refactorización del código junto con la integración de los registros del curso. Añadir nuevas mejoras de interacción con el usuario así como poder guardar los datos en ficheros cifrados. También realizar nuevas gráficas de los registros junto con sus opciones de filtrado.

### B.3. Catalogo de requisitos

Estos son los siguientes requisitos funcionales y no funcionales:

#### Requisitos funcionales

- **RF-1 Guardar los datos:** la aplicación tiene que ser capaz de guardar los datos para sus posteriores ejecuciones.
  - **RF-1.1 Almacenar en un fichero:** la aplicación debe almacenarlo en archivos locales.

- **RF-1.2 Cifrar los datos:** la aplicación debe encriptar los datos cuando se guarde.
- **RF-2 Recordar datos sesión:** la aplicación debe dar la opción de recordar el usuario y *host* en sucesivas ejecuciones.
- **RF-3 Opción de actualizar:** el usuario debe tener la posibilidad de elegir
- **RF-3 Filtros de visualización:** en la visualización de los registros tenga posibilidad de filtrar.
  - **RF-3.1 Filtrado por agrupaciones:** filtros agrupando por diferentes formas de tiempo.
  - **RF-3.2 Filtrado por fechas:** filtros según fechas del calendario.
- **RF-4 Aspectos de visualización de los usuarios:** mejorar la visualización de los participantes del curso.
  - **RF-4.1 Imágenes de los usuarios:** mostrar los fotos de los usuarios asociados en Moodle.
  - **RF-3.2 Últimos accesos:** mostrar los últimos accesos de los usuarios a la plataforma de Moodle.
- **RF-5 Integrar con otros proyectos:** debe poder integrar lo máximo posible de los proyectos UBUGrades y UBULogs.s
- **RF-6 Fechas de los registros:** los registros se deben guardar en la zona horaria del servidor, hay que transformar a la del servidor si el usuario está usando otra zona horaria .

## Requisitos no funcionales

- **RNF-1 Tiempos de respuesta:** la aplicación debe tener buenos tiempos de respuesta a las diferentes acciones del usuarios.
- **RNF-2 Gestión de memoria:** el consumo de memoria de la aplicación no debe ser excesiva.
- **RNF-3 Usabilidad:** el esfuerzo del usuario para aprender a usar la aplicación debería ser mínimo.
- **RNF-4 Soporte:** brindar al usuario una fácil instalación, documentaciones de uso y facilidad de actualización hacia versiones más recientes.

- **RNF-5 Internacionalización:** la aplicación debe poder añadir fácilmente otros idiomas.

## B.4. Especificación de requisitos

La aplicación solo posee un actor en los casos de uso, el usuario.

### Diagrama de casos de uso

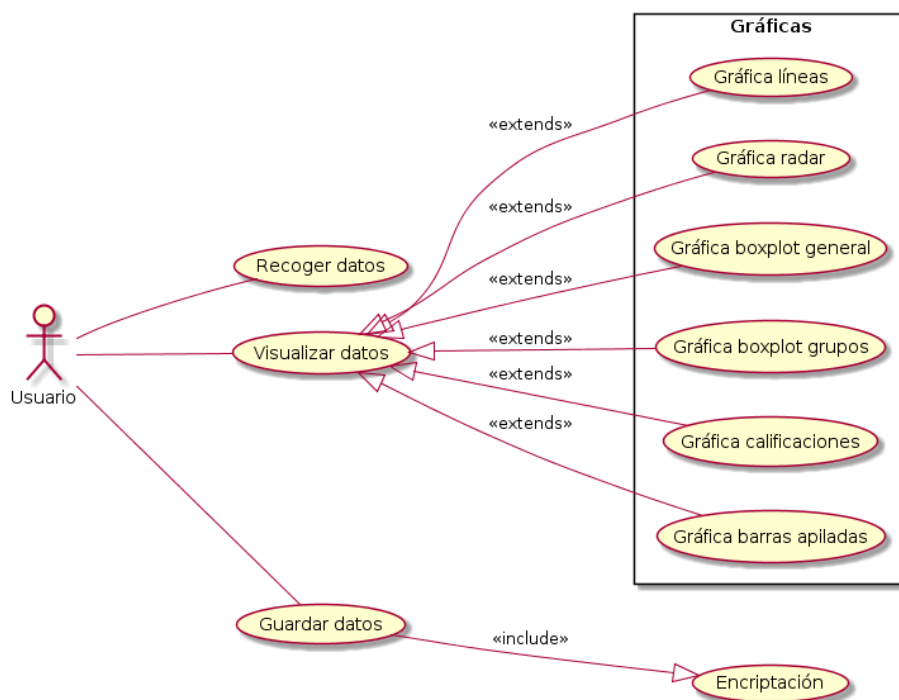


Figura B.1: Diagrama de casos de uso.



## *Apéndice C*

---

# Especificación de diseño

---

### C.1. Introducción

En este anexo se va a comentar los aspectos más relevantes del diseño de esta aplicación.

### C.2. Diseño de datos

Toda la información que se almacena en Moodle es con bases de datos con un campo numérico de identificador único (*ID*) para cada uno de los elementos (usuarios, cursos, módulos del curso). Para simular este efecto en Java se ha creado una clase que incluye varios `HashMap` para los elementos comentados anteriormente usando los identificadores de cada tipo como clave del `HashMap` tal y como se puede ver en la figura C.1. Esta clase también es el encargado de instanciar cada uno de los tipos con el objetivo de tener instancias únicas por cada identificador.

```

public class DataBase implements Serializable {

    /**
     *
     */
    private static final long serialVersionUID = 1L;
    private Map<Integer, Role> roles;
    private Map<Integer, Group> groups;
    private Map<Integer, EnrolledUser> users;
    private Map<Integer, Module> modules;
    private Map<Integer, Course> courses;
    private Map<Integer, GradeItem> gradeItems;
    private Map<Integer, CourseCategory> courseCategories;
}

```

Figura C.1: Simulación de base de datos.

### C.3. Diseño procedimental

En este apartado informa sobre las conexiones realizadas en la plataforma de Moodle durante la ejecución del producto. Comunicándose además de los servicios web de REST API, se inicia sesión también en la web para realizar acciones no disponibles como la descarga de los registros. Todas las peticiones HTTP y recibir las respuestas lo realiza librería jsoup.

Estas comunicaciones lo dividiremos en dos partes:

#### Inicio de sesión

Al introducir los datos de usuario, contraseña y *host* en la aplicación, se inicia sesión tanto en el servicio de REST API como en la web para *web scraping*.

Los pasos que sigue la aplicación en el inicio de sesión son:

1. Busca el *token* del REST API a partir del usuario y contraseña como parámetros de URL.
2. Busca el *login token* de inicio de sesión en la web.
3. Inicia sesión usando el *login token* como parámetro adicional si existe, guarda los *cookies* de sesión para futuras peticiones.
4. En la API recupera la información del usuario conectado usando la función del servicio web `core_user_get_users_by_field`.

5. Descarga la imagen del usuario conectado.
6. Recupera de los servicios web la información de los cursos matriculados por el usuario con la función **core\_enrol\_get\_users\_courses**.
7. Con la función de la API **core\_course\_get\_categories** devuelve la información de las categorías en el que se encuentran los cursos.

Estos pasos se visualizan también en el diagrama de secuencias de la figura **C.2**.

## Elección del curso

Después de elegir el curso, la aplicación descargará información de la REST API y también los registros del curso.

### REST API

De la información que se busca en los servicios web de Moodle:

1. A través de la función **core\_enrol\_get\_enrolled\_users** recogemos la información de los usuario matriculados del curso junto con sus roles y grupos.
2. Descargamos cada uno de las fotos de los usuarios matriculados.
3. Se busca la información de los módulos del curso que hay con la función **core\_course\_get\_contents**.
4. Con la función **gradereport\_user\_get\_grades\_table** nos proporciona los calificadores de todos los alumnos.

Esta información queda reflejada en el diagrama de secuencia de la figura **C.3**

## Descarga de registros

En la descarga de los registros es diferente si tiene el fichero de caché guardado previamente o no.

- Cuando existe el archivo caché

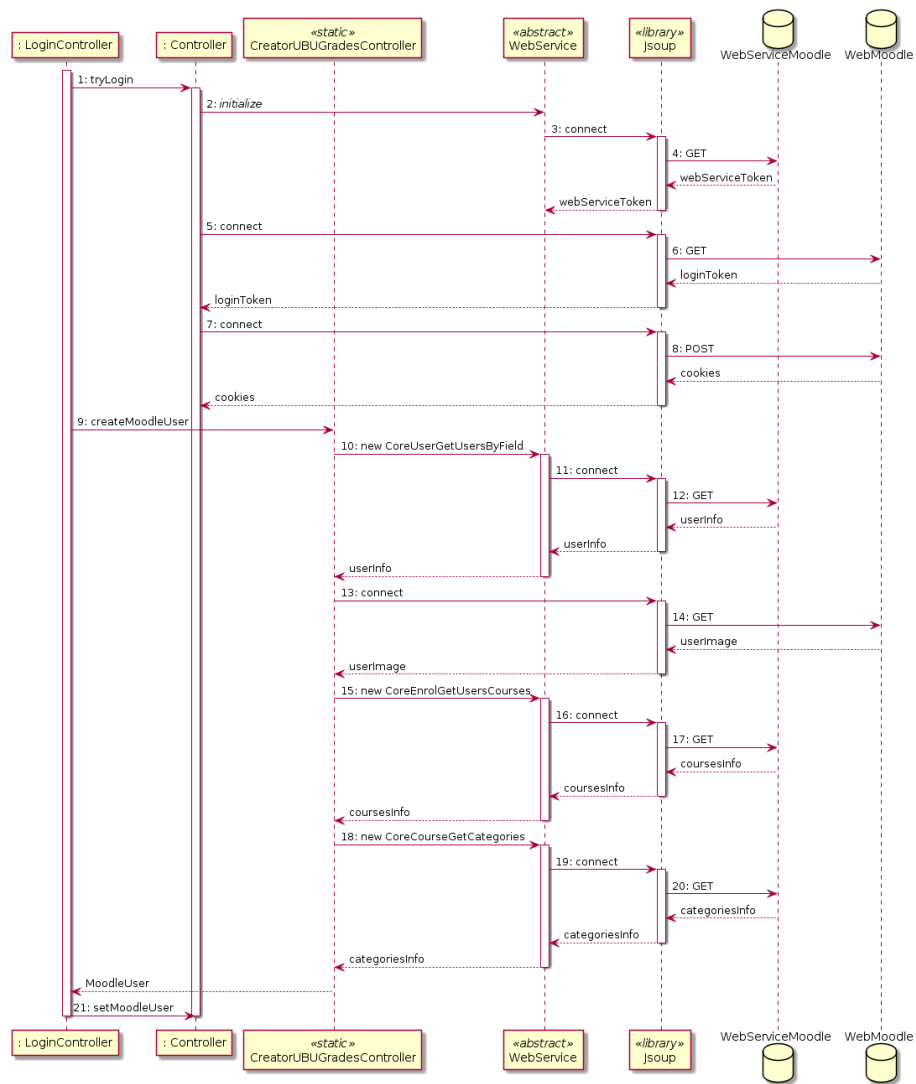


Figura C.2: Diagrama de secuencia del inicio de sesión.



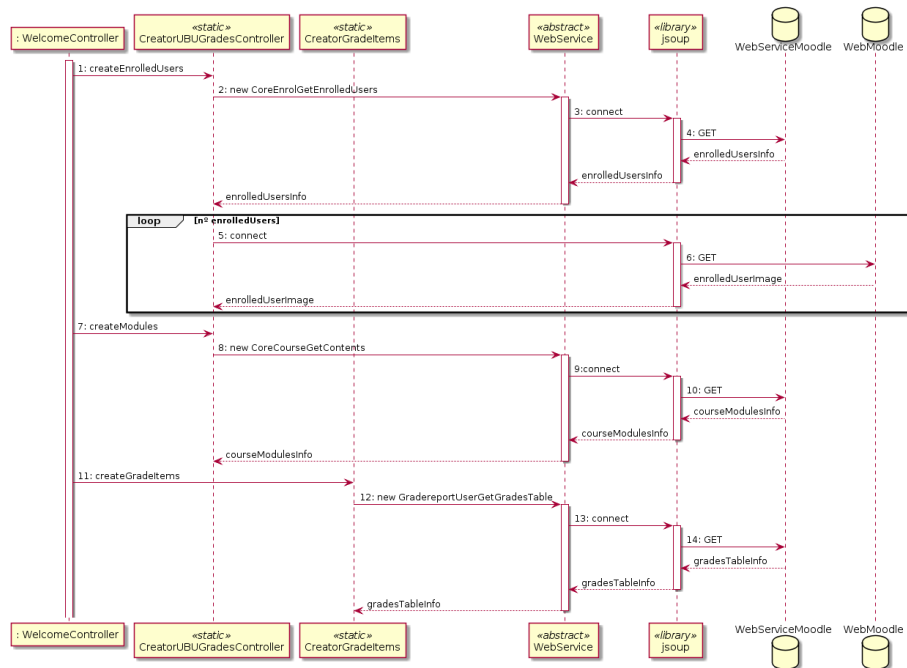


Figura C.3: Diagrama de secuencia de recogida de datos de los servicios web.

1. Si la zona horaria del usuario es la misma que el servidor (99) hay que buscarlo en el perfil del usuario mediante *web scraping*.
  2. Luego va descargando los registros de forma diaria desde la fecha del último registro hasta la fecha y hora actual.
- Cuando no existe el archivo caché
    1. Busca la zona horaria del servidor a partir del perfil del usuario.
    2. Descarga los registros completos del curso.

En el diagrama de secuencia de la figura C.4 recoge los pasos que se realiza en la descarga de registros.

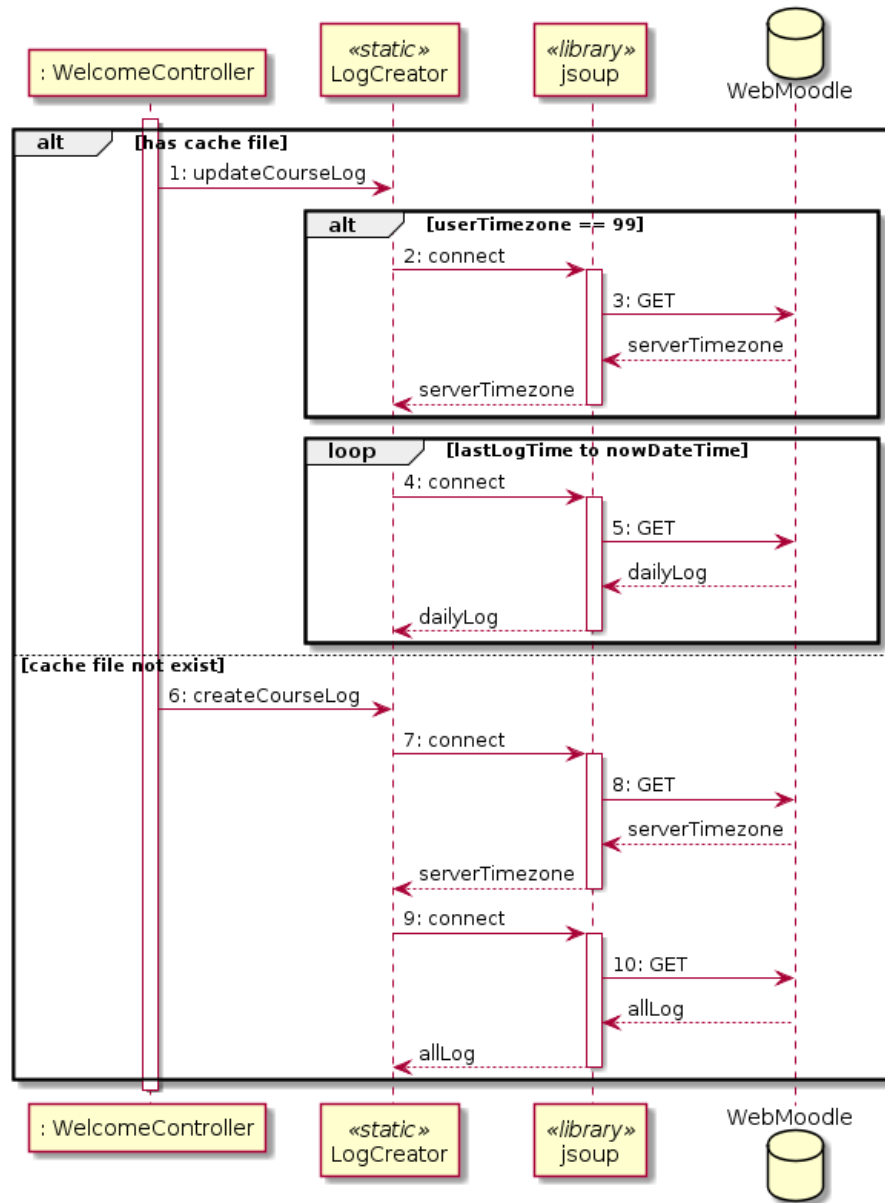


Figura C.4: Diagrama de secuencia de descarga de los registros del curso.

## C.4. Diseño arquitectónico

### Arquitectura cliente-servidor

Esta arquitectura consiste en un cliente que realiza peticiones a un servidor que le da respuestas. En este caso, el cliente sería la aplicación y el

servidor es la plataforma de Moodle y sus servicios web (REST API) y la web.

En la figura C.5 se visualiza un diagrama de despliegue que muestra la conexión de la aplicación con el servidor Moodle a través de Internet.

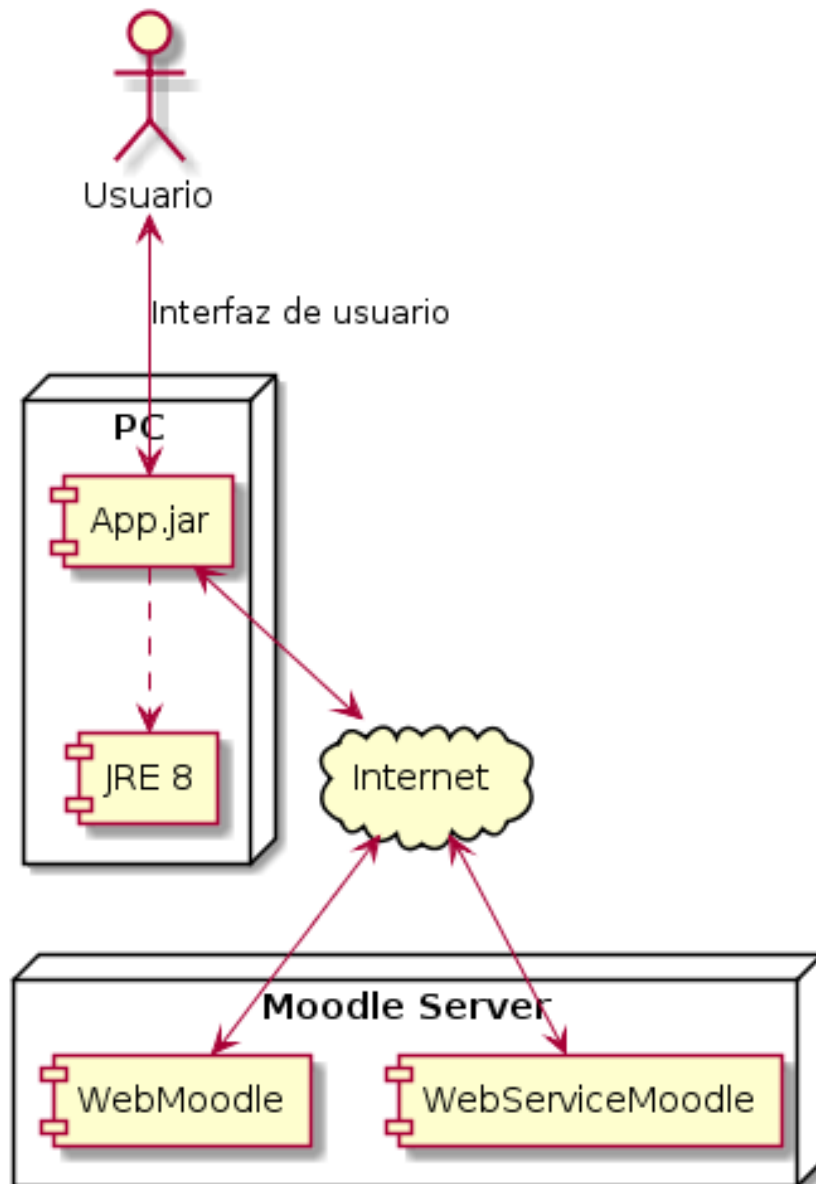


Figura C.5: Diagrama de despliegue de la aplicación.

## **Arquitectura MVC**

## **Patrones de diseño**

### **C.5. Diseño de la interfaz**

En cuanto al diseño de la interfaz se ha mantenido el del proyecto de UBUGrades añadiendo las nuevas características del presente desarrollo.

## *Apéndice D*

---

# Documentación técnica de programación

---

## D.1. Introducción

En esta sección del anexo se explicarán cómo están organizados los paquetes del proyecto; un manual para el programador sobre las herramientas que debe instalar para seguir el desarrollo; su compilación, instalación y ejecución del proyecto y las pruebas realizadas en la aplicación.

## D.2. Estructura de directorios

El repositorio del proyecto está distribuido de la siguiente forma:

- `/doc/`: los ficheros javadoc del proyecto.
- `/latex/`: los elementos necesarios para generar la memoria y el anexo, tanto los ficheros `.tex` como los `.pdf`. También se incluye las imágenes y la bibliografía en formato `.bib`.
- `/lib/`: librerías externas de Java.
- `/lib/src`: el código fuente de las librerías externas de Java.
- `/lib/docs`: los javadoc de las librerías externas de Java.
- `/python/`: varios archivos de utilidad para los registros del curso:

## APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN

- **Componentes y eventos.json**: fichero dividido en tres niveles que son Componentes, Eventos y Descripciones de los registros del curso.
  - **Contains checker.ipynb**: funciones de utilidad que comprueba en las Descripciones de los registros si existe o no un carácter o varios.
  - **Generador de código componentes y eventos.ipynb**: funciones que generan código Java automáticamente a partir del fichero JSON mencionado.
  - **Rastreador de componentes y eventos.ipynb**: programa de utilidad que escanea los registros del curso en formato CSV y con eso crea un fichero JSON con todos los Componentes, Eventos y Descripciones encontrados.
- 
- /resources/: recursos de apoyo para el programa.
  - /resources/css/: ficheros CSS que modifica la visualización y colores de la aplicación.
  - /resources/graphics/: ficheros HTML para dibujar las gráficas.
  - /resources/graphics/lib: librerías externas de JavaScript usado para las gráficas.
  - /resources/img/: todas las imágenes de la aplicación.
  - /resources/messages/: ficheros de internacionalización de textos.
  - /resources/view/: se encuentra los ficheros FXML.
  - /src/: código fuente del desarrollo.
  - /src/controllers/: paquete encargado del flujo de la ejecución.
  - /src/controllers/datasets/: dirige las generaciones de los datos en formato válido para Chart.js.
  - /src/controllers/ubugrades/: encargado de la generación de los cursos, usuarios matriculados, calificaciones, etc. Toda la parte de los servicios web.
  - /src/controllers/ubulogs/: gestiona la descarga de los registros y su posterior *parseo*.

- `/src/controllers/ubulogs/logtypes/`: las clases encargadas de gestionar los ids de las Descripciones del registro.
- `/src/model/`: representación de los datos, su lógica de negocio.
- `/src/model/mod`: todas las clases de los tipos de módulos del curso.
- `/src/persistence/`: encriptación y desencriptación de archivos de objetos Serializados.
- `/src/webservice/.../`: paquete y subpaquetes encargados de llamar a las funciones de Moodle añadiendo los parámetros necesarios a la URL y recibir las respuestas.

## D.3. Manual del programador

Este manual sirve de referencia para personas que tomen el proyecto en el futuro. Se explicarán como construir el entorno de desarrollo, qué es lo que necesitan instalar.

### Instalación de Java

La aplicación está desarrollada en Java 8, por lo tanto para trabajar en ella necesitará tener el JDK 8. Durante el proyecto se ha trabajado con la **versión 1.8.0\_201** pero sería recomendable trabajar con la última disponible. Realizamos los siguientes pasos si queremos descargar la última versión de JDK:

1. Entrar en la página de [descargas de Java](#).
2. Buscamos **Java SE 8uxxx** siendo las xxx el número de actualización de Java 8.
3. Descargamos el JDK.

## APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN

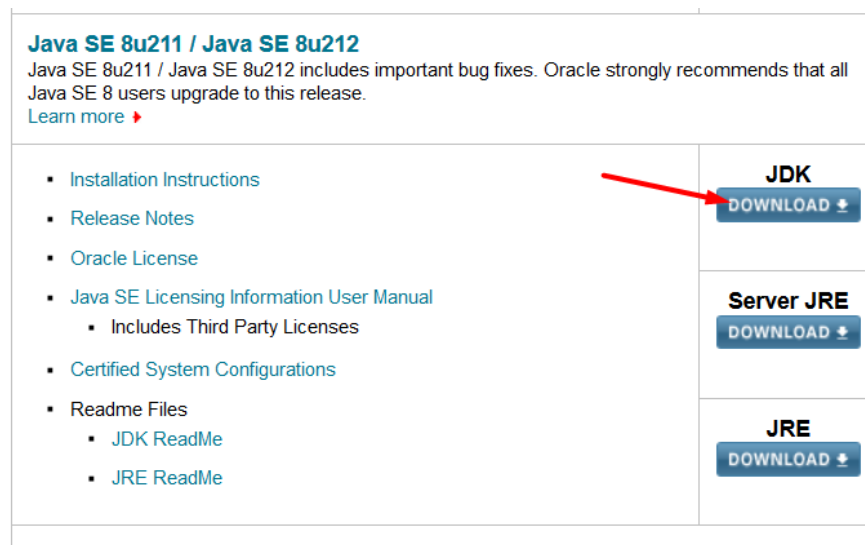


Figura D.1: Elegir la descarga de JDK.

4. Leemos la licencia de uso de Oracle y aceptamos las condiciones.
5. Cuando hayamos aceptado la licencia de uso descargamos el ejecutable según el sistema operativo que tengamos.

**Nota:** x86 es de 32 bits y x64 de 64 bits.

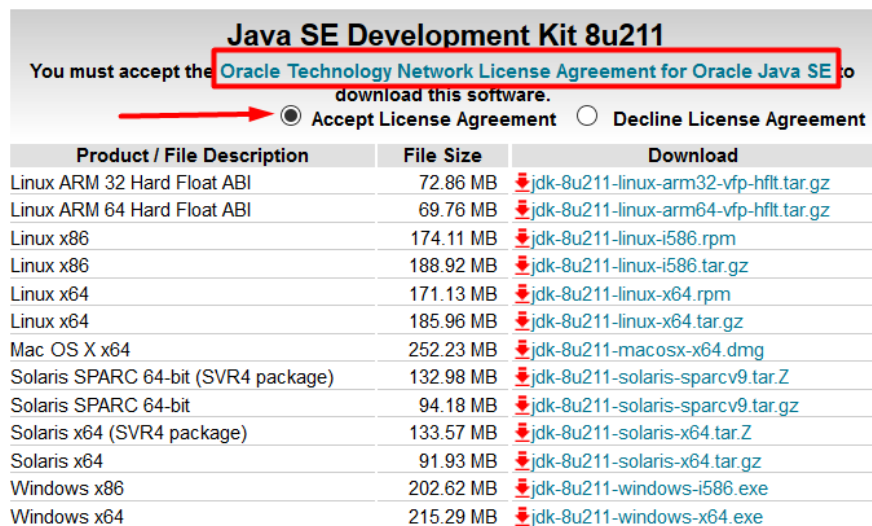


Figura D.2: Descarga del JDK 8



## 6. Seguir las instrucciones del instalador

Es recomendable desinstalar la versión antigua de Java si descargamos la última versión [6].

## Eclipse

Durante el desarrollo se ha empleado la versión **Oxygen.1a Release (4.7.1a)** de Eclipse aunque no habría muchos problemas usar una versión más reciente. A continuación se muestra el método de descargar una versión portable de Eclipse:

1. Entrar en la página web descarga de **paquetes de Eclipse**.
2. Descargamos el paquete **Eclipse IDE for Java Developers**, en el sistema operativo que se tenga.

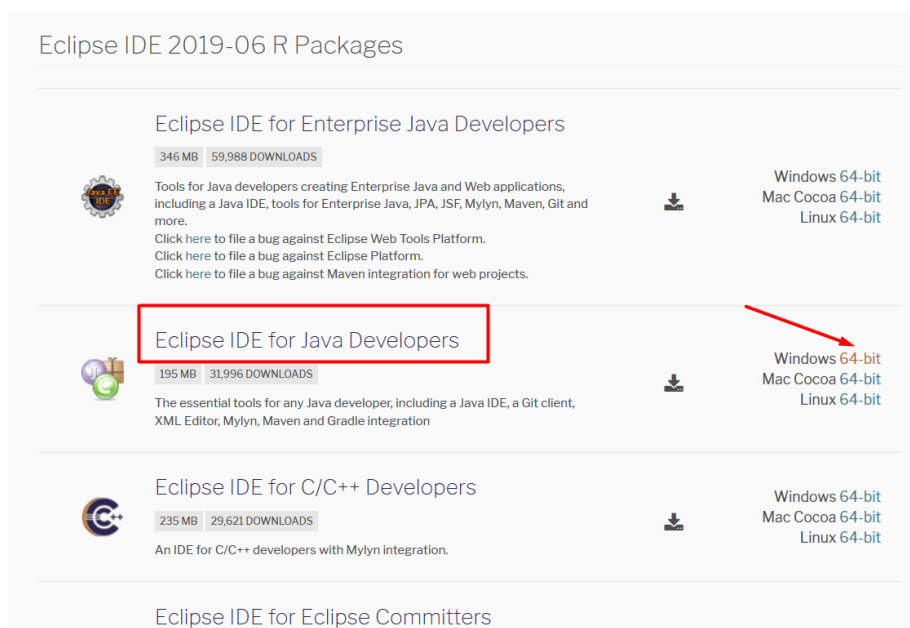


Figura D.3: Descarga del paquete de Eclipse.

3. Pulsamos el botón de descarga **Download**.

## APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN



Figura D.4: Pulsar el botón de descarga.

4. Descomprimir la carpeta dentro del ZIP.
5. Ejecutar **eclipse.exe** cuando se quiera usar.

### Instalación de e(fx)clipse

e(fx)clipse es una herramienta adicional de Eclipse que facilita el desarrollo de aplicaciones que usen JavaFX. Para instalar e(fx)clipse realizamos los siguientes pasos:

1. Seleccionamos el menú **Help** de la barra de herramientas de Eclipse.
2. Seleccionamos el elemento del menú **Install New Software...**

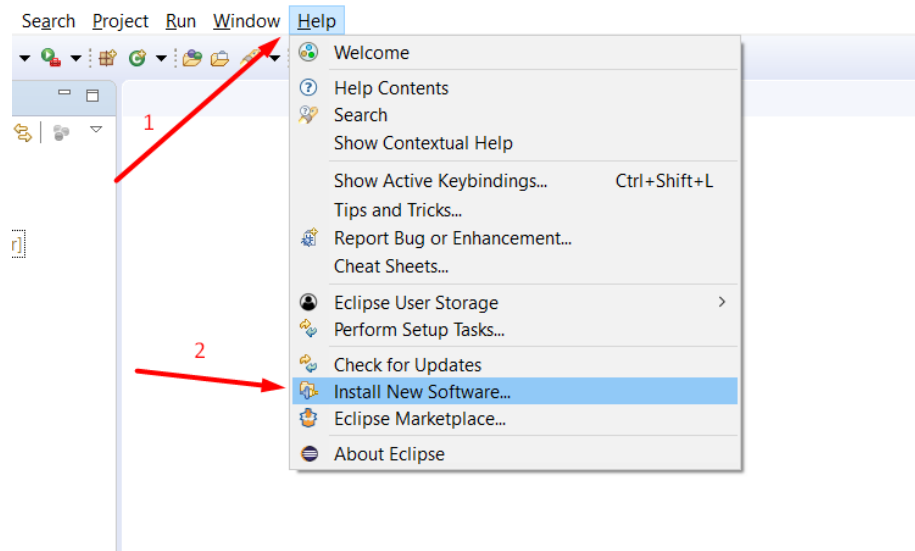


Figura D.5: Menú **Help** y seleccionar **Install New Software...**

3. Seleccionamos la opción **–All Available Sites–** de la lista desplegable de **Work with:**.
4. Escribimos **e(fx)clipse** en el cuadro de texto.
5. Marcamos solo la casilla que pone **e(fx)clipse - IDE**.
6. Pulsamos el botón **Next** y debería descargar e instalarse.

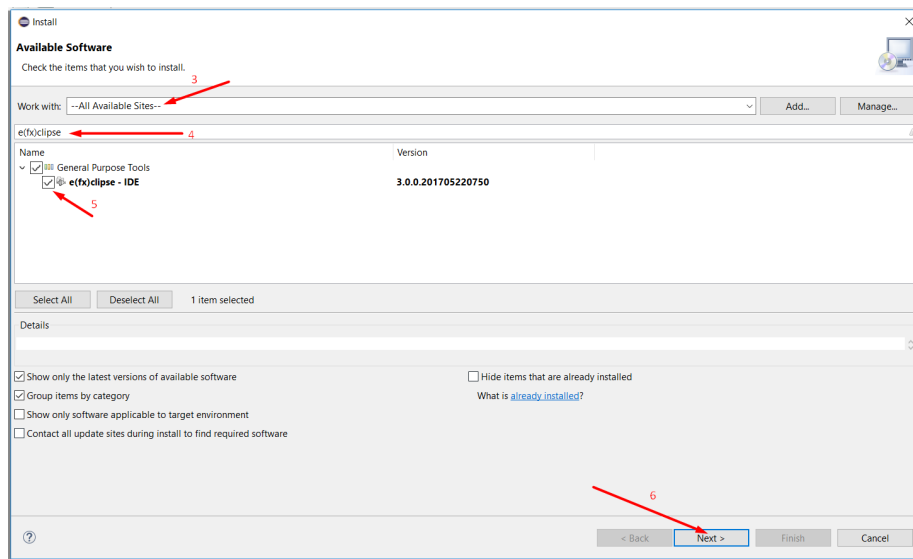


Figura D.6: Instalación de la herramienta.

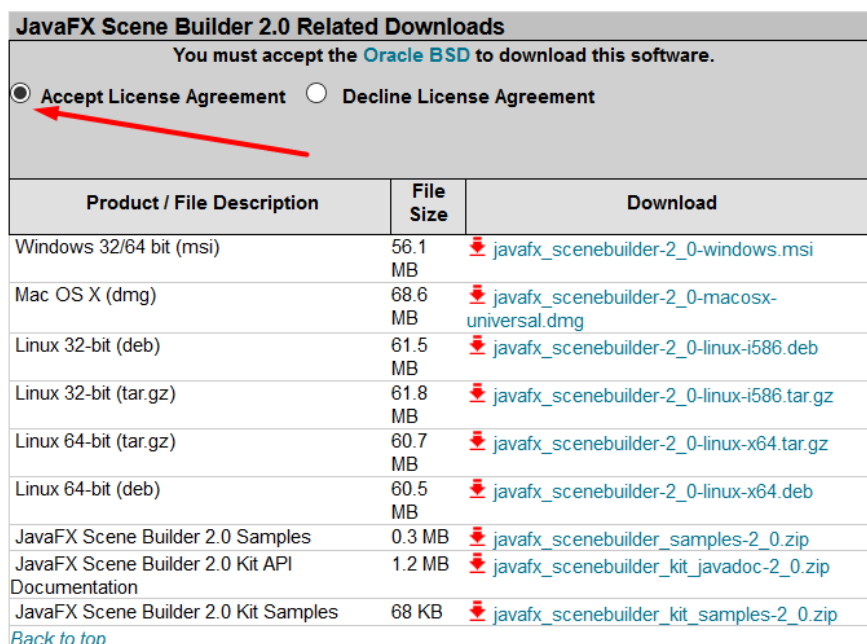
## Instalación de Scene Builder

La herramienta del Scene Builder (2.0) se puede conseguir desde:

<http://www.oracle.com/technetwork/java/javase/downloads/javafxscenebuilder-1x-archive-2199384.html>

Después de entrar a la página se busca la versión 2.0 de Scene Builder, se acepta la licencia y descargamos la versión según el sistema operativo que se use.

## APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN



JavaFX Scene Builder 2.0 Related Downloads		
You must accept the <a href="#">Oracle BSD</a> to download this software.		
<input checked="" type="radio"/> Accept License Agreement <input type="radio"/> Decline License Agreement		
Product / File Description	File Size	Download
Windows 32/64 bit (msi)	56.1 MB	<a href="#">javafx_scenebuilder-2_0-windows.msi</a>
Mac OS X (dmg)	68.6 MB	<a href="#">javafx_scenebuilder-2_0-macosx-universal.dmg</a>
Linux 32-bit (deb)	61.5 MB	<a href="#">javafx_scenebuilder-2_0-linux-i586.deb</a>
Linux 32-bit (tar.gz)	61.8 MB	<a href="#">javafx_scenebuilder-2_0-linux-i586.tar.gz</a>
Linux 64-bit (tar.gz)	60.7 MB	<a href="#">javafx_scenebuilder-2_0-linux-x64.tar.gz</a>
Linux 64-bit (deb)	60.5 MB	<a href="#">javafx_scenebuilder-2_0-linux-x64.deb</a>
JavaFX Scene Builder 2.0 Samples	0.3 MB	<a href="#">javafx_scenebuilder_samples-2_0.zip</a>
JavaFX Scene Builder 2.0 Kit API Documentation	1.2 MB	<a href="#">javafx_scenebuilder_kit_javadoc-2_0.zip</a>
JavaFX Scene Builder 2.0 Kit Samples	68 KB	<a href="#">javafx_scenebuilder_kit_samples-2_0.zip</a>

[Back to top](#)

Figura D.7: Aceptar la licencia de Oracle BSD y descargar.

Se sigue las instrucciones del instalador.

### Instalación de Moodle

Para instalar Moodle en el equipo personal, se puede descargar desde la página oficial con todo incluido para su funcionamiento:

<https://download.moodle.org/windows/>

Una vez descargado y descomprimido la carpeta, ejecutamos **StartMoodle.exe** que inicia la arranque del servidor.

Para acceder al sitio web de Moodle se usa el localhost: <http://localhost>. En la página web, la primera vez iniciará el proceso de instalación del servidor. Se sigue las instrucciones del instalador.

Por defecto no están activados los servicios web de Moodle. Para habilitarlo seguimos los siguientes pasos después de iniciar sesión con la cuenta del administrador:

1. Pulsamos el botón **Administración del sitio**.

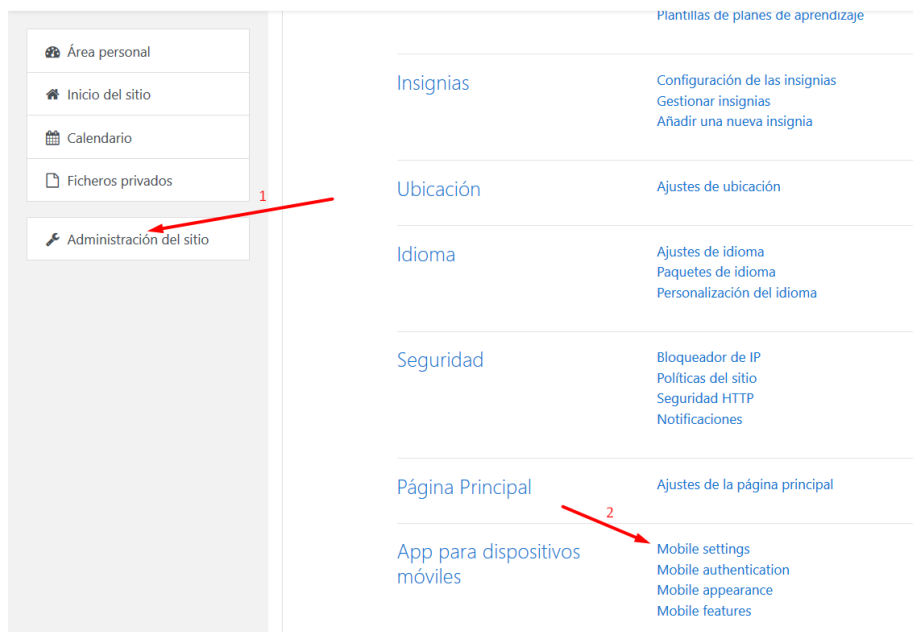
2. Entramos dentro de **Mobile Settings**.

Figura D.8: Habilitar servicios web.

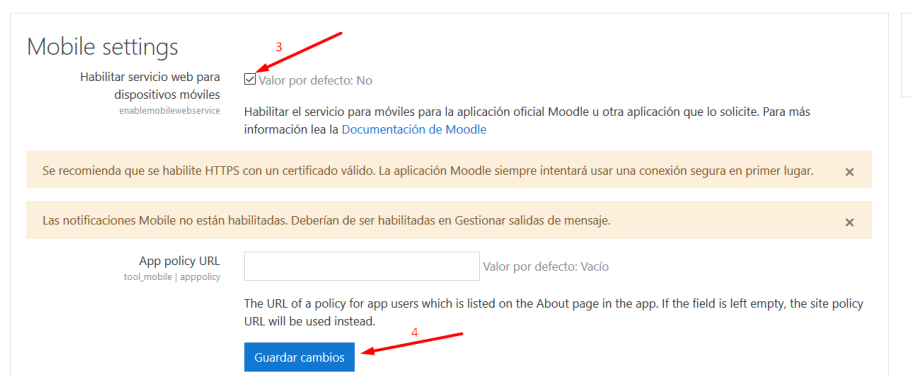
3. Marcamos la casilla de **Habilitar servicio web para dispositivos móviles**.4. Pulsamos en **Guardar cambios**

Figura D.9: Habilitar servicios web parte 2.

## APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN

### Listado de funciones de los servicios web de Moodle

Moodle ofrece también un listado de funciones del servicio web junto con información de parámetros y el esqueleto del JSON que devuelve. Para acceder a ella es necesario haber iniciado sesión con la cuenta del administrador.

1. Después de iniciar sesión con el administrador. Pulsamos **Administración del sitio** y luego a la pestaña de **Extensiones**.

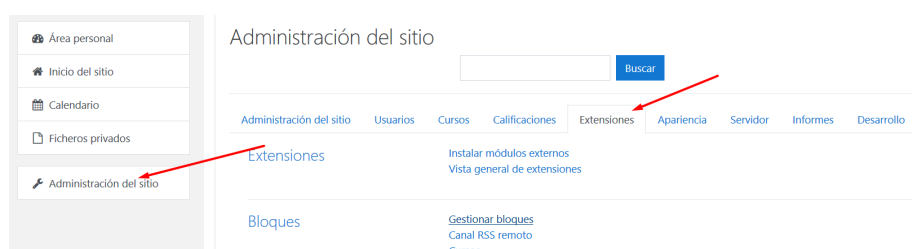


Figura D.10: Acceso al listado de funciones del servicio web de Moodle.

2. Bajamos abajo hasta la fila de **Servicios Web** y accedemos a **Documentación API**.

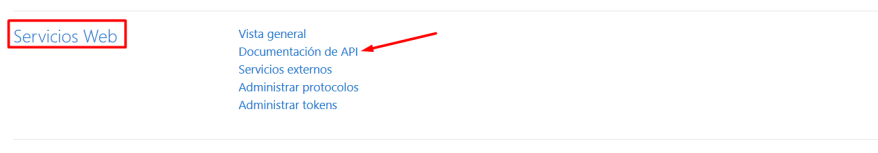


Figura D.11: Acceder a la Documentación API.

3. Muestra un listado de todas las funciones que al desplegar se puede visualizar los parámetros y esqueletos de la información que devuelve.

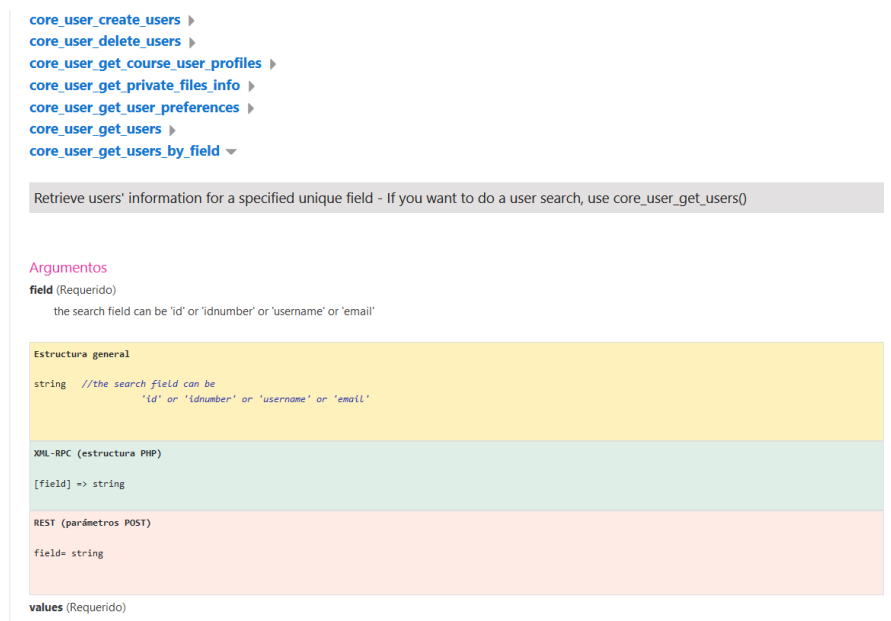


Figura D.12: Listado de funciones del servicio web de Moodle.

## Instalación de Jupyter Notebook

Según la página oficial de Jupyter Notebook [7], lo más recomendable es instalarlo a través de Anaconda en:

<https://www.anaconda.com/distribution/#download-section>.

La versión recomendable que se debe descargar es el que incluye Python 3.7 o superior debido a que los diccionarios están ordenados por orden de inserción.

## Clasificación de registros

Para clasificar se ha partido de la premisa de que las columnas **Componente** y **Nombre de Evento** usa un esqueleto de **Descripción** único.

A partir de expresiones regulares, se ha buscado las posibles los ids de las descripciones y con esos números pasado a un delegado (clase) en función del Componente y Evento.

## APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN

```
public class LogTypes {  
  
    private static final Map<ComponentEvent, ReferencesLog> TYPES = new HashMap<>();  
  
    static{  
  
        COMPONENTE EVENTO DELEGADO  
        TYPES.put(get(ACTIVITY_REPORT, ACTIVITY_REPORT_VIEWED), UserCourse.getInstance());  
        TYPES.put(get(ACTIVITY_REPORT, OUTLINE_REPORT_VIEWED), UserAffectedCourse.getInstance());  
  
        TYPES.put(get(ASSIGNMENT, A_SUBMISSION_HAS_BEEN_SUBMITTED), UserSubmissionCmid.getInstance());  
        TYPES.put(get(ASSIGNMENT, ALL_THE_SUBMISSIONS_ARE_BEING_DOWNLOADED), UserCmid.getInstance());  
        TYPES.put(get(ASSIGNMENT, AN_EXTENSION_HAS_BEEN_GRANTED), UserAffectedCmid.getInstance());  
        TYPES.put(get(ASSIGNMENT, ASSIGNMENT_OVERRIDE_CREATED), UserOverrideCmidAffected.getInstance());  
        TYPES.put(get(ASSIGNMENT, ASSIGNMENT_OVERRIDE_DELETED), UserOverrideCmidAffected.getInstance());  
        TYPES.put(get(ASSIGNMENT, ASSIGNMENT_OVERRIDE_UPDATED), UserOverrideCmidAffected.getInstance());  
        TYPES.put(get(ASSIGNMENT, COURSE_MODULE_INSTANCE_LIST_VIEWED), UserCourse.getInstance());  
        TYPES.put(get(ASSIGNMENT, COURSE_MODULE_VIEWED), UserCmid.getInstance());  
        TYPES.put(get(ASSIGNMENT, GRADING_FORM_VIEWED), UserAffectedCmid.getInstance());  
        TYPES.put(get(ASSIGNMENT, GRADING_TABLE_VIEWED), UserCmid.getInstance());  
        TYPES.put(get(ASSIGNMENT, SUBMISSION_CONFIRMATION_FORM_VIEWED), UserCmid.getInstance());  
        TYPES.put(get(ASSIGNMENT, SUBMISSION_FORM_VIEWED), UserCmid.getInstance());  
        TYPES.put(get(ASSIGNMENT, SUBMISSION_VIEWED), UserAffectedCmid.getInstance());  
        TYPES.put(get(ASSIGNMENT, THE_STATE_OF_THE_WORKFLOW_HAS_BEEN_UPDATED), UserAffectedCmid.getInstance());  
        TYPES.put(get(ASSIGNMENT, THE_STATUS_OF_THE_SUBMISSION_HAS_BEEN_UPDATED), UserSubmissionCmid.getInstance());  
        TYPES.put(get(ASSIGNMENT, THE_STATUS_OF_THE_SUBMISSION_HAS_BEEN_VIEWED), UserCmid.getInstance());  
        TYPES.put(get(ASSIGNMENT, THE_SUBMISSION_HAS_BEEN_GRADED), UserSubmissionAffectedCmid.getInstance());  
        TYPES.put(get(ASSIGNMENT, THE_SUBMISSIONS_HAVE_BEEN_LOCKED_FOR_A_USER), UserAffectedCmid.getInstance());  
        TYPES.put(get(ASSIGNMENT, THE_SUBMISSIONS_HAVE_BEEN_UNLOCKED_FOR_A_USER), UserAffectedCmid.getInstance());  
        TYPES.put(get(ASSIGNMENT, THE_USER_HAS_ACCEPTED_THE_STATEMENT_OF_THE_SUBMISSION), UserSubmissionCmid.getInstance());  
    }  
}
```

Figura D.13: Clasificación de los registros junto con sus delegados.

### Delegados

Los nombres de los delegados son códigos nemotécnicos en función el tipo de ids asigna. Por ejemplo en la siguiente imagen usa los códigos nemotécnicos User, Affected y Cmid. Está asignando el usuario, usuario afectado y el módulo del curso.



```

public class UserAffectedCmid extends ReferencesLog {

    /**
     * Instacia única de la clase UserAffectedCmid.
     */
    private static UserAffectedCmid instance;

    /**
     * Constructor privado de la clase singleton.
     */
    private UserAffectedCmid() {
    }

    /**
     * Devuelve la instancia única de UserAffectedCmid.
     * @return instancia singleton
     */
    public static UserAffectedCmid getInstance() {
        if (instance == null) {
            instance = new UserAffectedCmid();
        }
        return instance;
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public void setLogReferencesAttributes(LogLine log, List<Integer> ids) {
        setUserById(log, ids.get(0));
        setAffectedUserById(log, ids.get(1));
        setCourseModuleById(log, ids.get(2));
    }
}

```

Figura D.14: Un delegado UserAffectedCmid

En la tabla [D.1](#) se puede ver los códigos nemotécnicos usados y una pequeña descripción.

## Uso de las herramientas de utilidad Python

En este apartado se comentará las tres herramientas existentes en Jupyter Notebook. Estos facilitan al programador crear ciertos códigos que tienen que ver con los registros del curso.

### Rastreador de componentes y eventos

Programa que selecciona uno o varios CSV de los registros descargados en inglés. Y de forma opcional elegir el archivo JSON donde se actualizarán

## APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN

Código nemotécnico	Descripción
Affected	id del usuario afectado
Attempt	id del intento
Calendar	id del evento del calendario
Category	id de la categoría
Chapter	id del capítulo del libro
Choice	id de elección (choice)
Cmid	id del módulo del curso
Comment	id del comentario
Competency	id de competencia del curso
Course	id del curso
Discussion	id de la discusión del foro
Evidence	id de la evidencia
Field	id del campo
Files	número de ficheros
Glossary	id de la entrada al glosario
Grade	id de calificación
Gradeitem	id de grade ítem
Group	id del grupo
Grouping	id de grouping
Item	id de item
Note	id de note
Option	id de opción
Override	id de anulación (override)
Page	id de la página
Post	id del post del foro
Question	id de pregunta del cuestionario
Questioncategory	id de categoría de cuestionario
Record	id de record
Role	id del rol
Rule	id de regla
Scale	id de escala
Sco	id del scorm package
Section	número de sección
Submission	id de la entrega
Subscription	id de suscripción al evento
Tour	id de tour
User	id del usuario del usuario que realiza la acción
Words	número de palabras

Tabla D.1: Códigos nemotécnicos de los delegados.

todos los **Componentes**, **Eventos** junto con las **Descripciones**. Si no se selecciona el fichero JSON generará uno nuevo de los registros del CSV.

Se aporta ya un fichero JSON (**Componentes y eventos.json**) con **48 Componentes** y **193 Eventos** diferentes detectados hasta el momento.

Este fichero usa tres niveles:

1. En el primer nivel indica el Componente.
2. En el segundo nivel es el Evento.
3. En el tercer nivel es un listado del esqueleto de las Descripciones asociados al Componente y Evento.

En la figura D.15 podemos ver un extracto del JSON recuadrado los Componentes (en rojo) y los Eventos (en verde).

```
{
  "Activity report": {
    "Activity report viewed": [
      "The user with id 'INTEGER' viewed the outline activity report for the course with id 'INTEGER'."
    ],
    "Outline report viewed": [
      "The user with id 'INTEGER' viewed the outline report for the user with id 'INTEGER' for the course with id 'INTEGER'."
    ]
  },
  "Assignment": {
    "A submission has been submitted": [
      "The user with id 'INTEGER' has submitted the submission with id 'INTEGER' for the assignment with course module id 'INTEGER'."
    ],
    "All the submissions are being downloaded": [
      "The user with id 'INTEGER' has downloaded all the submissions for the assignment with course module id 'INTEGER'."
    ],
    "An extension has been granted": [
      "The user with id 'INTEGER' has granted an extension for the user with id 'INTEGER' for the assignment with course module id 'INTEGER'."
    ],
    "Assignment override created": [
      "The user with id 'INTEGER' created the override with id 'INTEGER' for the assign with course module id 'INTEGER' for the user with id 'INTEGER'."
    ],
    "Course module instance list viewed": [
      "The user with id 'INTEGER' viewed the instance list for the module 'STRING' in the course with id 'INTEGER'."
    ],
    "Course module viewed": [
      "The user with id 'INTEGER' viewed the 'STRING' activity with course module id 'INTEGER'."
    ],
    "Grading form viewed": [
      "The user with id 'INTEGER' viewed the grading form for the user with id 'INTEGER' for the assignment with course module id 'INTEGER'."
    ]
  },
}
```

Figura D.15: Extracto de Componentes y eventos.json

Los pasos a seguir añadir nuevos elementos son:

1. Abrir el archivo **Rastreador de componentes y eventos.ipynb** con Jupyter Notebook.
2. El programa está preparado para ejecutar todas las celdas seguidas, elegimos el menú **Cell** y luego la opción **Run All**.

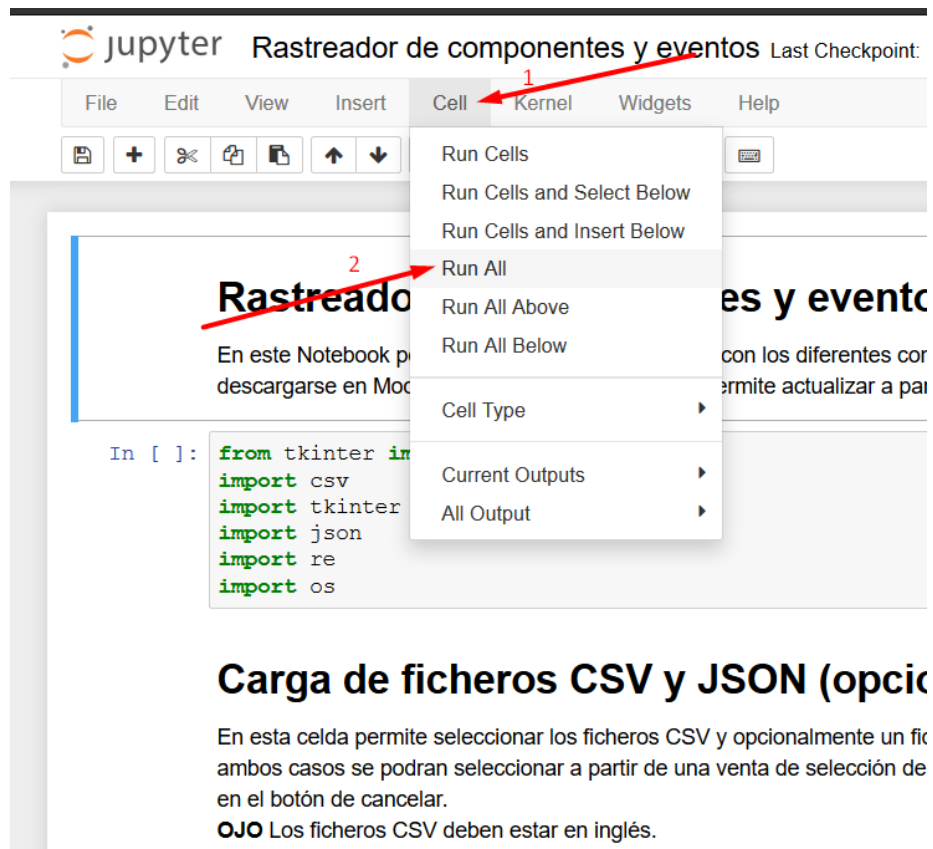


Figura D.16: Menú **Cell** y opción **Run All**

3. Aparecerá una primera ventana que pide seleccionar los ficheros CSV de los registros en inglés.

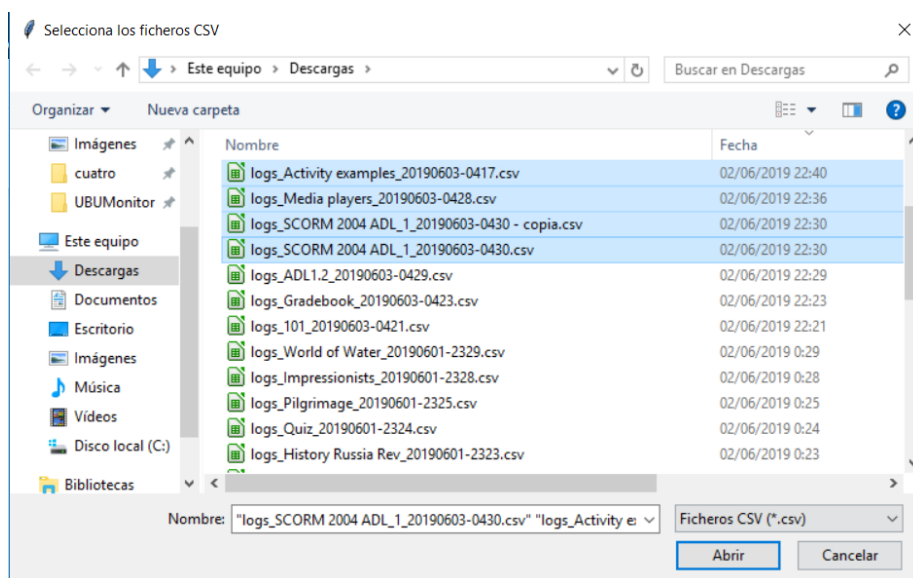


Figura D.17: Selección múltiple de los archivos CSV.

4. A continuación se mostrará una nueva ventana que pide el fichero JSON.

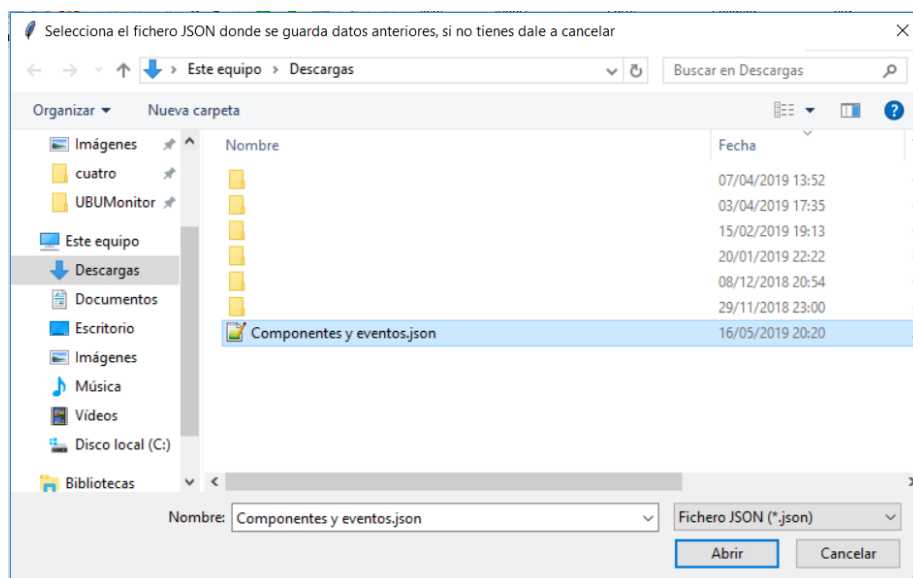


Figura D.18: Seleccionar el archivo JSON.

5. Si no hay ningún problema encontrado durante la ejecución de las

## APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN

celdas, ha salido todo bien.

Uno de los problemas que se puede encontrar es que haya elegido accidentalmente un CSV en otro idioma que no sea en inglés. Una de las celdas saltará un error y las siguientes no se ejecutarán.

Por ejemplo en la siguiente imagen se puede ver que salta un error de que no se encuentra la columna Component. Producido por el último fichero en *parsearse* indicado en un recuadro en rojo.

```
Parseando: C:/Users/PC/Downloads/logs Digital Literacy 20190625-1418.csv
Error en Fila 1 OrderedDict([('UserHora', '25/06/2019 14:18'), ('Nombre completo del usuario', 'Jeffrey Sanders'),
('Usuario afectado', '-'), ('Contexto del evento', 'Curso: Digital Literacy '), ('Componente', 'Registros'), ('Nombre
evento', 'Informe de registros visto'), ('Descripción', 'The user with id '13' viewed the log report for the course w
ith id '66'.'), ('Origen', 'web'), ('Dirección IP', '83.44.99.163')])

-----
KeyError                                Traceback (most recent call last)
<ipython-input-4-f1f0a18f00af> in <module>()
    11
    12
    13     try:
    14         component=row["Component"]
    15         event=row["Event name"]
    16         description=row["Description"]

KeyError: 'Component'
```

Figura D.19: Error producido al usar un CSV que no está en inglés.

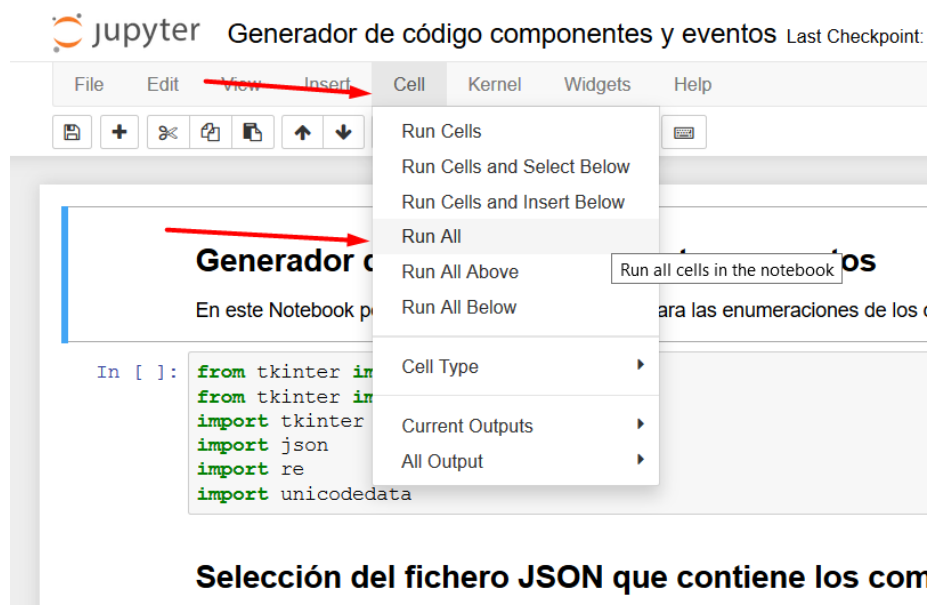
Para solucionar solo hay que volver ejecutar todas las celdas sin elegir ese archivo CSV que da problemas.

### Generador de código componentes y eventos

Este segundo programa usa el fichero JSON generado comentado anteriormente (D.3) y va mostrando en pantalla lo que hay que copiar y pegar en el proyecto.

Genera automáticamente las enumeraciones de los Componentes y Eventos, las traducciones del *Resource Bundle* en inglés y las combinaciones de Componente y Evento existentes.

1. Abrir el archivo **Generador de código componentes y eventos.ipynb** con Jupyter Notebook.
2. El programa está preparado para ejecutar todas las celdas seguidas, elegimos el menú **Cell** y luego la opción **Run All**.

Figura D.20: Menú **Cell** y opción **Run All**

3. Lo primero que mostrará una ventana que pide seleccionar el archivo JSON.

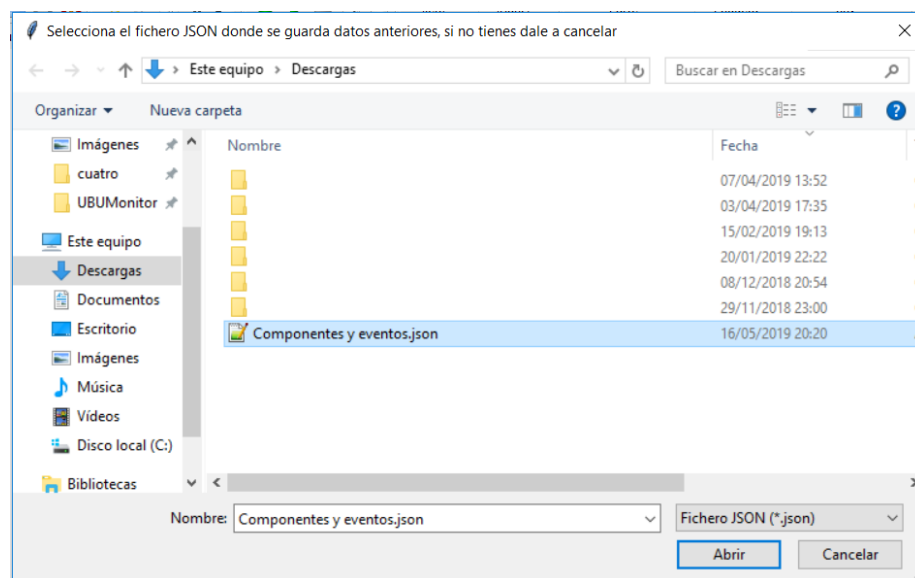


Figura D.21: Seleccionar el archivo JSON.

## APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN

4. En las siguientes celdas se parará en una que pide que copiemos los componentes del Resource Bundle en español desde:

/resources/messages/messages\_\_es.properties

### Copiar los componentes del Resource Bundle en Español ¶

1. Ir al fichero de Resource Bundle en Español y copiar los componentes

2. Pulsar al enter del input (el input bloquea el flujo de la ejecución)

3. Guardará en una variable lo que haya en el portapapeles

```
In [*]: #Ejecutar esta celda cuando tengas en el portapapeles los el listado de componentes en español
input("Pulsa enter cuando tengas en el portapapeles los componentes en español del resource bundle.")
pasted=root.clipboard_get()
pasted
Pulsa enter cuando tengas en el portapapeles los componentes en español del resource bundle.
```

Figura D.22: Celda que parará el flujo el ejecución.

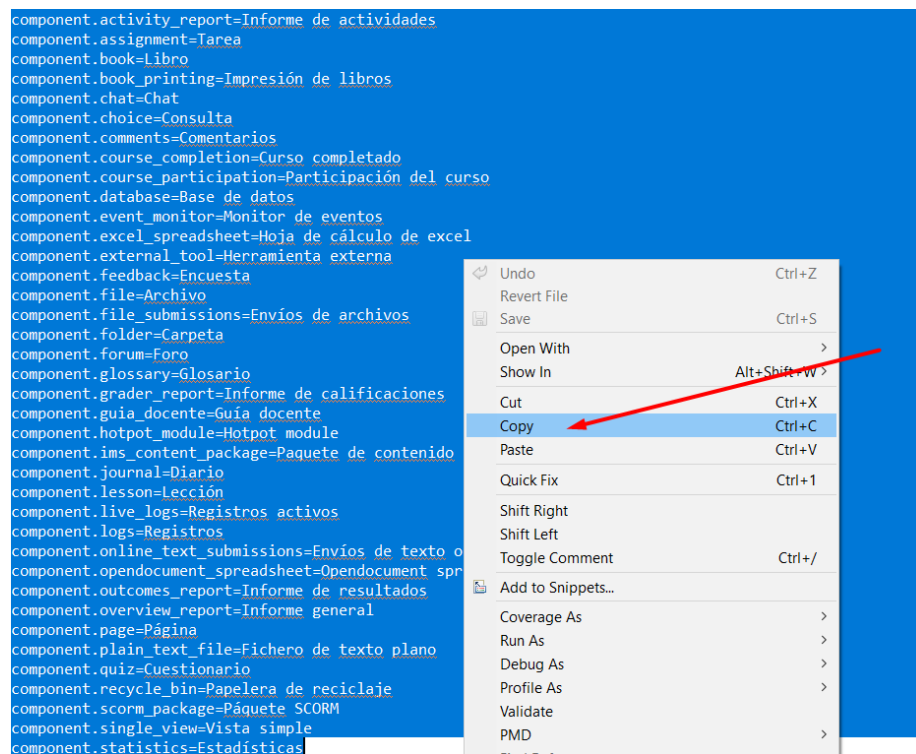


Figura D.23: Copiar los componentes que se encuentra en el Resource Bundle español.

5. Cuando lo hayamos copiado, en el cuadro de texto Notebook pulsamos la tecla **Enter** para que siga ejecutando las siguientes celdas.



6. Realizamos las mismas acciones pero ahora con los eventos del Resource Bundle en español.

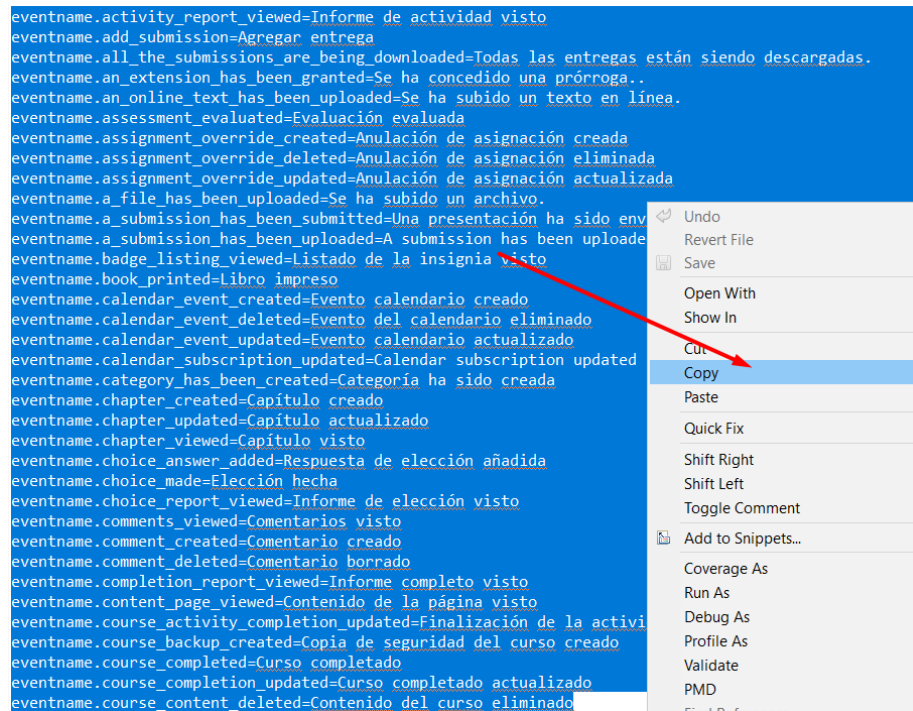


Figura D.24: Copiar los eventos que se encuentra en el Resource Bundle español.

7. Realizamos la misma acción con los tipos de logs que se encuentra en:  
**/controllers/ubulogs/logtypes/Logtypes.java**

## APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN

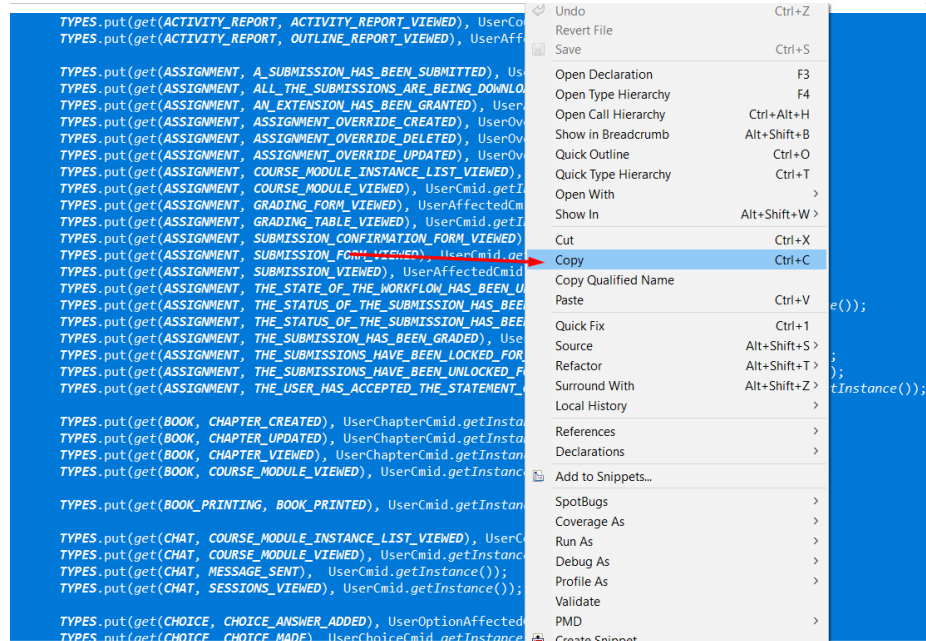


Figura D.25: Copiar los tipos de logs.

8. El último paso es solo ir copiando y pegando las salidas de las celdas al proyecto de Java.

### D.4. Compilación, instalación y ejecución del proyecto

#### Importar el proyecto a Eclipse

Lo primero de todo hay que descargar todo el ficheros fuentes desde el repositorio de GitHub y descargamos el ZIP:

<https://github.com/yjx0003/UBUMonitor>

Para importar en Eclipse seguimos los siguientes pasos:

1. Pulsar el menú **File** y elegir la opción **Import...**

#### D.4. COMPILACIÓN, INSTALACIÓN Y EJECUCIÓN DEL PROYECTO

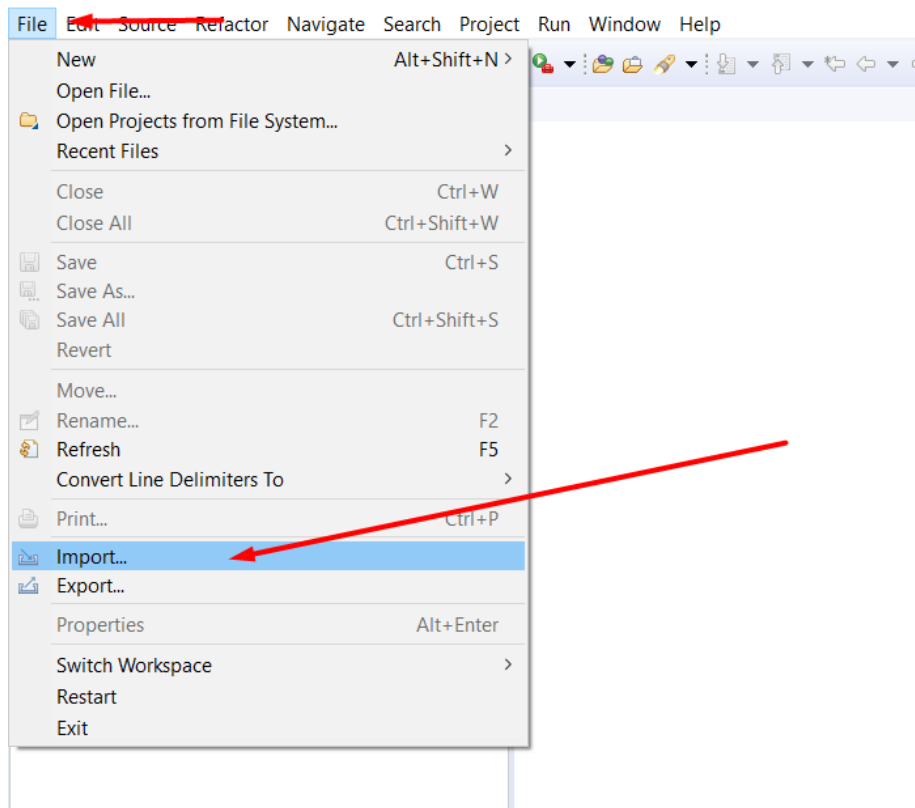


Figura D.26: Menú File y opción Import...

2. Seleccionar la opción de **Existing Projects into Workspace** y luego al botón **Next**.

## APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN

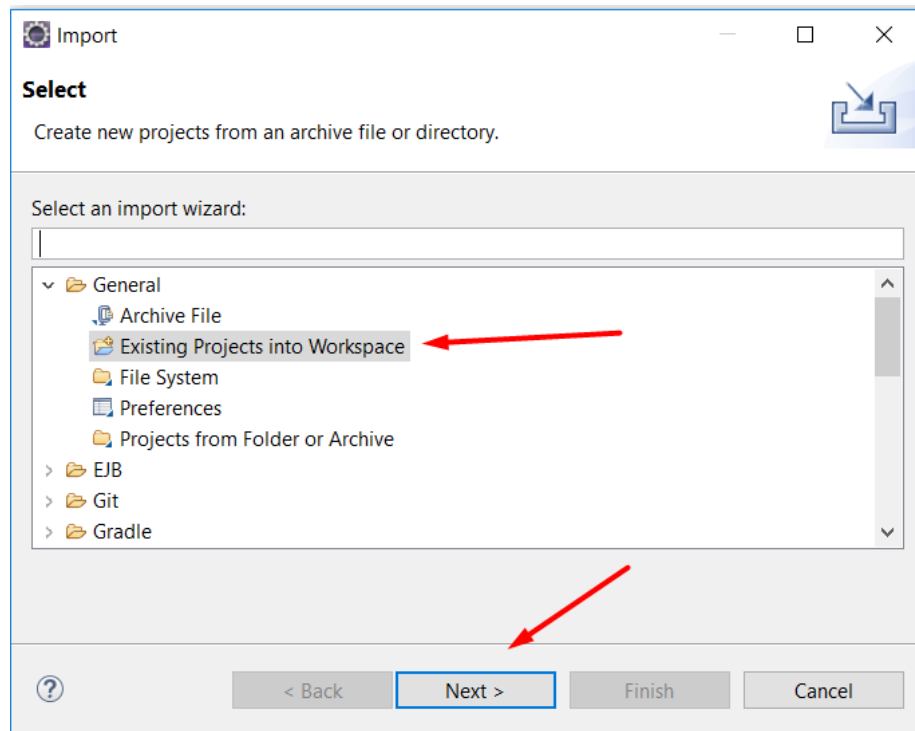


Figura D.27: Existing Projects into Workspace

3. Marcamos la opción **Select archive file** y seleccionamos el fichero ZIP descargado previamente del repositorio GitHub. Pulsamos Finish e importara el fichero.

#### D.4. COMPILACIÓN, INSTALACIÓN Y EJECUCIÓN DEL PROYECTO

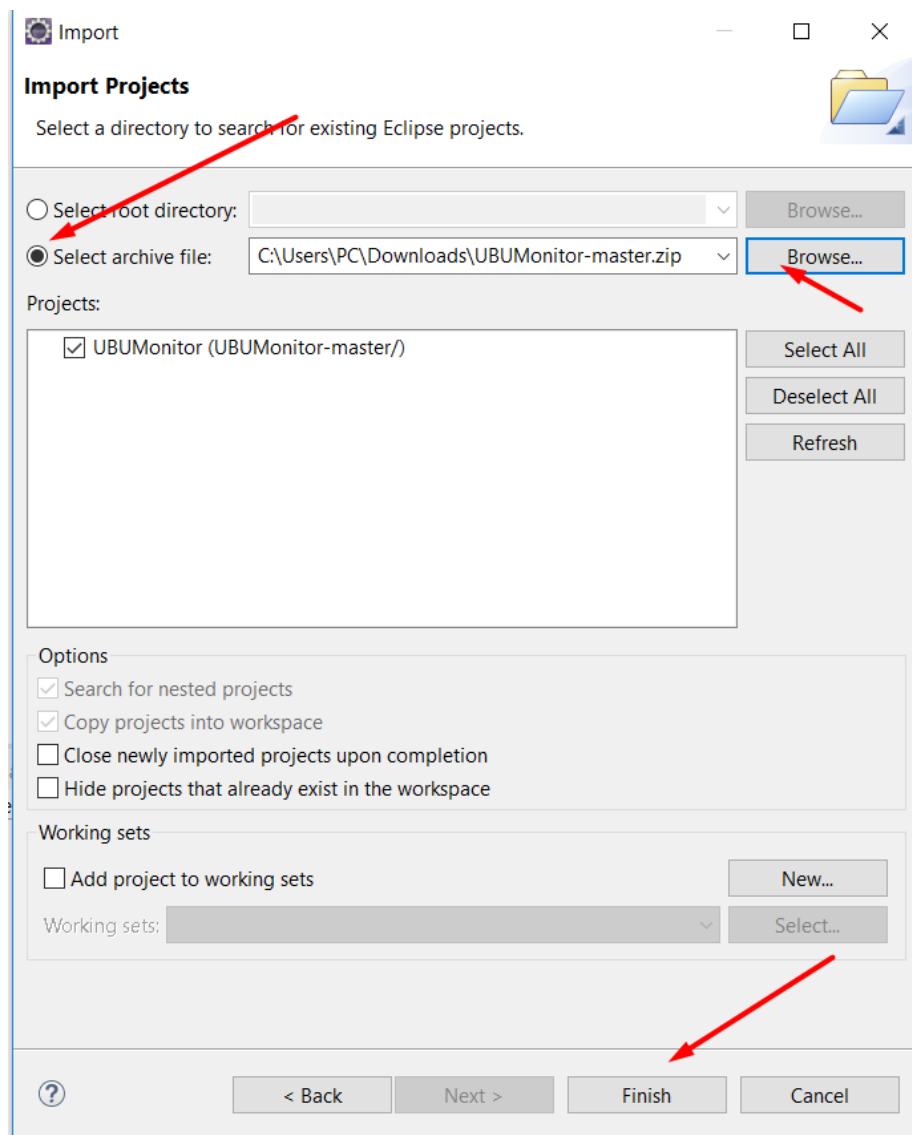


Figura D.28: Selección del archivo ZIP e importar proyecto.

### Generación del ejecutable de la aplicación

Para generar el ejecutable jar de la aplicación debemos seleccionar la opción **Exportar...** con click derecho del ratón al proyecto.

#### APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN

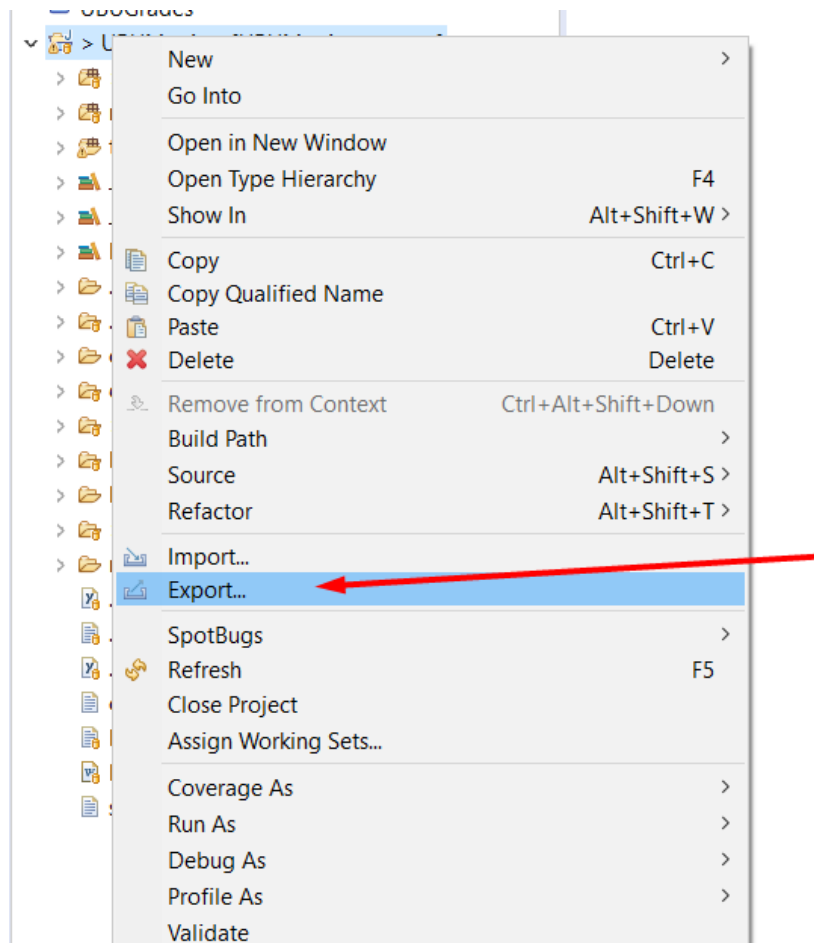


Figura D.29: Exportar jar

Seleccionamos la exportación del proyecto como **Runnable JAR file** dentro de la carpeta Java.

#### D.4. COMPILACIÓN, INSTALACIÓN Y EJECUCIÓN DEL PROYECTO

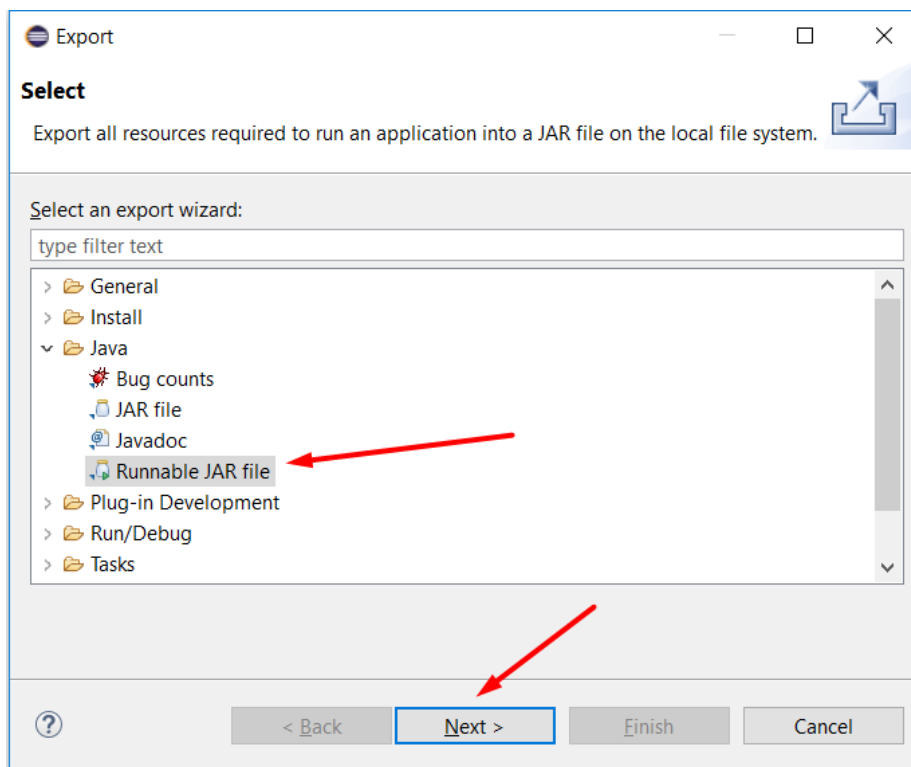


Figura D.30: Seleccionar Runnable JAR file.

Seleccionar al opción **Extract required libraries into generated JAR** y exportar.

## APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN

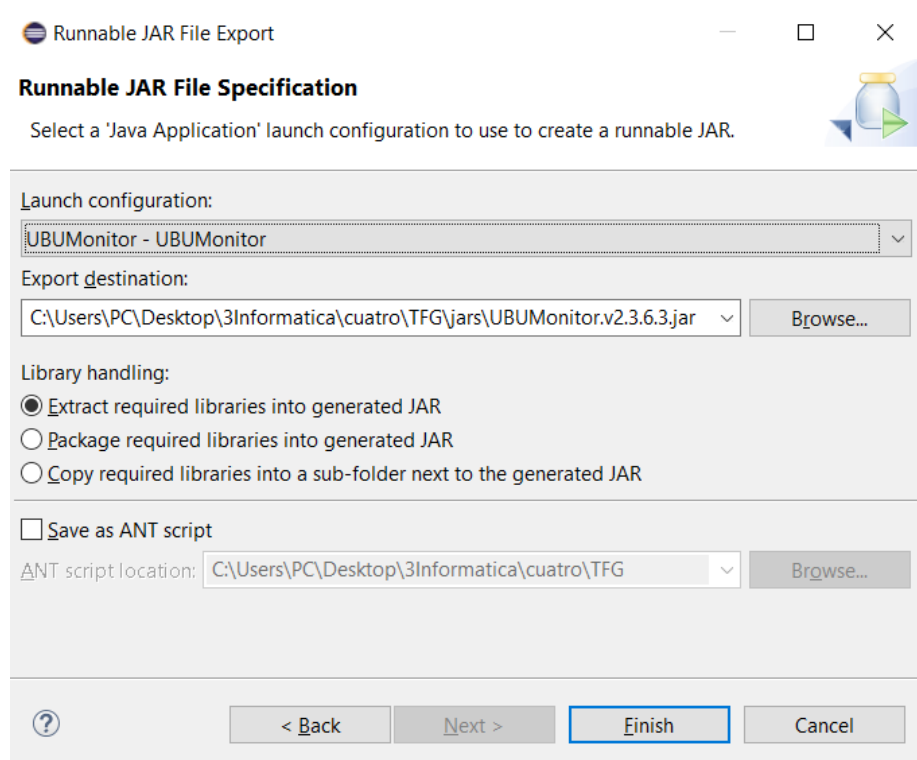


Figura D.31: Configuración del Runnable JAR.

### Ejecución de la aplicación

Para ejecutar el proyecto desde Eclipse: Buscamos el archivo java **UBU-Monitor** dentro del paquete **controllers** que es donde se encuentra el Main. Click derecho y **Run as... ->Java Application**



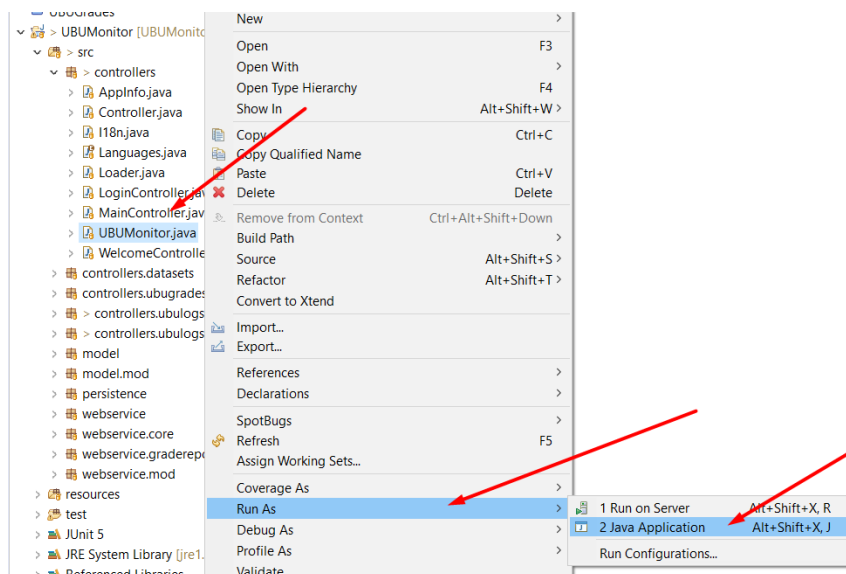


Figura D.32: Ejecución de la aplicación.

La ejecución del jar ejecutable es con doble *click* directamente.

## D.5. Pruebas del sistema

Las pruebas de las nuevas funcionalidades se realizaron contra la el Moodle instalado en el equipo en la versión **3.5.4+** y el servidor *demo* de Moodle **Mount Orange School** en la versión **3.7**.

Cuando se tenía unos cambios grandes, los tutores realizaban pruebas contra UBUVirtual con datos reales de alumnos. Con esto daban el visto bueno o informes de errores encontrados.



## *Apéndice E*

---

# Documentación de usuario

---

### E.1. Introducción

En este manual se explica los requerimientos de la aplicación, su ejecución en el ordenador e instrucciones de cómo usarlo correctamente.

### E.2. Requisitos de usuarios

- Tener instalado **Java 8**.
- Obtener el jar ejecutable de la última versión disponible en:  
<https://github.com/yjx0003/UBUMonitor/releases/latest>
- Debe disponer conexión a Internet.
- Una cuenta de Moodle con acceso a asignaturas con rol de Profesor.

### E.3. Instalación

La aplicación no necesita instalar nada más a parte de poseer Java 8.

### E.4. Manual del usuario

Después de ejecutar el jar previamente descargado, inicialmente nos mostrará una pantalla de inicio. Este manual de usuario también se encuentra en:

## Inicio de sesión

En esta primera pantalla es la de inicio de sesión. Ahí introducimos las credenciales de usuario, contraseña y en el campos host la URL de la página web de la plataforma Moodle.

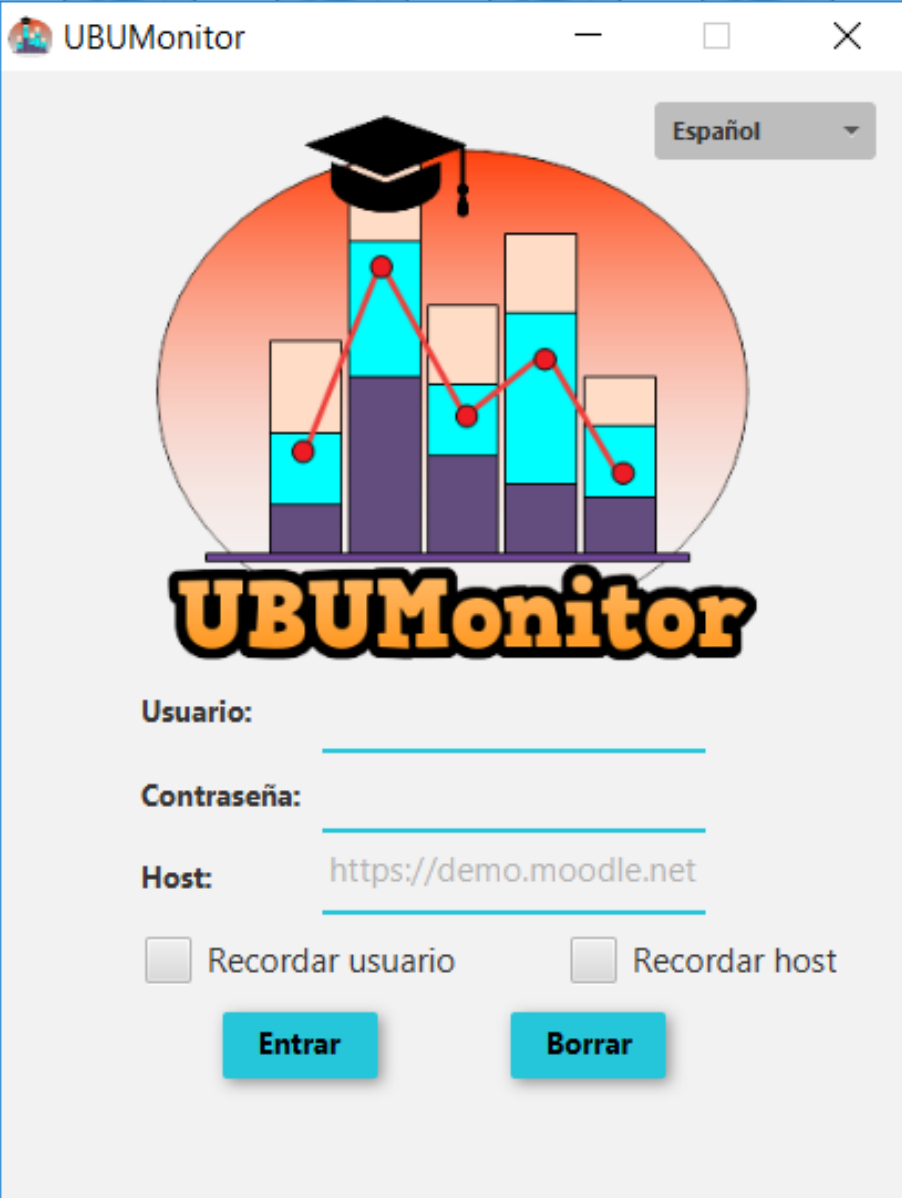


Figura E.1: Primera pantalla de inicio de sesión.

Da la opción de recordar el usuario y host para los siguientes usos de la

aplicación. Esa información se guarda en un fichero **config.properties** si marcamos una o ambas opciones.

En la esquina superior derecha nos permite cambiar el idioma de la aplicación y el botón **Borrar** para eliminar el texto de los tres campos.

Después de introducir las credenciales pulsamos el botón de **Entrar**. Si los datos son válidos cargará la siguiente pantalla y si no mostrará un mensaje de error.

## Elección del curso

En esta pantalla muestra un listado de cursos que está matriculado el usuario y permite seleccionar uno de ellos y acceder a la siguiente pantalla. Cuando se elige uno de los curso cambia la información de fecha de la última actualización en local. Esta fecha se muestra en función del fichero local que este guardado en el directorio **cache** con la opción de marcar o no la casilla de actualizar los datos. En caso contrario indica el texto **Nunca** y se deshabilita marca la casilla.

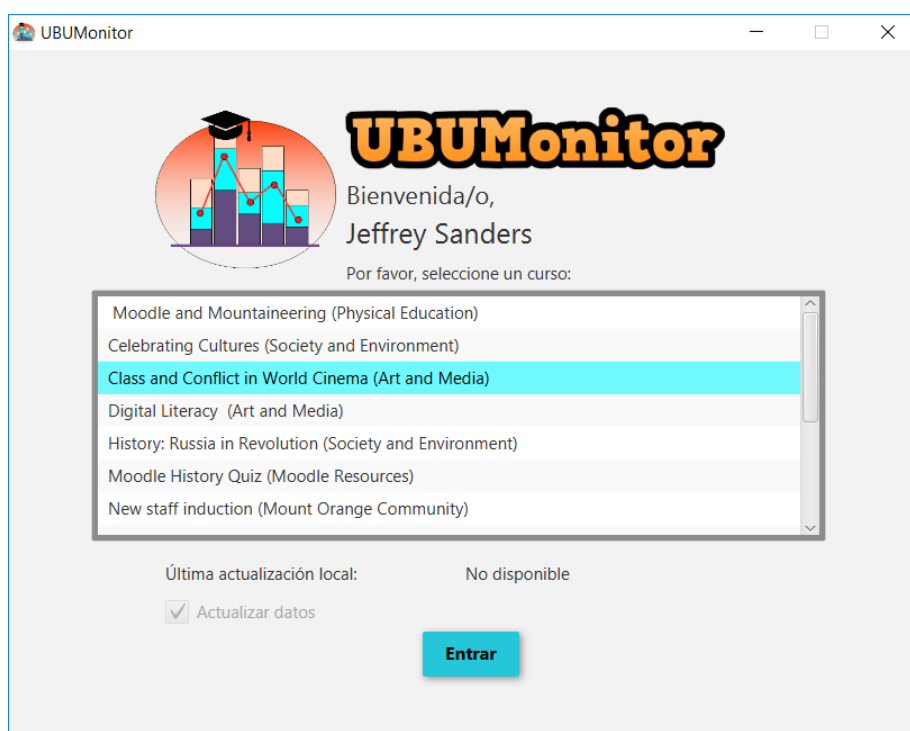


Figura E.2: Pantalla de selección de curso

Cuando se haya seleccionado un curso y pulsamos en el botón **Entrar**, si la casilla de actualizar los datos está seleccionado, carga fichero en local o crea en caso de que no exista y actualiza los datos del curso (usuarios matriculados, calificaciones, registros, etc...). El fichero local se **cifra** usando la contraseña de Moodle como clave. Este proceso puede tardar varios minutos si los registros son muy grandes. Si no está marcado descifra el fichero y lo carga de forma instantánea.

Carga la siguiente después de finalizar cualquiera de los dos procesos.

### Contraseña modificada

En caso de que se haya modificado la contraseña de Moodle, muestra una ventana emergente de que se ha detectado un cambio en la contraseña y que pide la anterior que tuvieras en la fecha señalada.

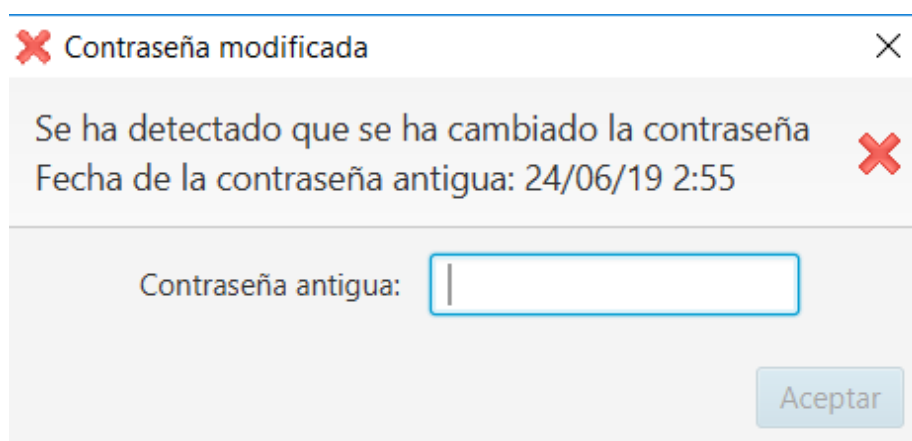
Una ventana emergente con un título que dice 'Contraseña modificada' precedido por un icono de 'X' roja. El cuerpo de la ventana contiene el mensaje 'Se ha detectado que se ha cambiado la contraseña' y 'Fecha de la contraseña antigua: 24/06/19 2:55', también precedido por un icono de 'X' roja. Debajo de este mensaje hay un campo de texto etiquetado 'Contraseña antigua:' con un cursor parpadeante. En la esquina inferior derecha hay un botón gris con el texto 'Aceptar'.

Figura E.3: Ventana emergente pidiendo la contraseña antigua.

Cuando se haya introducido correctamente la contraseña antigua, el fichero en cache local se guarda con la nueva.

## E.5. Visualización de gráficas

Ventana principal de visualización de gráficas del calificador y de los registros.

En esta ventana muestra los diferentes menús existentes, la lista de usuarios matriculados y las elecciones de diferentes gráficos.



Figura E.4: Pantalla inicial de la aplicación.

## Barra de herramientas

En la parte superior de la ventana muestra una barra de herramientas con diferentes opciones:

### ■ Archivo

- Cambiar asignatura: vuelve a la ventana de Elección del curso
- Guardar gráfico como...: Permite guardar en formato PNG el gráfico actual
- mostrado en la aplicación.
- Guardar todo como...: Permite guardar en formato HTML todos los gráficos.
- Cerrar sesión: Cierra la sesión y vuelve a la pantalla de inicio.
- Salir: Cierra la aplicación.

### ■ Editar

- Borrar selección: Quita la selección de todos los listados disponibles.

### ■ Ayuda

- Acerca de la aplicación: Abre en la ventana del navegador el proyecto de GitHub.

También muestra los datos del usuario actual, el curso seleccionado y la página de la plataforma de Moodle (host).

### Listado de usuarios matriculados

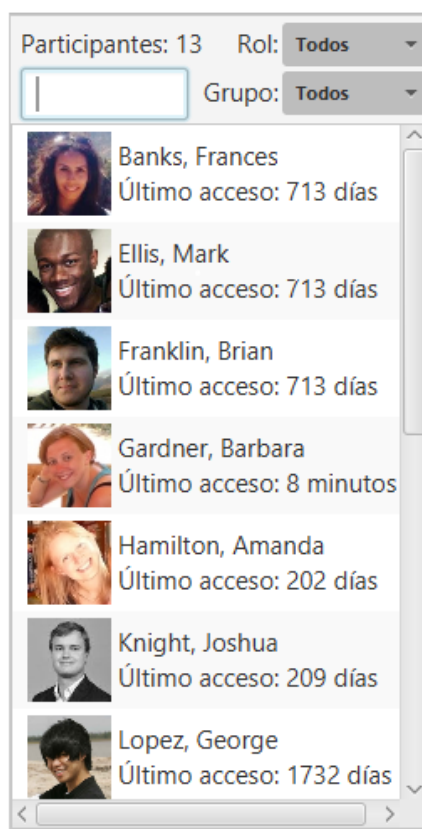


Figura E.5: Listado de usuario matriculados en el curso.

En este apartado se muestra el listado de los usuarios matriculados en el curso junto con su fotografía y el tiempo desde su último acceso a la plataforma de Moodle ya sea en minutos horas o días. Se toma como referencia la hora del sistema. Es de selección múltiple, elegir varios usuarios mientras se mantiene pulsando la tecla Control.

También da la opción de aplicar tres filtros juntos de forma conjunta:

- Campo de texto: filtrar el listado por nombre y/o apellidos.
- Selector de rol: seleccionar el rol por el que se quiere filtrar.
- Selector de grupo: grupo por el que se quiere filtrar.



## Gráficas de calificaciones

### ▪ Listado de elementos del calificador (Grade Items)

- Listado de elementos del calificador (Grade Items).

En la pestaña de calificaciones muestra la jerarquía del calificador junto con sus filtros. Es de selección múltiple. En cualquier momento se puede desplegar y hacer más grande esta zona para ver mejor los nombres.

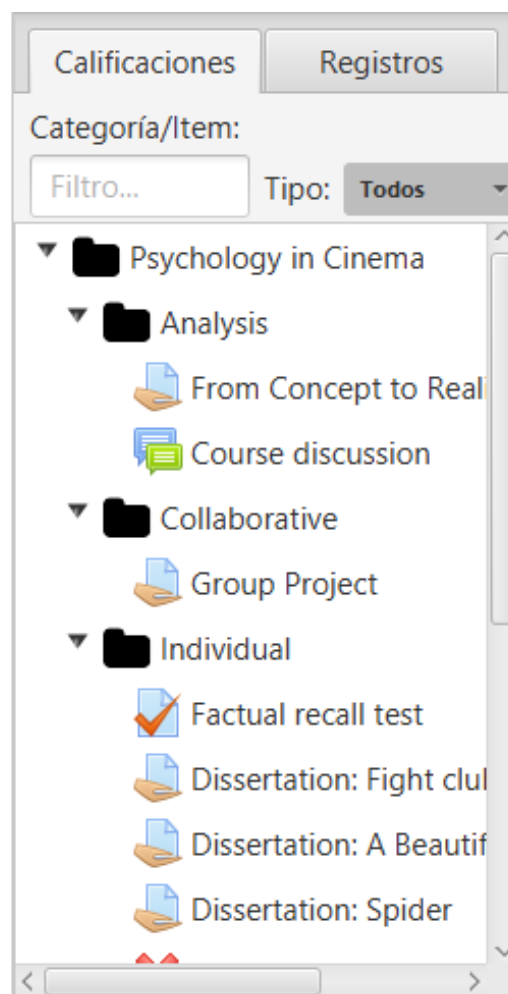


Figura E.6: Listado de calificaciones.

Existe dos tipos de filtros que se usan de modo conjunto:

- Campo de texto: por nombre del elemento del calificador.

- o Selector de tipo: según el tipo que sea (foro, tarea, cuestionario, et...)

También tiene varias pestañas de ocultar la leyenda, media general y media del grupo.

### ■ Gráfico de líneas

Muestra un gráfico de líneas donde se representan las notas de los alumnos para cada elemento del calificador seleccionado.

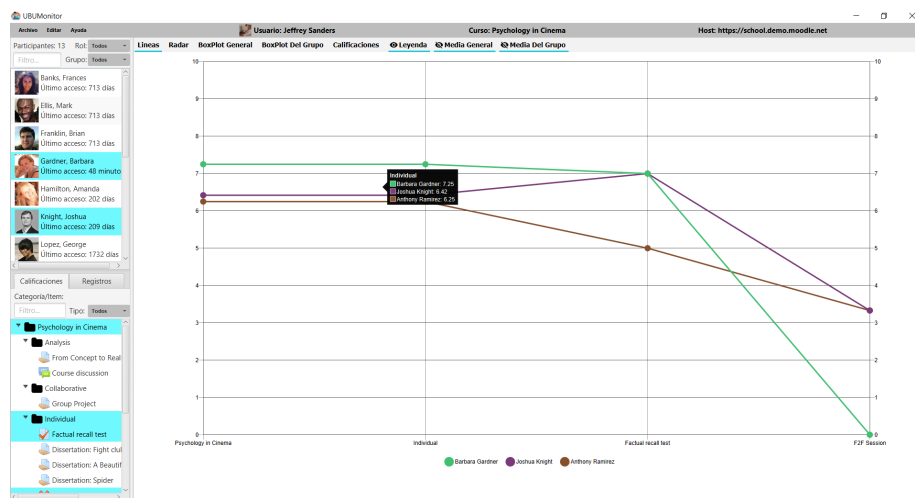


Figura E.7: Gráfico de líneas de las calificaciones.

### ■ Gráfico radar

Gráfico de tipo radar o también conocido como diagrama de araña. Muy útil para comparar dos participantes del curso.

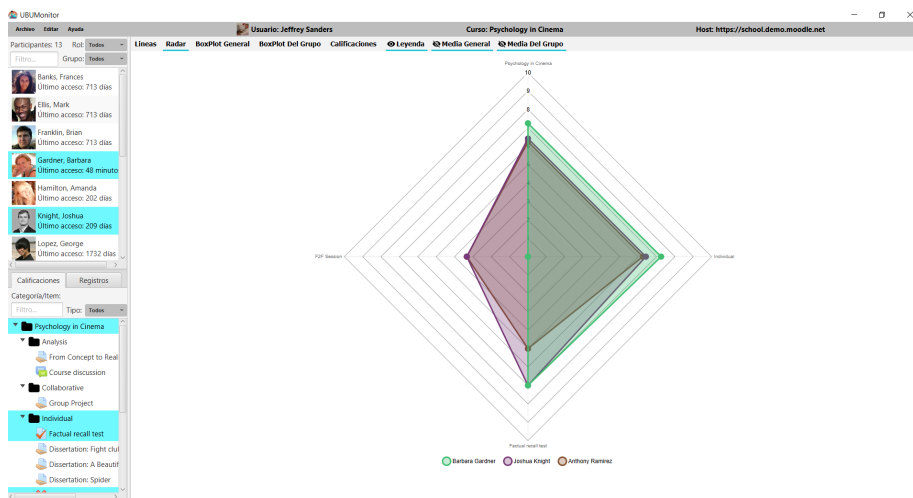


Figura E.8: Gráfico radar de las calificaciones.

### ■ Gráfico BoxPlot general

Muestra la información de un boxplot o diagrama de caja y bigotes mediante un gráfico de líneas. En este gráfico representa también los máximos, mínimos, la mediana así como los cuartiles primero y tercero.

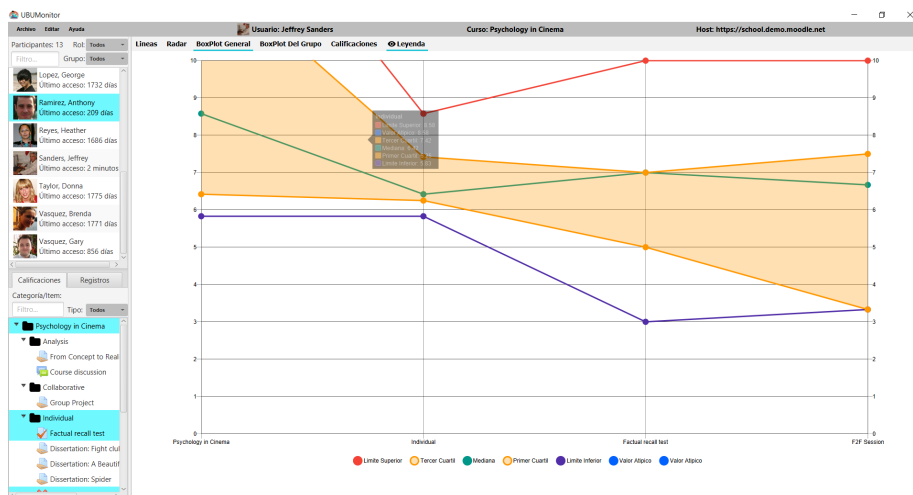


Figura E.9: Gráfico de BoxPlot general de las calificaciones.

### ■ Gráfico BoxPlot grupo

Muestra la misma información que el gráfico anterior pero para un grupo concreto. Para que este gráfico sea visible debemos seleccionar primero un grupo.

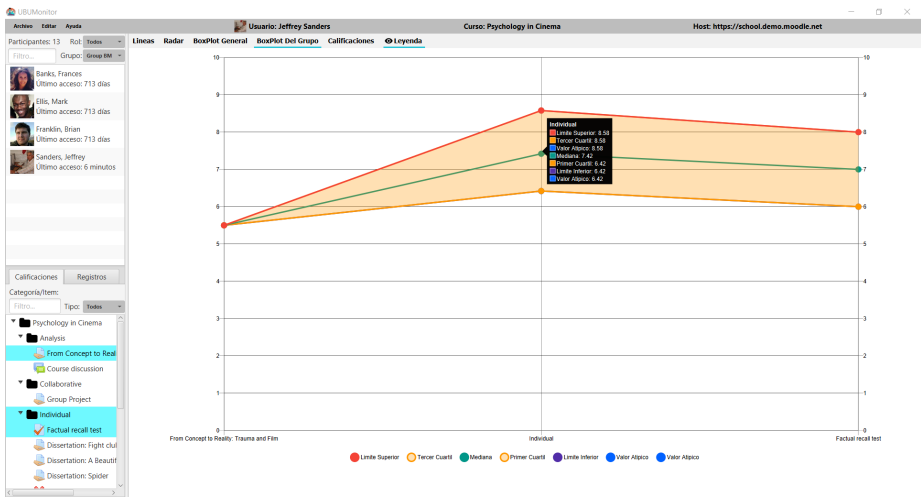


Figura E.10: Gráfico BoxPlot de grupo de las calificaciones.

■ **Tabla de calificaciones**

Muestra una tabla con las calificaciones así como la media general y la media de cada uno de los grupos del curso.

Lineas	Radar	BoxPlot General	BoxPlot Del Grupo	Calificaciones
				Notas
				From Concept to Reality: Trauma and Film
				Individual
				Factual recall test
Mark Ellis				55.0/100.0
Brian Franklin				77.0/120.0
Barbara Cardiner				103.0/120.0
Media				87.0/120.0
Media Grupo Group FC				6.89/10
Media Grupo Group IM				6.55/10
Media Grupo Group S				7.17/10
				6.50/10
				7.47/10
				6.19/10
				6.25/10

Figura E.11: Tabla de calificaciones de tres personas y las medias del grupo y general.

**Gráficas de los registros**

■ **Opciones de los registros**

Al pulsar la pestaña de Registros aparece varias opciones en la pantalla inferior.

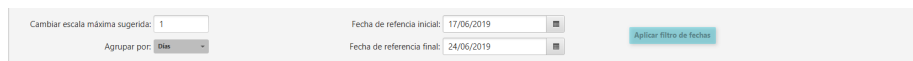
The image shows a horizontal filter bar with a light gray background. On the left, there is a text label 'Cambiar escala máxima sugerida:' followed by a text input field containing the number '1'. Below this, there is a label 'Agrupar por:' followed by a dropdown menu showing 'Día'. To the right of these, there are two date pickers. The first is labeled 'Fecha de referencia inicial:' and shows '17/06/2019'. The second is labeled 'Fecha de referencia final:' and shows '24/06/2019'. On the far right of the bar is a blue button with white text that says 'Aplicar filtro de fechas'.

Figura E.12: Opciones para las gráfica de registros.

- Cambiar escala máxima sugerida: permite modificar la escala máxima del eje Y. Se actualiza los valores al cambiar los componentes y/o eventos seleccionados o las otras opciones. Aporta información útil también al mostrar que existe un usuario con esa escala.
- Agrupar por: agrupa los registros en diferentes formas de tiempo (día, mes, año, etc...).
- Fecha de referencia inicial: la fecha de inicio orientativo que se muestra al agrupar.
- Fecha de referencia final: fecha límite orientativo que se muestra al agrupar.

Hay que tener en cuenta que la fecha de referencia inicial y final no toman todos los datos al realizar las agrupaciones. Por ejemplo al agrupar por meses y una fecha inicial del 17/06/2019 a 24/06/2019 no mostrará los registros entre esos dos días si no que mostrara todas las de junio. También las agrupaciones por Hora y Día de la semana no usa el selector de fechas, por lo tanto se deshabilitan.

#### ■ Componente

Lista los registros del curso en función del Componente que sea. Solo se visualizan componentes que hayan generado los usuarios matriculados en el curso actualmente.

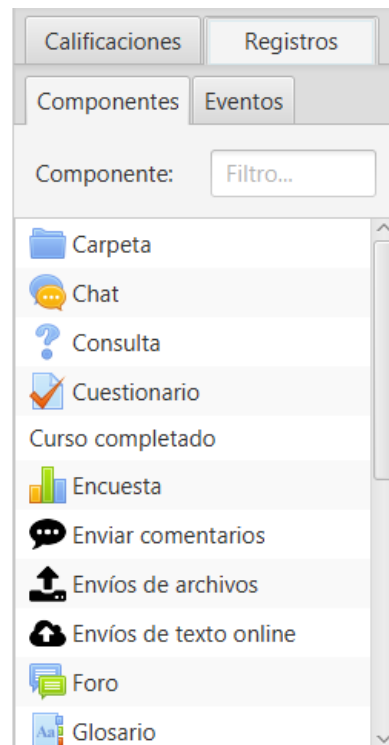


Figura E.13: Listado de componentes ordenados alfabéticamente.

#### ■ Evento

Lista los registros del curso en función del Componente y Evento que sea. Se muestran solo los componentes y eventos generados por los usuarios actuales matriculados en el curso.

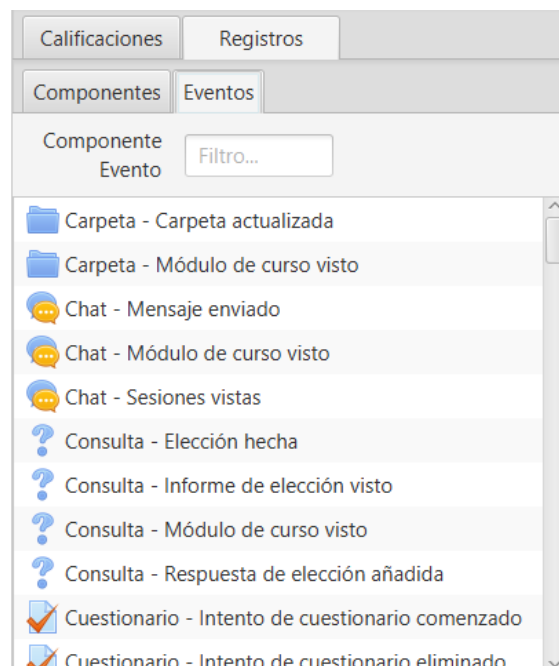


Figura E.14: Listado de componente y evento ordenados alfabéticamente por componente.

#### ■ Gráfica de barras apiladas

Para ambos tipos de subpestañas (Componentes y Eventos) la gráfica que se usa para mostrar los registros del curso son barras apiladas. Muestra en barras cada uno de los usuarios seleccionados con los componentes y/o eventos apilados con diferentes colores cada uno. También se apilan las líneas que indican el valor medio de los usuarios filtrados en ese momento.

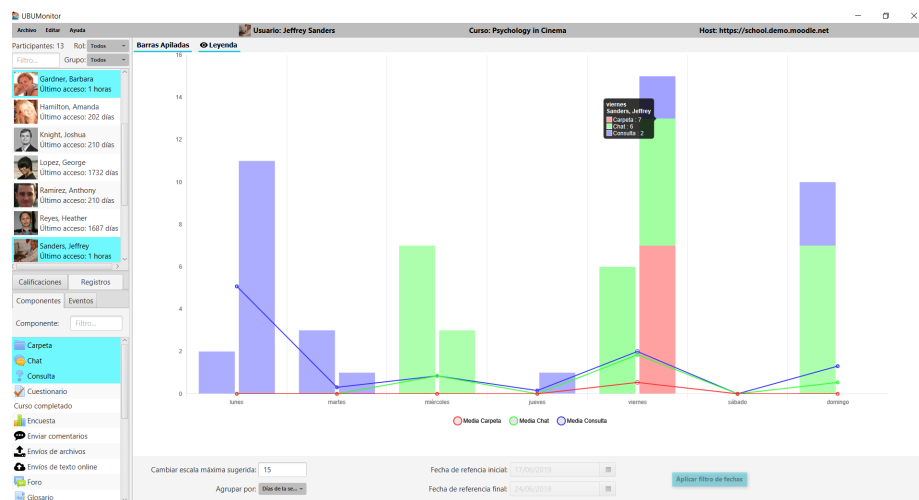


Figura E.15: Gráfica de barras apiladas con dos participantes y tres componentes agrupados por día de la semana.



---

## Bibliografía

---

- [1] *Font Awesome*. URL: <https://fontawesome.com/> (visitado 23-06-2019).
- [2] *Frequently Asked Questions / Charts*. en. URL: <https://developers.google.com/chart/interactive/faq?csw=1#offline> (visitado 23-06-2019).
- [3] *java.net (Java Platform SE 8 )*. URL: <https://docs.oracle.com/javase/8/docs/api/java/net/package-summary.html> (visitado 22-06-2019).
- [4] *jsoup Java HTML Parser, with best of DOM, CSS, and jquery*. URL: <https://jsoup.org/> (visitado 22-06-2019).
- [5] Ediciones El País. *Calculadora sueldo neto*. es. URL: <https://cincodias.elpais.com/herramientas/calculadora-sueldo-neto/> (visitado 23-06-2019).
- [6] *¿Por qué debo desinstalar las versiones anteriores de Java del sistema?* URL: [https://www.java.com/es/download/faq/remove\\_olderversions.xml](https://www.java.com/es/download/faq/remove_olderversions.xml) (visitado 28-06-2019).
- [7] *Project Jupyter*. Abril de 2019. URL: <https://www.jupyter.org> (visitado 25-06-2019).
- [8] Margaret Rouse. *What is Advanced Encryption Standard (AES)? - Definition from WhatIs.com*. en. Marzo de 2017. URL: <https://searchsecurity.techtarget.com/definition/Advanced-Encryption-Standard> (visitado 22-06-2019).
- [9] *Salarios para empleos de Programador/a junior en España | Indeed.es*. Junio de 2019. URL: <https://www.indeed.es/salaries/Programador/a-junior-Salaries?period=yearly> (visitado 23-06-2019).

- [10] *Schneier on Security: The Blowfish Encryption Algorithm*. URL: <https://www.schneier.com/academic/blowfish/> (visitado 18-06-2019).
- [11] *¿Sobran universidades en España?* es. Febrero de 2017. URL: <https://www.universidadsi.es/sobran-universidades-espana/> (visitado 27-06-2019).