UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

TFG del Grado en Ingeniería Informática

GII 20.09 Herramienta web repositorios de TFGII Documentación Técnica

Presentado por Diana Bringas Ochoa
en Universidad de Burgos — 13 de abril de 2021
Tutor: Álvar Arnaiz González y Carlos López Nozal

Índice general

Índice de figuras

Índice de tablas

Apéndice A

Plan de Proyecto Software

A.1. Introducción

En esta sección se detallará la planificación que se ha realizado, el estudio de viabilidad tanto de la parte económica como de la legal.

A.2. Planificación temporal

Sprint 0 (26/10/2020 - 25/11/2020)

Puesta a punto del proyecto, planteamiento de las herramientas con las que trabajar, búsqueda de alternativas y toma de contacto con las herramientas nuevas que se van a emplear. Las tareas que se realizaron fueron:

- Añadir la extensión ZenHub al navegador Se añadió la extensión Zenhub al navegador de Google Chrome para utilizarlo en el GitHub.
- Clonar el repositorio en local. Mediante la aplicación GitHub Desktop se clono el repositorio del Gestor de TFGs mediante el enlace HTTP que proporciona GitHub.
- Investigar sobre Vaadin. A través de la página oficial de Vaadin se realizo la instalación y me informe acerca de Vaadin.
- Actualización del README.md Se modifico el README.md del proyecto para que reflejará los cambios respecto a la versión de la que se hizo fork.

- Investigar LaTeX Se procedió a buscar información sobre cómo instalar y manejar Latex para realizar la documentación del proyecto posteriormente.

Se puede ver el trascurso de estas tareas gráficamente en la siguiente ilustración A.1.

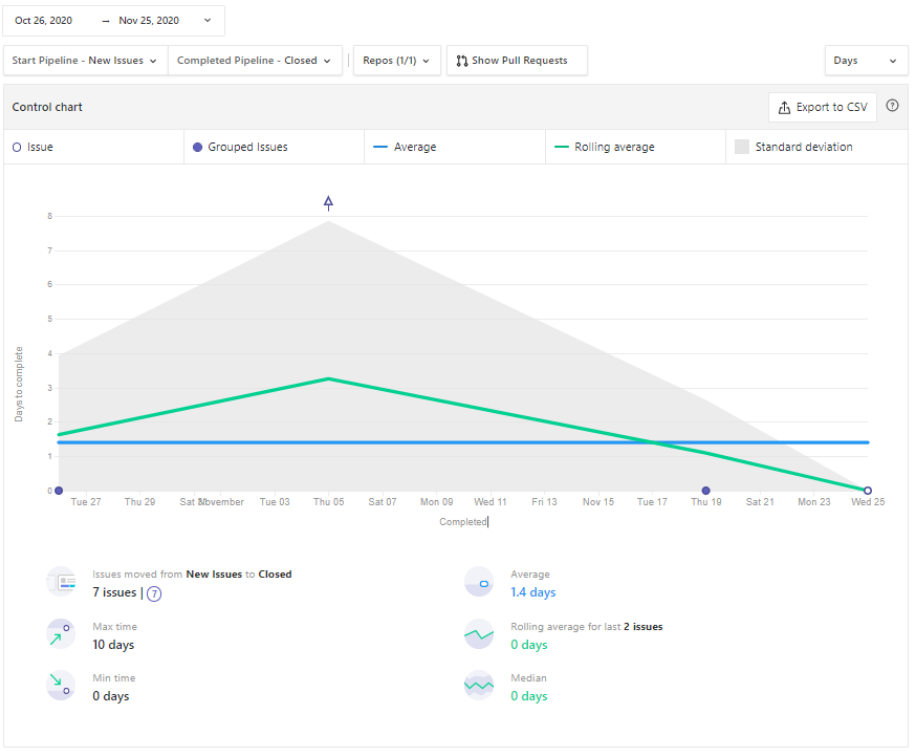

Figura A.1: Gráfica - Spring 0

Sprint 1 (25/11/2020 - 12/01/2021)

Generación de test unitarios, búsqueda de trabajos similares, cambio del driver para conectarse con el excel, información para obtener ideas de como realizar ciertas mejoras y comienzo de la documentación del proyecto. Mejora de la cobertura de la aplicación web.

Las tareas planteadas fueron:

- Instalación Miktex + TexStudio Tras la reunión del 25/11/20 se decicio que la mejor opción sería emplear un editor local, en vez del editor

online Overleaf. Para ello se instalo Miktex junto al editor de texto TexStudio.
- Se comienza la documentación en LaTeX - Spring 0 Creación de la documentación en LaTeX a partir de las plantillas.
- Generar nuevos test Para aumentar la covertura de la aplicación se definieron nuevos test.
- Cambiar driver JDBC Para poder usar los .ods se buscó una alternativa al driver para .csv que se empleaba.

Se puede ver el trascurso a través de la siguiente imagen A.2.

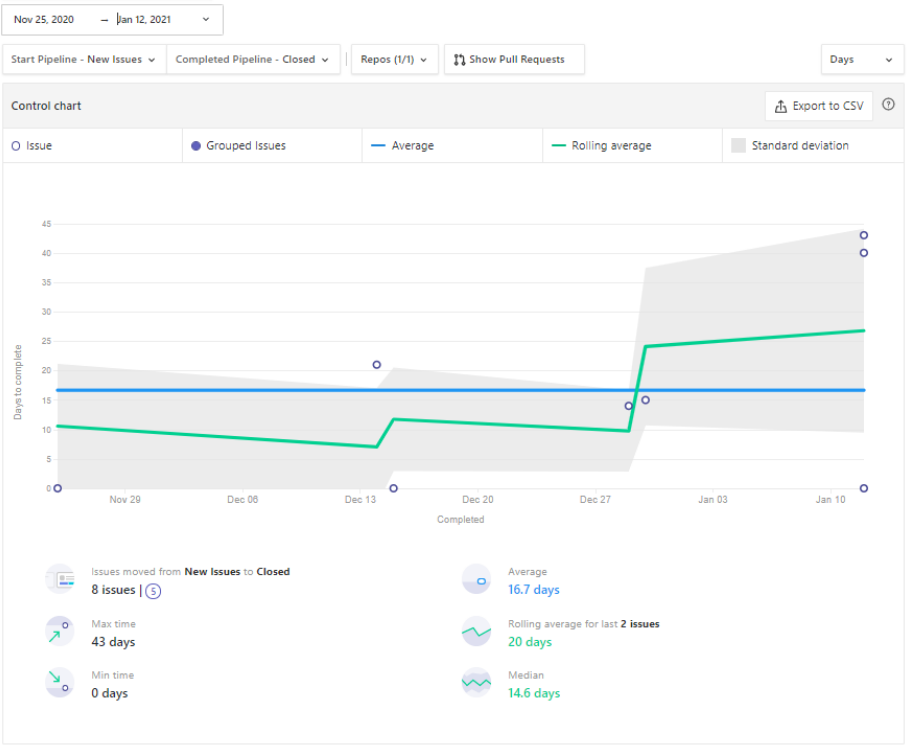

Figura A.2: Gráfica - Spring 1

Sprint 2 (12/01/2021 - 26/01/2021)

Búsqueda de nuevos driver que implementen JDBC para ficheros .xls. Prueba de opciones encontradas como Apache POI, SQLSheet, Fillo, Cdata JDBC for Excel y JdbcOdbcDriver. Integrar la API Fillo en el proyecto.

Las tareas planteadas fueron:

- Instalación del LibreOffice Se instaló Apache LibreOffice para manipular los fichero .xls y .csv que se emplean para la parte de datos de la aplicación.
- Cambiar de formato al fichero "**BaseDeDatosTFG.ods**". Se modificó el formato del fichero "**BaseDeDatosTFG.ods**" a "**BaseDeDatosTFG.xls**".
- Anexos - Modificación para poder cambiar el tamaño de las imágenes. Cambios en la plantilla de **anexos.tex** para poder modificar el tamaño de las imágenes al deseado.
- Probar el Apache POI Para verificar el funcionamiento de Apache POI, se incluyo en el proyecto de prueba **HolaMundoVaadin** y se realizaron varios ejemplos de prueba tomando como referente el proyecto principal.
- Error al intentar ejecutar la imagen del proyecto gabrielstan/gestion-tfg Bug realizado al intentar desplegar el proyecto gabrielstan/gestion-tfg.
- Desplegar proyectos relacionados con la gestión de TFG Proceso por el cual se probó a desplegar los proyectos relacionados con **GII 20.09 Herramienta web repositorios de TFGII**.
- Memoria - Mejoras Se realizaron varios cambios en la memoria.
- Incorporación del driver para fichero Excel Prueba de la API Fillo en el proyecto de prueba **HolaMundoVaadin**, y tras verificar su funcionamiento se procedió a incluirlo en el proyecto principal.
- Modificación de los nombres de los fichero csv Se cambiaron los nombres de los ficheros .csv para que coincidieran con los nombres de las hojas del fichero .xls.
- Memoria - Documentación Spring 2 Comienzo de la documentación de lo realizado en el Spring 2. En el cual se detallan las tareas realizadas.
- Modificaciones en los test Para probar la incorporación de la API Fillo se modificaron el test "**SintInfDataTest**".
- Realización de cambios para el funcionamiento de la nueva conexión para los fichero .xls Para poder usar la API Fillo se han tenido que realizar multitud de cambios en el proyecto. Aunque tanto el "**CsvDriver**" y "**Fillo**" emplean lenguaje SQL, "**CsvDriver**" emplea JDBC puro mientras que "**Fillo**" usa funciones y clases propias.

La siguiente imagen A.3 muestra cómo se han desarrollado las tareas a lo largo del tiempo.

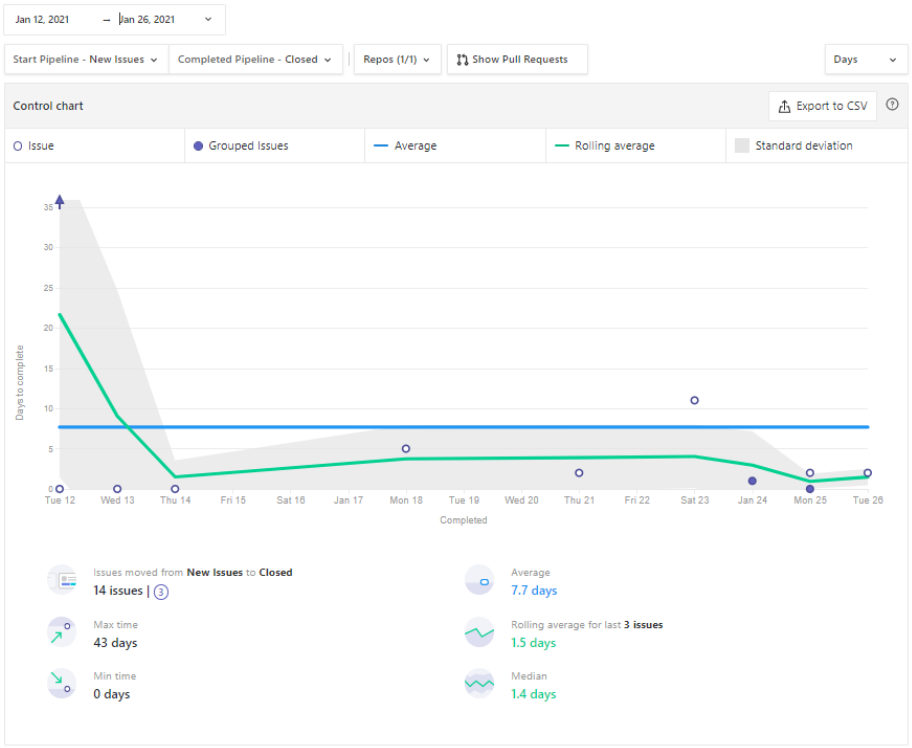

Figura A.3: Gráfica - Spring 2

Sprint 3 (26/01/2021 - 11/02/2021)

Las tareas planteadas fueron:

- Investigar opciones de hosting para el despliegue Se comparan varias opciones gratuitas para desplegar la aplicación, las cuales son: GitHub Pages, LucusHost, Awardspace, RunHosting, FreeHostia, X10Hosting y Heroku. Eligiendo la opción de Heroku, entre otras razones, porque proporciona la opción de conectar el despliegue con GitHub.
- Rediseño fachada y vistas Se elimina la parte vinculada a los datos de las vistas incluyéndola en a las clases fachada.
- Cambiar nombres a inglés. Se traducen los nombres de las variables, métodos y clases a inglés.
- Actualización de la Memoria y Anexos. Se realizan cambios y ampliaciones en el contenido de la documentación, en la Memoria y el Anexo.
- Instalación Heroku CLI Para realizar el despliegue del proyecto se instala el terminal de Heroku, denominado Heroku CLI. La instalación

se llevo a cabo según el tutorial de instalación de la página oficial de Heroku (https://devcenter.heroku.com/articles/heroku-cli).

- Incorporación del patron Factory Se crea una nueva clase con la función de seleccionar el tipo de acceso de datos, ya sea la clase fachada encargada de los ficheros .csv o, la que gestiona los ficheros .xls.
- Creación branch de prueba Rama del repositorio donde se almacena el proyecto de prueba formularioVaadin, el cual posteriormente será usado para realizar una prueba de despliegue en Heroku.
- Modificación de Test JUnit Se cambian los test para que sirvan para las funciones de la clase fachada para los ficheros .xls.
- Probar el despliegue del proyecto de prueba Se emplea el proyecto de prueba, formularioVaddin, para realizar una prueba de despliegue en Heroku. Se intentará realizarlo tanto, a través de Heroku CLI como, mediante la interfaz la página de Heroku.

En la siguiente imagen se enseña gráficamente el desarrollo de las issues A.4.

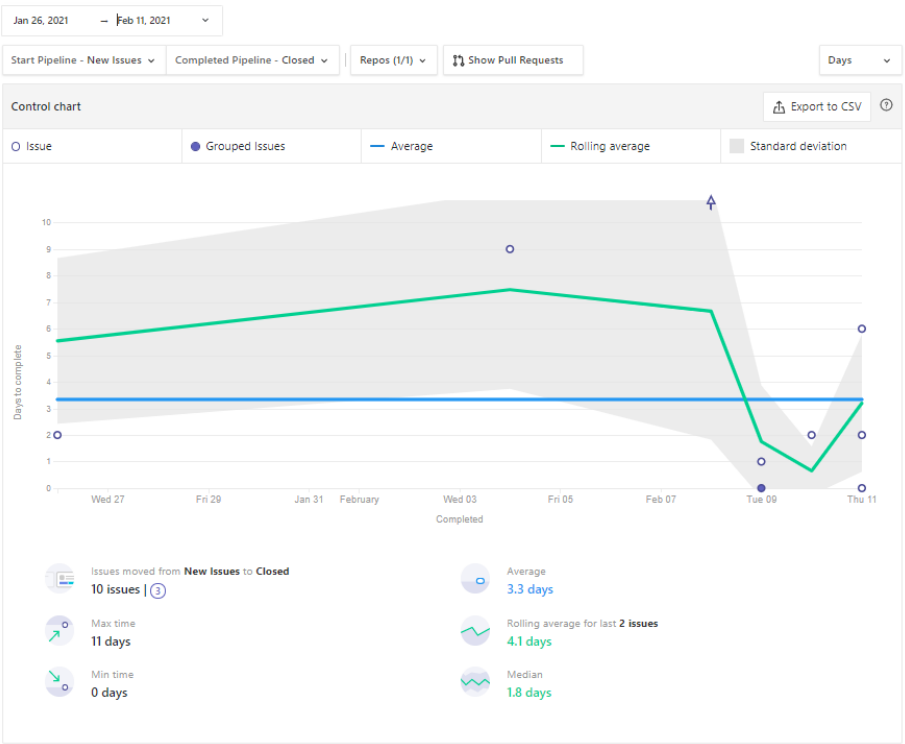

Figura A.4: Gráfica - Spring 3

Sprint 4 (11/02/2021 - 23/02/2021)

Se intentará incorporar la integración continua. Realizar el despliegue de la aplicación con Heroku. Añadir una columna de rankings en el Histórico. Cambiar las notas del histórico a privadas.

Las tareas planteadas fueron:

- Realizar la Integración Continua. A través de la opción GitHub Actions se incorporará la opción de compilar y ejecutar los test cuando se realiza un cambio (ya sea un push o un pull request) en el repositorio. Tiene como objetivo detectar fallos eficazmente mediante la integración automática frecuentemente de un proyecto.
- Actualización bibliografía Se incorporan los enlaces de las páginas o datos bibliográficos que se han empleado hasta ahora en la realización del proyecto.
- Cambiar las notas del Histórico a privadas Se elimina las notas de la columnas de "Notas.[en] la tabla de Históricos.
- Añadir dependencia Apache Commons Math
- Crear ranking de notas en Histórico
- Actualización del despliegue del proyecto de prueba en Heroku
- Mejora de test JUnit

A continuación, se expondrá una ilustración donde se aprecia el trascurso del Spring y el desarrollo de las tareasA.5.

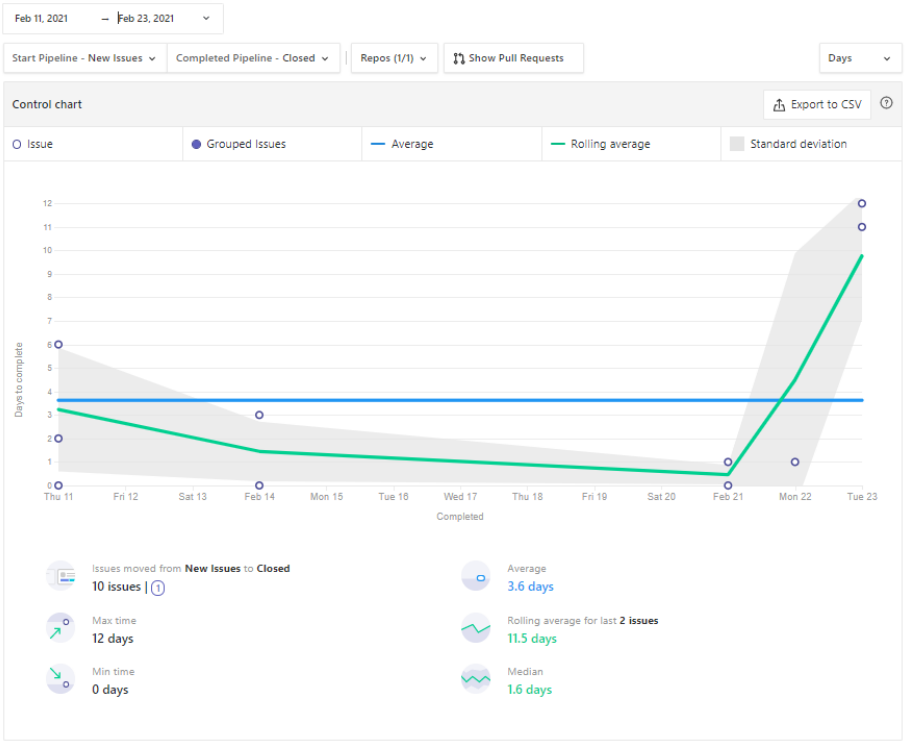

Figura A.5: Gráfica - Spring 4

A.3. Estudio de viabilidad

Viabilidad económica

En este apartado se detallan los costes que llevaría realizar este proyecto.

Coste del personal

Coste hardware

Referente a los costes del equipo utilizado en el desarrollo del trabajo. Teniendo en cuenta el precio del ordenador empleado de aproximadamente 700 euros.

Coste software

Referente a los costes de las herramientas software no gratuitas empleadas en el proyecto. Como es el caso del Sistema Operativo Windows o el Microsoft Office 365.

Viabilidad legal

En este apartado se detallaran las licencias de cada dependencia que se ha utilizado en el proyecto

Software	**Licencia**
Vaadin	Apache License 2.0
Vaadin Maven Plugin	Apache License 2.0

Tabla A.1: Dependencias del proyecto

Apéndice B

Especificación de Requisitos

B.1. Introducción

Los requisitos[**?**] del proyecto son los puntos clave que se deben cumplir.

B.2. Objetivos generales

El objetivo del proyecto es la mejora de la aplicación Web centrándose en los siguientes puntos:

- Incorporación y validación de un nuevo tipo de datos (XLS)
- Realización de la métricas con los resultados del análisis de la calidad del código de los TFG (SonarCloud)
- Mejoras en la estética de la aplicación Web.
- Modificación del Login para conectarse con la cuenta de la Universidad.

B.3. Catalogo de requisitos

Se describirán los requisitos específicos, funcionales y los no funcionales.

B.4. Especificación de requisitos

Requisitos funcionales

- **RF-1 Adicción nueva capa de datos**: la aplicación permite emplear dos tipos de archivos, xls y csv, como entrada de datos.
  - **RF-1.1 Subida de datos:** el usuario que tenga permisos podrá subir archivos de datos, tanto en el formato xls como en el csv.
- **RF-2 Login**: el sistema permitirá la subida de datos a ciertos usuarios que se autentificarán a través del login de la aplicación.

Requisitos no funcionales

- **RNF-1 Seguridad**: la aplicación solamente debe permitir la subida de datos a los usuarios con permisos.
- **RNF-2 Mantenibilidad**: mejora de la aplicación para permitir la escalabilidad y incorporación de nuevas modificaciones en el futuro de forma sencilla. Un ejemplo de esto sería la actualización de Vaadin a una versión nueva para conservar el soporte en el futuro.
- **RNF-2 Mejora diseño**: se realizarán mejoras para que la aplicación sea más atractiva e informativa. Se optará siempre por opciones intuitivas y sencillas de utilizar.

Apéndice C

Especificación de diseño

C.1. Introducción

C.2. Diseño de datos

C.3. Diseño procedimental

C.4. Diseño arquitectónico

Apéndice D

Documentación técnica de programación

D.1. Introducción

En esta sección se explica la estructura del proyecto, el proceso de instalación del framework y herramientas necesarias para desarrollar el trabajo, cómo realizar la compilación, instalación y ejecución del proyecto y las pruebas que se han llevado a cabo.

D.2. Estructura de directorios

D.3. Manual del programador

A continuación se explicará cómo realizar la instalación de los programas necesarios para el desarrollo de la aplicación.

Instalación de Java

Como en el proyecto se usa Vaadin 8, se debe emplear la **versión 8 de Java**, en concreto se ha usado la jdk1.8.0_271.

Para ello se deberá ir a la página de descargas de Oracle Java SE 8.0 y descargar la versión de JDK 8 correspondiente con tu sistema operativo y su arquitectura, ya sea de 64 o 32 bits. Ver imagen D.1.

Tras escoger la versión según nuestro SO se deberán de leer y aceptar las licencias de uso de Oracle, como se muestra en la figura D.2, y dar a descargar.

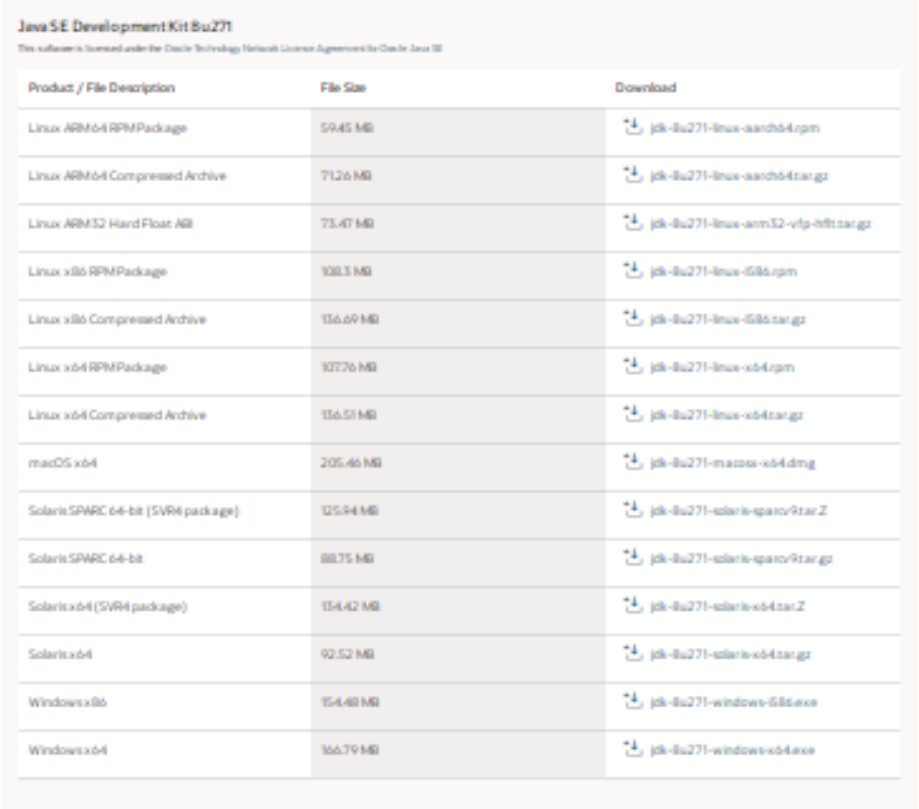

Figura D.1: Descarga de JDK 8

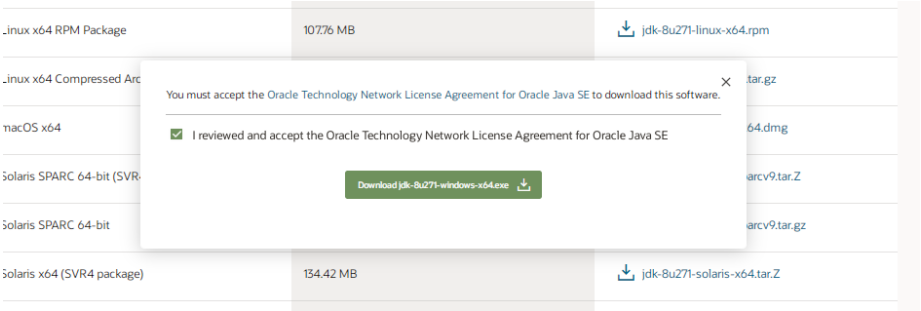

Figura D.2: Descarga JDK 8 Licencia

Una vez descargado, se deberá ejecutar el instalador y seguir el proceso de instalación del asistente.

Instalación de Eclipse

A continuación se instalará un entorno de desarrollo integrado(IDE) para Java, en este caso se ha utilizado **Eclipse IDE for Enterprise Java Developers** en la versión 2020-06.

Para descargar el IDE se accederá a la página de descargas de Eclipse y descargar la opción correspondiente a nuestro sistema operativo del **Eclipse Installer 2020-06 R**. Ver imagen D.3.

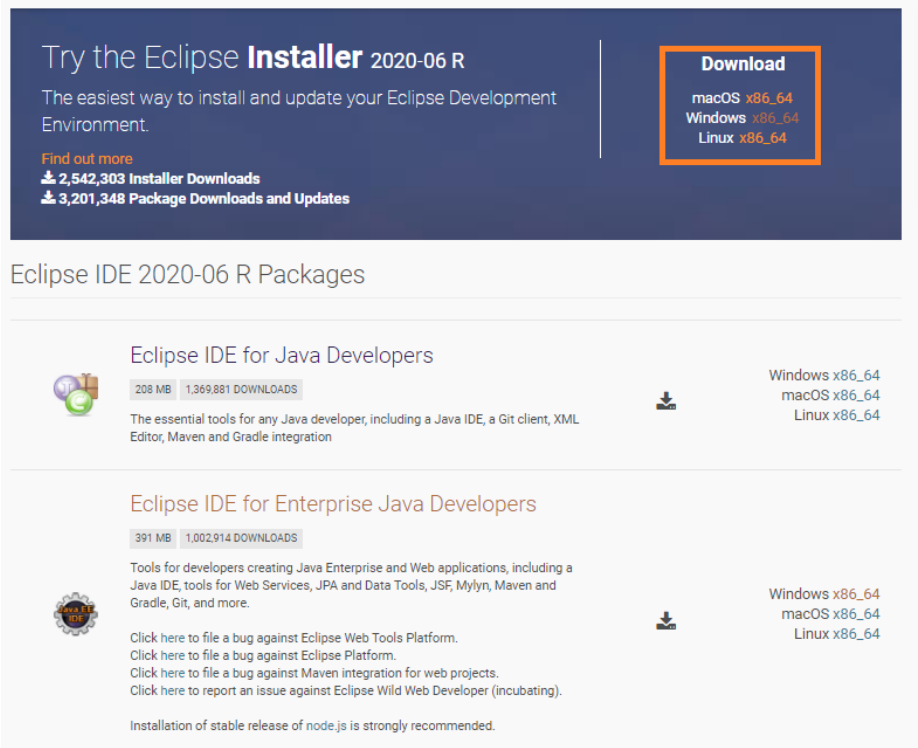

Figura D.3: Descargar IDE Eclipse

En el caso de los sistemas operativos Windows se descargará un archivo ejecutable que se deberá ejecutar como administrador. Una vez ejecutado se deberá seleccionar la opción "***Eclipse IDE for Enterprise Java Developers***".

En el siguiente paso, en el apartado de "**Java 1.8 + VM**" se deberá seleccionar la carpeta donde se encuentra el JDK 8, instalado anteriormente.

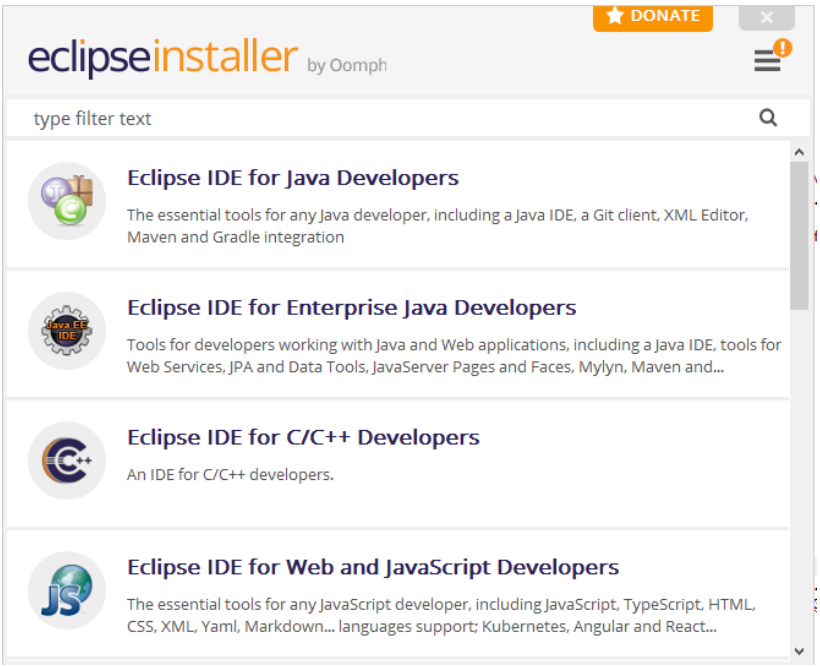

Figura D.4: Seleccionar Eclipse

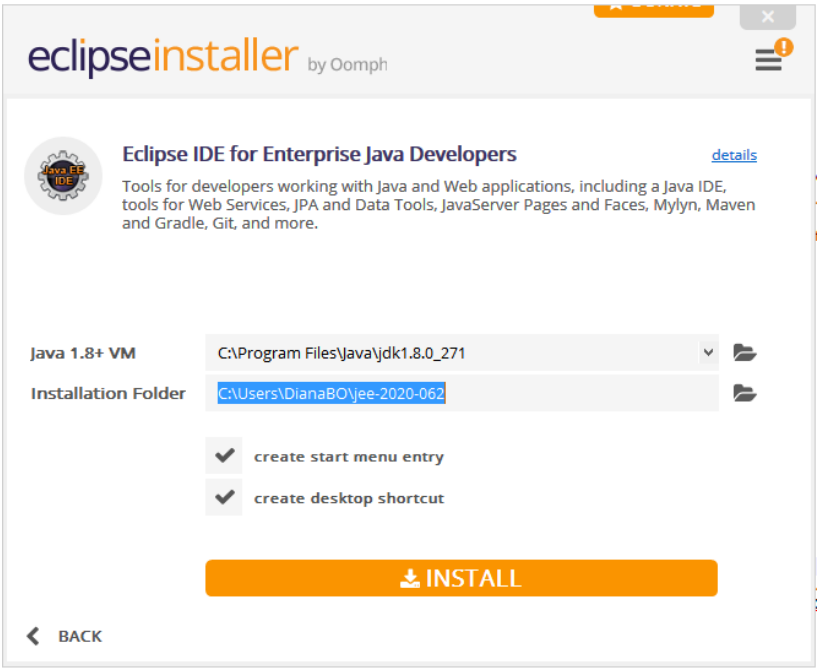

Figura D.5: Seleccionar JDK que usará el IDE

Instalación del *plugin de Vaadin* para Eclipse

Una vez se haya instalado Eclipse, se procederá a añadir el plugin de Vaadin para Eclipse. Esto se realizará mediante el **Eclipse Marketplace de Eclipse**, el cual se encuentra en la opción de "***Help/Eclipse Marketplace...***" de la barra de herramientas.

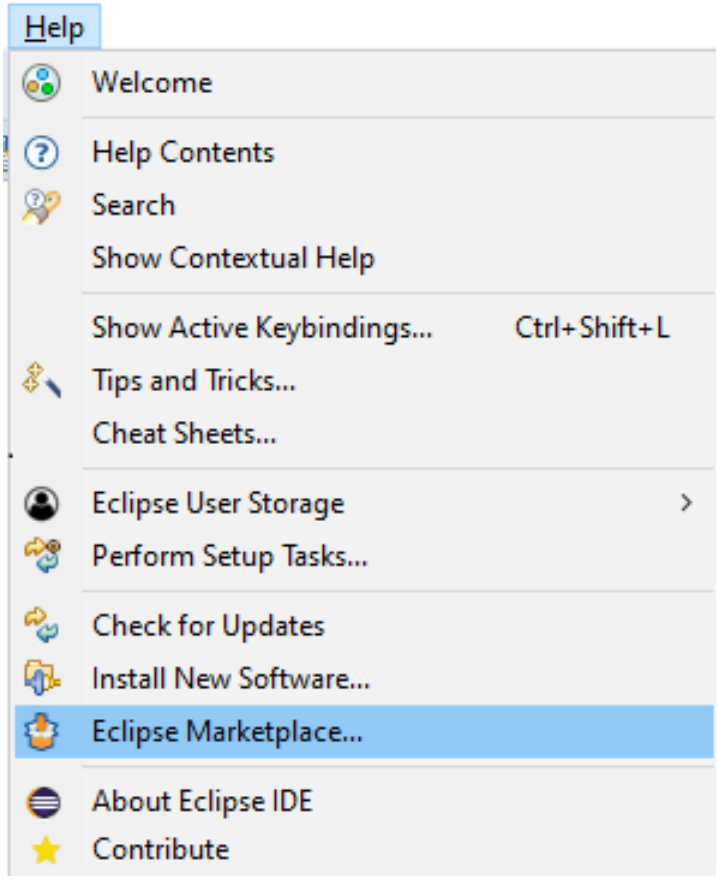

Figura D.6: Eclipse marketplace

Una vez en el Eclipse Marketplace, se buscará "**Vaadin**" y se pulsará "**Go**". Tras salir el plugin "***Vaadin Plugin for Eclipse***", se dará a "**Install**" y comenzará la instalación del plugin.

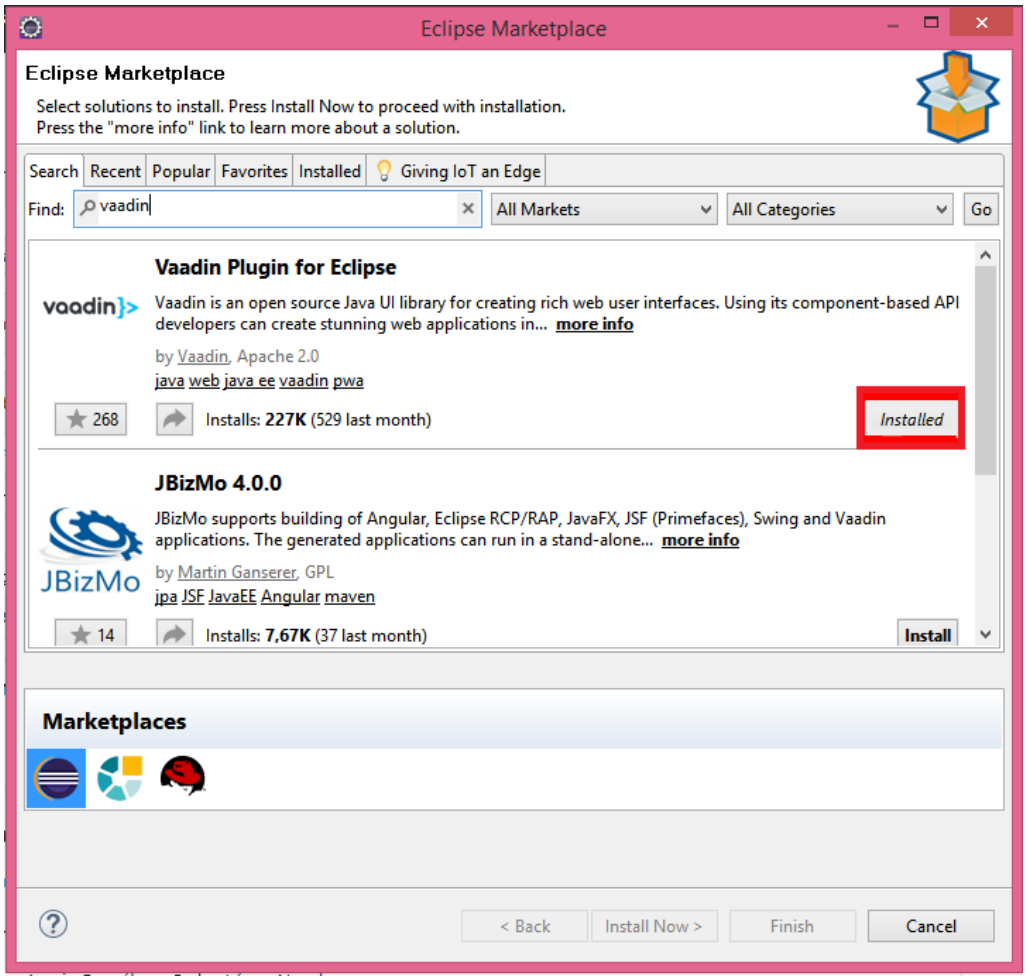

Figura D.7: Plugin Vaadin

D.4. Compilación, instalación y ejecución del proyecto

Se explicará como compilar, instalar e ejecutar el proyecto. En el caso de la ejecución, se detallará como hacerlo con el términal de Windows y mediante Eclipse (IDE).

Descarga del repositorio

El código fuente se encuentra en el repositorio del proyecto en GitHub. Para descargarlo se deberá hacer click en "***Code***" y copiar la URL que

aparece en el apartado de "**HTTP**". Con esta URL deberemos ir al "**GitHub Desktop**" y clonar el repositorio.

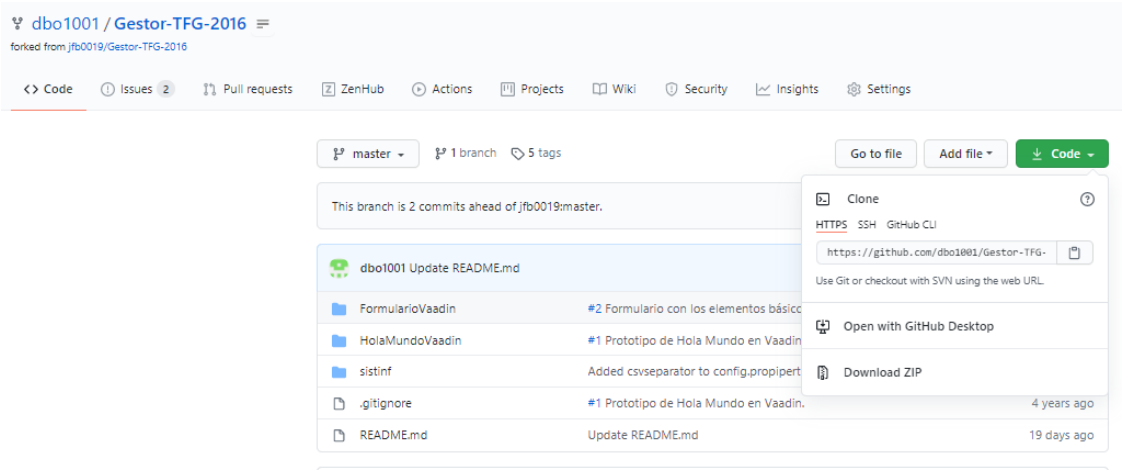

Figura D.8: Copiar URL repositorio

Si se desea tener código en local se deberá descargar el zip "***Download ZIP***" en la opción "***Code***" anteriormente mencionada. Una vez descargado el zip se descomprimirá y abrirá con Eclipse.

Para abrir el proyecto con Eclipse se seleccionara en la barra de herramientas ***File/Import...***. Aparecerá una ventana en la que se optará por la opción "***Projects from Folder or Archieve***" y se hará click en "***Next***".

Después se hará click en "***Directory...***" y se seleccionará la carpeta del proyecto con nombre "**sistinf**" y términaremos la importación con "***Finish***".

D.5. Pruebas del sistema

Apéndice E

Documentación de usuario

E.1. Introducción

E.2. Requisitos de usuarios

E.3. Instalación

No se requiere ninguna instalación por parte del usuario, simplemente puede acceder a la aplicación web a través del enlace .

E.4. Manual del usuario

Bibliografía