UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

TFG del Grado en Ingeniería Informática

título del TFG
Documentación Técnica

Presentado por nombre alumno
en Universidad de Burgos — 2 de julio de 2019
Tutor: nombre tutor

Índice general

Índice de figuras

Índice de tablas

Apéndice A

Plan de Proyecto Software

A.1. Introducción

El plan del proyecto es un documento que recoge el tiempo, esfuerzo y el dinero que supondrá la realización del proyecto. Este plan se divide en dos partes:

- Planificación temporal
- Estudio de la viabilidad

El objetivo de la primera es estimar el tiempo y el esfuerzo que se requieren para la realización del proyecto, La segunda parte se centra en un análisis de la viabilidad del proyecto. Se objetivo es estimar si el proyecto se podría realizar con éxito. El análisis de viabilidad se puede ha dividido en dos apartados:

- **Viabilidad económica**: Análisis del coste y del beneficio que supondría la realización del proyecto.
- **Viabilidad legal**: Análisis de las leyes que se aplicarían desde el comienzo del proyecto.En un proyecto software tienen especial importancia las licencias y la Ley de Protección de Datos.

A.2. Planificación temporal

No se ha realizado una planificación temporal del proyecto. Pero se podría decir que se han tratado de seguir los 12 principios del manifiesto ágil y el modelo SCRUM [4]:

- Se ha aplicado un desarrollo incremental y evolutivo.
- Se han realizado iteraciones (sprints) de dos semanas. Al finalizar un sprint se realizaba una reunión entre el tutor y el alumno que daba comienzo al siguiente sprint y que consta de dos partes:
  - Una parte de revisión del sprint en la que se exponía una parte operativa del producto.
  - Otra de planificación del siguiente sprint en la que se determinaba el trabajo y los objetivos a alcanzar durante el sprint siguiente. Esto quedaba reflejado como una pila de tareas que se debían completar durante el sprint y que han sido registradas en el sistema de gestión de incidencias de GitLab.

Sprints

A continuación se definen los sprints y sus respectivas pilas de tareas que se llevaron a cabo durante la realización del proyecto. Se pueden visualizar las issues en orden ascendente de creación a partir de esta URL:

Inicio del proyecto

El proyecto comenzó el 1 de octubre de 2018. Las primeras tareas que se definieron fueron de investigación y configuración del entorno de trabajo, no se definían de forma muy clara ya que aun no se sabía que caminos escoger para la realización del proyecto.

A.3. Estudio de viabilidad

Viabilidad económica

Viabilidad legal

Apéndice B

Especificación de Requisitos

B.1. Introducción

B.2. Objetivos generales

B.3. Catalogo de requisitos

B.4. Especificación de requisitos

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

Este documento detalla asuntos técnicos de programación.

D.2. Estructura de directorios

El código fuente presenta la siguiente estructura:

- ***/.gitignore*** Contiene los ficheros y directorios que el repositorio git no tendrá en cuenta
- ***/.gitlab-ci.yml*** Contiene las etapas y trabajos que se han definido para que se ejecuten en una máquina virtual proporcionada por GitLab (*runner*) tras hacer un commit. Permite la integración y el despliegue continuo.
- ***/README.md*** Fichero con información relevante sobre el proyecto.
- ***/codacy-coverage-reporter-4.0.5-assembly.jar*** Ejecutable Java que necesario para una de las tareas de la integración continua: El informe de cobertura de las pruebas. Se ejecutará en una de las tareas definidas en el fichero /.gitlab-ci.yml.
- ***/pom.xml*** Fichero de configuración del proyecto maven.

- ***/system.properties*** Fichero con propiedades del proyecto. Ha sido necesario su uso para el despliegue en Heroku.
- ***/MemoriaProyecto*** Memoria del proyecto según la plantilla definida en https://github.com/ubutfgm/plantillaLatex.
- ***/src/test/resources*** Datos almacenados en ficheros CSV para proporcionar datos a test parametrizados.
- ***/src/test/java*** Casos de prueba JUnit para la realización de pruebas. Se organiza de la misma forma que /src/main/java
- ***/src/main/webapp/VAADIN/themes/MyTheme*** Tema principal utilizado por la aplicación. Generado por Vaadin.
- ***/src/main/webapp/frontend*** Ficheros *.css* utilizados por la interfaz gráfica.
- ***/src/main/webapp/images*** Imágenes que se muestran en la interfaz gráfica.
- ***/src/main/resources*** Ficheros de configuración de la aplicación. En este caso el fichero log4j2.properties para configurar el log.
- ***/src/main/java*** Contiene todo el código fuente
- ***/src/main/java/app/*** Contiene fachadas que conectan la interfaz de usuario con el resto de componentes que componen la lógica de la aplicación.
- ***/src/main/java/app/listeners*** Contiene observadores y eventos utilizados por la aplicación
- ***/src/main/java/datamodel*** Contiene el modelo de datos de la aplicación.
- ***/src/main/java/exceptions*** Contiene excepciones definidas en la aplicación.
- ***/src/main/java/gui*** Contiene la interfaz de usuario.
- ***/src/main/java/gui/views*** Contiene páginas y componentes de Vaadin que componen la interfaz gráfica de la aplicación.
- /src/main/java/metricsengine Define el motor de métricas.

- ***/src/main/java/metricsengine/numeric_value_metrics*** Métricas definidas por el programador y sus respectivas fábricas (Patrón de diseño método fábrica[1]). Todas las métricas tienen resultados numéricos.
- ***/src/main/java/metricsengine/values*** Valores que devuelven las métricas.
- ***/src/main/java/repositorydatasource*** Framework de conexión a una forja de repositorios como GitLab.

D.3. Manual del programador

Se explica en este apartado algunas bases para entender como continuar la programación de la aplicación y los puntos de extensión que se han definido.

Motor de métricas

El motor de métricas se ha desarrollado con una base inicial a una solución propuesta en *Soporte de Métricas con Independencia del Lenguaje para la Inferencia de Refactorizaciones*[2]. El diseño se puede observar en **??**.

[1] https://refactoring.guru/design-patterns/factory-method

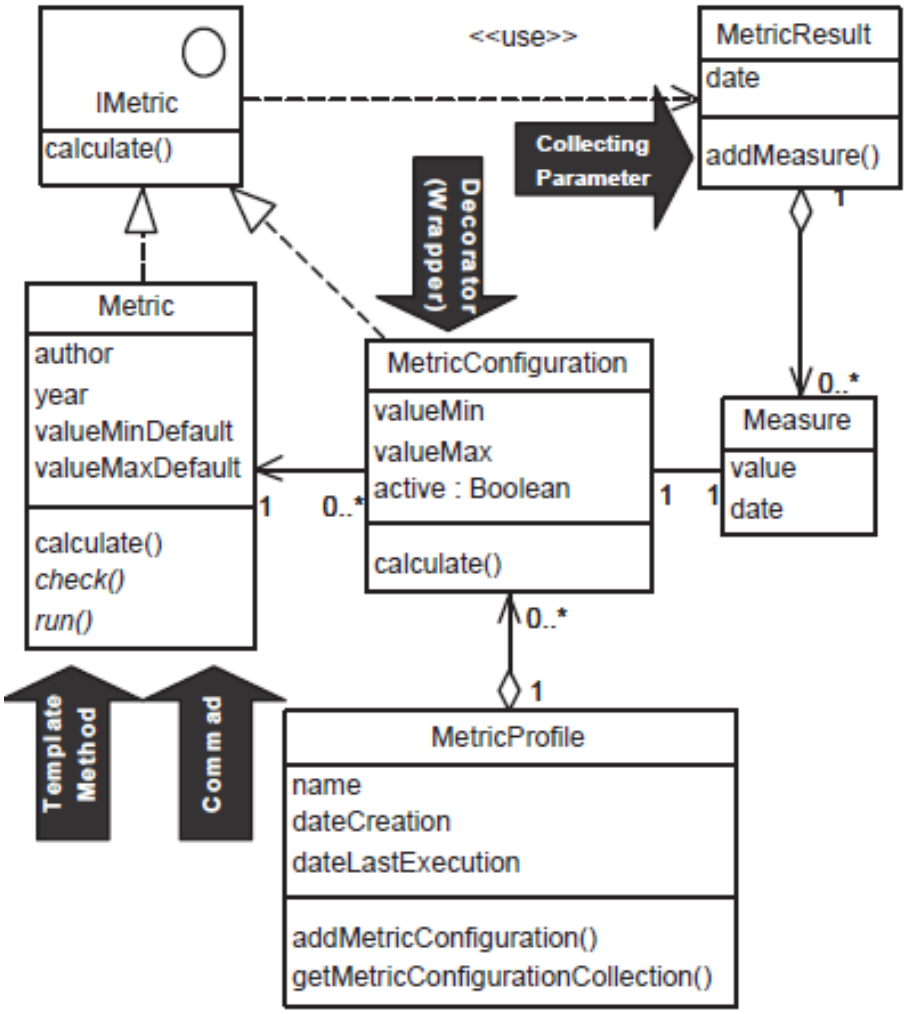


Figura D.1: Diagrama del Framework para el cálculo de métricas con perfiles.

Framework de conexión

Interfaz gráfica

D.4. Compilación, instalación y ejecución del proyecto

D.5. Pruebas del sistema

Apéndice E

Documentación de usuario

E.1. Introducción

Este documento detalla cómo un usuario, puede utilizar la aplicación una vez desplegada en un servidor.

E.2. Requisitos de usuarios

Los requisitos para poder utilizar la aplicación son:

- Tener la aplicación desplegada en algún servidor.
- Disponer de conexión al servidor y tener instalado un navegador web con el que poder acceder a la aplicación. Se recomienda:
  - Google Chrome Versión 75.0.3770.100 o superior
  - Firefox Quantum Versión 67.0.4 o superior
  - IE11 Versión 11.829.17134.0 o superior
  - Opera Versión:62.0.3331.18 o superior

E.3. Instalación

Al ser una aplicación web no requiere instalación. Solo es necesario desplegar la aplicación en un servidor.

E.4. Manual del usuario

La aplicación permite añadir proyectos de GitLab y evaluarlos mediante métricas de evolución.

Conceptos

Para utilizar la aplicación es importante entender los siguientes conceptos:

Medición
El proceso de medición es un proceso en el que se asignan números o símbolos a atributos de entidades del mundo real, de tal forma que los caracteriza a través de reglas.

Métrica
Medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado (IEEE, 1993).

Indicador
Métrica o combinación de métricas que proporcionan una visión profunda del proceso, del proyecto o del producto.

Métrica de evolución
Es una métrica que mide un atributo del proceso de desarrollo de un producto software.

Evaluación
Es uno de los objetivos del proceso de medición. Consiste en determinar el estado de un proyecto en relación con otros proyectos de la misma naturaleza.

Proyecto (software)
Proyecto en el cual se desarrolla un producto software.

Repositorio de código
Lugar dónde se almacena el código de un proyecto software. A menudo cuentan con un sistema de control de versiones.

Sistema de control de veriones (VCS - Version Control System)
Sistema que registra los cambios que se producen sobre los ficheros de un proyecto software almacenados en un repositorio de código.

Sistema de seguimiento de incidencias (Issue tracking system)
Sistema que gestiona las diferentes tareas o incidentes que se definen en un proyecto software y que pueden ser asignadas a colaboradores del proyecto.

Las métricas que se gestionan en la aplicación

Las métricas que se calculan de los proyectos son un conjunto de métricas que proceden de la Master Tesis titulada *sPACE: Software Project*

Assessment in the Course of Evolution [3].

I1 - Número total de issues

- **Categoría**: Proceso de Orientación
- **Descripción**: Número total de issues creadas en el repositorio
- **Propósito**: ¿Cuántas issues se han definido en el repositorio?
- **Fórmula**: NTI. NTI = número total de issues
- **Fuente de medición**: Repositorio de un gestor de repositorios
- **Interpretación**: NTI >= 0. Valores bajos indican que no se utiliza un Sistema de seguimiento de incidencias, podría ser porque el proyecto acaba de comenzar
- **Tipo de escala**: Absoluta
- **Tipo de medida**: NTI = Contador

I2 - Commits por issue

- **Categoría**: Proceso de Orientación
- **Descripción**: Número de commits por issue
- **Propósito**: ¿Cuántos commits realizados por cada issue?
- **Fórmula**: CI = NTC/NTI. NTI = Numero total de issues, NTC = Número total de commits
- **Fuente de medición**: Repositorio de un gestor de repositorios
- **Interpretación**: CI >= 0, Si se acerca a 1 se definen bien las issues, si alto: no se definen bien las issues, si bajo: desarrollo del proyecto lento
- **Tipo de escala**: Ratio
- **Tipo de medida**: NTI, NTC = Contador

I3 - Porcentaje de issues cerradas

- **Categoría**: Proceso de Orientación
- **Descripción**: Porcentaje de issues cerradas
- **Propósito**: ¿Qué porcentaje de issues definidas en el repositorio se han cerrado?
- **Fórmula**: PIC = NTIC*100/NTI. NTIC = Número total de issues cerradas, NTI = Numero total de issues
- **Fuente de medición**: Repositorio de un gestor de repositorios
- **Interpretación**: 0 <= PIC <= 100. Cuanto más alto mejor
- **Tipo de escala**: Ratio

- **Tipo de medida**: NTI, NTIC = Contador

TI1 - Media de días en cerrar una issue

- **Categoría**: Constantes de tiempo
- **Descripción**: Media de días en cerrar una issue
- **Propósito**: ¿Cuánto se suele tardar en cerrar una issue?
- **Fórmula**: MDCI = SUM(DCI) / NTIC . NTIC = Número total de issues cerradas, DCI = Días en cerrar la issue
- **Fuente de medición**: Repositorio de un gestor de repositorios
- **Interpretación**: MDCI >= 0. Cuanto más pequeño mejor.
- **Tipo de escala**: Ratio
- **Tipo de medida**: NTI, NTIC = Contador

TC1 - Media de días entre commits

- **Categoría**: Constantes de tiempo
- **Descripción**: Media de días que pasan entre dos commits consecutivos
- **Propósito**: ¿Cúanto tiempo suele pasar desde un commit hasta el siguiente?
- **Fórmula**: MDEC = [Sumatorio de (TCi-TCj) desde i=1, j=0 hasta i=NTC] / NTC. NTC = Número total de commits, TC = Tiempo de Commit
- **Fuente de medición**: Repositorio de un gestor de repositorios
- **Interpretación**: MDEC >= 0. Cuanto más pequeño mejor.
- **Tipo de escala**: Ratio
- **Tipo de medida**: NTC = Contador; TC = Tiempo

TC2 - Días entre primer y último commit

- **Categoría**: Constantes de tiempo
- **Descripción**: Días transcurridos entre el primer y el ultimo commit
- **Propósito**: ¿Cuantos días han pasado entre el primer y el último commit?
- **Fórmula**: DEPUC = TC2- TC1. TC2 = Tiempo de último commit, TC1 = Tiempo de primer commit.
- **Fuente de medición**: Repositorio de un gestor de repositorios
- **Interpretación**: DEPUC >= 0
- **Tipo de escala**: Absoluta
- **Tipo de medida**: TC = Tiempo

TC3 - Ratio de actividad de commits por mes

- **Categoría**: Constantes de tiempo
- **Descripción**: Muestra el número de commits relativos al número de meses
- **Propósito**:¿Cuál es el número medio de cambios por mes?
- **Fórmula**: RCM = NTC / 12
- **Fuente de medición**: Repositorio de un gestor de repositorios
- **Interpretación**: RCM >0. Cuanto más alto mejor
- **Tipo de escala**: Ratio
- **Tipo de medida**: NTC = Contador

C1 - Número de commits en el mes pico

- **Categoría**: Constantes de tiempo
- **Descripción**: Número de commits en el mes que más commits se han realizado en relación con el número total de commits
- **Propósito**: ¿Cuál es la proporción de trabajo realizado en el mes con mayor número de cambios?
- **Fórmula**: CCP = NCMP / NTC. NCMP = Número de commits en el mes pico, NTC = Número total de commits
- **Fuente de medición**: Repositorio de un gestor de repositorios
- **Interpretación**: 0 <= CCP <= 1. Mejor valores intermedios
- **Tipo de escala**: Ratio
- **Tipo de medida**: NCMP, NTC = Contador

Establecer una conexión a GitLab

Una vez arrancada la aplicación se mostrará un diálogo que le pedirá elegir un tipo de conexión, tal y como se muestra en la figura E.1.

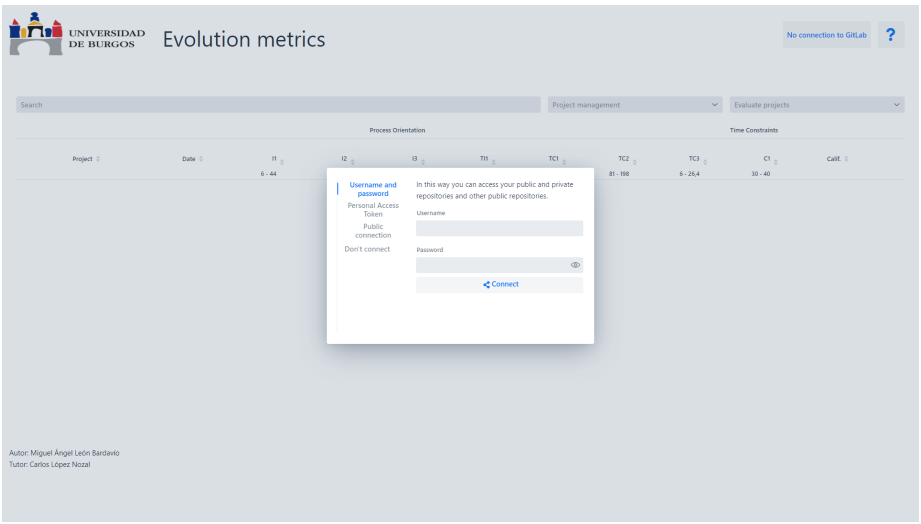
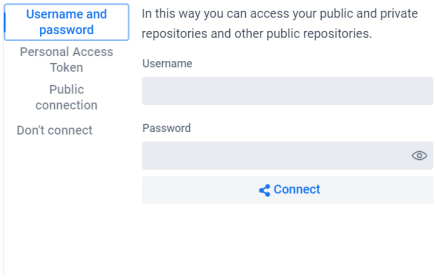


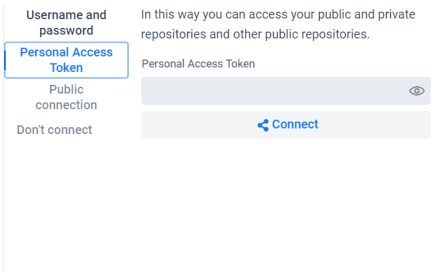
Figura E.1: Diálogo de conexión

Hay 5 posibilidades, que permiten establecer una conexión con la sesión iniciada, una conexión pública o utilizar la aplicación sin conexión:

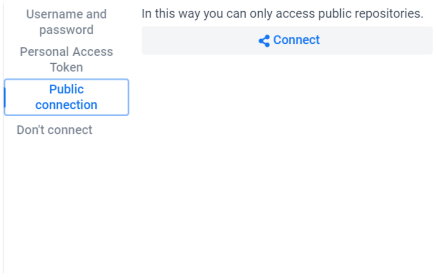
- **Iniciar sesión en GitLab mediante usuario y contraseña.** Se establece una conexión a GitLab iniciando sesión mediante un nombre de usuario y una contraseña. De esta forma se puede acceder a todos los repositorios públicos y privados accesibles desde ese usuario. Ver figura E.2a.
- **Iniciar sesión en GitLab mediante *Personal Access Token*.** Se establece una conexión a GitLab iniciando sesión mediante un *Personal Access Token*. De esta forma se puede acceder a todos los repositorios públicos y privados accesibles desde el usuario, como ocurría en el caso anterior. Si se accede a GitLab desde una cuenta externa a GitLab como Google o GitHub, esta opción es la única manera de iniciar sesión con su cuenta de GitLab. Ver figura E.2b.
  Para generar un *Personal Access Token* desde GitLab hay que iniciar sesión desde la web y entrar en la configuración del usuario. En el apartado de *Access Token* se debe dar un nombre, opcionalmente una fecha de expiración y los permisos. Para utilizar la aplicación se necesitan estos permisos: *api*, *read_user*, *read_repository*, *read_registry*. Una vez finalizado, pulsar sobre el botón "*Create personal access token*", copiar el token y utilizarlo. Una vez se salga de la ventana en la que se muestra el token, no volverá a aparecer, por lo que se recomienda copiarlo en algún lado. Ver figura E.3.



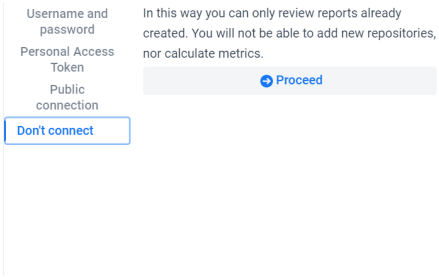
(a) Conexión mediante usuario y contraseña



(b) Conexión mediante *Personal Access Token*



(c) Conexión publica



(d) Sin conexión

Figura E.2: Distintas formas de establecer una conexión

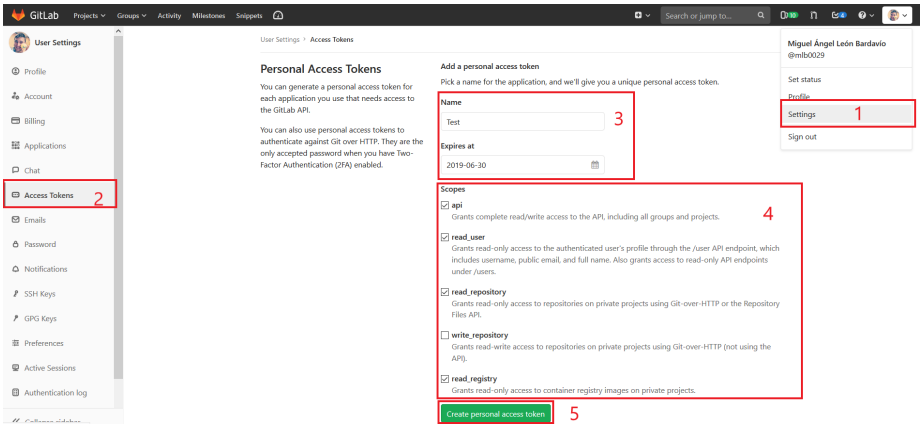


Figura E.3: Crear un *Personal Access Token* desde GitLab

- **Usar una conexión pública hacia GitLab.** Se establece una conexión pública a GitLab sin iniciar sesión, por lo que solo se podrá acceder a repositorios públicos. Ver figura E.2c.
- **No utilizar ninguna conexión.** No se realizará ninguna conexión a GitLab. Solo se podrá trabajar con proyectos y perfil de métricas

importados y con el perfil de métricas por defecto. Ver figura E.2d.

Página principal

Figura E.4: Página principal

Una vez elegida por primera vez el tipo de conexión deseado se accede a la página principal, como se observa en la figura E.4.
En la parte superior se puede observar un botón que indica el tipo de conexión actual.

- Si se ha iniciado sesión mediante usuario y contraseña o mediante un *personal access token*, se mostrará la imágen del usuario y el texto: "Connected as: <nombre de usuario>"
- Si se ha establecido una conexión pública, se mostrara el texto: "Using a public connection"
- Y si no se ha establecido ninguna conexión, el texto mostrado será: "No connection to GitLab".

Para cambiar el tipo de conexión es obligatorio cerrar la conexión actual. Por ello se muestra el diálogo de la figura E.5a si existe una conexión y el diálogo de la figura E.5b si no existe conexión al pulsar sobre el botón de conexión. Al pulsar sobre "*Connect*" o sobre "*Close connection*" se abrirá el diálogo de conexión de la figuras E.1 y E.2.
A la derecha del botón se encuentra un botón que da acceso a este manual en la Wiki del proyecto.

(a) Cerrar conexión

(b) Establecer conexión

Figura E.5: Modificar tipo de conexión

Listado de proyectos

En el centro de la página principal se pueden gestionar los proyectos. Consta de una barra de búsqueda, dos menús, y una tabla que visualiza las métricas de los proyectos que se añadan.
En el cuadro de búsqueda se filtrarán los repositorios por su nombre mientras se vaya escribiendo.
En el menú de "*Project management*" existen estas opciones, como se muestra en la figura E.6a:

- ***Add new.*** Permite añadir uno o varios proyectos.
- ***Import.*** Permite importar proyectos a partir de un fichero previamente exportado.
- ***Export.*** Permite exportar todos los proyectos existentes a un fichero, lo que permitirá su posterior importación. Se almacena en un fichero con foromato ".emr".
- ***Export to CSV.*** Permite generar un fichero CSV que contenga toda la información de la tabla de proyectos. Este fichero no servirá para importar los proyectos posteriormente.

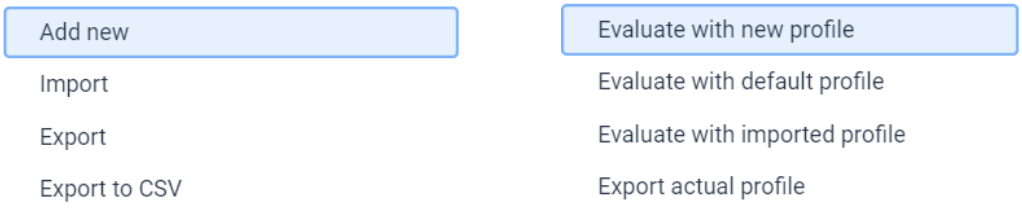
En el menú de "*Evaluate projects*" existen estas opciones, como se muestra en la figura E.6b:

- ***Evaluate with new profile.*** Permite evaluar los proyectos calculando los valores mínimos y máximos de cada métrica a partir de los repositorios actuales.
- ***Evaluate with default profile.*** Permite evaluar los proyectos con un perfil por defecto creado a a partir de un conjunto de datos[1] de un

[1] https://github.com/clopezno/clopezno.github.io/blob/master/agile_practices_experiment/DataSet_EvolutionSoftwareMetrics_FYP.csv

estudio empírico de las métricas de evolución del software en trabajos finales de grado[1].

- ***Evaluate with imported profile.*** Permite evaluar los proyectos a partir de un perfil de métricas previamente exportado.
- ***Export actual profile.*** Permite exportar el perfil de métricas actual para su posterior importación. Se almacena en un fichero con foromato ".emmp".



(a) Menú: "*Project management*" (b) Menú: "*Evaluate projects*"

Figura E.6: Menús del listado de repositorios

Añadir un proyecto

Importar proyectos

Exportar proyectos

Evaluar los proyectos

Exportar perfil de métricas

Bibliografía

[1] Carlos López. Portal web para la gestión de la asignatura de trabajos fin de Grado de Ingeneiría : clopezno/clopezno.github.io, June 2019. original-date: 2016-09-27T09:36:06Z.

[2] Raúl Marticorena, Yania Crespo, and Carlos López. Soporte de Métricas con Independencia del Lenguaje para la Inferencia de Refactorizaciones. In *X Jornadas Ingenieria del Software y Bases de Datos (JISBD 2005), Granada, Spain ISBN: 84-9732-434-X*, pages 59–66, September 2005.

[3] Jacek Ratzinger. *sPACE: Software Project Assessment in the Course of Evolution*. PhD Thesis, 2007.

[4] Iubaris Info 4 Media SL. Scrum Manager: Temario Troncal I. Versión 2.6.1:94, January 2019.