



Universidad
Rey Juan Carlos

GRADO EN INGENIERÍA EN TECNOLOGÍA DE LA
TELECOMUNICACIÓN

Curso Académico 2020/2021

Trabajo Fin de Grado

IDENTIFICACIÓN Y ANÁLISIS DE PROYECTOS
DE INGENIERÍA EN GITHUB

Autor : Carlos Morón Barrios

Tutor : Dr. Gregorio Robles Martínez

Trabajo Fin de Grado

Identificación y Análisis de Proyectos de Ingeniería en GitHub

Autor : Carlos Morón Barrios

Tutor : Dr. Gregorio Robles Martínez

La defensa del presente Trabajo Fin de Grado se realizó el día de
de 2020, siendo calificada por el siguiente tribunal:

Presidente:

Secretario:

Vocal:

y habiendo obtenido la siguiente calificación:

Calificación:

Fuenlabrada, a de de 2020

*“Programar sin una arquitectura o
diseño en mente es como explorar una
gruta sólo con una linterna: no sabes
dónde estás, dónde has estado ni
hacia dónde vas”.*

Danny Thorpe

Agradecimientos

Quiero agradecer a mis padres su apoyo incondicional, sin el cual habría sido casi imposible llegar hasta aquí. También quiero agradecer a mi hermano su ayuda y compañía, sobre todo en los momentos difíciles y complicados, que los ha habido.

La carrera no habría sido lo mismo sin mis compañeros, que han hecho más ligeros los madrugones para ir a estudiar. Aunque con el tiempo nos hemos desperdigado por los diferentes cursos y asignaturas, siempre hemos conseguido reunirnos para esas tapas y escapadas.

Gracias también a mis amigos porque sin ellos, nada sería lo mismo. Su apoyo e interés también es una razón para seguir adelante y poder compartir con ellos los buenos momentos.

Resumen

El objetivo de este Trabajo Fin de Grado es emplear diversas herramientas estadísticas como Random Forest y el clasificador basado en la puntuación, para la clasificación de repositorios GitHub.

GitHub es una plataforma que se utiliza para alojar proyectos y código con control de versiones, dando la posibilidad de colaboración y de trabajo en grupo.

En el mundo actual centrado en el software, algunos repositorios de software a gran escala, como por ejemplo SourceForge y GitHub, contienen una enorme cantidad de proyectos e información. Es por ello, que tanto los científicos como los ingenieros están interesados en analizar esta información tanto porque buscan proyectos y código para reutilizar como por probar sus hipótesis. Sin embargo, la extracción sistemática de datos relevantes de estos repositorios y el análisis de dichos datos para probar hipótesis es difícil. Hay que tener en cuenta, que muchos de los proyectos de GitHub son trabajos de clase y pequeños proyectos particulares poco rigurosos, por lo que nosotros nos centraremos en aquellos proyectos que llamaremos de ingeniería porque se realizan de manera más ingenieril (tanto en proceso como en producto).

En este trabajo se pretende analizar los repositorios de GitHub para seleccionar los proyectos de software de ingeniería e identificarlos. Para ello nos hemos basado en un trabajo previo de Munaiah y colaboradores [17], que utilizan una serie de parámetros para identificar los proyectos de ingeniería como son: Comunidad (evidencia de colaboración), Integración continua (evidencia de calidad), Documentación (evidencia de mantenibilidad), Historia (evidencia de evolución sostenida), Gestión de erratas (evidencia de la gestión del proyecto), Licencia (evidencia de responsabilidad) y Pruebas unitarias (evidencia de calidad). Una vez obtenidos estos

parámetros, se han utilizado dos clasificadores distintos, el basado en la puntuación y Random Forest.

De esta forma se ha realizado un muestreo de 2.316.524 repositorios y vemos que en los repositorios de propósito general el clasificador basado en puntuación ha identificado un 71,02 % de proyectos de ingeniería y un 28,98 % que no eran de ingeniería, mientras que Random Forest ha identificado un 23,51 % como de ingeniería y un 76,49 % que no. Por ello, aunque sobre el papel Random Forest es más preciso, ha dado peor resultado, quizá debido a que como existe una correlación entre las dimensiones se favorecen los grupos más pequeños sobre los más grandes. Además, estos resultados son similares a los obtenidos por Munaiah y colaboradores del 69.61 % y 24.07 % que sí obtenían de ingeniería con el clasificador basado en puntuación y Random Forest respectivamente.

Summary

The objective of this Final Degree Project is to use various statistical tools such as Random Forest and the score-based classifier for the classification of GitHub repositories.

Github is a platform that is used to host projects and code with version control, giving the possibility of collaboration and group work.

In today's software-centric world, large-scale software repositories such as SourceForge and GitHub, contain a huge amount of projects and information. That is why both scientists and engineers are interested in analyzing this information both out of curiosity and to test their hypotheses. However, the systematic extraction of relevant data from these repositories and the analysis of such data to test hypotheses is difficult. Keep in mind that many of the GitHub projects are class assignments and small private projects that are not very rigorous, so we will focus on those projects that we will call engineering because they are carried out in a more engineering way (both in process and in product).

This work aims to analyze the GitHub repositories to select engineering software projects and identify them. For this we have based on a previous work by Munaiah and collaborators [17], who use a series of parameters to identify engineering projects such as: Community (evidence of collaboration), Continuous integration (evidence of quality), Documentation (evidence of maintainability), History (evidence of sustained evolution), Questions (evidence of project management), License (evidence of responsibility) and Unit tests (evidence of quality). Once these parameters were obtained, two classifiers have been used different, score based and Random Forest.

In this way, 2,316,524 repositories have been sampled and we can see that in the general purpose repositories the score-based classifier has identified 71.02 % of engineering projects and 28.98 % that were not from engineering, while Random Forest has identified 23.51 % as engineering and 76.49 % as not. For this reason, although on the Random Forest paper it is more precise, it has given a worse result, perhaps because, as there is a correlation between the dimensions, the smaller groups are favored over the larger ones. Furthermore, these results are similar to those obtained by Munaiah and collaborators of the 69.61 % and 24.07 % that they obtained from engineering with the score-based classifier and Random Forest respectively.

Índice general

1. Introducción	1
1.1. Análisis de los datos	2
1.2. Estructura de la memoria	4
2. Objetivos	7
2.1. Objetivo general	7
2.2. Objetivos específicos	7
2.3. Planificación temporal	8
3. Estado del arte	9
3.1. GitHub	9
3.2. Git	9
3.3. GHTorrent	12
3.4. Machine Learning	13
3.5. Clasificador basado en la puntuación	14
3.6. Random Forest	15
3.7. Métricas de un repositorio	16
3.8. Herramientas informáticas	19
4. Diseño e implementación	23
4.1. Metodología	23
4.2. Pasos seguidos	24
4.3. Procedimientos informáticos	24

5. Resultados	29
5.1. Pruebas preliminares	29
5.2. Clasificadores	33
5.3. Validación	33
5.3.1. Establecer la verdad fundamental	34
5.3.2. Validación interna	34
5.3.3. Validación externa	35
5.4. Aplicación de los clasificadores	35
5.5. Discusión	36
5.6. Puntos débiles del estudio	37
6. Conclusiones	47
6.1. Consecución de objetivos	47
6.2. Aplicación de lo aprendido	48
6.3. Lecciones aprendidas	48
6.4. Futuros trabajos	48
A. ENGINEERED	49
Bibliografía	111

Índice de figuras

1.1. Fases de análisis. Imagen obtenida del trabajo de Yeray [20]	3
2.1. Planificación temporal del Trabajo Fin de Grado. Elaboración propia.	8
3.1. Imagen extraída de guides.github.com [14]	10
3.2. Imagen extraída de guides.github.com [14]	10
3.3. Imagen extraída de guides.github.com [14]	11
3.4. Métrica SLOC de un repositorio [5]	17
4.1. Esquema de la base de datos utilizada	25
5.1. Número de repositorios en la organización agrupados por lenguajes de programación.	30
5.2. Número de repositorios en la utilidad del conjunto de datos agrupados por lenguajes de programación.	31
5.3. Distribución del número de puntuación de los repositorios; (a) en la organización y (b) utilidad del conjunto de datos	32
5.4. Distribución de las dimensiones de repositorios en el conjunto de datos de la organización	38
5.5. Distribución de las dimensiones obtenidas de los repositorios en el conjunto de datos de utilidad.	39
5.6. ρ de Spearman entre pares de dimensiones en la organización (a) y conjuntos de datos de utilidad (b) con - (guión) representando las correlaciones estadísticamente insignificantes.	40
5.7. Distribución de las dimensiones de los repositorios en el conjunto de validación.	41

5.8. Número de repositorios obtenidos por los clasificadores basados en la puntuación y Random Forest agrupados por lenguajes de programación (ORGANIZACIÓN).	42
5.9. Número de repositorios obtenidos por los clasificadores basados en la puntuación y Random Forest agrupados por los lenguajes de programación (UTILIDAD).	43
5.10. Distribución de las dimensiones de los repositorios en el conjunto de validación.	44
5.11. Comparación de la distribución de las dimensiones de los repositorios que contienen proyectos informáticos de ingeniería propiedad de organizaciones y usuarios.	45

Capítulo 1

Introducción

La captura y posterior tratamiento de datos correspondientes a empresas tecnológicas e investigación en ingeniería puede revelar información valiosa para la organización relacionada con sus productos, descubrimiento clientes potenciales y grandes avances en una tecnología determinada.

Por otro lado, la revolución surgida en la última década por las redes sociales, dispositivos móviles, sensores, el internet de las cosas, etc., ha dado como resultado la aparición de un gran volumen de datos (Big Data), y la necesidad de poder realizar su análisis y tratamiento [16]. Ahora el problema reside en cómo analizar eficazmente los datos.

Además, aunque el análisis de esa gran cantidad de datos (existen muchos repositorios) no es sencillo, cada vez hay más científicos que se dedican a ello para sacar información del software y para probar sus hipótesis científicas. Ejemplos de esos repositorios son SourceForge (más de 324.000 proyectos) y GitHub (más de 69.000.000 proyectos).

EL problema surge al intentar separar y discernir cuáles de todos los proyectos alojados en esas plataformas son los que particularmente nos interesa para la labor investigadora o de desarrollo que se está llevando a cabo particularmente en un momento determinado.

Aquí es donde entramos nosotros, ya que pretendemos buscar y encontrar de una forma eficiente cuales de los proyectos alojados en GitHub tienen que ver con ingeniería y cuales no. Además, una vez identificados los proyectos de ingeniería se pretende saber cuales de ellos pertenecen a grandes organizaciones y cuales tienen un propósito general.

A la hora de poder identificar los proyectos de ingeniería nos podemos fijar en un amplio abanico de características: determinadas palabras claves, número de visualizaciones, etc.

Para la realización de este Trabajo Fin de Grado nos hemos basado en el trabajo de Munaiah y colaboradores [17], que utilizan una serie de parámetros para identificar los proyectos de ingeniería como son: Comunidad (evidencia de colaboración), Integración continua (evidencia de calidad), Documentación (evidencia de mantenibilidad), Historia (evidencia de evolución sostenida), Gestión de erratas (evidencia de la gestión del proyecto), Licencia (prueba de responsabilidad) y Pruebas unitarias (prueba de calidad). Aunque existen más características que se podrían haber utilizado (presencia en el código de determinadas palabras clave, número de visualizaciones, etc.), Munaiah y colaboradores se quedaron con esos siete parámetros y obtuvieron resultados muy buenos y consiguieron analizar más de 3.000.000 de repositorios. Además, este trabajo ha sido citado 72 veces y tiene 2.974 visualizaciones.

Los parámetros que eligieron y que nosotros también hemos utilizado cubre un amplio abanico de características propias de proyectos de ingeniería como es la presencia de una licencia, el tamaño del proyecto y la evidencia de que se produce un mantenimiento prolongado en el tiempo.

1.1. Análisis de los datos

Hay que tener en cuenta a la hora de desarrollar nuestro proyecto que estamos hablando de revisar millones de repositorios, con lo cual vamos a tener en nuestras manos mucha cantidad de datos, lo cual va a influir en la utilización de plataformas informáticas adecuadas [6], [12] y [10].

Además, para poder analizar dichos datos, es necesario organizarlos y depurarlos adecuadamente para poder sacar la información relevante buscada. Por esto las empresas y diversos organismos oficiales (Universidades, centros de investigación, ministerios...) invierten cada vez más recursos en la formación de profesionales y en investigación y desarrollo [50]. En este último trabajo [3], se proponen las fases del análisis que se especifican en la Figura 1.1.

En este trabajo nos van a interesar únicamente los proyectos de ingeniería, para ponernos al día en los desarrollos tanto técnicos como de software que los distintos desarrolladores van

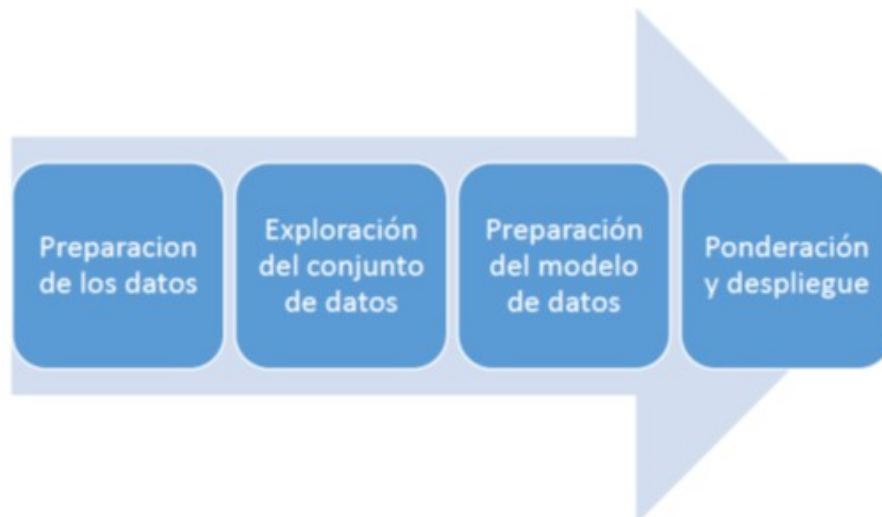


Figura 1.1: Fases de análisis. Imagen obtenida del trabajo de Yeray [20]

creando y poder avanzar en una técnica o vía de investigación determinada. Para ello vamos a tener que analizar las bases de datos contenidas en los repositorios y saber discernir qué proyectos son de ingeniería y cuales no. Por eso es interesante saber primero, qué entendemos nosotros por proyecto de ingeniería.

El concepto de proyecto es lo suficientemente amplio y ambiguo como abarcar muchos ámbitos de la vida cotidiana; proyecto de vida, proyecto político, proyecto de viaje, ... Sin embargo, en la RAE nos encontramos con dos acepciones importantes para nosotros porque son las que más se asemejan al concepto que en ingeniería se tiene por proyecto: Conjunto de escritos, cálculos y dibujos que se hacen para dar idea de cómo ha de ser y lo que ha de costar una obra de arquitectura o de ingeniería; primer esquema o plan de cualquier trabajo que se hace a veces como prueba antes de darle la forma definitiva.

Sin embargo, aunque la mayoría de las veces un proyecto nace de la necesidad, la sucesión de las distintas actividades es lo que va a dar forma al proyecto. Entre dichas actividades tenemos la toma de decisiones, iteraciones, compensación de recursos, influencia del entorno, planificación, calidad, diseño... Y es aquí, en estas actividades u otras similares donde nos vamos a fijar nosotros para poder averiguar si un proyecto es de ingeniería o no.

Se han realizado diversos estudios a lo largo de estos años en los que realizando distintas encuestas a usuarios de GitHub, se han analizado las mismas para entender cómo afectan las distintas características sociales de los usuarios [7], [8] y [21]. Esos estudios han concluido que los usuarios de GitHub se forman impresiones y sacan conclusiones sobre las actividades

y el potencial de otros desarrolladores y proyectos, de forma que los usuarios interiorizan esas conclusiones para decidir a quién hacer un seguimiento, o dónde realizar una contribución.

En el proceso de análisis de los repositorios, los investigadores van aprendiendo y mejorando sus técnicas, de manera que lo que desarrollan a posteriori se va a ver influenciado y muchas veces mejorado. Si esto lo aplicamos a desarrolladores de software, obtenemos que dicho análisis tiene una influencia sobre la calidad del código que desarrollan y el producto final [4].

Sin embargo, el mundo del software no es ajeno a repositorios que introducen mucha información intrascendente que poco tiene que ver con la ingeniería (tareas de trabajo a domicilio, archivos de texto, imágenes, copias de seguridad personales...) [9], por esto los investigadores utilizan diversos filtros para identificarla. Uno de los más utilizados es el de la popularidad, ya que se cree que la popularidad está correlacionada con la calidad, aunque hay muchos detractores del mencionado filtro [11], [2], [22], [15] y [23] que crean sus propios filtros.

Por ejemplo, Kalliamvakou y colaboradores [9] muestrearon manualmente 434 repositorios de GitHub y encontraron que solo el 63.4% (275) de ellos contenían proyectos de software, mientras que los 159 repositorios restantes se utilizaron con fines experimentales, de almacenamiento o académicos, estaban vacíos o ya no eran accesibles. Sin embargo, el método de seleccionar manualmente una muestra no es factible dado el gran volumen de repositorios alojados en GitHub.

1.2. Estructura de la memoria

- En el capítulo 1 se hace una introducción al proyecto y al análisis de los datos.
- En el capítulo 2 se describen los objetivos generales y específicos del trabajo y la planificación temporal del mismo.
- En el capítulo 3 se contextualiza el trabajo con el estado del arte y se introduce el concepto de Git y el funcionamiento de GitHub.
- En el capítulo 4, presentamos las siete dimensiones utilizadas para analizar los repositorios de nuestro estudio, la metodología seguida y los clasificadores utilizados para catalogar si un repositorio contiene un proyecto de ingeniería o no.

- En el capítulo 5, se presenta la validación del procedimiento y los resultados obtenidos de una muestra de 2.316.524 repositorios GitHub analizados.
- Por último, en el capítulo 6 se presentan las conclusiones y los futuros trabajos.

Capítulo 2

Objetivos

2.1. Objetivo general

Este trabajo fin de grado consiste en identificar los proyectos de ingeniería de los repositorios de software libre alojados en la plataforma GitHub.

2.2. Objetivos específicos

Para conseguir el objetivo principal, ha sido necesario realizar los siguientes pasos:

- Conseguir y analizar la base de datos de GHTorrent.
- Seleccionar un número determinado de datos para identificar los campos de la base de datos.
- Poner las cabeceras correspondientes a cada fichero como se indica en el esquema de la página GHTorrent.
- Calcular los parámetros correspondientes a cada dimensión.
- Seleccionar los proyectos que se consideran ingeniería en función de las dimensiones elaboradas.
- Comprobar que los proyectos seleccionados son los correctos.

2.3. Planificación temporal

A continuación, detallo en la figura 2.1 la temporalidad de las diferentes fases del trabajo que empezó en enero de 2020.

Trabajo Fin de Grado	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
Planteamiento												
Base de datos GHTorrent												
Identificar campos												
Calcular dimensiones												
Seleccionar Proyectos												
Comprobación												
Escritura TFG												

Figura 2.1: Planificación temporal del Trabajo Fin de Grado. Elaboración propia.

Como se puede ver en la figura 2.1, una vez que el tutor me planteó el trabajo, durante los meses de enero y febrero me propuse estudiar sobre el tema y me dediqué a recopilar toda la información que pude. A continuación, una vez conseguida toda la información el siguiente paso fue estudiar y manejar la base de datos GHTorrent. El primer paso fue identificar los campos involucrados, para poder calcular las dimensiones que se han utilizado para poder seleccionar los proyectos de ingeniería. Se hizo una primera comprobación con un conjunto de repositorios seleccionados manualmente y que se sabía que contenían proyectos de ingeniería y a posteriori se aplicó al caso real de 2.316.524 repositorios.

Capítulo 3

Estado del arte

3.1. GitHub

GitHub es un sitio de alojamiento de código colaborativo construido sobre el sistema de control de versiones de git. GitHub introdujo un modelo en el que los desarrolladores crean su propia copia de un repositorio y envían una solicitud de extracción cuando quieren que el mantenedor del proyecto lleve sus cambios a la rama principal. Además del alojamiento de código, la revisión colaborativa del código y el seguimiento integrado de erratas, GitHub tiene funciones sociales integradas. Los usuarios pueden suscribirse a la información “viendo” los proyectos y “siguiendo” a usuarios, lo que da como resultado una fuente de información sobre esos proyectos y usuarios. Los usuarios también tienen perfiles que se pueden completar con información de identificación y además contienen su actividad reciente dentro del sitio.

El manejo de GitHub es bastante amigable [18] y nos permite desde crear un repositorio con la descripción del mismo (Figura 3.1), crear una bifurcación (Figura 3.2), realizar cambios que se denominarán “Commits” (Figura 3.3) hasta enviar una solicitud de extracción a un usuario, para que tenga en cuenta tu versión, la revise y la agregue a su código fuente o versión master.

3.2. Git

Git es un software de control específico de versión de código abierto creado por el ingeniero Linus Torvalds en 2005, conocido por ser el creador del kernel de Linux. Esto quiere decir que si activamos el control sobre la carpeta donde está nuestro código el sistema se encargará

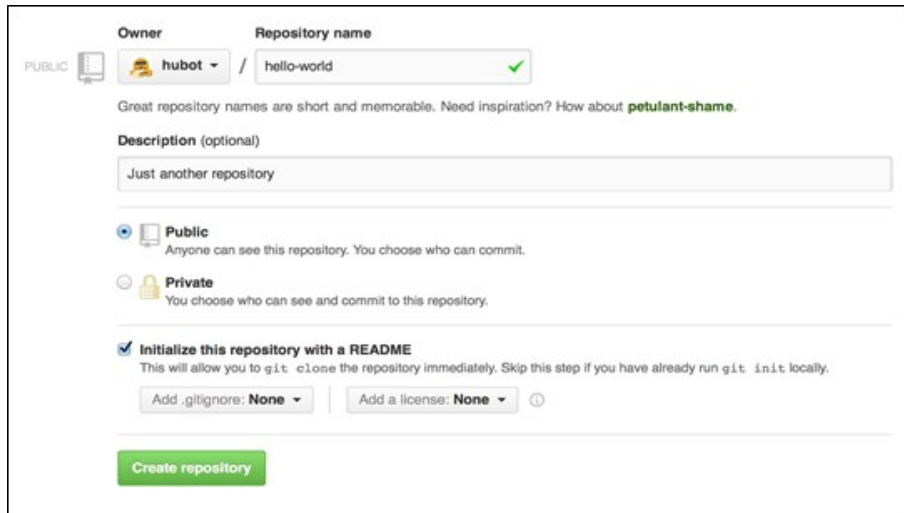


Figura 3.1: Imagen extraída de guides.github.com [14]

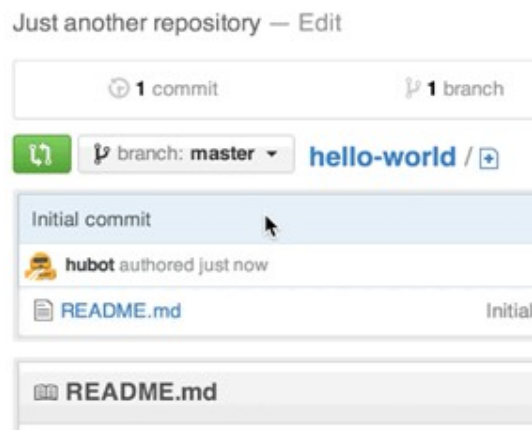


Figura 3.2: Imagen extraída de guides.github.com [14]

de controlar los cambios en los archivos, con lo que tanto la base del código entero como su historial se encuentran disponibles en los ordenadores de todos los desarrolladores, lo cual permite un fácil acceso a las bifurcaciones y fusiones [13].

El principal objetivo de Git es llevar un estricto control de los cambios que varias personas realizan al tiempo sobre un archivo de computadora [18].

Sus principales características son:

1. El diseño de Git se basa en BitKeeper y en Monotone.
2. Además de ofrecer apoyo al desarrollo no lineal, proporciona rapidez en la gestión de ramas y fusión de diferentes versiones.

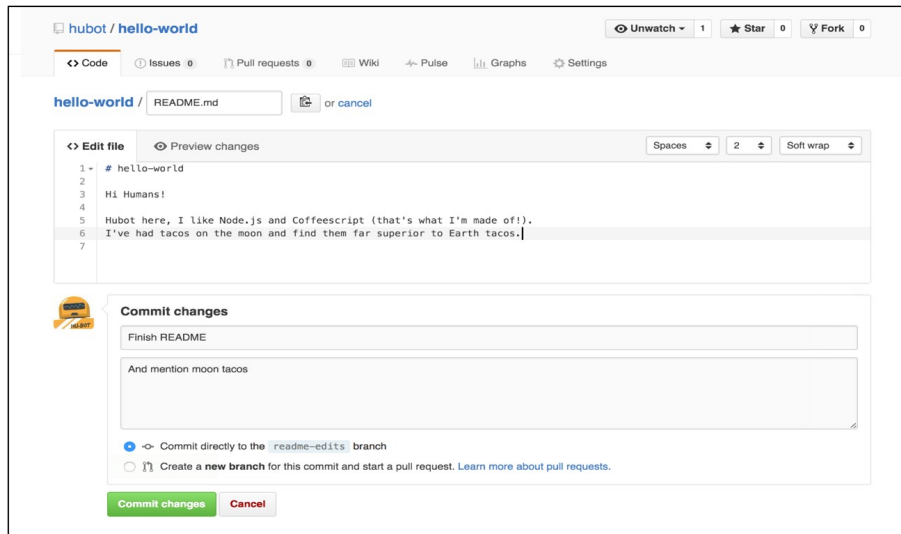


Figura 3.3: Imagen extraída de guides.github.com [14]

3. Proporciona a cada programador una copia local del historial del desarrollo entero, y los cambios se propagan entre los repositorios locales.
4. Los almacenes de información pueden publicarse mediante HTTP, FTP, rsync o un protocolo nativo.
5. Los repositorios Subversion y svk se pueden usar directamente con git-svn.

Además, cuando se quieran marcar los cambios se realizará lo que se conoce como commit, que consiste en describir los cambios realizados, apuntándolos en este registro. De esta forma, podremos movernos entre los diferentes commits, por ejemplo, para volver a una versión anterior de nuestro proyecto.

También, los sistemas de control de versiones, entre los que se encuentra Git, son la base en la actualidad para los proyectos de equipo ya que podemos trabajar a la vez múltiples programadores en un mismo proyecto, incluso en un mismo archivo de una manera fácil [19].

Por otro lado, con el portal Kinsta, el sistema de control de versiones permite a los desarrolladores administrar el código fuente de un programa y habilitarlo para que se hagan modificaciones a través de la bifurcación y la fusión. La bifurcación te permite crear una copia de una parte del código, para que los desarrolladores puedan modificarla de forma segura sin que los cambios afecten la versión original (Nextu, [18]). Esto evita que cualquier error afecte el software final. La fusión permite al desarrollador unir su versión de código al código fuente una

vez que ha comprobado su buen funcionamiento, aunque posteriormente el sistema de control de versiones les permitirá a los administradores revertir cualquier cambio.

Para nosotros es importante Git porque GitHub es donde se aloja el código colaborativo construido sobre el sistema de control de versiones de Git.

3.3. GHTorrent

GHTorrent (<https://ghtorrent.org/>) [23] es un proyecto que ofrece metadatos de Git y GitHub disponibles a través de la REST API de GitHub y fue diseñado para usar el almacenamiento en caché (para evitar solicitudes duplicadas) y también para ser distribuido (para permitir que múltiples usuarios recuperen datos en paralelo).

La API de GitHub admite dos tipos de consultas y los resultados de la consulta de recursos se pueden almacenar en la memoria caché de manera muy eficiente, ya que, por definición, nunca cambian. Las consultas de rango son más difíciles de almacenar en caché, ya que su resultado puede cambiar a medida que el proyecto evoluciona (nuevos compromisos, nuevos seguidores, etc.); afortunadamente, por defecto, GitHub proporciona primero los resultados más nuevos, por lo que es suficiente pasar por las primeras páginas de resultados solo para recuperar los datos actualizados. Para almacenar en caché los resultados por entidad, GHTorrent utiliza una base de datos MongoDB, que habilita la consulta en los datos sin procesar. El almacenamiento en caché también se utiliza en la capa de solicitud HTTP; GHTorrent serializa automáticamente las respuestas HTTP en el disco, lo que evita la recuperación de páginas antiguas.

El algoritmo de duplicación se basa en un proceso de resolución de dependencia recursiva. Para cada elemento recuperable, se especifica un conjunto de dependencias, ya que lógicamente fluyen desde el esquema de datos. Por ejemplo, para recuperar un proyecto, es necesario recuperar primero el usuario propietario; del mismo modo, para recuperar una solicitud de extracción, el proyecto debe recuperarse primero. Si falla cualquier paso de la resolución de dependencia, todo el elemento se marca como no recuperado. El proceso fue diseñado desde el principio para ser idempotente: cada paso de la resolución de dependencia puede fallar, pero una vez que tiene éxito, siempre devolverá el mismo resultado. Esta elección de diseño es muy importante ya que hace que se agreguen los datos y los resultados de cada paso y, por lo tanto, que sean almacenables en caché.

El proyecto GHTorrent es similar al proyecto GitHub Archive, ya que ambos comienzan con el marco temporal de los eventos públicos de GitHub. Mientras que el proyecto GitHub Archive simplemente registra los detalles de un evento GitHub, el proyecto GHTorrent recupera exhaustivamente los contenidos del evento y los almacena en una base de datos. Además, los conjuntos de datos GHTorrent están disponibles para descargar, ya sea como volcados incrementales de MongoDB o como un único volcado de MySQL, lo que permite el acceso sin conexión a los metadatos. Se ha optado por utilizar el volcado de MySQL que se descargó y restauró en un servidor local. En el resto del documento, cada vez que se usa el término base de datos nos referimos a la base de datos GHTorrent.

3.4. Machine Learning

El machine learning se puede definir como la automatización mediante algoritmos de la identificación de patrones en un conjunto de datos, por lo que resulta crucial elegir bien el algoritmo adecuado para conseguir el objetivo propuesto. El machine learning ha aumentado su importancia con los años debido a su aplicación en la robótica, vehículos autónomos, la toma de decisiones inteligentes y la inteligencia artificial. De hecho es una rama de la inteligencia artificial que empezó a despuntar en la década de los años 80.

El objetivo principal del machine learning es la creación de un modelo que permita conseguir el objetivo y, después del entrenamiento del modelo con una gran cantidad de datos, consiga aprender y termine siendo capaz de hacer predicciones. Según la tarea que se quiera realizar, será más adecuado trabajar con un algoritmo u otro. En nuestro caso, estamos ante un problema de clasificación de repositorios para detectar los que tienen proyectos de ingeniería.

De entre los diferentes tipos de implementación de machine learning que existen, nosotros estamos ante un aprendizaje supervisado, ya que los algoritmos van a trabajar con datos etiquetados.

Por ello se van a utilizar dos clasificadores que son los más utilizados por la mayoría de los investigadores en este campo. Dichos clasificadores son el basado en puntuación y el aleatorio.

3.5. Clasificador basado en la puntuación

El clasificador basado en la puntuación es una herramienta matemática que sirve para implementar el marco de evaluación de los datos [1]

$$f(r) = \begin{cases} \text{verdadero} & \text{si } score(r) \geq score_{ref} \\ \text{falso} & \text{cualquier otro caso} \end{cases} \quad (3.1)$$

$$score(r) = \sum_{d \in D} h_d(M_d, t_d) \times w_d \quad (3.2)$$

donde:

- r es el repositorio a clasificar.
- D es un conjunto de dimensiones a lo largo del cual se evalúa el repositorio, r .
- M_d es la métrica que cuantifica la evidencia del repositorio, r , empleando una cierta práctica de ingeniería de software en la dimensión d . Por ejemplo, la proporción de líneas de comentario a líneas de origen cuantifica la documentación.
- t_d es un umbral que debe ser satisfecho por la métrica correspondiente, M_d , para que el repositorio, r , sea considerado ingeniería en la dimensión d .
- $h_d(M_d, t_d)$ es una función heurística que evalúa a 1 si el valor métrico, M_d , satisface el umbral correspondiente, t_d , 0 en caso contrario.
- w_d es el peso que especifica la importancia relativa de cada dimensión d .
- $score_{ref}$ es la puntuación de referencia i.e. la puntuación mínima a la que un repositorio debe evaluar para que se considere que contiene un proyecto de software diseñado.

En el caso del clasificador basado en puntuación, el conjunto de datos de prueba se usa para determinar los umbrales, t_d , y calcular la puntuación de referencia, $score_{ref}$. Para todos los repositorios en cada uno de los dos conjuntos de datos de prueba se recopilan los valores de las siete dimensiones que se describen en la sección de métricas de un repositorio en este mismo capítulo.

3.6. Random Forest

Random Forest es una técnica de agregación desarrollada por Leo Breiman (breiman, 2001), que mejora la precisión en la clasificación mediante la incorporación de aleatoriedad en la construcción de cada clasificador individual. Esta aleatoriedad puede introducirse en la partición del espacio (construcción del árbol), así como en la muestra de entrenamiento.

En este tipo de clasificador hay que tener en cuenta que, si los datos contienen atributos correlacionados de relevancia similar para el rendimiento, entonces los grupos más pequeños están favorecidos sobre los grupos más grandes.

Para el análisis de los repositorios se ha utilizado el programa que se encuentra alojado en <https://github.com/RepoReapers/reaper>. Para ejecutar dicho programa ha sido necesario instalar python3. Este programa usa el fichero `batch_score.py` que se tiene que ejecutar de la siguiente manera:

```
batch_score.py -c <config> -r <repos_path> -m <manifest> -s <sample_file>
```

donde

- `<config>`: es una parte de `config.json`.
- `<repos_path>`: es la dirección de un directorio donde el programa puede comprobar los ficheros fuente de un proyecto.
- `<manifest>`: es una parte de `manifest.json` que contiene información sobre qué atributos deben ejecutarse.
- `<sample_file>`: es una lista de los identificadores de los proyectos de GHTorrent que se van a analizar.

A continuación el fichero `config.json` contiene las siguientes claves y valores:

- `Threshold` (define el umbral por el cual el sistema considera que un repositorio contiene un proyecto de software): es un número positivo.
- `persistResult` (decide si los resultados obtenidos deben guardarse en la fuente de datos especificada): `true` or `false`.
- `Datasource` (Configuración para conectarse a la base de datos de GHTorrent): `object`.

- `github_tokens` (es una lista de los identificadores de los autores): list.

Después de realizar dichos cambios, el programa que tenía el nombre de `config.json.skel` se guardará con el nombre de `config.json`. Si en `persistResult` se ha puesto `True`, debe existir una tabla de base de datos en la que el programa pueda escribir los resultados. Esta tabla debe llamarse `reaper_results` y debe contener al menos una columna con el nombre de `Project_id` que debe tener los identificadores del proyecto, y otra columna llamada `score` para almacenar la puntuación de un repositorio. Además, debe haber una columna para cada atributo que se desee almacenar.

Además, en este proyecto he utilizado Anaconda, que es una distribución libre y abierta de Python que se utiliza en aprendizaje automático y Jupyter Notebook, que es una aplicación web de código abierto que permite crear, compartir y editar documentos en los que se puede ejecutar código python, hacer anotaciones, insertar ecuaciones, visualizar resultados y documentar funcionalidades.

3.7. Métricas de un repositorio

En la métrica de los repositorios vamos a usar diversas características de los mismos, como son: Comunidad (evidencia de colaboración), Integración continua (evidencia de calidad), Documentación (evidencia de mantenibilidad), Historia (evidencia de evolución sostenida), Gestión de erratas (evidencia de la gestión del proyecto), Licencia (prueba de responsabilidad) y Pruebas unitarias (prueba de calidad). Además de esto, es necesario tener en cuenta el tamaño del repositorio, utilizando la métrica de las líneas de código fuente (SLOC), ya que hay que considerar que el tamaño influye en las otras dimensiones. En este trabajo se ha utilizado la utilidad CLOC [5] para recopilar la métrica SLOC de un repositorio (Figura 4.1).

- Comunidad: La presencia de un conjunto grande de desarrolladores de software de código abierto indica que hay alguna forma de colaboración y cooperación involucrada en el desarrollo del software y nos da cierta idea de que un repositorio puede contener un proyecto de ingeniería del software. Se van a calcular las contribuciones totales contando el número de confirmaciones realizadas en un repositorio cuando se registra en la base de datos. Luego agrupamos las confirmaciones por autor y seleccionamos los primeros n autores para los cuales el número acumulado de confirmaciones representaba el 80 % de las contribuciones totales. El valor de n representa la métrica de los contribuyentes principales.
- Integración continua: La métrica para la integración continua se puede definir como una función por partes como se muestra a continuación:

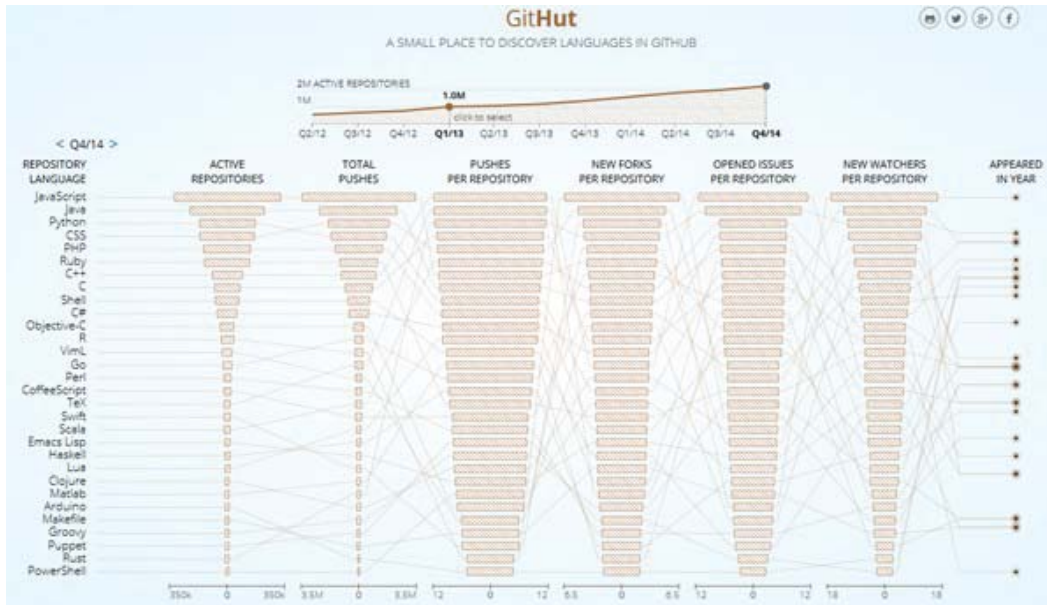


Figura 3.4: Métrica SLOCC de un repositorio [5]

$$M_{ci}(r) = \begin{cases} 1 & \text{si el repositorio } r \text{ usa un servicio de integración continua} \\ 0 & \text{en cualquier otro caso} \end{cases} \quad (3.3)$$

- Documentación: Nos limitamos a la documentación en forma de comentarios de código fuente. Se propone una relación métrica de comentarios para cuantificar la extensión de la documentación del código fuente de un repositorio, que es la relación entre el número de líneas de código de comentarios (CLOC) y el número de líneas de código fuente (SLOC) que no están en blanco en un repositorio r .

$$M_d(r) = \frac{cloc}{sloc + cloc} \quad (3.4)$$

Se puede observar que la proporción de comentarios solo cuantifica la extensión del código fuente en la documentación contenida en un repositorio. No se considerará la calidad, la antigüedad o la relevancia de la documentación.

- Historia: La presencia de un cambio sostenido indica que software se está modificando para garantizar su viabilidad. Así, un commit es la unidad por la cual se puede cuantificar el cambio del código fuente de un repositorio. Se propone por tanto que la frecuencia del commit sea una métrica de los cambios que sufre un repositorio. Con todo esto tendremos que la frecuencia del commit va a ser el número promedio de commit por mes.

$$M_h(r) = \frac{1}{m} \sum_{i=1}^m c_i \quad (3.5)$$

(3) donde:

- c_i es el número de commit para el mes i
- m es el número de meses entre el primer y el último commit del repositorio r

Cada c_i se calculará contando el número de commit registrados en la base de datos para el mes i . Sin embargo, m se calculó como la diferencia, en meses, entre la fecha del primer commit y la fecha del último commit del repositorio. Si m da como resultado 0, el valor de la métrica debe ser 0 también.

- **Gestión de erratas:** Una “errata” en GitHub puede estar asociado con una variedad de etiquetas personalizables que podrían alterar la interpretación del repositorio. Con el tiempo se han venido utilizando muchas herramientas que simplifican la gestión de grandes proyectos, por lo que un proyecto de software que emplea herramientas de gestión de proyectos es representativo de un proyecto de ingeniería. Se puede tener en cuenta entonces que el uso de herramientas de gestión de proyectos en un repositorio es un indicativo de un proyecto de ingeniería. Sin embargo, no hay una única forma de integrar esas herramientas en un repositorio y puede haber otros repositorios que utilicen esa herramienta para otros fines y que den como resultado una identificación falsa (errata) como proyecto de ingeniería.

En este proyecto se asume que el uso sostenido de la función GitHub Issues indica la gestión en un repositorio de código fuente. Por eso se propone como métrica la frecuencia de la errata para cuantificar el uso sostenido de GitHub Issues en un repositorio. Así, la frecuencia del problema va a ser la media del número de erratas ocurridos por mes.

$$M_i(r) = \frac{1}{m} \sum_{i=1}^m s_i \quad (3.6)$$

(4) donde:

- s_i es el número de erratas para el mes i
- m es el número de meses entre el primer y el último commit del repositorio r

Cada s_i se calculará contando el número de erratas registrados en la base de datos para el mes i . Sin embargo, m se calculará como la diferencia, en meses, entre la fecha del primer commit y la fecha del último commit del repositorio. Si m es 0, el valor de la métrica debe ser 0 también.

- **Licencia:** La presencia de una licencia en un repositorio de código fuente se evalúa utilizando la licencia API de GitHub. Así, se identifica la información de la licencia buscando en los archivos del repositorio las 12 licencias de código abierto más populares en GitHub.

La métrica en este caso viene dada por la función:

$$M_l(r) = \begin{cases} 1 & \text{si el repositorio tiene una licencia} \\ 0 & \text{en cualquier otro caso} \end{cases} \quad (3.7)$$

- **Pruebas unitarias:** La evidencia de las pruebas implica que los desarrolladores han dedicado tiempo y esfuerzo para garantizar que el producto tenga el comportamiento previsto. Sin embargo, la presencia de pruebas no es una medida suficiente para concluir que el software funciona correctamente, ya que influye la idoneidad de esas pruebas.

Esencialmente, para obtener la métrica de estas pruebas se requiere la ejecución de las pruebas unitarias y como métrica vamos a usar la razón entre el número de líneas fuente de código en los archivos de prueba y el número de líneas de código fuente en todos los archivos fuente.

$$M_u(r) = \frac{slotc}{sloc} \quad (3.8)$$

donde,

- *slotc* es el número de líneas de código fuente en los archivos de prueba en el repositorio *r*.
- *sloc* es el número de líneas de código fuente en todos los archivos fuente en el repositorio

Para calcular el *slotc*, primero debemos identificar los archivos de prueba. Logramos esto buscando patrones específicos del lenguaje y del marco de pruebas en el repositorio.

Resumiendo, en el cuadro 3.1 se pueden ver las distintas dimensiones utilizadas en este trabajo con sus métricas correspondientes.

3.8. Herramientas informáticas

Python es un lenguaje de programación interpretado cuya principal filosofía es que sea legible por cualquier persona con conocimientos básicos de programación. Además, posee una serie de características que lo hacen muy particular y que, sin duda, le aportan muchas ventajas y están en la raíz de su uso

DIMENSIÓN	MÉTRICA
Comunidad	Colaboración
Integración continua	$M_{ci}(r) = \begin{cases} 1 & \text{si el repositorio } r \text{ usa un servicio de integración continua} \\ 0 & \text{en cualquier otro caso} \end{cases}$
Documentación	$M_d(r) = \frac{cloc}{sloc+cloc}$
Historia	$M_h(r) = \frac{1}{m} \sum_{i=1}^m c_i$
Gestión de erratas	$M_i(r) = \frac{1}{m} \sum_{i=1}^m s_i$
Licencia	$l(r) = \begin{cases} 1 & \text{si el repositorio tiene una licencia} \\ 0 & \text{en cualquier otro caso} \end{cases}$
Prueba unitaria	$M_u(r) = \frac{slotc}{sloc}$

Cuadro 3.1: Dimensiones y métricas utilizadas.

tan extendido: es gratuito, se puede ejecutar en todas las plataformas y utiliza un lenguaje muy flexible y fácil de aprender.

Teniendo como eje común el identificador, se han relacionado con un mismo identificador las distintas variables con Anaconda Distribution, que es una distribución de Python que funciona como un gestor de entorno y paquetes, y posee una colección de más de 720 paquetes de código abierto.

Además, se ha utilizado Jupyter Notebook que es una aplicación web de código abierto que permite crear, compartir y editar documentos en los que se puede ejecutar código python, hacer anotaciones, insertar ecuaciones, visualizar resultados y documentar funcionalidades.

Para el desarrollo del proyecto se ha tenido que importar las siguientes librerías:

Pandas: es una librería de código abierto que aporta a Python unas estructuras de datos fáciles de usar, junto con un gran número de funciones esenciales para el análisis de datos. Con la ayuda de Pandas podemos trabajar con datos estructurados de una forma más rápida y expresiva.

Numpy: es el paquete fundamental para la computación científica en Python y se utiliza porque es muy eficiente para almacenar y manipular datos.

Matplotlib: es una librería muy popular en Python que se usa para visualizaciones y gráficos.

Scikit-learn: es la principal librería que existe para trabajar con Machine Learning e incluye la implementación de un gran número de algoritmos de aprendizaje. Se puede utilizar para clasificaciones, extracción de características, regresiones, agrupaciones, reducción de dimensiones, selección de modelos, o preprocesamiento. Posee una API que es consistente en todos los modelos y se integra muy bien con el resto de los paquetes científicos que ofrece Python. Esta librería también nos facilita las tareas de

evaluación, diagnóstico y validaciones cruzadas ya que nos proporciona varios métodos de fábrica para poder realizar estas tareas en forma muy simple.

Capítulo 4

Diseño e implementación

4.1. Metodología

En nuestro estudio se han utilizado un conjunto de repositorios que se han clasificado manualmente como que contienen proyectos de ingeniería de acuerdo con alguna definición específica de un proyecto de ingeniería. En este contexto, se van a utilizar las dos definiciones siguientes de proyecto de ingeniería:

Organización: se dice que un repositorio contiene un proyecto de ingeniería si es similar a los repositorios propiedad de organizaciones populares de ingeniería.

Utilidad: se dice que un repositorio contiene un proyecto de ingeniería si es similar a los repositorios que tienen una utilidad de propósito general para los usuarios. Por ejemplo, un repositorio que contiene un plug-in de Chrome se considera que tiene una utilidad de propósito general, sin embargo, un repositorio que contiene una aplicación móvil desarrollado por un estudiante como un proyecto de curso no puede considerarse que tenga una utilidad de propósito general.

El proceso seguido en la realización de este trabajo ha sido el siguiente:

- Para empezar se realizaron una serie de pruebas preliminares, para las que se eligieron manualmente 1000 repositorios de forma que la mitad contenían un proyecto de ingeniería y la otra mitad no. En estas pruebas preliminares primero evaluamos las 7 dimensiones y después aplicamos los dos criterios de clasificación, el basado en puntuación y Random Forest. De esta forma obtenemos una asignación de engineered o non-engineered que posteriormente podemos evaluar.
- A continuación se hicieron unas pruebas de validación sobre un conjunto de 300 repositorios para los que se estableció manualmente la verdadera clasificación para evaluar el rendimiento de la clasificación.

- El siguiente paso fue establecer la verdad fundamental para lo que se cogieron esos 300 repositorios para evaluar el trabajo de los clasificadores basados en puntuación y Random Forest.
- Una vez establecida la verdad fundamental se realizaron la validación interna, mediante la utilización de los conjuntos de datos de organización y utilidad, y la validación externa, mediante los datos obtenidos por otros investigadores.
- Por último, se aplicaron los clasificadores a la totalidad de los repositorios en activo (2.316.524) el día de la medida (22 de febrero de 2020) y se obtuvieron los resultados definitivos del trabajo.

4.2. Pasos seguidos

Para la obtención de las dimensiones se ha utilizado el software reaper alojado en la dirección web <https://github.com/RepoReapers/reaper>. Una vez obtenida la base de datos de los repositorios alojados en GHTorrent, lo que se hace es aplicar reaper y obtener las tablas con los valores de las dimensiones.

El cuadro 4.1 representa los resultados obtenidos al ejecutar el programa reaper en la base de datos que anteriormente hemos sacado de GHTorrent. De esta forma, obtenemos, entre otros, los valores de las 7 dimensiones. A su vez, esta tabla la podemos relacionar con los distintos repositorios de GitHub a través del project-id.

En la dirección <https://gthorrent.org/files/schema.pdf> se puede ver el esquema de la base de datos que se ha utilizado (Figura 4.1). Se han utilizado los datos id y type de la tabla de users, que representa el identificador del usuario y el tipo de usuario (organización o utilidad) respectivamente.

Por otro lado, los parámetros id, owner-id, language y updated-at de la tabla de projects relaciona el identificador con el usuario que lo creó, con el lenguaje de programación del proyecto y cuando se modificó por última vez.

Por último, se han utilizado los parámetros id, autor-id y project-id de la tabla commits para relacionar el identificador con el usuario que realizó ese commits y con el proyecto en donde se realizó.

4.3. Procedimientos informáticos

El proceso que se ha seguido es el siguiente:

- Primero hay que llamar a unas librerías que el programa necesitará:

```
import pandas as pd
```

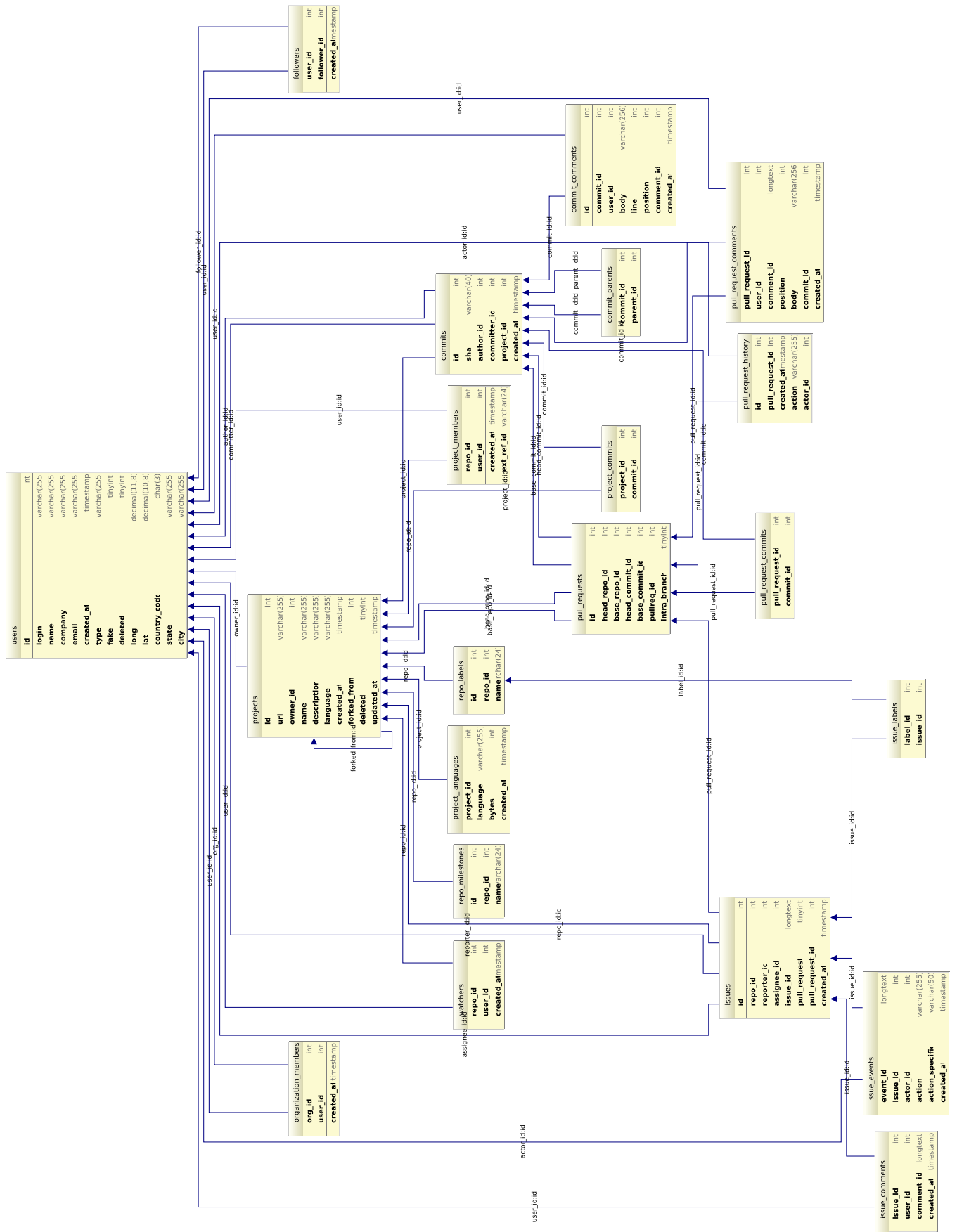



Figura 4.1: Esquema de la base de datos utilizada

reaper results	
project_id	int(11)
architecture	double
community	double
continuous_integration	double
documentation	double
history	double
license	double
management	double
project_size	double
state	double
stars	double
unit_test	double
score	double

Cuadro 4.1: Parámetros obtenidos al ejecutar reaper en la base de datos.

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
from sklearn.model_selection import train_test_split
```

- Se carga el fichero de datos que contiene entre otros valores los de las siete dimensiones (formato de texto, csv):

```
Datos = pd.read_csv('dataset.csv')
```

- Se eliminan los datos con valores missing porque Python no puede hacer árboles con datos missing:

```
Datos[state] = Datos[state].map(active: 1, dormant: 0)
```

```
datoslimpios = Datos.dropna()
```

- Se crea la muestra de entrenamiento y de test, tanto para predictores como para la variable objetivo, siendo test el 20 % y de entrenamiento el 80 %.

```
X= datoslimpios.drop(columns=['project_id','state']).values
```

```
Y= datoslimpios ['state'].values
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2)
```

- Se importa desde `sklearn.ensemble`, el algoritmo de Random Forest:

```
from sklearn.ensemble import RandomForestClassifier
```

- Se inicializa el algoritmo Random Forest e indicamos el número de árboles que vamos a construir, en este caso se ha utilizado 15 debido a la velocidad de cómputo aunque cuantos más árboles se construyan más precisión habrá en la clasificación (se suelen crear 100).

```
clasificador = RandomForestClassifier(n_estimators=15)
```

- Construimos el modelo sobre los datos de entrenamiento:

```
modelo = clasificador.fit(X_train, Y_train)
```

- Se dibuja el clasificador Random Forest de la siguiente manera:

```
plt.bar(range(len(modelo.feature_importances_)), modelo.feature_importances_)
```

```
plt.show()
```

- Para casos de predicción se puede sacar el índice Accuracy Score, que es la precisión de las predicciones correctas. Sin embargo, para poder obtener dicho índice, primero hay que importar la librería y luego ejecutar la función como se muestra a continuación:

```
import sklearn.metrics
```

```
sklearn.metrics.accuracy_score(tar_test, predicciones)
```

- También se puede sacar la matriz de confusión de las predicciones del grupo Test que nos permite ver los aciertos y los errores que comete nuestro modelo y en python se realiza de la siguiente manera:

```
sklearn.metrics.confusion_matrix(tar_test, predicciones)
```


Capítulo 5

Resultados

En este capítulo se presentan los resultados de la validación de los clasificadores para identificar proyectos de software de ingeniería en una muestra de 2.316.524 de los 2.634.807 repositorios de GitHub que estaban activos en el momento en que se realizó el análisis. Como es bien sabido, hay muchos más millones de repositorios alojados en GitHub, pero se ha estado limitado por la velocidad de procesamiento de los equipos y la capacidad de almacenamiento y tratamiento de los datos. Esa fue la causa por la que nos quedamos con ese número de repositorios. Además, como ya se ha comentado, se han utilizado dos clasificadores (basado en la puntuación y Random Forest) y dos conjuntos de datos diferentes (organización y utilidad).

5.1. Pruebas preliminares

Para empezar a probar nuestro método, de los 1.000 repositorios seleccionados se escogieron 500 pertenecientes a Organización y 500 a Utilidad.

Organización: se han examinado manualmente repositorios de organizaciones conocidas como Amazon, Apache, Facebook, Google, Microsoft, Mozilla, Oracle, Telefónica, etc. y se han elegido manualmente un conjunto de 500 repositorios que cumplen alguna de estas características (incluidos en el apéndice de la memoria): tiene licencia de código abierto, usa comentarios para documentar el código, usa integración continua y contiene pruebas unitarias. De esos 500 repositorios, se han elegido 250 con proyecto de ingeniería y 250 que no contienen proyecto de ingeniería. En la figura 5.1 se muestra el número de repositorios de cada tipo (“engineered” y “non-engineered”) agrupados por lenguaje de programación en la organización, donde no proyecto significa en este caso que no son repositorios de organizaciones.

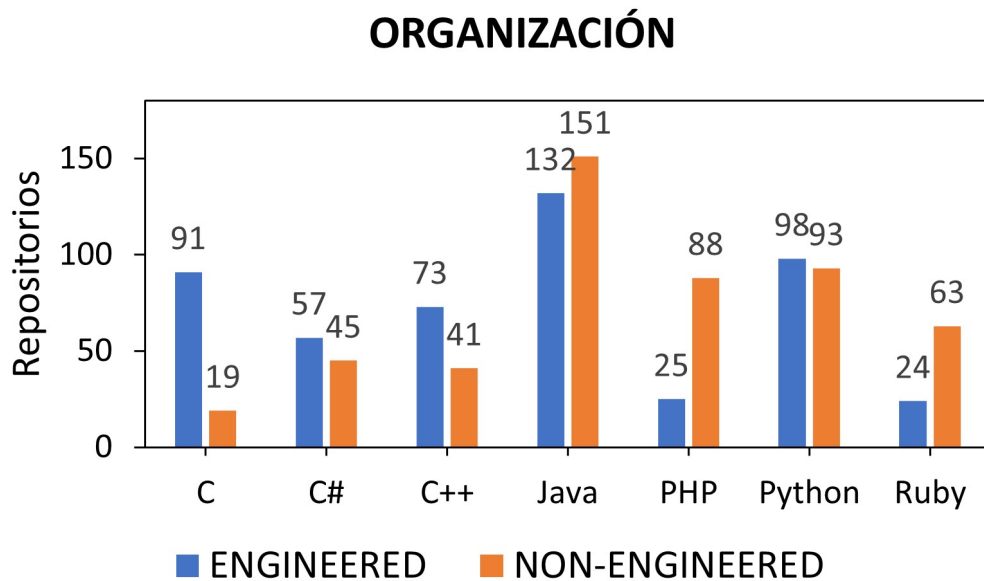


Figura 5.1: Número de repositorios en la organización agrupados por lenguajes de programación.

Utilidad: estos repositorios se escogieron manualmente de una muestra aleatoria de 2.316.524 repositorios. Se seleccionaron también 500 repositorios (incluidos en el apéndice de la memoria) de la misma forma que en el caso de organización, es decir, 250 de ellos con proyecto de ingeniería y 250 que no contienen proyecto de ingeniería. Se eligieron siguiendo estos criterios:

- El repositorio tiene documentación suficiente que permita a muchos usuarios utilizarlo para distintas aplicaciones.
- El repositorio contiene una aplicación o servicio que puede ser utilizado por muchas personas distintas como por ejemplo un repositorio sobre programación de Arduino.
- El repositorio no contiene indicaciones que indiquen que es simplemente un trabajo personal de un alumno.

En la figura 5.2 se muestra el número de repositorios de cada tipo (“engineered” y “non-engineered”) agrupados por lenguaje de programación en el conjunto de datos de utilidad.

Por otro lado, en la figura 5.3 se puede ver la distribución del número de repositorios de cada tipo (“engineered” y “non-engineered”) pertenecientes a organización y utilidad.

En las figuras 5.4 y 5.5 se muestran las distribuciones de las siete dimensiones obtenidas de los repositorios en la organización y los conjuntos de datos de utilidad, respectivamente.

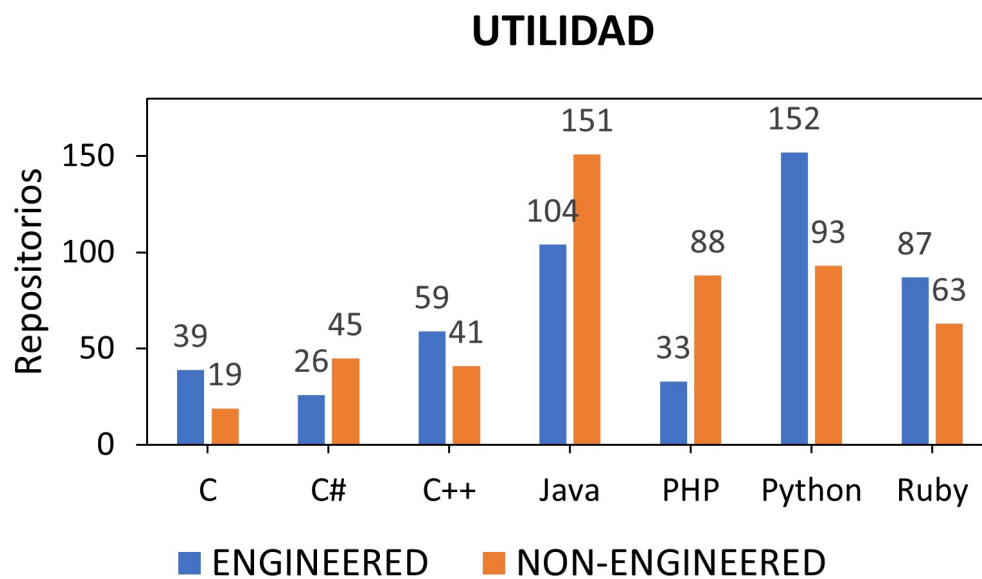


Figura 5.2: Número de repositorios en la utilidad del conjunto de datos agrupados por lenguajes de programación.

Una vez que ya sabemos calcular las dimensiones de los repositorios, nos surge la duda de qué relación puede haber entre las distintas dimensiones y si que una sea mayor que otra tiene alguna influencia en el resultado final de catalogar a ese repositorio como ingeniería de software o no. Para saber esto hemos recurrido a métodos estadísticos como la correlación de Spearman Rank Co-efficient (ρ) para evaluar la correlación entre las diferentes dimensiones.

El coeficiente de correlación de Spearman (ρ) es una medida no paramétrica de la correlación de rango (dependencia estadística del ranking entre dos variables). Se utiliza principalmente para el análisis de datos, así como para medir la fuerza y la dirección de la asociación entre dos variables clasificadas.

Con otras palabras, es una medida de la correlación (la asociación o interdependencia) entre dos variables aleatorias (tanto continuas como discretas). Para calcular ρ , los datos son ordenados y reemplazados por su respectivo orden.

Dicho coeficiente ρ viene dado por la ecuación:

$$\rho = 1 - \frac{6 \sum D^2}{N(N^2 - 1)} \quad (5.1)$$

donde N es el número de parejas de datos y D es la diferencia entre los correspondientes estadísticos de orden de $x - y$.

En la figura 5.6 se muestran los valores de la ρ de Spearman, cuando es estadísticamente significativa

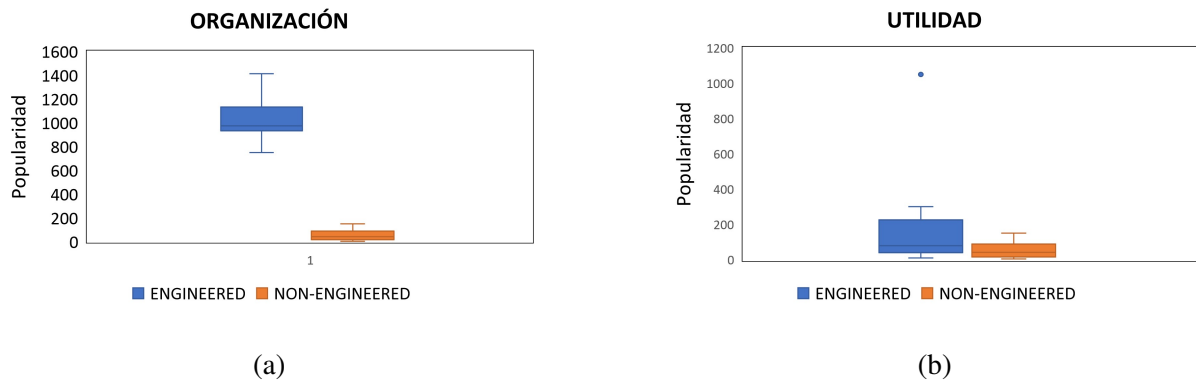


Figura 5.3: Distribución del número de puntuación de los repositorios; (a) en la organización y (b) utilidad del conjunto de datos

en el p -valor < 0.05 , entre pares de dimensiones en los conjuntos de datos de organización y utilidad. También se pueden observar en dicha figura las correlaciones que no son estadísticamente significativas y se han representado con un guión.

Como se ve en la figura 5.6, hay una correlación de moderada a fuerte entre las diversas dimensiones en el conjunto de datos de la organización. Como se ve en dicha figura, a excepción de la dimensión de integración continua en el conjunto de datos de utilidad, el tamaño del repositorio es estadísticamente significativo (p -valor < 0.05) asociado con las dimensiones de valor binario con tamaño de efecto medio. Los resultados de la asociación indican que es más probable que un repositorio más grande tenga una integración y/o licencia.

DIMENSIÓN	PESO(wd)	Umbral(td)	
		Organización	Utilidad
Comunidad	20 %	2 %	2 %
Integración continua	5 %	1 %	1 %
Documentación	10 %	0.0929 %	0.0583 %
Historia	20 %	2.0895 %	0.1500 %
Tamaño del repositorio	10 %	190 %	160 %
Gestión de erratas	5 %	0.3111 %	0.1611 %
Licencia	20 %	1 %	1 %
Prueba unitaria	10 %	0.0506 %	0.0260 %

Cuadro 5.1: Pesos y umbrales de las dimensiones utilizadas.

5.2. Clasificadores

Ahora que ya sabemos calcular los distintos parámetros (dimensiones) y sabemos correlacionarlos, siempre con un conjunto de repositorios elegidos manualmente y que estamos seguros pertenecen a los dos tipos de repositorios Organización y Utilidad, tenemos que utilizar los dos tipos de clasificadores para conseguir seleccionar los repositorios que son de ingeniería y los que no.

El valor umbral y los pesos relativos correspondiente a cada una de las siete dimensiones que hemos utilizado se basa en los utilizados por otros investigadores [17] y se pueden ver en el cuadro 4.1. Dichos pesos se pueden cambiar para tener un control más fino sobre la clasificación y seleccionar repositorios que se adapten a otros estudios.

5.3. Validación

En esta sección, se presenta el enfoque y los resultados de la validación de los clasificadores basados en la puntuación y Random Forest entrenados con conjuntos de datos de organización y utilidad. La validación se ha realizado en un conjunto de 300 repositorios llamado conjunto de validación, para el que se estableció manualmente la verdadera clasificación. Se consideró la validación desde dos perspectivas: interna, en la que se validaba el rendimiento de los propios clasificadores, y externa, en la que se comparaba el rendimiento de los clasificadores con el de un esquema de clasificación que utilizaron Ray y colaboradores como criterio (Ray et al. 2014), como por ejemplo, el tamaño del proyecto, tamaño del

CLASIFICADOR	FPR	FNR	PRECISIÓN	RECALL	F-MEASURE
BASADO EN PUNTUACIÓN	20 %	37 %	73 %	60 %	66 %
RANDOM FOREST	5 %	58 %	85 %	43 %	57 %

Cuadro 5.2: Resultados de los clasificadores Basado en Puntuación y Random Forest testeados con los datos de organización

equipo y tamaño del commit. Se utilizó tasa de falsos positivos (FPR), tasa de falsos negativos (FNR), precisión, memoria y F-medida para evaluar el rendimiento de la clasificación.

5.3.1. Establecer la verdad fundamental

La evaluación del trabajo de cualquier clasificador normalmente implica el uso del clasificador para evaluar un conjunto de muestras para las que se conoce la clasificación de la verdad fundamental. En líneas similares, para evaluar el trabajo de los clasificadores basados en la puntuación y Random Forest, compusimos manualmente un conjunto de 300 repositorios, de los cuales se sabe a ciencia cierta que 150 de ellos tienen proyectos de ingeniería y los restantes no.

En la figura 5.7 se muestra la distribución de las siete dimensiones recogidas de los repositorios en el conjunto de validación. Como se ve en la figura, los repositorios que contienen proyectos de software de ingeniería tienden a tener valores medianos más altos en casi todas las dimensiones.

5.3.2. Validación interna

En este tipo de validación, se evalúan los clasificadores basados en la puntuación y de Random Forest que han sido entrenados mediante la utilización de los conjuntos de datos de organización y utilidad.

Conjunto de datos de organización. En el cuadro 5.1 se puede apreciar que el clasificador basado en la puntuación tiene mejores resultados que el clasificador Random Forest en términos de medida F. Si se desea una tasa de falsos positivos más baja, el clasificador Random Forest puede ser más adecuado ya que tiene una tasa de falsos positivos considerablemente más baja que el clasificador basado en la puntuación.

Conjunto de datos de utilidad. Claramente, el modelo de Random Forest funciona mejor que el modelo basado en la puntuación. Como se puede observar en el cuadro 5.8 hay una gran tasa de falsos positivos del clasificador basado en la puntuación. La gran tasa de falsos positivos indica que el clasificador puede haber clasificado casi todos los repositorios como que contienen un proyecto de software de

ingeniería.

CLASIFICADOR	FPR	FNR	PRECISIÓN	RECALL	F-MEASURE
BASADO EN PUNTUACIÓN	76 %	1 %	52 %	98 %	68 %
RANDOM FOREST	20 %	17 %	80 %	86 %	83 %

Cuadro 5.3: Resultados de los clasificadores Basado en Puntuación y Random Forest testeados con los datos de utilidad.

5.3.3. Validación externa

La validación externa consiste en comparar los clasificadores basados en la puntuación y Random Forest con el clasificador basado en citas que utilizaron otros investigadores (Ray et al. 2014). Así, utilizando la popularidad como parámetro, diversos investigadores como Jarcky et al. (2014) y Guzmán et al (2014) decidieron si un repositorio contiene un proyecto de ingeniería en función del número de citas y/o número de estrellas. De esta forma, Ray et al. (2014) lograron ordenar los 50 repositorios más populares en 19 idiomas, donde la popularidad se estima con el número de estrellas.

En nuestro estudio, para realizar la validación externa comparamos los resultados obtenidos con los clasificadores basados en la puntuación y Random Forest con los obtenidos mediante el clasificador basado en estrellas (citas). De esta forma, diremos que un repositorio contiene un proyecto de ingeniería si tiene 1.000 o más estrellas. Además, probamos otros umbrales como 500, 50 y 10 para poder comparar cómo influye el umbral elegido en los resultados (ver cuadro 5.4).

Como se puede ver, para umbrales altos del orden de 1.000 y 500 los resultados son malos, ya que se obtiene que ningún repositorio tiene proyecto de ingeniería, lo cual es falso. Además, según disminuye el umbral, los resultados van mejorando. Estos resultados son sorprendentes, ya que se tiende a creer, por intuición, que a mayor número de estrellas, es decir, mayor popularidad, es más probable que contenga un proyecto de ingeniería.

5.4. Aplicación de los clasificadores

En esta sección, se presentan los resultados de la aplicación de los clasificadores basados en la puntuación y Random Forest para identificar proyectos de software de ingeniería en una muestra de 2.316.524 repositorios GitHub. En el cuadro 5.10 se pueden observar el número de repositorios tanto de los datos de Organización como de Utilidad que tienen un proyecto de software de ingeniería cuando

UMBRAL	FPR	FNR	PRECISIÓN	RECALL	F-MEASURE
1.000	0 %	100 %	NA	0 %	NA
500	0 %	100 %	NA	0 %	NA
50	0 %	85 %	100 %	16 %	28 %
10	1 %	64 %	94 %	30 %	45 %

Cuadro 5.4: Rendimiento del clasificador basado en la popularidad (puntuación de estrellas) en función del umbral mínimo exigido.

DATOS	CLASIFICADOR	Nº REPOSITORIOS	PORCENTAJE
-2*ORGANIZACIÓN	BASADO EN PUNTUACIÓN	244.624	10,56 %
	RANDOM FOREST	145.246	6,23 %
-2*UTILIDAD	BASADO EN PUNTUACIÓN	1.645.196	71,02 %
	RANDOM FOREST	544.615	23,51 %

Cuadro 5.5: Número de repositorios que contienen un proyecto de software de ingeniería en función de los clasificadores utilizados con los datos pertenecientes a organización y utilidad.

se usan los clasificadores Random Forest y el Basado en Puntuación. En dicha tabla se puede observar que el número de repositorios que tienen una utilidad de propósito general obtenido por el clasificador basado en la puntuación es considerablemente alto, lo cual podría deberse al número bajo de proyectos analizados del conjunto de datos utilizados (Munaiah et al., 2017). Una agrupación de resultados más detallada se puede observar en las figuras 5.8 y 5.9, donde se pueden ver el número de repositorios por lenguaje de programación para los dos conjuntos de datos (organización y utilidad).

5.5. Discusión

En este estudio, se han identificado los repositorios que contienen proyectos de software de ingeniería de acuerdo con dos definiciones diferentes del término. La aplicación de una de las definiciones incluía la capacitación de dos clasificadores que utilizaban repositorios en el conjunto de datos de la organización. Se podría suponer que el resultado de la aplicación de estos clasificadores puede ser igualado porque todos los repositorios de cualquier organización en GitHub contienen un proyecto de software de ingeniería. Sin embargo, esto no siempre es así.

El conjunto de validación contiene 300 repositorios de los cuales 90 pertenecen a organizaciones.

Se eligieron 150 que contienen proyectos de software de ingeniería y los 150 restantes no contienen proyectos de software de ingeniería.

En la figura 5.10 se muestra una comparación entre la distribución de las siete dimensiones recogidas de repositorios propiedad de organizaciones, pero con diferentes etiquetas de clasificación manual. Como se puede ver en esta figura, la diferencia en la distribución de las dimensiones proporciona pruebas cualitativas que respaldan la idea de que no todos los repositorios de propiedad de las organizaciones son similares entre sí. En líneas similares, comparamos la distribución de las siete dimensiones recogidas de repositorios conocidos por contener proyectos de software de ingeniería, pero con el subgrupo de organizaciones y usuarios.

La comparación se muestra en la figura 5.11, donde las medianas de la mayoría de las dimensiones son comparables entre los repositorios de propiedad de los usuarios y los de propiedad de las organizaciones. En este estudio se puede observar, un número considerable de repositorios clasificados como que contienen proyectos de software de ingeniería que son propiedad de usuarios individuales. Por otra parte, un número considerable de repositorios clasificados como que no contenían un proyecto de software de ingeniería eran propiedad de organizaciones. El filtrado de repositorios basado únicamente en que el propietario es una organización puede dar lugar a la exclusión de repositorios potencialmente pertinentes, de propiedad de los usuarios, o a la inclusión de repositorios que pueden no contener proyectos informáticos de ingeniería o ambos.

5.6. Puntos débiles del estudio

Aunque ya se ha comentado que el criterio elegido en este trabajo para elegir, tanto las dimensiones como los pesos específicos y los umbrales, en el clasificador basado en la puntuación ha sido el de trabajos previos realizados con anterioridad por otros investigadores, es cierto que dicho criterio es subjetivo y, por lo tanto, se podrían elegir otros distintos a pesar de que los resultados que hemos obtenidos son satisfactorios, ya que están en concordancia con los obtenidos en esos otros trabajos. Por ejemplo, también se podría pensar en utilizar algoritmos de aprendizaje automático o un clasificador basado en popularidad ponderado.

Al describir las dimensiones medidas por Reaper Primera vez que hablas de Reaper en toda la memoria en la Sección 3.6, se describen las limitaciones en dicho estudio para recopilar la métrica de dimensiones de un repositorio. Estas limitaciones pueden llevar a la inducción de sesgos en los repositorios seleccionados.

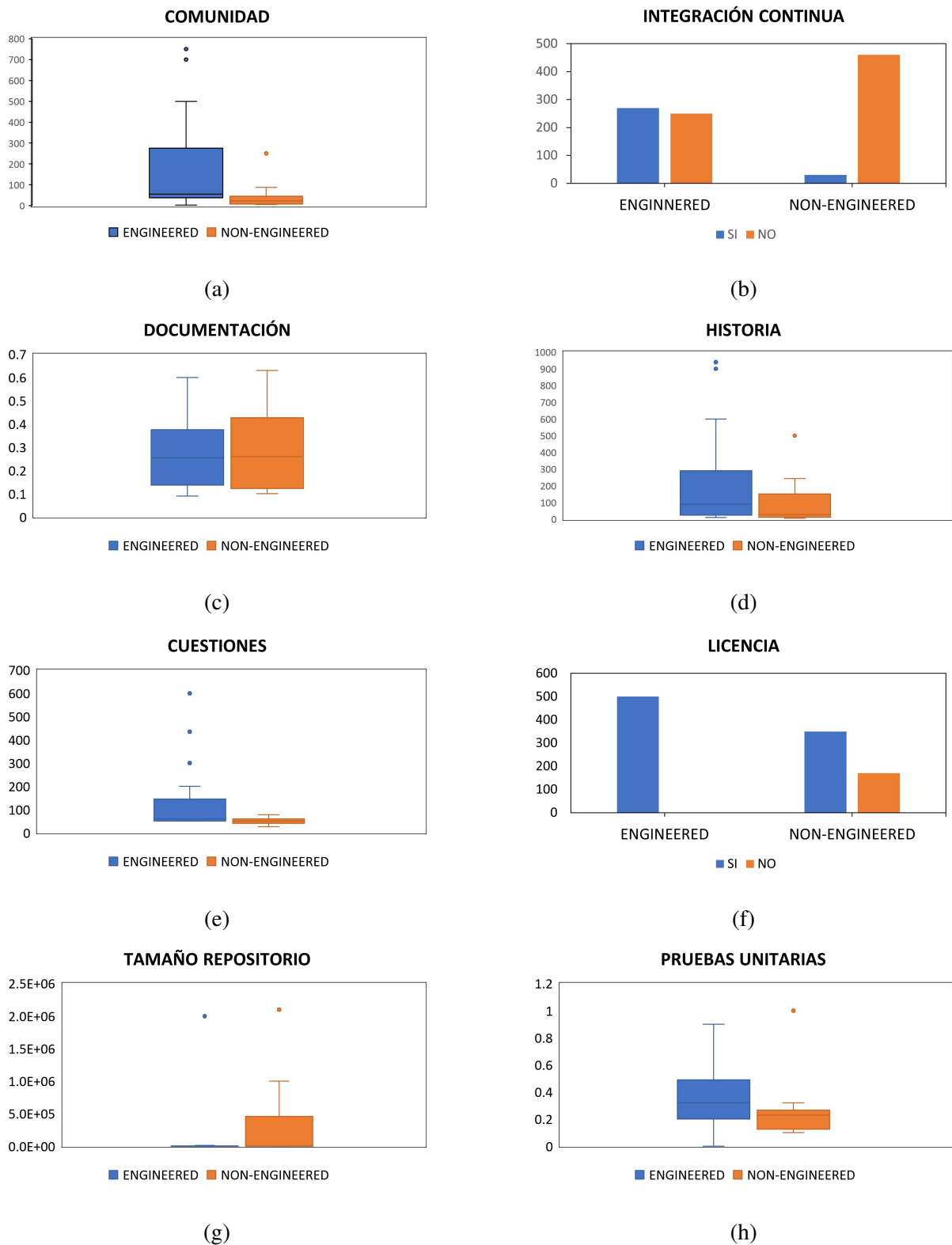


Figura 5.4: Distribución de las dimensiones de repositorios en el conjunto de datos de la organización

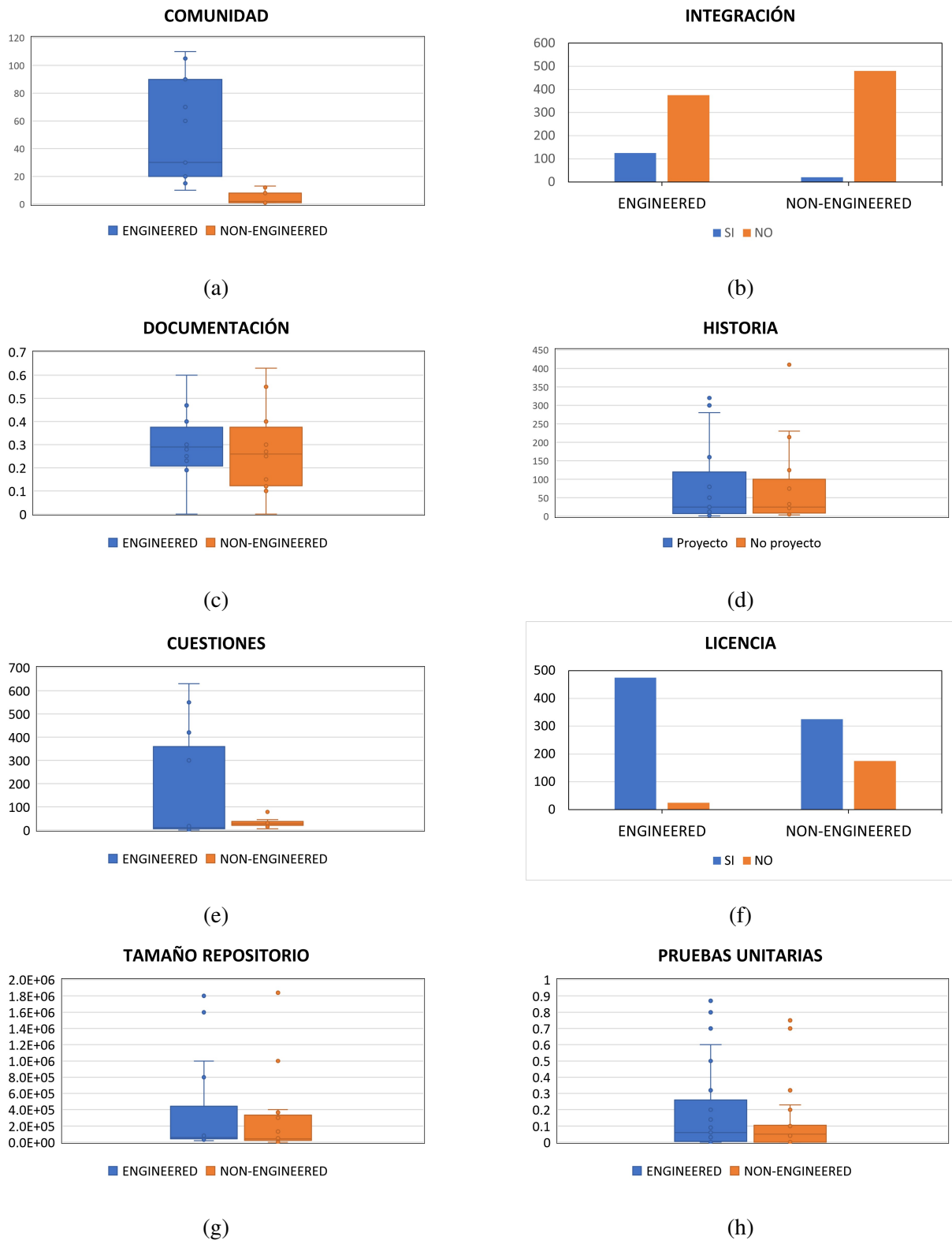


Figura 5.5: Distribución de las dimensiones obtenidas de los repositorios en el conjunto de datos de utilidad.

Organización						
Pruebas Unitarias						1.00
Tamaño del Repositorio					1.00	0.1429
Cuestiones				1.00	0.2381	0.2619
Historia			1.00	0.5000	0.0952	0.3571
Documentación		1.00	0.5714	0.4286	-	0.1429
Comunidad	1.00	-	-	-	0.6190	0.1429
	Comunidad	Documentación	Historia	Cuestiones	Tamaño del Repositorio	Pruebas Unitarias

(a)

Utilidad						
Pruebas Unitarias						1.00
Tamaño del Repositorio					1.00	-
Cuestiones				1.00	0.3333	0.0714
Historia			1.00	0.0476	0.5952	-
Documentación		1.00	-	0.5238	0.0476	-
Comunidad	1.00	0.5476	0.2381	0.6905	0.0476	-
	Comunidad	Documentación	Historia	Cuestiones	Tamaño del Repositorio	Pruebas Unitarias

(b)

Figura 5.6: ρ de Spearman entre pares de dimensiones en la organización (a) y conjuntos de datos de utilidad (b) con - (guión) representando las correlaciones estadísticamente insignificantes.

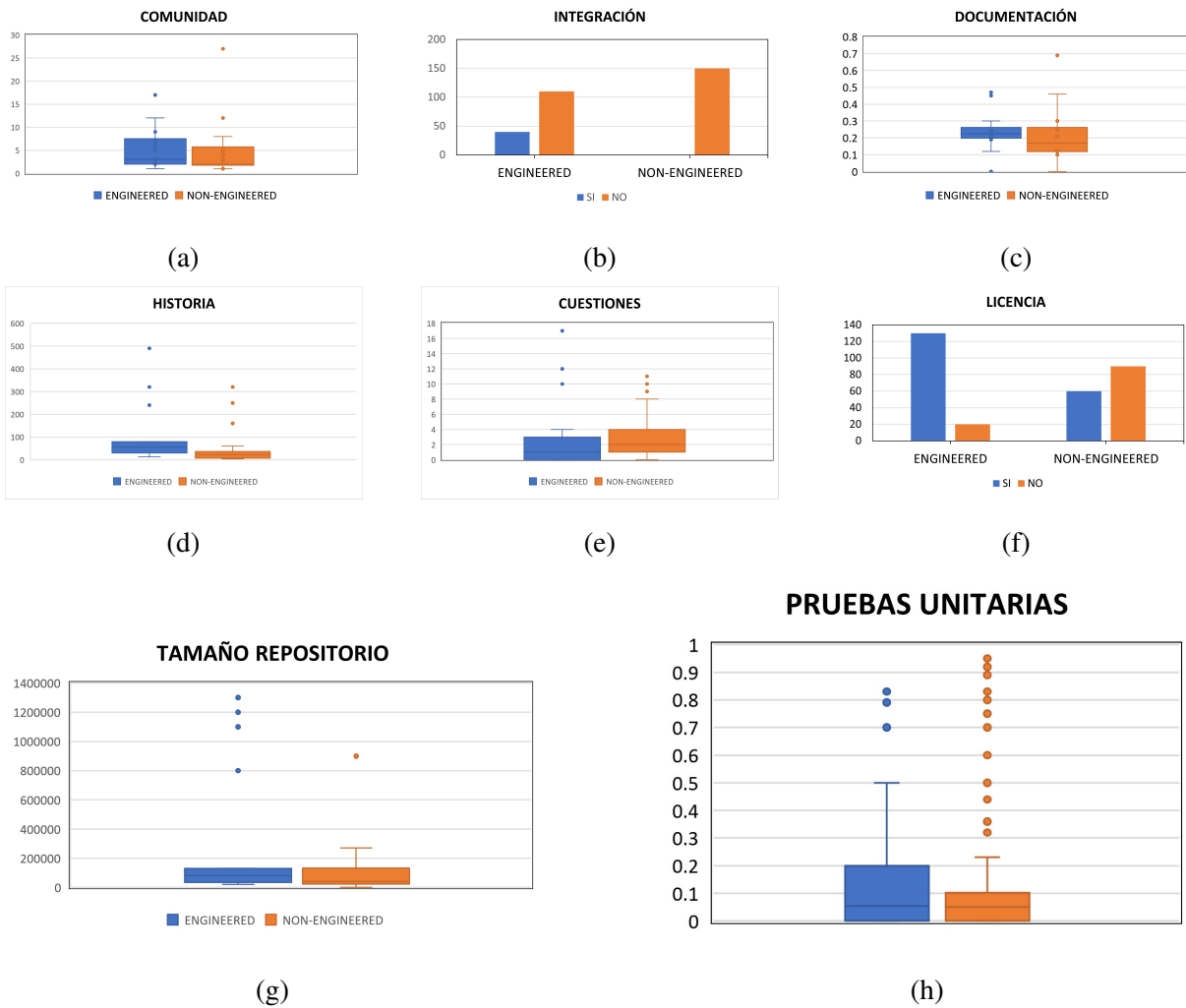


Figura 5.7: Distribución de las dimensiones de los repositorios en el conjunto de validación.

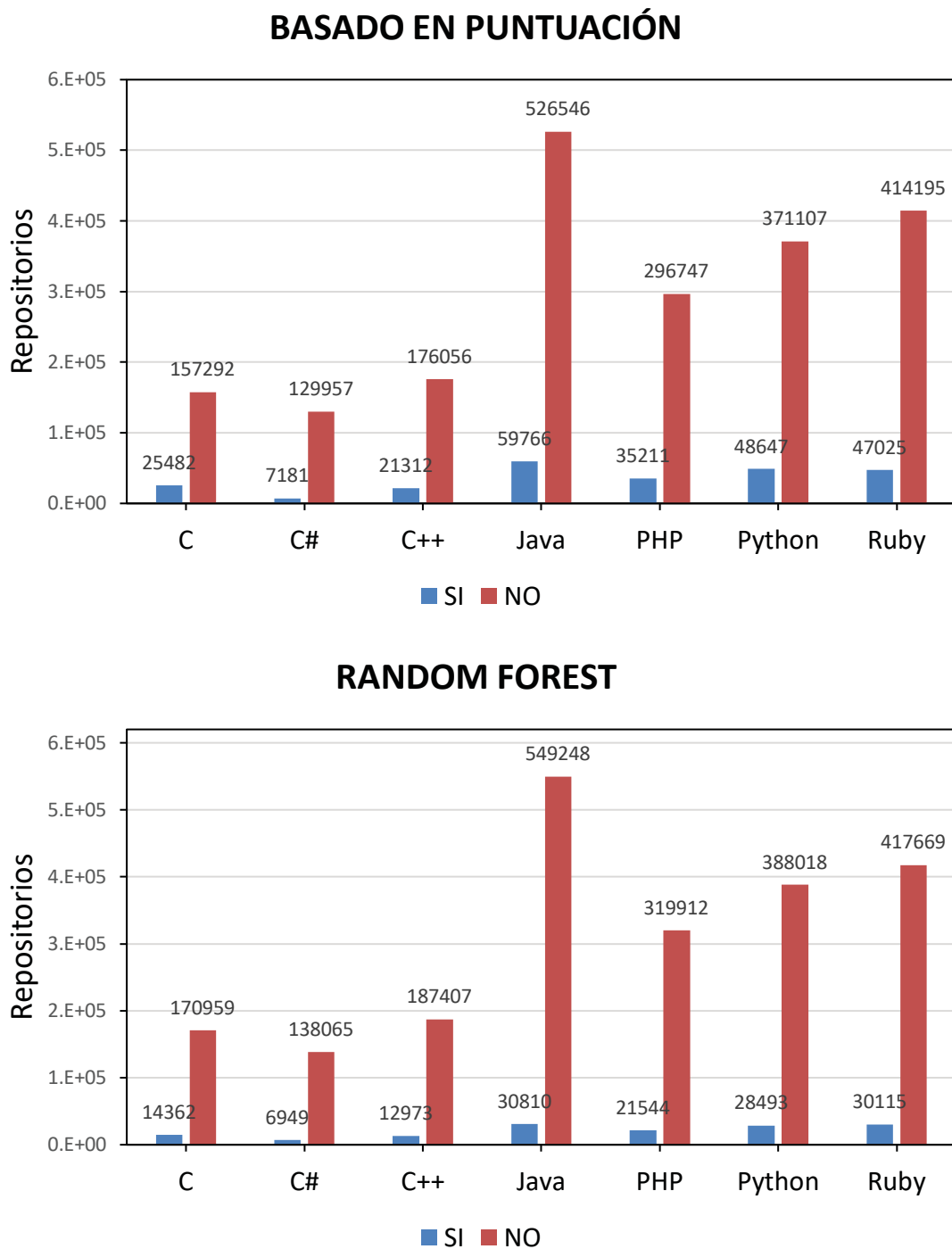


Figura 5.8: Número de repositorios obtenidos por los clasificadores basados en la puntuación y Random Forest agrupados por lenguajes de programación (ORGANIZACIÓN).

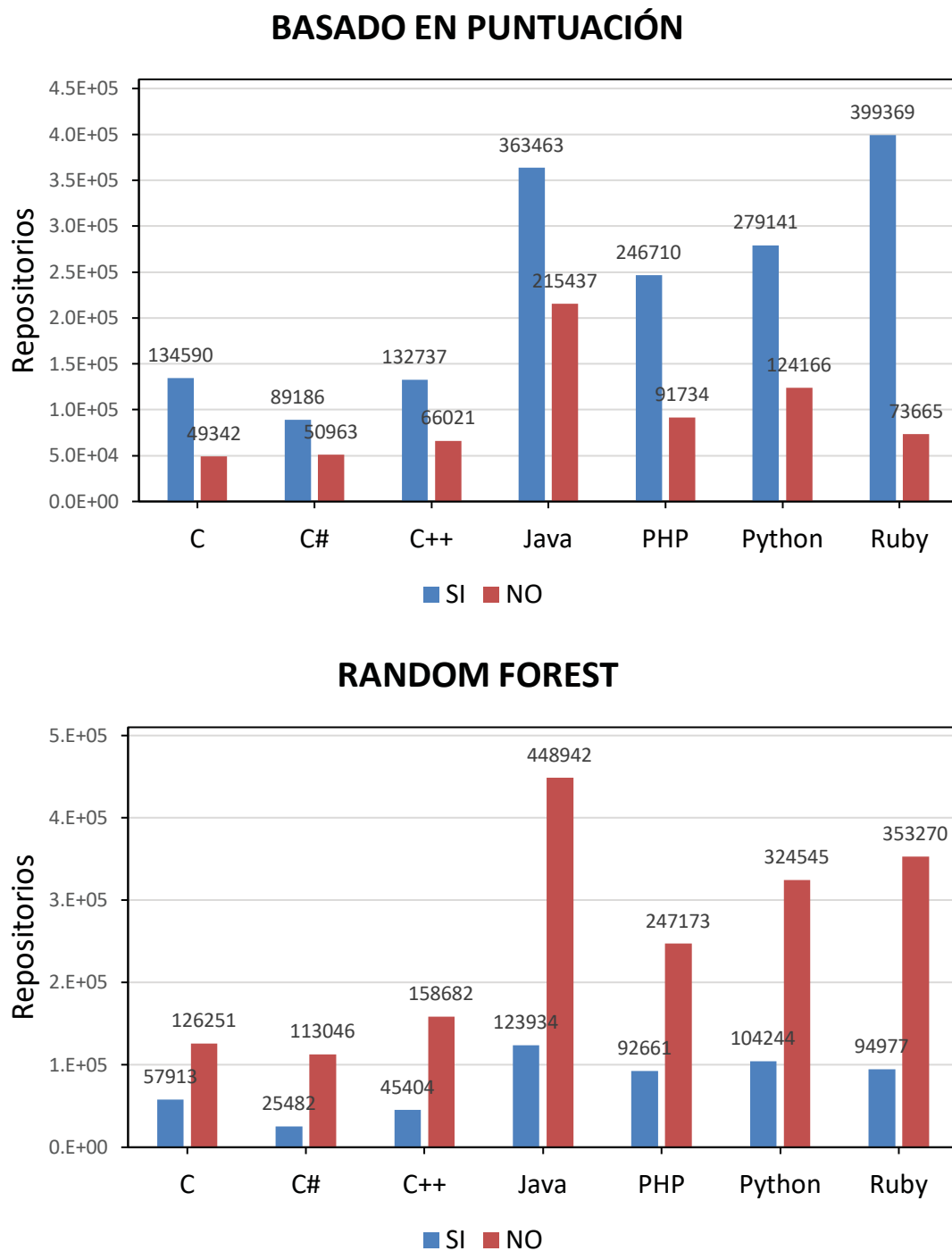


Figura 5.9: Número de repositorios obtenidos por los clasificadores basados en la puntuación y Random Forest agrupados por los lenguajes de programación (UTILIDAD).

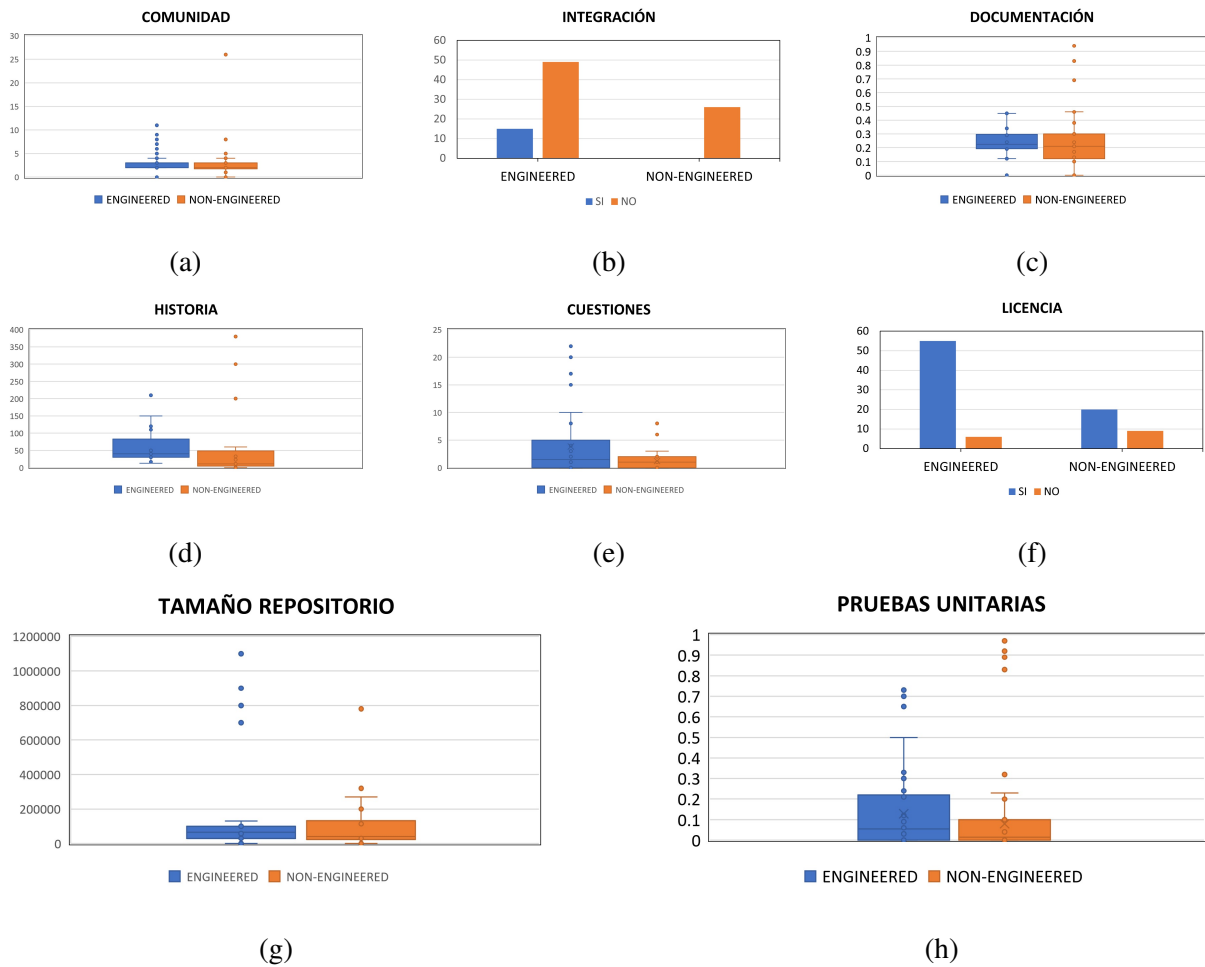


Figura 5.10: Distribución de las dimensiones de los repositorios en el conjunto de validación.

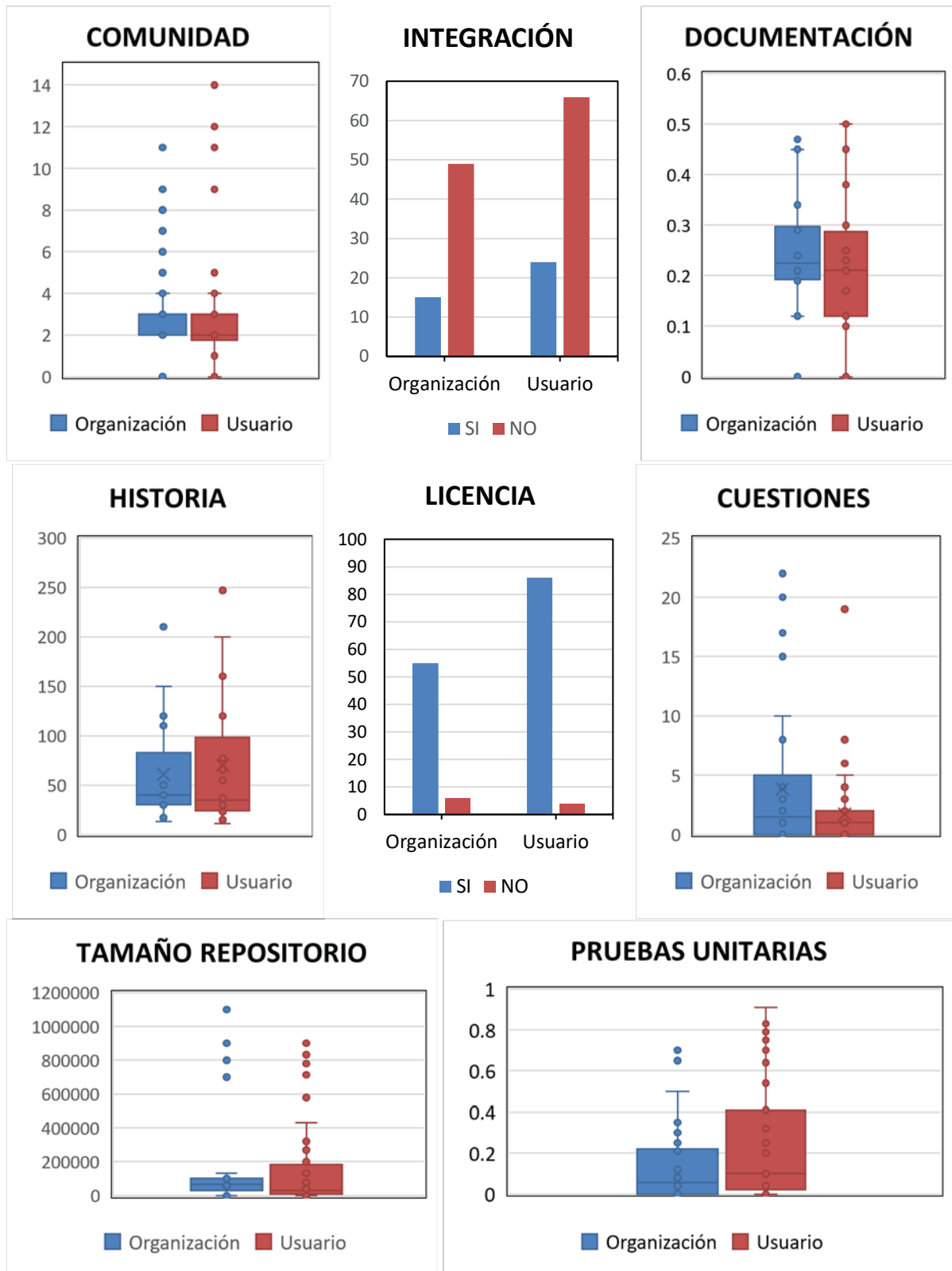


Figura 5.11: Comparación de la distribución de las dimensiones de los repositorios que contienen proyectos informáticos de ingeniería propiedad de organizaciones y usuarios.

Capítulo 6

Conclusiones

6.1. Consecución de objetivos

El objetivo principal de este trabajo era entender los elementos que constituyen un proyecto de software de ingeniería para poder identificar en repositorios de GitHub dichos proyectos de los que no son de ingeniería. Se han propuesto siete elementos, llamados dimensiones: comunidad, integración continua, documentación, historia, gestión de erratas, licencias y pruebas unitarias, para poder identificar dichos proyectos y se han realizado dos conjuntos de repositorios, cada uno de los cuales correspondía a una definición diferente de un proyecto de software de ingeniería que estaba formado y entrenado por el clasificador basado en la puntuación y por el clasificador Random Forest.

Aquí tengo que puntualizar que se han logrado tanto el objetivo principal, como los específicos planteados. Los clasificadores se utilizaron para identificar todos los repositorios de la muestra de 2.316.524 repositorios de GitHub que eran similares a los que se ajustan a las definiciones del proyecto de software de ingeniería. Nuestro modelo Random Forest ha dado el mejor resultado: predijo que el 23,51 % de 2.316.524 repositorios GitHub contienen proyectos de ingeniería de software.

Aunque el final ha sido bueno, tengo que decir que el proceso ha sido complicado, ya que, dado el gran volumen de repositorios tratados, conseguir abrir el fichero de datos ha sido difícil, así como relacionar los distintos archivos disponibles entre sí, ya que venían todos sin la cabecera y ésta ha sido necesario construirla en base a un archivo pdf explicativo en forma de diagrama disponible para los usuarios.

6.2. Aplicación de lo aprendido

Para la realización de este trabajo me han sido de gran ayuda varias asignaturas del Grado en Ingeniería en Tecnologías de las Telecomunicaciones tales como Estadística, ya que la base de los métodos estadísticos utilizados la he aprendido en dicha asignatura. Por otro lado, el análisis de los repositorios se ha llevado a cabo en Python, con lo que la asignatura de Servicios y Aplicaciones Telemáticas cursada en Grado me ha resultado imprescindible para el desarrollo de todo el trabajo. Además, la asignatura Ingeniería de Sistemas de Información me ha resultado muy útil en el manejo de la base de datos.

6.3. Lecciones aprendidas

En este trabajo he aprendido que Excel no permite abrir documentos con un número de filas superior a 1.048.576, lo cual me ha complicado en exceso el tratamiento de los datos. Además, para el análisis de estos he tenido que aprender a manejar la distribución Anaconda Python y la aplicación Jupyter Notebook, que es una aplicación web que sirve a modo de puente constante entre el código y los textos explicativos.

6.4. Futuros trabajos

Como ya adelanté al final del capítulo anterior, tanto las dimensiones utilizadas para representar repositorios de código fuente en el modelo de clasificación, como los umbrales y pesos utilizados en el clasificador basado en la puntuación son subjetivos. Aunque el resultado es razonablemente bueno, en futuros trabajos se podrían utilizar esquemas de ponderación alternativos para mitigar parte de esa subjetividad. Además, sería bueno probar otros enfoques alternativos como el uso de algoritmos de aprendizaje automático para evaluar la importancia de las dimensiones utilizando repositorios en un conjunto de datos de entrenamiento, una ponderación uniforme entre dimensiones, o un esquema de ponderación basado en la popularidad.

Apéndice A

ENGINEERED

A continuación se relacionan los repositorios que son de ingeniería y que se han analizado en este trabajo.

<https://github.com/AgilTec/cadenero>

<https://github.com/brenoc/opentracks>

<https://github.com/onaio/onadata>

<https://github.com/alu0100536829/prct11>

<https://github.com/clementine-player/Android-Remote>

<https://github.com/rafallo/p2c>

<https://github.com/gfx/Android-HankeiN>

<https://github.com/Ydle/RoomBundle>

<https://github.com/zeronullity/SDRwatchdog>

<https://github.com/linchproject/linch-servlet>

<https://github.com/tuanhiep/mqtt-jmeter>

https://github.com/zeisler/active_mockers

<https://github.com/NSLS-II/pyRafters>

<https://github.com/liquidise/Quickbase-Gem>

<https://github.com/franksl/ncsvlib>

<https://github.com/sangotaro/my-boxen>

<https://github.com/marsender/atoll-digital-library>

<https://github.com/mikesname/blueprints-sql-graph>

https://github.com/SomethingExplosive/android_frameworks_av

<https://github.com/wardrobecms/locales>

<https://github.com/lbitonti/liquibase-hana>

<https://github.com/Querela/ekIRC>

<https://github.com/tbruyelle/HappyContacts>

<https://github.com/videolan/x265>

<https://github.com/xenserver/xsconsole>

<https://github.com/higanworks-cookbooks/mruby>

<https://github.com/muchomasfacil/WysiwygBundle>

<https://github.com/jgauffin/SipSharp>

<https://github.com/jessy1092/jackpad>

<https://github.com/spring-projects/spring-net-codeconfig>

<https://github.com/Skobayashi/Weather>

<https://github.com/gpac/gpac>

<https://github.com/rhq-project/wildfly-cassandra>

<https://github.com/NathanSweet/dnsmadeeasy>

<https://github.com/Stibbons/pyyaml>

<https://github.com/emersion/bups>

<https://github.com/aelarabawy/glib>

<https://github.com/sphaero/uae4all-rpi>

<https://github.com/DragonSpawn/Json2Class>

<https://github.com/wangduoxiong/Egg>

<https://github.com/chef/knife-ec2>

<https://github.com/crbanman/AstroidEscape>

https://github.com/android-ia/platform_external_libsepol

<https://github.com/Serneum/jousting-core>

<https://github.com/davidkempers/django-tasks>

<https://github.com/Taapeli/ProtoLoader>

https://github.com/vzvu3k6k/mcg_source_list

<https://github.com/mthli/Tweetin>

<https://github.com/uProxy/obfuscation>

<https://github.com/hadleyrich/GerbLook>

<https://github.com/usgs/icoast>

<https://github.com/structured-commons/tools>

<https://github.com/ntuosproj/fastalg-nfqueue>

<https://github.com/OCA/banking-addons>

<https://github.com/ldrummy/good-talk>

<https://github.com/tanel/bugsnag-qt>

<https://github.com/gabepolk/double-dog>

<https://github.com/bryanjswift/simplenote-android>

<https://github.com/russellsimpkins-nyt/varnish-mmdb-vmod>

<https://github.com/BobKingstone/Pedlar-Cart>

<https://github.com/semantic-dependency-parsing/toolkit>

<https://github.com/numat/threeflex>

<https://github.com/slowmoVideo/slowmoVideo>

<https://github.com/TroyShaw/troykanoid>

<https://github.com/uakatt/kaikifs>

<https://github.com/Twisol/anachronism>

<https://github.com/jruby/jruby-ldap>

<https://github.com/collegedesis/bidwars>

https://github.com/II-OpenLink/ti-utils_soldel_maintenance

<https://github.com/jittat/caffe-grader-judge-scripts>

<https://github.com/zopefoundation/zope.app.publication>

<https://github.com/metabrainz/libdiscid>

<https://github.com/cemagg/sucem-fem>

<https://github.com/zfsrogue/spl-crypto>

<https://github.com/packfire/concrete>

<https://github.com/Elive/emodule-productivity>

<https://github.com/ckw-mod/ckw-mod>

<https://github.com/AgencyPMG/PMG-WP-Core>

<https://github.com/vivid-planet/kwf-newsletter-demo>

<https://github.com/liu-chong/micropolis>

<https://github.com/CollectorsQuest/magnify-sdk>

<https://github.com/NESCent/Taxonomy-Ontology-Tool>

<https://github.com/mteodori/jira-git-plugin>

<https://github.com/sanguinariojoe/ocland>

<https://github.com/grate-driver/grate>

https://github.com/singuerinc/puppet-font_explorer_pro

<https://github.com/kzoll/ztlogger>

<https://github.com/couchbase/couchbase-net-client>

<https://github.com/colszowka/phantomjs-gem>

<https://github.com/djblets/djblets>

<https://github.com/hhru/tornado>

<https://github.com/proycon/pynlpl>

<https://github.com/sonatype/plexus-compiler>

<https://github.com/tgjones/ormongo>

<https://github.com/herumi/cybozulib>

<https://github.com/mkraft/fides>

<https://github.com/cerb-plugins/wgm.facebook>

<https://github.com/alanbem/Josser>

<https://github.com/ramusus/kinopoiskpy>

<https://github.com/FrankHB/yplib>

<https://github.com/jagregory/fluent-nhibernate>

https://github.com/niw/iphone_opencv_test

<https://github.com/veg/hyphy>

https://github.com/kennyma/health_graph

<https://github.com/microcai/gentoo-zh>

<https://github.com/samirahmed/Iris-Voice-Automation>

<https://github.com/lex-lingo/lingo>

<https://github.com/pgoergler/Quartz>

<https://github.com/nicanorperera/xaver-template>

<https://github.com/jimlindstrom/xbrlware-ruby19>

<https://github.com/aamattos/GMF-Tooling-Visual-Editor>

https://github.com/mconf/bigbluebutton_rails

<https://github.com/ramen/phply>

<https://github.com/steveliles/dsl4xml>

<https://github.com/abulrim/siscode>

<https://github.com/otubo/qemu>

<https://github.com/martynsmith/lg4l>

<https://github.com/jburman/ZeroG>

<https://github.com/achiu/rack-recaptcha>

<https://github.com/CRAVA/crava>

<https://github.com/PowerKiKi/qTranslate>

<https://github.com/vext01/hgd>

<https://github.com/SRombauts/SQLiteCpp>

<https://github.com/opencog/opencog>

https://github.com/Ariloy/redmine_messenger

<https://github.com/Shuyang/uic-automatic-reviewer>

<https://github.com/darioquintana/NHibernate-Shards>

<https://github.com/lukecampbell/h5py>

<https://github.com/RedTurtle/cciaa.modulistica>

<https://github.com/mpaladin/java-dirq>

<https://github.com/mgkimsal/zfkit>

<https://github.com/ddsc/ddsc-worker>

<https://github.com/mensfeld/FB-Video-URL-Converter>

<https://github.com/dalmirdasilva/RaspberryIO>

<https://github.com/seblin/launchit>

<https://github.com/alx/reprap-arduino-firmware>

<https://github.com/phillord/mathjax-latex>

<https://github.com/smcameron/opencscad>

<https://github.com/zopefoundation/zope.app.interface>

<https://github.com/PuzzleOpenDataHackdayTeam/finsta-deck>

<https://github.com/verdigris/HappyNewYear>

https://github.com/cmonterrosa/search_routes

<https://github.com/gbagnoli/openphoto-utils>

<https://github.com/Youscribe/opencv-cookbook>

<https://github.com/englishtown/stash-hook-mirror>

<https://github.com/lpotter/libalignedtimer>

<https://github.com/sonatype/maven-guide-en>

<https://github.com/CorlenS/ROIMemberLoot>

<https://github.com/mirego/bourgeois>

<https://github.com/charliemorning/weibocrawler>

https://github.com/tomana/ofxONI1_5

<https://github.com/tjfontaine/node-addon-layer>

<https://github.com/rcbops-cookbooks/swift-private-cloud>

<https://github.com/jaapverloop/knot>

<https://github.com/goccy/p5-Compiler-CodeGenerator-LLVM>

https://github.com/BadrIT/seo_pages

<https://github.com/fish2000/h5dj>

<https://github.com/chbrown/pi>

<https://github.com/ScottMcMichael/lronacPipeline>

<https://github.com/dreikanter/pyke>

<https://github.com/iancook75/oscillocalc>

<https://github.com/novapost/django-ticketoffice>

<https://github.com/php-carteblanche/tool-form>

<https://github.com/erdavila/git-svn-diff>

<https://github.com/robertf224/pyTunes>

https://github.com/etherdev/markdown_datafier

<https://github.com/BradStevenson/Project-SciSearcher>

<https://github.com/generators-io-projects/generators>

<https://github.com/yhteentoimivuuspalvelut/ckanext-ytp-drupal>

<https://github.com/bingoohuang/buka>

<https://github.com/melizalab/arfx>

<https://github.com/barterli/barter.li>

<https://github.com/giucam/termistor>

<https://github.com/langner/cclib>

<https://github.com/taky/joanne>

<https://github.com/apanzerj/zit>

<https://github.com/kzeleny/script.tmdb>

<https://github.com/jlagerweij/swagger-springweb-maven-plugin>

<https://github.com/openSUSE/osc-plugin-factory>

<https://github.com/DevCabin/rootless>

<https://github.com/gagle/raspberrypi-openmax-jpeg>

<https://github.com/yoshizow/global-pygments-plugin>

<https://github.com/oguna/SharpXFileParser>

<https://github.com/altstone/doorscenter>

<https://github.com/liulhdarks/darks-codec>

<https://github.com/briggittewarner/trailblaze>

<https://github.com/schjan/RailNet>

<https://github.com/yahim91/FloatingContent>

<https://github.com/futureimperfect/autopkg-notify>

<https://github.com/hcs/hcs-cloud>

<https://github.com/nsimplex/ktools>

<https://github.com/takawitter/robodova>

<https://github.com/ambitioninc/django-entity-emailer>

<https://github.com/Khan/frankenserver>

<https://github.com/safaci2000/google-voice-java>

https://github.com/XPerience-NXT/android_build2

<https://github.com/jltjohanlindqvist/jltflash>

<https://github.com/M7S/dockbarx>

<https://github.com/WSULib/quicksearch>

<https://github.com/PigeonPack/pigeon-pack>

https://github.com/siwilkins/bible_passage

https://github.com/UltraSabreman/Free_Sharp_Player

<https://github.com/v3l0c1r4pt0r/HistoriaPojazdu>

<https://github.com/RobinRadic/laravel-bukkit-swiftapi>

<https://github.com/waltervargas/prestashop-perconnect>

<https://github.com/fperez/coulomb-3body>

<https://github.com/versionone/V1TortoiseSVN>

<https://github.com/koolkode/bpmn>

<https://github.com/snemetz/puppet-dripstat>

<https://github.com/nbari/py-sessions>

https://github.com/adejoux/file_sharer

<https://github.com/dsoprea/M2CryptoWin32>

<https://github.com/godiard/imageviewer-activity>

<https://github.com/non117/yuyushiki>

<https://github.com/bwildenhain/air-quality-sensor>

<https://github.com/akrasic/souschef>

https://github.com/blueplanet/task_chute

<https://github.com/MeldCE/wp-gallery-hierarchy>

<https://github.com/lucasr/dspec>

<https://github.com/inmoon/ProListView>

<https://github.com/Jugendhackt/hackspace-dashboard>

<https://github.com/AfterTheRainOfStars/QQStars>

https://github.com/os6sense/sunra_ffs_relay

<https://github.com/Kalbintion/Kdkbot>

<https://github.com/NativeScript/nativescript-cli-tests>

<https://github.com/nano-byte/LightTag>

<https://github.com/ZachOhara/Bukkit-Location-Manager>

<https://github.com/smarpension/signable>

<https://github.com/fenicks/rubysoul-ng>

<https://github.com/jjbunn/MultipathODL>

<https://github.com/humangeo/DateSense>

<https://github.com/globocom/content-gateway-ruby>

<https://github.com/PiDyGB/android-slidinglayout>

<https://github.com/relldoesphp/com.aghstrategies.giftmemberships>

<https://github.com/blockchain/api-v1-client-php>

<https://github.com/libdynd/libdynd>

<https://github.com/halmd-org/cuda-wrapper>

<https://github.com/lazymaniac/LexSem>

<https://github.com/open-epicycle/Epicycle.Math-cs>

<https://github.com/coreone/tex-boxen>

<https://github.com/ausaccessfed/aaf-lipstick>

<https://github.com/LSST/imgserv>

<https://github.com/eetac/android-logging-log4j>

<https://github.com/openregister/entry>

https://github.com/android-ia/platform_external_chromium_org_third_party_libj

<https://github.com/cloudcopy/seafiler>

<https://github.com/RetroShare/ChatServer-Frontend>

<https://github.com/calico-g/coraline>

<https://github.com/kapilt/contentmirror>

<https://github.com/Philippe2201/alura-provas2>

https://github.com/manmeetmlt/my_first_rails_app

<https://github.com/Chabadsuite/com.chabadsuite.batchactivityadd>

<https://github.com/fritzmonkey/esquery>

<https://github.com/diegocstn/NumberPickerView>

<https://github.com/frozenith/CredKing>

<https://github.com/hikmahealth/bahmni-core>

<https://github.com/frozenith/CredSniper>

<https://github.com/frozenith/PowerWebShot>

<https://github.com/rorychristianmurray/animal-kingdom>

<https://github.com/frozenith/DPAT>

<https://github.com/Blaine876/chatroomAPI>

<https://github.com/bassuny3003/ShareX>

https://github.com/Vladimir547/Glasses_By_ILYUSHIN_VLADIMIR

<https://github.com/rsmsnot/poppin-bottles>

<https://github.com/DraconianLore/poppin-bottles>

<https://github.com/yynickel/poppin-bottles>

<https://github.com/zhuzhenping/faceswap>

<https://github.com/Djuwita/calculating-lines-data-science-intro->

<https://github.com/ToyaMitchell/101-personal-site>

<https://github.com/glenpgd/glenpgd-github.com>

<https://github.com/anandangalig/react-context-system>

<https://github.com/gitter-badger/DabEngine>

<https://github.com/lauralaura14/css-kitten-wheelbarrow-lab-v->

<https://github.com/ghlxm/HotSpot-JVM-Linux-x86-Research>

<https://github.com/ElementAI/NAF>

<https://github.com/davidmnagy27/Ajax-Dictionaries>

<https://github.com/miklo88/User-Interface>

<https://github.com/joshlong90/System-Calls>

<https://github.com/nicholasstano/operators-nyc-web->

<https://github.com/diomedes314/PATA>

https://github.com/stivenson/run_script

<https://github.com/kjannenga/flex-fishy>

<https://github.com/kristoferjoseph/restafarian>

<https://github.com/masayoshi-toku/papper-tweet>

<https://github.com/seckisecki/SupervisedLearning-FindingDonorsForCharityML>

<https://github.com/jkanchelov/PIXI.TextInput>

<https://github.com/cleborys/materials>

<https://github.com/Yimiao123/Coursera-ML-AndrewNg-Notes>

<https://github.com/christianjgreen/bno055>

<https://github.com/bimlas/bash-mosh>

<https://github.com/tomoakimiura/php-qa-plaza>

<https://github.com/reyesc27/Proyect1>

<https://github.com/twichtendahl/TempConverter>

<https://github.com/mcc85s/ADRecon>

<https://github.com/ianstafford/qtpylib>

https://github.com/OmarElsebaey/Project_1_boston_housing

<https://github.com/tommasoauerbach/looping-times-nyc-web-060319>

<https://github.com/ohosseinmardi/austin-nyc>

https://github.com/narendraanupoju/MRI_data_preparation

<https://github.com/Lutterus/t2SisOp>

<https://github.com/orar/play-silhouette>

<https://github.com/mvegaxx/monat>

https://github.com/sergioquadros/Machine_Learning_Tutorials

<https://github.com/pragmatux/mkos-mlo-common>

<https://github.com/AustinMorrill/Open-Trivia-Quiz>

https://github.com/SarahQiong/WBC_Matlab

<https://github.com/adamstirtan/trailheadx19>

<https://github.com/sukimsiriam/fbtools>

<https://github.com/Fraks51/CppHW>

<https://github.com/abramp20/CashMachineWeekend>

<https://github.com/vagnerfonseca182/LIKEFOOD>

<https://github.com/GgeekFreak/Depression-diagnostic-system-desktop-based>

<https://github.com/davidson-lee/personal-site>

<https://github.com/seshajay/FunWithML>

<https://github.com/JuniorDugue/The-Book-APP>

<https://github.com/anticomputer/semantic>

<https://github.com/Fe-Ordan/ofxIisu>

<https://github.com/garikoitz/nipype>

<https://github.com/crowdbotics-apps/g-way->

<https://github.com/IgnoredRhyme520/Minecraft-Server-Jars>

<https://github.com/sibonnet15/excel-to-python-data-science-intro->

<https://github.com/ramsestrejo/ShSDD-ICE2>

<https://github.com/sohelsheikh91/Sample-Project>

<https://github.com/WalczRobert/Lazy-Raider>

<https://github.com/svngoku/javascript>

<https://github.com/Raaj108/rnews-webapp>

<https://github.com/VadimS4/fewpjs-js-fundamentals-variables-lab-seattle-web-care>

<https://github.com/wylcetal/ejercicios>

<https://github.com/ron-ny2/rclone>

<https://github.com/thshorrock/ionic>

<https://github.com/seckisecki/UnsupervisedLearning-CreateCustomerSegments>

<https://github.com/Martin444/React-hospedajes>

https://github.com/HapCuji/Build_Compilers

<https://github.com/MehmetRamiz/nodejs-3-Weather-App>

https://github.com/cha10vd/coroutines_concurrency

<https://github.com/ryantillis/cs-checkstyle>

<https://github.com/shabigit/leetcode>

https://github.com/AA-CubeSat-Team/soci_cdh_rtos

<https://github.com/reedlex98/Clockesthic>

<https://github.com/molecular-cell-biomechanics-lab/DiTaxa>

<https://github.com/shabigit/LeetCodeAnimation>

<https://github.com/jkoenig134/wysiwyg-e-java>

<https://github.com/rshirani/bazel>

<https://github.com/tiffanyrivas/GifTastic>

<https://github.com/Callum-Mitchell/Chromavolt>

<https://github.com/rsbondi/lnet>

<https://github.com/wadegilmer/techjobs-console-java>

<https://github.com/keunyop/IntuneDocs.ko-kr>

<https://github.com/Bleizingard/material-components-android>

<https://github.com/hellsee/adding-up>

<https://github.com/mlxnle/Mycat-Server>

<https://github.com/vianafn/Pop-Corn-o-Grupo-7-WEB>

<https://github.com/astewart27/react-github-user-profile>

<https://github.com/emptyopen/reactotron>

https://github.com/Eadancel/test_move

<https://github.com/jibrel/openmrs-config-jib>

<https://github.com/molecular-cell-biomechanics-lab/MicroPheno>

<https://github.com/kgullion/pugBot>

<https://github.com/akuholla/camera>

https://github.com/yijun-zhang/stream_file_upload

[https://github.com/Djuwita/single-variable-regression-lab-data-science-intro-](https://github.com/Djuwita/single-variable-regression-lab-data-science-intro)

https://github.com/alttabs/Boston_House_Price_Prediction

https://github.com/leomhcorrea/ep2_aed_1

<https://github.com/viniciusleati/studyingR>

<https://github.com/zhengpingwan/awesome-python>

<https://github.com/Beaulieu527/js-basics-control-flow-lab-online-web-pt->

<https://github.com/ditointernet/dito-spectacle-theme>

<https://github.com/wxlost/KeepMyGoogleVoice>

<https://github.com/dave-89/angulat-strict-css>

<https://github.com/molecular-cell-biomechanics-lab/dimotif>

<https://github.com/leechongchong/Credit-Card-Fraud-Detection-System>

<https://github.com/zhengpingwan/system-design-primer>

<https://github.com/KevinKelley/zdog>

https://github.com/reactome-fi/reactome_cancer

<https://github.com/longniansheng/gallery>

<https://github.com/jacklee20151/nlp-tutorial>

<https://github.com/kjersey690/news>

<https://github.com/kernel-dev/inteloops>

<https://github.com/daeyounglim/rainbow-bash-prompt>

[https://github.com/Djuwita/applying-gradient-descent-lab-data-science-intro-](https://github.com/Djuwita/applying-gradient-descent-lab-data-science-intro)

<https://github.com/zhengpingwan/public-apis>

<https://github.com/Clew27/VG-Bundle-Algorithm-Prototyping>

https://github.com/Decman84/mkvserver_mk2

<https://github.com/axeltux/api-laravel>

<https://github.com/ianstafford/pandas>

<https://github.com/frozenith/learn-python3>

<https://github.com/mauriciogontijo/gentelella>

<https://github.com/zhengpingwan/Python>

<https://github.com/frozenith/learn-python>

<https://github.com/dalems4/dsc-instance-variables-lab-seattle-ds-career->

<https://github.com/znzn2013/task1>

<https://github.com/ysnacrk/quicktranslater>

<https://github.com/zhengpingwan/thefuck>

<https://github.com/mike-roberts/Blueshift-iOS-SDK>

<https://github.com/zinebkhanjari/ProjetJavaUml>

<https://github.com/6stringninja/inav>

<https://github.com/aenygma/hashdex>

<https://github.com/hanbanana/Color-Picker>

<https://github.com/Hikari0418/leetcode>

<https://github.com/Jason-Cooke/plyr>

<https://github.com/amajidsinar/ETL>

<https://github.com/dtychero/core>

<https://github.com/PythonDarkeningProjects/proyect>

<https://github.com/losdor/fb-brute>

<https://github.com/szieglerICF/aws-account-summary-windows>

<https://github.com/clearxiaofeifei/cube-ui>

<https://github.com/answjddnr/windows-cis-profile>

<https://github.com/danielmarcgardner/portfolio-site>

<https://github.com/tdbuchanan/sfdx-project>

<https://github.com/sergaka/Personal-Quiz-Example->

<https://github.com/zhengpingwan/python-patterns>

<https://github.com/phmiranda/senai-banco>

<https://github.com/ZoraSantos/exercicio-blogapp>

<https://github.com/bkakilli/RANSAC-circle-python>

https://github.com/Maksim1990/Laravel_Dockerize_OLD_Meet_Mate_APP

<https://github.com/aschams/dsc-introduction-section-intro-dc-ds->

<https://github.com/zhengpingwan/pandas>

<https://github.com/cassiuscampos/AWSHelpers>

<https://github.com/cfuxiang/nl2sql>

<https://github.com/TheLonestar1/kj>

<https://github.com/Tirth1200/jQueryCodeSnippets>

<https://github.com/richardleach/whocalls>

<https://github.com/richardli515/phylogenes-api>

<https://github.com/dtychero/coreclr>

<https://github.com/CerfVert94/Kernel-Module>

<https://github.com/manojrajuladevi9/EF6-DBFirst-Demo>

<https://github.com/dtychero/corefx>

<https://github.com/xMizu/my-select-nyc-web-062419>

<https://github.com/joelcoxokc/underbar-training>

<https://github.com/ghost11886/FAEDER>

<https://github.com/Coolerian/gamelifa>

<https://github.com/AlvisS66/VincentTV-Code>

<https://github.com/vescamilla/devops-essentials-sample-app>

<https://github.com/NinjaNymo/AltiumLib>

<https://github.com/ssss-38438-org/click-to-deploy>

<https://github.com/yj679/Dementia>

https://github.com/Gilbert-Adu/my_currency_converter

<https://github.com/hickeye/Talks-and-Presentations>

<https://github.com/francois-drielsma/InstrumentKit>

<https://github.com/JMVCoelho/m19>

<https://github.com/Il1SmithDa/salesforceNotes>

<https://github.com/lawsonhung/parrot-ruby-dumbo-web->

<https://github.com/MahdiRahbar/ChatterBot>

<https://github.com/orar/silhouette>

<https://github.com/Hyperfine/pyTHM1176>

https://github.com/farfalle211/weather_transportation_app

<https://github.com/Byteflux/askyblock>

<https://github.com/sjinks/winston-mail-lite>

<https://github.com/howardjohn/file-based-istio>

<https://github.com/MarkPartlett/dashboardcharts>

<https://github.com/yuyime/how-does-navicat-encrypt-password>

<https://github.com/joelcoxokc/toyproblems>

<https://github.com/subhashdasyam/spring-boot-rest-example>

<https://github.com/blackfist/CacheOnlyKeyWrapper>

<https://github.com/jwickens/postgraphile>

<https://github.com/cesarhdz/estatico-php>

https://github.com/maurotoro/UMI2018_ToroBergomi

<https://github.com/aschams/dsc-problems-data-science-can-solve-dc-ds->

<https://github.com/kargo-k/nyc-pigeon-organizer-seattle-web-060319>

<https://github.com/EduBic/Cpp-A-Refreshing>

<https://github.com/adsj1982/jsignpdf>

<https://github.com/hajsong/personal>

https://github.com/teespring/omniauth_openid_connect

<https://github.com/bmacauley-reward/example-aws>

https://github.com/ashish-ucsb/ece_283_machine_learning

<https://github.com/estorey11/simple-partial-lab-online-web-ft->

<https://github.com/Benk0033/Constructor-Word-Guess>

<https://github.com/Onboard-Informatics/postman-collections>

<https://github.com/marcospereira/lagom-samples>

<https://github.com/koudyk/ohbm>

<https://github.com/GgeekFreak/Chronic-disease-detection-system-desktop-based>

<https://github.com/gerardoMed/practica2>

<https://github.com/phmiranda/senai-clinica>

<https://github.com/nkatwesigye/OAuth2.0>

<https://github.com/othaderek/rack-dynamic-web-apps-dumbo-web->

<https://github.com/cesarhdz/SeamLESS-PHP>

<https://github.com/jslight90/nixpkgs-channels>

<https://github.com/dguilmezian/paw-integrador>

<https://github.com/Roelifela/ListaContato>

<https://github.com/ffiaux/AlunosApp>

<https://github.com/Rogerch99/Projects>

<https://github.com/juliocezarfs/Project-site-UEG-work>

<https://github.com/ThomasNigro/owen-home>

<https://github.com/din-lab-train-data-v3/>

<https://github.com/jalywang123/flask-video-streaming>

https://github.com/andreas-h/line_profiler-feedstock

<https://github.com/T-Gio/sfdx-project>

<https://github.com/hexaforce/webpack-vue>

https://github.com/jhuxiang/LeetCode_Python

<https://github.com/Vinicius01i/Quiz>

<https://github.com/sixtysix-Team/fbbrute>

<https://github.com/Depau/swin>

<https://github.com/smankoo/backup-gdrive>

<https://github.com/srini100/grpc.io>

<https://github.com/TallanGroberg/flash-cards>

<https://github.com/sangel217/planetary-age>

<https://github.com/ceanver/sourceCode>

<https://github.com/escuelainformatica/mayo31>

<https://github.com/colesam/typescript-ci>

<https://github.com/reyesc27/Proyect2>

<https://github.com/mengqiy/play-kubebuilder>

<https://github.com/Agonec-bozhij/angular-bazel-example>

<https://github.com/T1V/SystemKit>

<https://github.com/Engitano/fs2-gcp>

<https://github.com/Nain22/Courier>

<https://github.com/jzheaux/spring-security-oauth-to-5-translation-dictionary>

<https://github.com/bsiegel/azure-sdk-for-net>

<https://github.com/Nexusflamehart/orcpub>

<https://github.com/kolson256/sfdx-project>

<https://github.com/alex-polosky/cs-dotnet-ef-npgsql-multi-db-error>

<https://github.com/cesarhdz/flag-icon-css>

<https://github.com/BevanR/php-coding-task>

<https://github.com/raypage/sfdx-project>

<https://github.com/judith-dev/react-child-proptypes>

A continuación se relacionan los repositorios que no son de ingeniería y que se

<https://github.com/shirleyshz/shirleyshz.github.io>

<https://github.com/thelukestor/learn-co-sandbox>

https://github.com/ErnisEr/Test_

<https://github.com/DeadSending/hello-world>

<https://github.com/malteado/malty>

<https://github.com/dalilahannouche/tournerpage>

<https://github.com/lucaspada894/chat-app>

<https://github.com/subtlymoney/0225Git>

https://github.com/Traci7822/RENAME_ME

<https://github.com/zainabroo/airbnbchallenge>

<https://github.com/marcelogaray/auto-GA-v06>

<https://github.com/EnricoJohn/CodeNation>

<https://github.com/juanlozano24/concesionario>

<https://github.com/ZeyadOsama/SoleekLabTask>

<https://github.com/DerspaB/Programa-Parcial>

https://github.com/OVGULIU/tensor_derivative

https://github.com/k-bosko/predicting_catalog_demand

<https://github.com/natomasvillageapartments/natomasvillageapartments.github.io>

<https://github.com/wagner-luis97/AM>

<https://github.com/larowlan/larowlan.github.io>

<https://github.com/kre64/2D>

<https://github.com/git2e/5378522d-2785-446a-b468-c200898f7835>

<https://github.com/JefferyQ/el-repository-MFr>

<https://github.com/joelyustiz/SEP-Components>

<https://github.com/luangm123/Tech-Jalsa>

<https://github.com/lipeA/clinica>

https://github.com/ujwalarora34/c0754449_Midterm_s2019mad3463

https://github.com/alex93skr/Euler_Project

<https://github.com/LeonardoVivasAndrade/BolivarApp>

<https://github.com/slalomchris/sfdx-project>

<https://github.com/FFizzZZ/deeplearning.ai>

<https://github.com/jacobmoore324/eclipse-repo>

<https://github.com/Billdozer1547/sfdx-project>

<https://github.com/lslutskaya/Insider2>

<https://github.com/lautaumpierrez/CookieManager>

<https://github.com/YevKozarchuk/test.github.io>

<https://github.com/boniface100/Realm-App-assignment-3>

<https://github.com/soeunlang/Spoon-Knife>

<https://github.com/mvid/simpleforce>

https://github.com/HanVTran/Hello_World

<https://github.com/cearo/C195>

<https://github.com/railcheg/hello-world>

<https://github.com/noltron000/three-collector>

https://github.com/gholzrib/treinamento-ios_jogo_da_memoria

<https://github.com/ahmedengu/private-ai>

<https://github.com/HanVTran/Data-Pipelines-with-Airflow>

https://github.com/vipulgarg192/c0753362_MidTerm_s2019MAD3463

<https://github.com/IsaacMolinari/teste>

https://github.com/Freelive2425/Clase_Funda_S10

<https://github.com/maksymbogdanov/SLAExtension>

<https://github.com/charneff/oo-tic-tac-toe-online-web-ft->

<https://github.com/cc5212/2019-frequent-costars-imdb>

<https://github.com/AnuragDharNEU/csye6225-summer2019-template>

<https://github.com/claridiva2000/webdb-challenge>

<https://github.com/smolaka/sfdx-project>

<https://github.com/pkj-m/VertexSafeGraphs.jl>

<https://github.com/alienturtle87/Devploit>

<https://github.com/molecular-cell-biomechanics-lab/Deep-Proteomics>

<https://github.com/charles201311/cms>

<https://github.com/adioe3/noapp-please>

<https://github.com/markbucko12341/massiveCARpackHQ>

<https://github.com/ezequieloscarescobar/hamburguers>

<https://github.com/NafeeJ/TextBlockMaker>

<https://github.com/cesar-yoab/portfolio>

<https://github.com/gmarshall5/Learning>

https://github.com/JonathanC3/minisuper_nolep

https://github.com/yosrathala/ProgoLineScrap_Crawl

<https://github.com/mitsuaki1/test190601>

<https://github.com/WilliamChou06/awesome-code>

https://github.com/Sebaolivares11/utem_thesis_template

<https://github.com/littlelight2019/java1>

<https://github.com/jaroslawkid/dick>

https://github.com/fr3dn3t/card10_sequins

<https://github.com/ali-mozafari/FR-SRGAN>

https://github.com/emilylauyw/Detect_Dangerous_Driver

<https://github.com/hybalov-viacheslav/callfrodrops>

<https://github.com/jamgochiana/RLPlayground>

<https://github.com/liyanage/kicad-library-liyanage>

https://github.com/kiuwong/ARStudio_ChickenHat

<https://github.com/hmalik88/cryptoexchange>

<https://github.com/stephane19/Boulder-Dash-Group1>

<https://github.com/CallMeTheMasterHacker/Ring-of-Elysium>

<https://github.com/brpiank/hello-world>

<https://github.com/pythonpete32/AGPs>

<https://github.com/autotester-one/integration-tests->

<https://github.com/gorchels/esm262baseball>

https://github.com/JohanPortilla/OP_CorrerBloque

https://github.com/lslutskaya/Technologies_lab

<https://github.com/aulaalmir/TrabalhoAlmirAd>

<https://github.com/ObertoIsOBS/discordjs>

<https://github.com/flamelitface/ZolaHomeworks>

<https://github.com/KuzeIII/OSCP-Archives>

<https://github.com/tuflix/practical1>

https://github.com/agirrearri/3eval_ende

<https://github.com/otarampinelli/card>

<https://github.com/benalavi/pop-dark-syntax>

<https://github.com/lialsoftlab/minreq>

<https://github.com/tyzbit/awake-docker>

<https://github.com/tim-smart/awesome-contentful>

https://github.com/HakonaLabs/hello_world

<https://github.com/JoseDFS/TallerDos>

<https://github.com/amritsinghrbc/Automation-Assignment>

https://github.com/JOSEALVESPEREIRA/pet_shop

<https://github.com/zvorgiygeonka/hello-world>

<https://github.com/JohnCC330/webpy>

<https://github.com/sml920505/rongge>

<https://github.com/VolleyTKE/cuckoo>

<https://github.com/francesquini/test>

https://github.com/ucd-rundergrad-club/S_Cheng_Rundergrad

<https://github.com/wbroome14/eventsPLUS-privacy-policy>

<https://github.com/khounvandy68/supreme-dollop>

<https://github.com/gunjanpatil23/ParticleFilterSLAM>

<https://github.com/angelaaaateng/uchicago-msca-club.github.io>

<https://github.com/Mady974/Mazed>

<https://github.com/Vencesland/my-first-blog>

<https://github.com/dumbbond/leanback-assistant>

<https://github.com/glu251982/FlyTour>

<https://github.com/osirrc/anserini-docker>

<https://github.com/noltron000/three-snowball>

<https://github.com/caiogirao/hello-world>

<https://github.com/WittyRejoinder/sfdx-project>

<https://github.com/BlakeGardner/cockpit-docs>

<https://github.com/vguillet/aqmen>

<https://github.com/bot-luc/LuC>

<https://github.com/tarabhawnani/sfdx-project>

<https://github.com/WillPoulson/node-database-example>

<https://github.com/YurBurGer/2nd-keyboard>

<https://github.com/pointgenesis/pointgenesis.io>

<https://github.com/lowliver/LPoint>

<https://github.com/vervallsweg/cast-web-api-android>

<https://github.com/alejuaquez/cs-technical-curriculum-public>

https://github.com/yixinmao/example_python_unittest

<https://github.com/Neumann789/JavaGuide>

<https://github.com/1slutskaya/Technologies>

<https://github.com/act1991/Python-100-Days>

https://github.com/ShanKwanCho/Song_Lyric_Finder

<https://github.com/JayTrask/JayTrask.github.io>

<https://github.com/MahdiRahbar/awesome-cpp>

<https://github.com/stenpiren/plxt-guides>

<https://github.com/Ejra92/memes>

<https://github.com/rwchisholm/DX>

<https://github.com/amojics/contactMeWeb>

<https://github.com/josivantarcio/backup-github>

https://github.com/egecan12/JS_Guess_Who

<https://github.com/liqingshuchina/mtjiu>

<https://github.com/distortedsignal/awesome-erlang>

<https://github.com/zampolitx/test>

<https://github.com/taylormcknight/weybolt>

<https://github.com/wildewalkingtours/site>

<https://github.com/rphilson/sfdx-project>

<https://github.com/calimero2018/code1>

<https://github.com/GoryMoon/Bingo>

<https://github.com/xXHack/Dark>

<https://github.com/victorhamon/sige-production>

<https://github.com/steve86/TopologicalDefect>

<https://github.com/MichlosBSB/infra-test>

<https://github.com/DeeGee1015/library.gym>

<https://github.com/daticomunebologna/opendata>

<https://github.com/unmisha/mycode>

<https://github.com/hflicka/django-autoload>

<https://github.com/mitchellzen/intrstmap>

<https://github.com/rstoyanchev/dispatch-test>

<https://github.com/jh9027/konqr>

<https://github.com/iKolt/sample>

<https://github.com/hasenj/arabic-writer>

<https://github.com/dhellmann/metering-prototype>

<https://github.com/nskrypnik/Pyramid-file-storage-app>

<https://github.com/JorisSpekreijse/boekhoud>

https://github.com/henning1004/AddSomeFurniture_ReWrite

<https://github.com/crodas/Haanga2>

https://github.com/mikehamer/ardrone_tutorials

<https://github.com/kvm/BlueWhale>

https://github.com/darrensemail/hw2_rottenpotatoes

<https://github.com/jefigo/Bootcamp>

https://github.com/makavellious/sample_app

<https://github.com/ekingery/Bar-Napkin>

<https://github.com/gra/nikitina>

<https://github.com/Jim-Holmstroem/SQLightning>

<https://github.com/WinSir/WinSir.Tools.Markdown.PhotosPost>

<https://github.com/danlex/bstexample>

<https://github.com/altheredbull/FerrisWheel>

<https://github.com/benshope/w5d4>

<https://github.com/pngisnotgif/python>

<https://github.com/Carlosmr/NaturalLanguageProcessing>

<https://github.com/rewbs/ljcu.findbugs.ext>

<https://github.com/pdonlon/Sudoku-Solver-2>

<https://github.com/jerbio/My24HourTimerWPF>

<https://github.com/richistron/slim-services>

<https://github.com/andreeasandu/wim>

<https://github.com/Mehdi23/TpJava>

<https://github.com/nate-d-olson/MBWG>

https://github.com/novoalg/demo_app

<https://github.com/loboweissmann/groovy-grails-na-pratica>

<https://github.com/jataggart/seven-languages>

<https://github.com/aangarita/invent>

https://github.com/garrettlord/MHacks_Pill_Delivery_Quad

<https://github.com/liuhaodong/hw0-haodong1>

<https://github.com/lukefx/AccessManager2>

<https://github.com/ruedigergad/qca-qt5>

<https://github.com/pulkitsinghal/TestJavaClientsForCouchDB>

<https://github.com/triplethreatg/elec424-lab2>

<https://github.com/Yogendra0Sharma/Python-Design-Patterns>

<https://github.com/iso32/EncryptedMessage>

<https://github.com/rohan1020/songsite>

<https://github.com/amarnatht/p2.amar15.biz>

<https://github.com/rwieckowski/demo-dynamic-proxy>

<https://github.com/philfromfmf/kovalevskiy>

https://github.com/bilastom/demo_app

<https://github.com/numerodix/pybits>

<https://github.com/jeconje/sms>

<https://github.com/kevin-ww/text.classification>

<https://github.com/castoridae/naval-battle-sim>

<https://github.com/fazie/algorithms>

<https://github.com/jw84/eddie>

<https://github.com/reigner-yrastorza/htdigest-php-web-frontend>

<https://github.com/hinedavid/dasesedus>

<https://github.com/UTAustin/php-site-spring2013>

<https://github.com/knighthunter09/XPEDIA>

<https://github.com/stanlemon/lemon-app>

<https://github.com/neutrino-git/TelerikAcademy>

<https://github.com/drackows/poc-rmi>

<https://github.com/mohitsehgal/MapsApp2>

<https://github.com/torypayne/exercise4>

<https://github.com/MZ992/Movie-rental-Store>

<https://github.com/lmtreser/Python>

<https://github.com/timbooker/Simple.Data.Linqpad.Tracelistener>

<https://github.com/r-julien/txl-project>

<https://github.com/praetorius/demo1>

<https://github.com/manuelblanch/ebreescool>

<https://github.com/Jorgtre/Cpp>

<https://github.com/mmcguinn/PoolVisor>

https://github.com/nogrudge/first_app

<https://github.com/AlexandreAbraham/movements>

<https://github.com/thiendv/GSO>

<https://github.com/PaulStoffregen/k72memtest>

<https://github.com/zwgirl/summer>

<https://github.com/crisguitar/ProjectEulerProblems>

<https://github.com/rikikun/SpringExample>

<https://github.com/xieyilong/test>

https://github.com/valerie7869/sample_app4

<https://github.com/gastitan/Criaturas>

<https://github.com/robertbachmann/openbsd-libcrypto>

<https://github.com/nicolasfaure/evaluacion>

<https://github.com/Youngs285/Chapter5Problems>

<https://github.com/msoltysik/Flask-Mega-Tutorial>

<https://github.com/johnnyshih99/pulseMVCsample>

https://github.com/smellytofu/customer_relationship_applications

<https://github.com/jpleau/hockeystreams-ruby>

<https://github.com/t-larsen/test>

<https://github.com/anthonylife/EventRecommendation>

<https://github.com/zfh1005/demoPackage>

<https://github.com/Leozzer/Python-for-DB>

<https://github.com/yangjisheng/footballlotterycrawler>

<https://github.com/hectormoreno87/FinditOut>

<https://github.com/davidcrotty/DependancyInjection>

<https://github.com/jack2606/spotify-voter>

<https://github.com/Robbi-Blehdose/StarTrekMod>

<https://github.com/lizuqiliang/AssassinPlatformerGame>

<https://github.com/ycj0808/Travel>

<https://github.com/lalorincon/patron-mvc>

https://github.com/mizoguchi-y/first_app

<https://github.com/MaikGrunwl/Maik>

https://github.com/RUAN0007/Crisis_Management_System

<https://github.com/dmzadorin/prime>

<https://github.com/DarkmatterVale/Pocket-Spacecraft>

<https://github.com/mpaskew/isc-dev>

<https://github.com/demon7452/C>

<https://github.com/biwin/eleven>

<https://github.com/fijiaaron/qa.perfected>

<https://github.com/Impulser/OpenRSS>

<https://github.com/lejund/etsydemo>

<https://github.com/Avoduhcado/Press-X-to-Win>

https://github.com/gskielian/Android_01_Java

<https://github.com/notcompletelylow/Lab08forBoshart>

<https://github.com/Dorga/mymuseum>

https://github.com/icotanchev/scripting_lang_first_project

<https://github.com/Gribblesnibit8/NVSE-Docs-Manager>

<https://github.com/korugant/nbad-xtra-assignment>

<https://github.com/TrinhThiThuyDung/shopping>

<https://github.com/oorion/chess>

<https://github.com/moroojaldeeb/xceed>

<https://github.com/fernandomsilva/Final-Project--RJ-bus->

<https://github.com/JcFowles/MathCalculators>

<https://github.com/jos3pir3s/fofinho-de-bico-js>

https://github.com/Vallium/get_next_line42

<https://github.com/kandx/OA-PHP>

<https://github.com/vin047/helloworld-java>

<https://github.com/EminemJeff/dimsumapp>

<https://github.com/stevewitman/rails-practice-nested-resources>

<https://github.com/yusuke1116/VastariCsharpcode>

<https://github.com/mkrzman/KaraokeFinder>

<https://github.com/akhilabs/Distributed-Computing>

https://github.com/ItWorksOnMyMachine/HuntTheWumpus_SHS

<https://github.com/mpratt/Weg>

<https://github.com/XenoBlaze/hello-world>

<https://github.com/tatelucas/mayan>

<https://github.com/matthewhoendorf/hoendomd-lab06-IntList-master>

https://github.com/svoges/toy_app

<https://github.com/mohamedahmedabdefattah/System-Outage>

<https://github.com/theoriginallazybum/Algebra-Solver2>

<https://github.com/esevi/exam01>

<https://github.com/insane-stdios/WebCalculator>

https://github.com/nawalathar/Lab4_NawalAthar

<https://github.com/justanotheruser/Diplom>

<https://github.com/asiunov/utils>

<https://github.com/Marghytis/SarahsWorld>

<https://github.com/bleakwood/ducking-tyrion>

<https://github.com/tiptobi/HLM-App>

<https://github.com/cursorcl/trustee.android.app>

<https://github.com/dschobel/movies>

<https://github.com/vivekrk/RobolectricSample>

https://github.com/jamessha78/BloombergHack_App

<https://github.com/SomeSmallProjects/MyCode>

<https://github.com/hepl-3infogra-quelu/ToDo>

<https://github.com/khuongnt/luongcms>

<https://github.com/kapaseker/mysitetutorial>

<https://github.com/classic-city-rails/isislvx-rthw>

<https://github.com/KostadinStefanov/Team-PAPAYA->

<https://github.com/threeStone313/productCatalogManagementSystem>

<https://github.com/no-wa-ke/BSPareidolia>

<https://github.com/zakhar-herych/refresher-cpp>

<https://github.com/mikaeldui/PyAvanza>

<https://github.com/mk44/javapublic>

https://github.com/gulatiak/sample_app

<https://github.com/robert-school/ludum-dare22>

<https://github.com/FedorSelitsky/ruby-mccabe-halstead>

<https://github.com/xionon/hacounter>

<https://github.com/liorkesos/drupalcamp>

https://github.com/eurismarpires/teste_git

<https://github.com/vindula/vindula.blog>

<https://github.com/ElGringov2/swrpgcompanion>

<https://github.com/nomack84/spring-security-rest-example>

<https://github.com/dgilbert85/gitimmersion>

<https://github.com/respinar/athletes>

<https://github.com/ik11235/AOJcontest-counter>

<https://github.com/naXa777/dummy-web-app-2>

<https://github.com/NataBangun/Riskesdas>

<https://github.com/bteng22/string-calculator-kata>

<https://github.com/cougarTech2228/sensorSystem>

https://github.com/eyantra/CS684_Rangoli-Bot_2011

<https://github.com/Bambooy/12-Seconds-Gamejam>

<https://github.com/spapageo0x01/dioskrS>

https://github.com/xiaolonw/nips14_loc_seg_testonly

<https://github.com/onogithub/NovelIDE>

<https://github.com/guts2014/StockRun>

<https://github.com/wznshuai/AntiLoseDevice>

<https://github.com/xiongjunliang/topcoder>

https://github.com/Evervolv/android_device_htc_common

<https://github.com/ArtyomRusak/blog>

<https://github.com/pavankumar2203/SnippetsForAmazonWebServices>

<https://github.com/Josie-zou/Project>

<https://github.com/davidmalcolm/smoketest>

<https://github.com/mer-packages/qtgraphicaleffects>

<https://github.com/aasheer/depot>

<https://github.com/jimlambie/smooze-store>

<https://github.com/endowdly/EngineeringInALockedCloset>

https://github.com/hartzellt/first_app

<https://github.com/kuuji/android-supsup>

<https://github.com/Harwkk/laboxid-base-devel>

<https://github.com/DzonyKalafut/LWJGL3DEngine>

https://github.com/DSBell/cs190lab5_bell99

<https://github.com/keiji0/libobj>

https://github.com/criddell/demo_app

<https://github.com/guihsabino/Validar-IP-e-Mascara>

<https://github.com/cis-yogendra/cis-cart>

https://github.com/strawlab/visualization_common

<https://github.com/slorange/Diablo3-AuctionHouse-Bot>

<https://github.com/JackHolland/Tap>

<https://github.com/troyastorino/16.83-power-subsystem>

https://github.com/morenomariscal/clase_10

<https://github.com/jlas/misc>

<https://github.com/lalanne/Scientific-Hpc>

<https://github.com/ratnakarrao-nyros/inplace>

<https://github.com/cseslam/simply-ubuntu>

<https://github.com/Kvothes/Test>

<https://github.com/mcansky/-bearded-octocat>

<https://github.com/wooga/fb-payment-v2-demo-ruby>

<https://github.com/Stabledog/bin-pub>

<https://github.com/wikimedia/operations-debs-vips>

<https://github.com/rjewell/disclosed.org>

<https://github.com/jinyb09017/crawler>

<https://github.com/rcbau/hacks>

<https://github.com/TerryThreatt/react-yelp-clone>

<https://github.com/platipy/publications>

<https://github.com/utsavgupta7/myfirst>

<https://github.com/torokp/udb>

<https://github.com/Janberk/MediaBoxMngr>

https://github.com/toop/pyramid_doc_chinese

<https://github.com/yurek-0/yurek-0.github.io>

https://github.com/funtiik/sample_app

<https://github.com/willdc/project3>

<https://github.com/YukihiroMoriyama/SplitTheCost>

https://github.com/gallaghd27/demo_app

<https://github.com/angela2020/Tripadvisor>

https://github.com/makuto72/ns-3_tir

<https://github.com/jaimeceballos/Tor-k>

<https://github.com/sdecricri/JoinTextFiles>

<https://github.com/jordankomnick/HW02Final>

<https://github.com/CodigoMonki/MonoAndroidSamples>

https://github.com/HiroakiKamon/demo_app

<https://github.com/hubeicaolei/whereislover>

<https://github.com/gnilkreb/Fowler>

<https://github.com/siephen/dbt-helper>

<https://github.com/software-engineering-amsterdam/sea-of-ql>

<https://github.com/hellrage/NachGeom>

<https://github.com/lihuaiqiu/lihuaiqiu.github.io>

https://github.com/JoaoPirolo/App_RachandoAConta

https://github.com/luxal99/market_app

<https://github.com/lambda-labs-13-stock-price-2/preprocessing-pipeline>

<https://github.com/webNeat/Quiddich>

<https://github.com/KarinaMontesTrevino/codeup.dev>

<https://github.com/bmacauley-reward/example-zeit-now>

<https://github.com/Anoopc444/test>

<https://github.com/williamwang0/opencv>

<https://github.com/melunekip/SHIELD>

https://github.com/Jessiewithani/Check_Yo_Self

<https://github.com/jindradev/vscode>

<https://github.com/patrickw14/cse190>

<https://github.com/Deep15P/Portfolio-4>

<https://github.com/ffbesp/pstepup>

https://github.com/acvan11/-GA-repractice-1my_first_express_app

https://github.com/bellcom/bosa_attendees

<https://github.com/dtslocum/hello-world>

https://github.com/danielfdzperez/shooter_consola_2_jugadores

<https://github.com/dhochbaum/js-basics-online-shopping-lab-js-intro>

<https://github.com/ryjones/indy-sdk>

<https://github.com/fmmartinez/paper4>

<https://github.com/todrobbins/cabal-desktop-mini>

<https://github.com/our-power/inner-warehouse-monitor>

<https://github.com/johndpope/pretrained-weights>

<https://github.com/GreenCoin-Project/Forum>

<https://github.com/EloneSampaio/odoo12-custom-report>

<https://github.com/QuizzesGroup1/Quizzes1>

<https://github.com/takezoe/fluency>

<https://github.com/Karasuma1412/programs>

<https://github.com/lawsonhung/method-scope-lab-dumbo-web->

<https://github.com/KyleAMathews/testing-medium-export>

<https://github.com/42IN11EWd/RAS-Android>

<https://github.com/Barrymuch/WifiSMS>

<https://github.com/lucasmv/desenvolvedor.io>

<https://github.com/stuartraetaylor/diydog-beerxml>

<https://github.com/ahujaprabhpreet/csye6225-summer2019-template>

<https://github.com/mayurisinkar/esdl-mayuri-sanika>

<https://github.com/br0wnie/gulp-size>

<https://github.com/VijayMVC/MasterSchedule>

<https://github.com/AprilHGraves/aA-W7D5>

<https://github.com/DannyGH/pi>

<https://github.com/myagley/mlog>

<https://github.com/Moneyalls123/Moneyfalls123>

<https://github.com/santiagocardo80/pull-request>

<https://github.com/whitftyl/sfdx-project>

<https://github.com/Daniel-Acosta-L/cosa>

<https://github.com/game-5/finale>

<https://github.com/atfost3r/AutomateTheBoringStuff>

<https://github.com/sharonk515/calculating-distance-data-science-intro->

<https://github.com/BrandonDBell/trailheadx19-sfdx-bootcamp>

<https://github.com/liqingshuchina/vue-first>

<https://github.com/wuyunxiangwyx/oie-resources>

<https://github.com/prnta/duvida-tipagem-js>

<https://github.com/tyunegov/Explorer>

<https://github.com/lucaspolo/universodev>

<https://github.com/niekang/JavaScriptCore>

<https://github.com/VivienYH/hello-world>

<https://github.com/hafssaelloiaabane/dattt>

<https://github.com/Gavin-Kimberlin/Gavins-Portfolio>

<https://github.com/edivldo/junior>

<https://github.com/rb670586/background-generator>

<https://github.com/hunsakerjeff/sfdx-project>

<https://github.com/Niffery/Q-Master>

<https://github.com/IrKor/awesome-ton>

<https://github.com/mashhour04/mern-boilerplate>

<https://github.com/robertliscano/challenge>

<https://github.com/funmi5/Banka>

<https://github.com/LiuJin3042/OSX-KVM>

<https://github.com/pragmatux/ptux-sdk-boneblack>

<https://github.com/Vindicia/vindicia.github.io>

<https://github.com/anderspitman/iobio-backend>

<https://github.com/mateusz-piotrowski/mateusz-piotrowski.github.io>

<https://github.com/rswgnu/keycastr>

<https://github.com/Fengyu94/homework>

<https://github.com/Adahel/LomLib>

Bibliografía

- [1] E. Alpaydin. *Introduction to Machine Learning*. MIT Press, 2010.
- [2] V. F. P. D. Baishakhi Ray, Daryl P Posnett. A large scale study of programming languages and code quality in github. In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*, pages 155–165. ACM, 2014.
- [3] D. C. N. . B. M. Batista. Big data. anÃ;lisis de datos., June 2013.
- [4] N. M. B. G. H. D. P. Bird, C.; Nagappan. Donât touch my code!: examining the effects of ownership on software quality. In *Proceeding of the 19th ACM SIGSOFT symposium and the 13th European conference on foundation of software engineering*, pages 4–14. ACM, 2011.
- [5] A. Danial. Cloc â count lines of code, 2014.
<http://cloc.sourceforge.net/>.
- [6] S. Dean J, Ghemawat. Mapreduce: simplified data processing on large clusters.
- [7] P. Dourish and V. Bellotti. Awareness and coordination in shared workspaces. In *Proceedings of the 1992 ACM conference on Computer-supported cooperative work*, pages 107–114, 1992.
- [8] R. Dyer, H. A. Nguyen, H. Rajan, and T. N. Nguyen. Boa: A language and infrastructure for analyzing ultra-large-scale software repositories. In *2013 35th International Conference on Software Engineering (ICSE)*, pages 422–431. IEEE, 2013.
- [9] K. B. L. S. D. M. G. D. E. D. Eirini Kalliamvakou, Georgios Gousios. The promises and perils of mining github. In *Proceedings of the 11th Working Conference on Mining Software Repositories*, pages 92–101. MSR, 2014.
- [10] G. H. L. S. Ghemawat, S. The google file system. 37:29–43, 2003.

- [11] J. O. H. Sajjani, V. Saini and C. V. Lopes. Is popularity a measure of quality? an analysis of maven components. In *IEEE International Conference on Software Maintenance and Evolution*, pages 231–240. IEEE, 2014.
- [12] S. G. J. Dean, 2010. Patent No 7.650.331.
- [13] Kinsta, 2020.
<https://kinsta.com/es/base-de-conocimiento/que-es-github/>.
- [14] C. manejar GitHub, 2020.
<http://guides.github.com>.
- [15] C. A. S. Miltiadis Allamanis. Mining source code repositories at massive scale using language modelling. In *Proceedings of the 10th Working Conference on Mining Software Repositories*, pages 207–216. MSR, 2013.
- [16] C. M. Minelli, M. and A. Dhiraj. *Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses*. Jon Wiley Sons, 2013.
- [17] N. Munaiah, S. Kroh, C. Cabrey, and M. Nagappan. Curating github for engineered software projects. *Empirical Software Engineering*, 22(6):3219–3253, 2017.
- [18] Nextu. ¿qué es github?, Feb. 2020.
<https://www.nextu.com/blog/que-es-github>.
- [19] Openwebinars, 2020.
<https://openwebinars.net/blog/que-es-github/>.
- [20] Y. P. Peraza. Big data y la visualización en el ámbito educativo, July 2017.
Trabajos de Fin de Grado en Ingeniería Informática. Universidad de La Laguna.
- [21] R. Pike, S. Dorward, R. Griesemer, and S. Quinlan. Interpreting the data: Parallel analysis with sawzall. *Scientific Programming*, 13(4):277–298, 2005.
- [22] L. J. L. R. J. K. T. F. Bissyandé, D. Lo and Y. L. Traon. Got issues? who cares about it? a large scale investigation of issue trackers from github. In *Proceedings of 24th International Symposium on Software Reliability Engineering (ISSRE)*, pages 188–197. IEEE, 2013.
- [23] D. A. Wheeler. Hall of fame, 2016.
<http://ghtorrent.org/halloffame.html>.