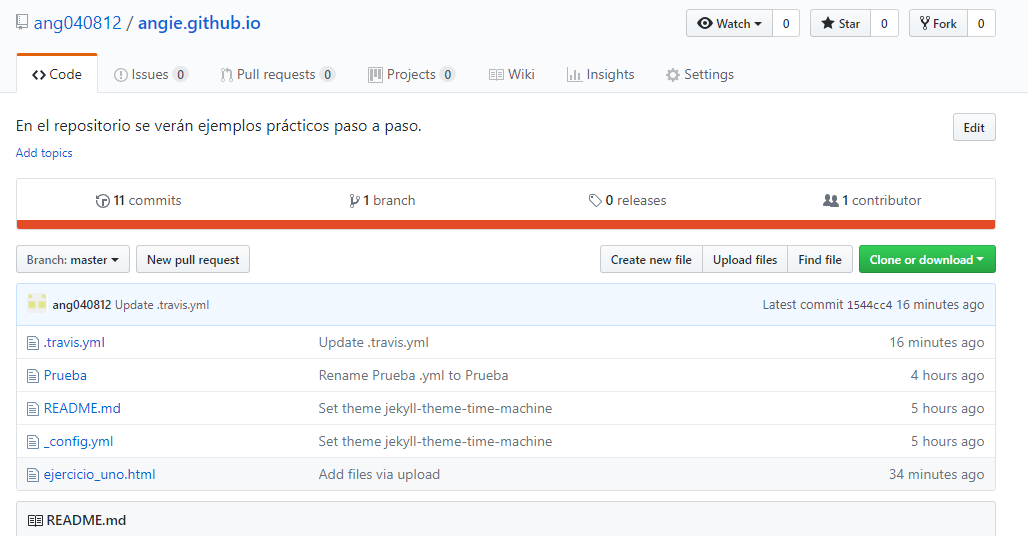
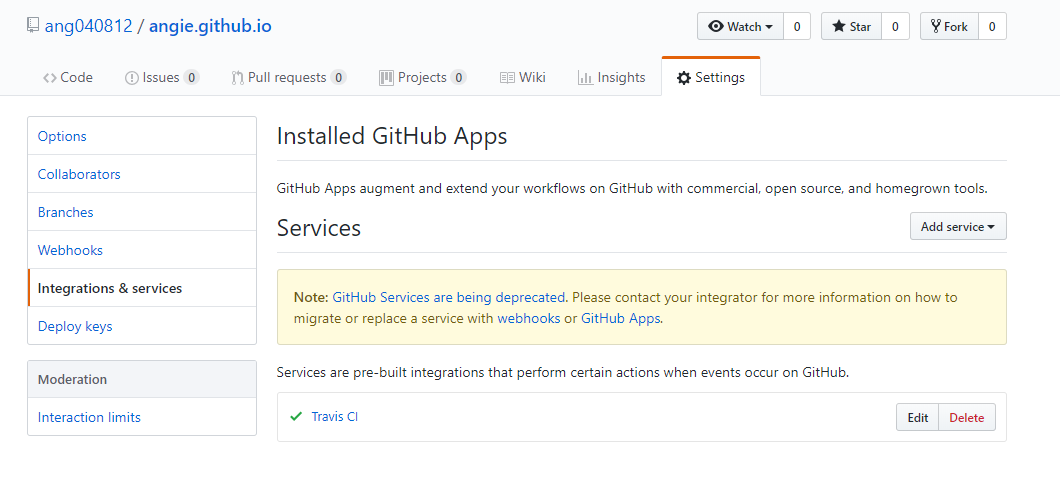
Buenas noches profesor;

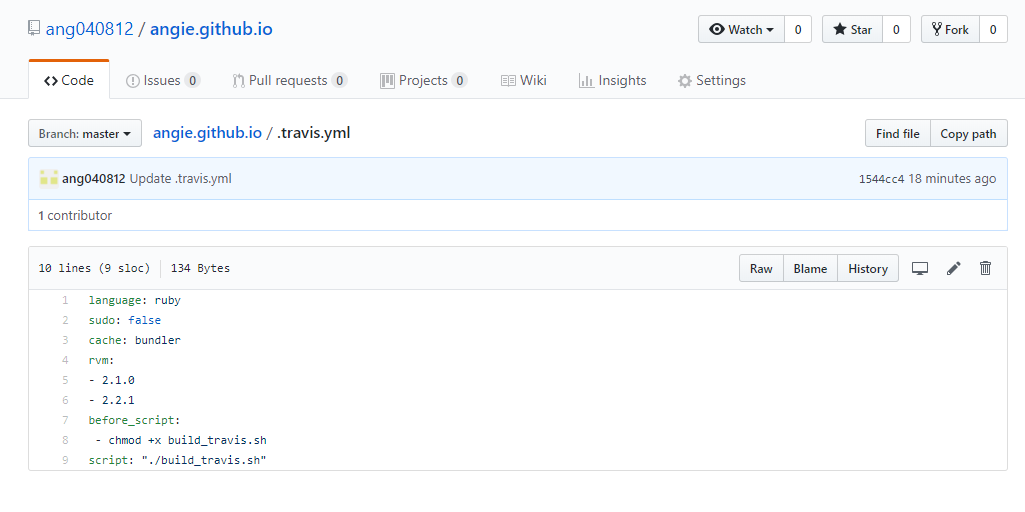
A continuación envio las respuestas de las preguntas que usted realizo y la evidencia de las pruebas realizadas en github con la evidencia de los pasos generados en la activiad.

A continuación envio la ruta del repositorio.

<https://github.com/ang040812/angie.github.io>







**Que son los archivos .yml**

Son Documentos que permiten que los desarrolladores guarden información con extensión .yml lo cual genera que no se tenga que usar un solo lenguaje de programación ya que por medio de estos archivos es posible programar en Ruby, C / C ++, Python, Perl, PHP, Java, JavaScript, AJAX, C # y así sucesivamente. Por ejemplo, un archivo con la [.yml](https://www.reviversoft.com/es/file-extensions/yml) como extensión puede almacenar contenido y servir como un archivo de base de datos implementada con RoR (Ruby on Rails) funcionalidades de programación, y este archivo de base de datos se puede utilizar para almacenar información sobre las conexiones de red cada vez que los usuarios tratan de conectarse a una base de datos de red.

Estos documentos se guardan en formato de texto plano y se adjuntan con él [.yml](https://www.reviversoft.com/es/file-extensions/yml) extensión.

Estos archivos también pueden ser leídos por cualquier editor de texto desarrollado para crear, abrir y editar archivos de texto sin formato, ya sea la edición de texto de software para sistemas basados ​​en Microsoft Windows, como Microsoft Bloc de notas y Microsoft WordPad, o para las plataformas Mac como el software de Apple TextEdit. Bibliotecas YAML también se pueden utilizar para integrar el formato YML en varios lenguajes de programación.

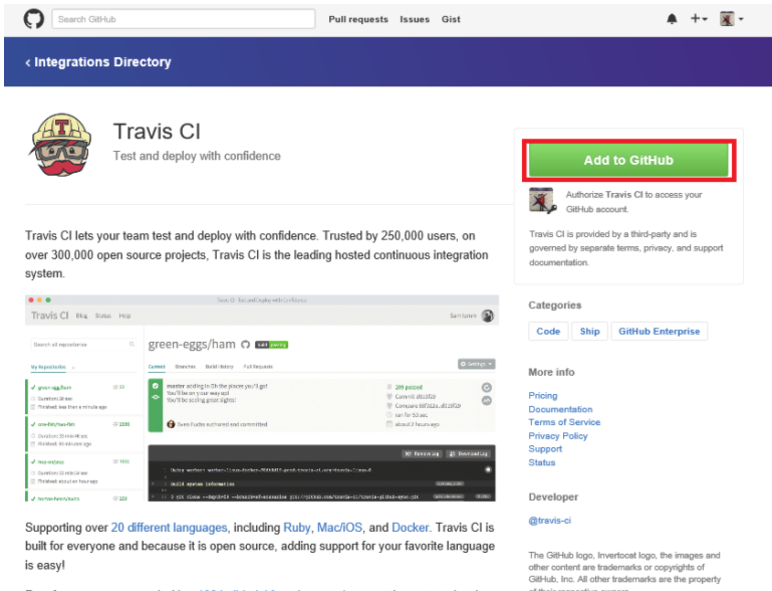
**Travis configuración Github**

[Travis-CI](https://travis-ci.org/) es un sistema de [Integración Continua](https://www.federico-toledo.com/diferencia-entre-continuous-integration-delivery-y-deployment/), gratuita para proyectos Open Source y de pago para proyectos privados. Se integra sin problemas con GitHub y automáticamente ejecuta el pipeline definido en cada push o pull requests. Testea y buildea aplicaciones escritas en Ruby, Node, Objective-C, Go, Java, C# y F#, entre otras (que corran en Linux).

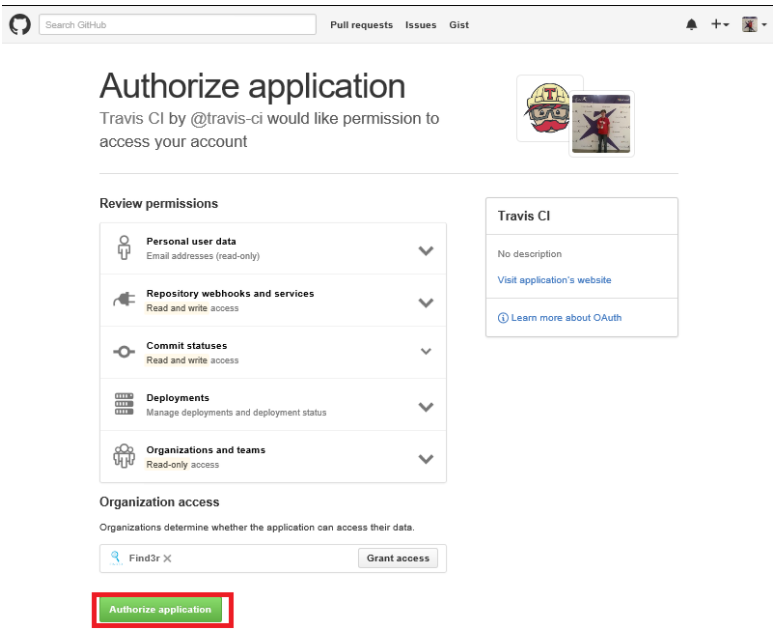
Pasos a seguir

El primer paso es crear un repositorio GitHub que contenga el código de la aplicación a la cual queremos incorporar en Travis-CI.

Una vez el repositorio es creado, debemos dar permisos a Travis para conectarse con el mismo. Para esto debemos ir a las configuraciones del repositorio, hacer click sobre “Servicios” y seleccionar Travis-CI. Una vez allí seleccionar “Add to GitHub”.



Luego de esto, en la próxima página debemos otorgar todos los permisos requeridos a Travis-CI.



Una vez en Travis-CI, seleccionar el “+”, buscar el repositorio sobre el cual queremos trabajar y hacer click sobre el switch.



Volver al repositorio GitHub y crear el archivo ***.travis.yml***. Este archivo debe estar alojado en la raíz del proyecto. Le indicará a Travis-CI lo que debe realizar cada vez que ejecuta un Build. En el archivo incluir al menos las siguientes líneas:

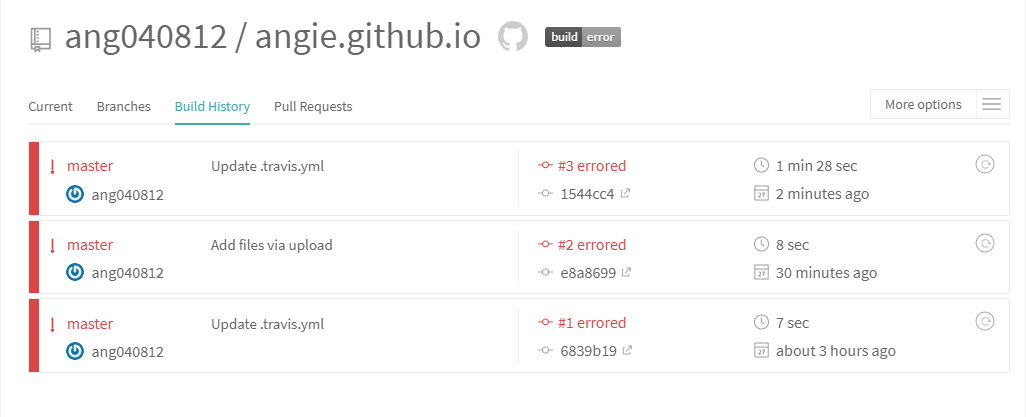
Language: java

Jdk: oraclejdk8

Estas líneas indican el lenguaje de programación y la versión de la JDK que se debe utilizar para hacer el build (en este caso la aplicación lo necesita).

Luego de este paso hacemos el commit correspondiente. En el momento que sea realiza el commit, Travis-CI va a ejecutar el Build, por lo que si abrimos la pestaña veremos que hay una tarea corriendo.

En caso de que el Build sea satisfactorio, Travis-CI presentará los resultados en verde, de lo contrario los mismos se verán rojos. En cualquiera de los dos casos, hay una pequeña descripción.



Estos han sido los pasos mínimos para configurar Travis-CI. Se pueden agregar muchas tareas más como notificaciones por email, por Slack, correr otros tipos de pruebas, hacer build con varios SDK distintos, entre otras. Todo esto es configurable mediante el archivo .travis.yml y en la documentación oficial explica todo lo que podemos hacer (<https://docs.travis-ci.com/>).

**CodeShip**

Codeship es básicamente un ambiente digital IC/EC para programadores en la nube con aplicaciones como GitHub, Bitbucket y GitLab. Con él, es posible configurar y ejecutar builds automáticamente, puedes hacer deploys en cualquier ambiente.

El plan Basic, o sea, el gratuito, es indicado para suportar el trabajo de “Docker” con una plataforma de integración disponible, conectando registros públicos y privados con facilidad.

El plan Pro, pago en relación con servicios usados, puedes trabajar con compilación robotizada listas para el uso en aplicaciones pre-instaladas.

**Características Principales**

* Configuración de builds y testes desde cualquier plataforma (Tablet/PC/Smart) debido al almacenamiento en nube de los repositorios;
* Repositorios integrados (GitHub; Bitbucket; GitLab);
* Red comunitaria de programadores que pueden agregar contribuciones a proyectos;
* Guías de lenguaje de programación, desarrollo de software de implementación y configuración.
* Soportes a los siguientes lenguajes: Ruby; Node; PHP; Python; Java y Go;