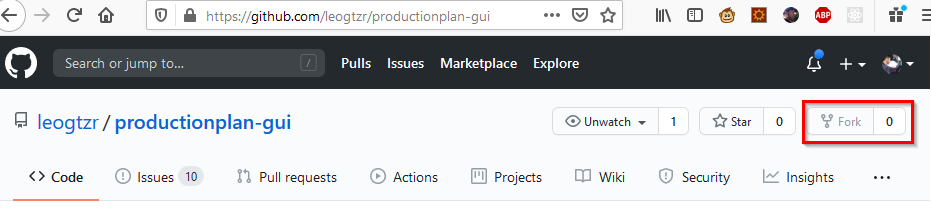
## Dónde está el código?

<https://github.com/leogtzr/productionplan-gui/>

Todo el código se encuentra ahí.

Miriam, te recomiendo que crees una cuenta en Github, eso permitirá tener el proyecto y manejar su ciclo de vida ahí. El próximo desarrollador tendrá que crear una cuenta también.

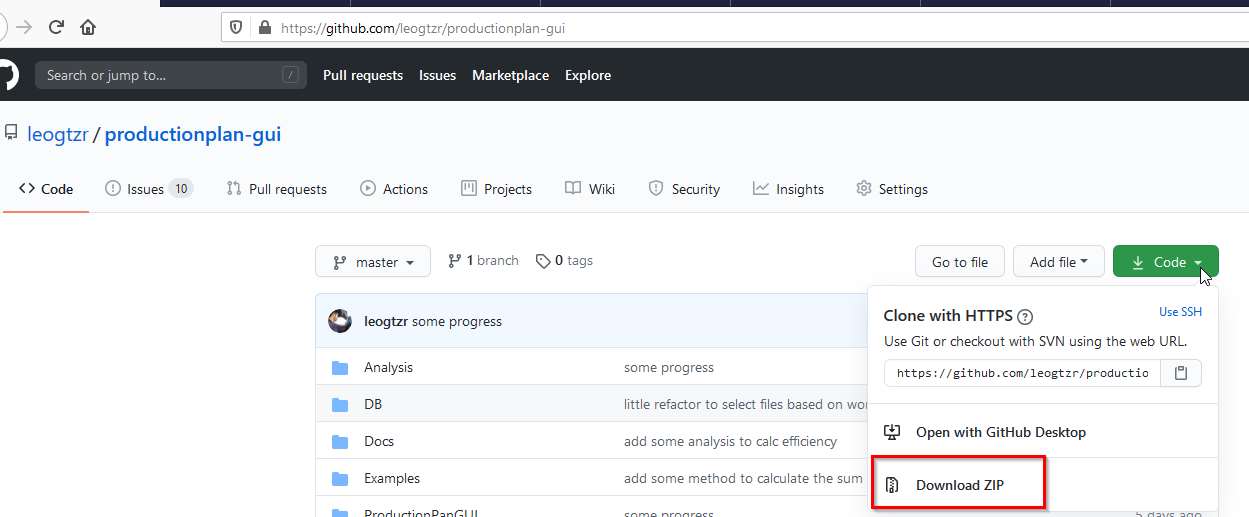
Recomiendo que el desarrollador haga un *Fork* de mi proyecto después de crearse su propia cuenta.



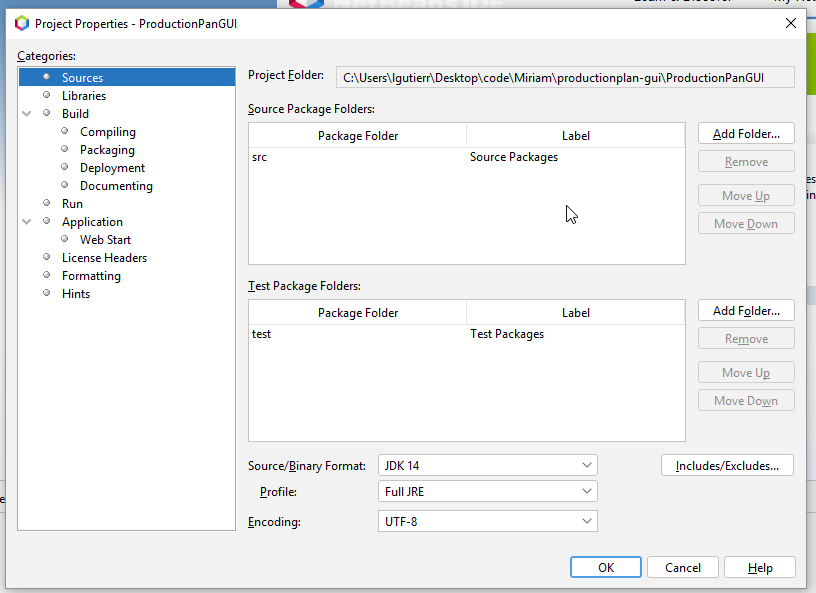
Así él podrá trabajar el código en su propio repo.

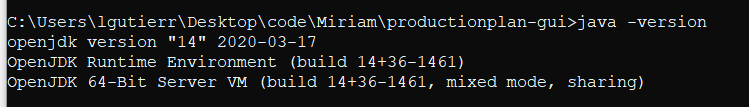
## Puedo bajar el código en formato ZIP?

Sí, de la siguiente manera:



## ¿Qué programas eran utilizados para el desarrollo?

* NetBeans 12.0
  + <https://netbeans.apache.org/download/nb120/nb120.html>
* JDK 14
  + 

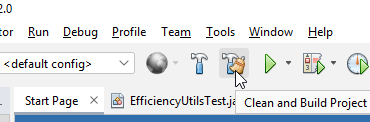


JDK 14 se puede descargar desde aquí:

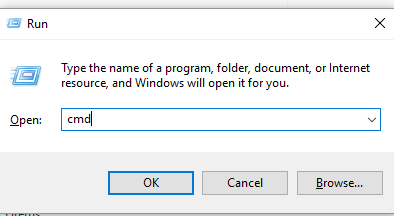
<https://jdk.java.net/14/>

## ¿Cómo correr el código?

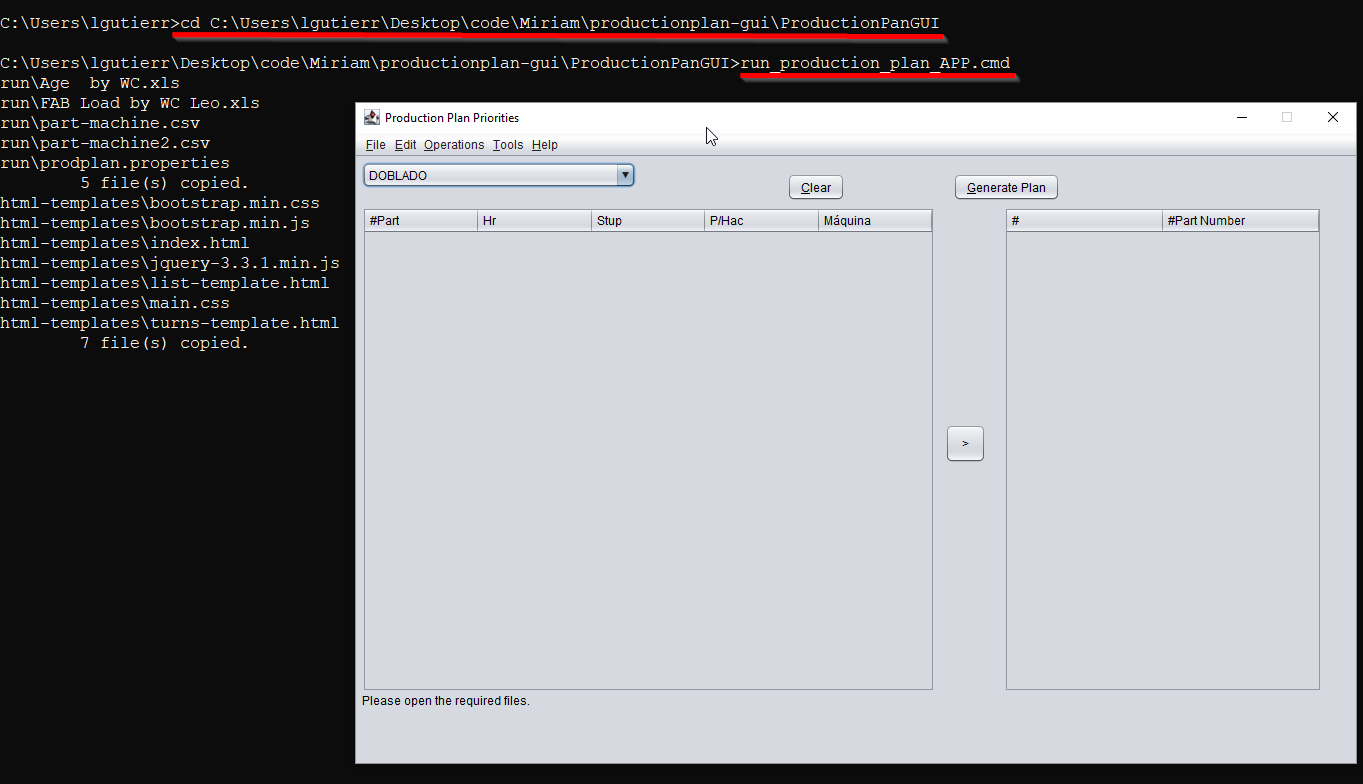
1. Clean & Build primero



1. Abrir una ventana de comandos (CMD) y correr el script “run\_production\_plan\_APP.cmd”:

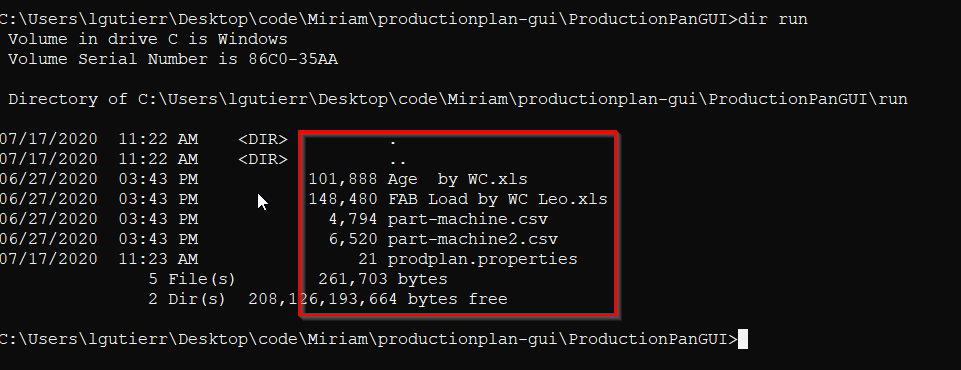


Ir con “cd” a la ruta donde se haya bajado o clonado el repositorio de código:



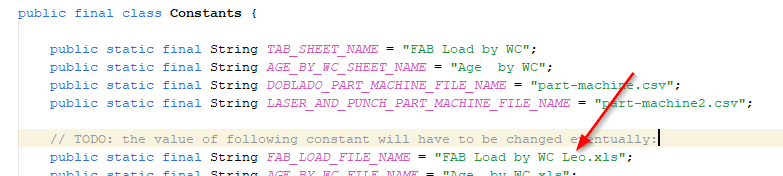
El script ya pone los archivos necesarios en el directorio donde el programa los espera para solo tener que teclear la combinación “Control + 3”.

Con “Control + 3” el programa interpretará los archivos que están dentro del directorio “run”:



Cuando estaba desarrollando esto pensamos que en lugar de estar explorando los archivos uno a uno, quizás sería mejor si ponemos los archivos en un directorio que el programa reconoce y que luego con una combinación de teclas el programa los buscaría específicamente ahí.

Una de las cosas pendientes es cambiar el nombre de los archivos, como se puede observar arriba, uno de los archivos tiene en su nombre la palabra “Leo”, eso hay que removerlo y hacer los cambios correspondientes en la clase Constants.java:



## Un poco más de información

El código ya se encarga de *parsear* los archivos de entrada y ponerlos en sus respectivas estructuras de datos.

Recomiendo empezar a analizar el contenido del método generatePlanBtnActionPerformed() dentro de la clase MainWindow.

Eso dará una idea de la cadena de ejecución hasta conseguir el HTML. OJO: ese código no funciona ya que no toma en cuenta el Split de ordenes por eficiencia. Ver:

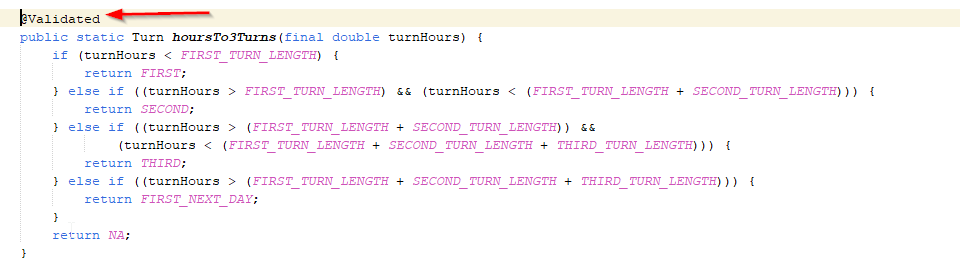
<https://github.com/leogtzr/productionplan-gui/issues/12#issuecomment-662102709>

El algoritmo para eficientar tiene que ser realizado desde 0, es ahí donde me quedé sin tiempo.

## Tests

Gran parte del código ha sido verificado con Unit Tests. *Ver las clases* UtilsTests.java, HTMLFormatTests.java y EfficiencyUtilsTests.java.

Para cada método que empezaba realizaba su correspondiente prueba unitaria. Cuando estaba seguro que funcionaba marcaba el método como validado con la anotación @Validated.



La intención del programa era generar código HTML para luego ser ubicado en un Shared Drive. Recomiendo mirar la sección de “Issues” para entender las cosas que faltan y las dudas siendo trabajadas.

<https://github.com/leogtzr/productionplan-gui/issues>