**Proposal to use Git and the Github platform in the development of algorithms (Concepts) in the Detection of medical claims.**

**Author:** *ProDev Developr Team*

**Problems that exist in the development of algorithms:**

* Spaghetti code
* Comments within the same code (Give example).
* Code that previously worked is impossible for the developer to obtain again because it was overwritten.
* Compare different versions of the same code to determine the error.
* Due to the nature of the data that is constantly updated, at one point the data is obtained from a version of the data and then as time passes (months, days) no data is obtained.
  + This casts doubt on the effectiveness of the QA and developer's work.
* Code in different formats (small letters, large letters, chamfered notation, etc.) and styles that make it very difficult for other developers to understand.
* A huge learning curve.
* Algorithms (BigQuery script) without documentation (White Paper) which does not allow studying or knowing the logic of the algorithm.

**Solution proposal**

Use [git](https://www.atlassian.com/git/glossary) and [github](https://github.com/) as a system and repository and version control.

- Use github actions as a Continous Integration / Continuous Deployment (CI/CD) system for validation and checking of the source code of the algorithms.

- Proposed directory hierarchy and source code:

PORA

**README.md**

**<general\_docs>**

* ***sql\_code\_style.md***

**<schema\_name\_in\_BigQuery>**

* *<algorithm\_name\_1>*
  + ***algorithm\_name\_1.sql***
  + **<WP>**
    - *algorithm\_name\_idAlgoritm\_wp.docx*
  + **<TABLES>**
    - *finalPool\_v2.csv*
  + **<OUTPUTS>**
    - *result\_algorithm\_name.csv*

…..

* *<algorithm\_name\_n>*

**Glossary:**

**<general\_docs>:**

***sql\_code\_style.md***

**<schema\_name\_in\_BigQuery>:**

*<algorithm\_name\_1>*

***algorithm\_name\_1.sql***

**<WP>**

*algorithm\_name\_idAlgoritm\_wp.docx*

**<TABLES>**

**<OUTPUTS>**

**Rama master en github:**

Cada issue del Jora esta asociado a una rama de nombre #isssue\_nombre del algoritmo que de master la estructura principal y el resto del código.

* Al desarrollar un algoritmo siempre que se haga un commit en el texto se debe color al final el #issue asociado al Jira para que quede comentado cualquier avance y sea fácil relacionar y acceder desde el Jira.
* Para desarrollar un algoritmo nuevo o arreglar un bugs de alguno ya creado:
  + Respectar las normas de code style para SQL.
  + Seguir la estructura en el árbol de directorios propuesta anteriormente.
  + Cuando se termine la realización del algoritmo se debe crear un pull request contra la rama master.
  + En el punto anterior si el proceso de QA a finalizado entonces se aprueba el PR, no sin antes revisar el cumplimiento del code style y otros requerimientos.
  + Mientras se encuentra en el proceso el PR el código fuente de un algoritmo un team developer lead o personal designado revisara el código y chequeara el cumplimiento con el code style y podrá hacer sugerencias al desarrollador.
  + Solo un personal autorizado o con privilegios hará la mezcla sobre master (en caso que se decida pueda ser el mismo desarrollador por el nivel de confianza, aunque no se sugiere).
* Realizar un push sobre la rama en la que se desarrolla el código fuente de un algoritmo.
  + Si se tienen tablas de pruebas se debe crear en el esquema de pruebas.
  + Crear o actualizar el algoritmo en el esquema de pruebas.
  + Ejecutar el algoritmo para validar que el código este sintácticamente correcto.
  + Obtener la salida para crear Sample\_File o Final File
  + Salvar la salida en archivos de tipo csv de tipo Sample\_File