**LISTA DEFINITIVA DE CARACTERÍSTICAS**

* **LABEL**

**completed** : DICHO INTENTO ACABÓ EN ÉXITO O FRACASO

* **USUARIO**

**percentage\_tutorial** : porcentaje de puzzles tutorial completados

**percentage\_intermediate**: porcentaje de puzzles intermedios completados

**percentage\_advanced**: porcentaje de puzzles avanzados completados

**attemps\_per\_puzzle**: numero de intentos por cada puzzle completado

**user\_elo**: ELO del jugador

* **PUZZLE**

**puzzle\_difficulty**: Dificultad del puzzle ( levels of difficulty)

**puzzle\_elo**: Dificultad del puzzle ( ELO )

* **INTENTO**

**percentileEvents**: Número de eventos normalizado respecto a ese puzzle (persistenceByAttempt)

***percentileAtt*:** Número de submits normalizado respecto a ese puzzle (persistenceByAttempt)**.**

**percentileComposite ¿?** ( Sería solo con events y Att, y no con activetime como en PersistenceByAttempt. A lo mejor sustituiría a las dos anteriores )

**persistence ¿? :** En función de completed y percentileComposite ( para mi no tiene sentido ya que el modelo detectaría y tendría en cuenta estas dos características )

***n\_breaks*:** Número de breaks ( inactividad de más de 15 segundos )

***n\_snapshot*:** Número de eventos de snapshots.

***n\_rotate\_view*:** Número de rotaciones.

**avg\_time\_between\_submissions ¿? :** Tiempo medio entre submits

**qualityFigures**: Evaluar figuras creadas( cuantas, son correctas? Cuanto dista de la solución?) (misconceptions)

**betterSubmit**: Mejor submit ( porcentaje de vistas correcta (0,33,66,100) ) (misconceptions)

**FUNCIONES A IMPLEMENTAR**

* **timeLimitEvents:** Dado unos dataEvents y un timestamp, se queda con los eventos ocurridos antes del timestamp indicado ( HECHO )
* **avgTimeByPuzzle**: Función para sacar los percentiles (HECHO)
* **userHistorial**: Función para obtener porcentajes de puzzles completados para un usuario y número de intentos por puzzle. (HECHO)
* **ELO**: Dado un usuario y un puzzle objetivo, devuelve el ELO del usuario y el puzzle ( HECHO )
* **computeFeatures**: Conseguir meter todas las características recogidas en un solo dataframe. Hacemos una primera pasada a todos los eventos, de donde obtenemos toda la información que necesitamos sobre cada intento. Después llamamos al resto de funciones que nos dan para cada intento información sobre el usuario y el puzzle.