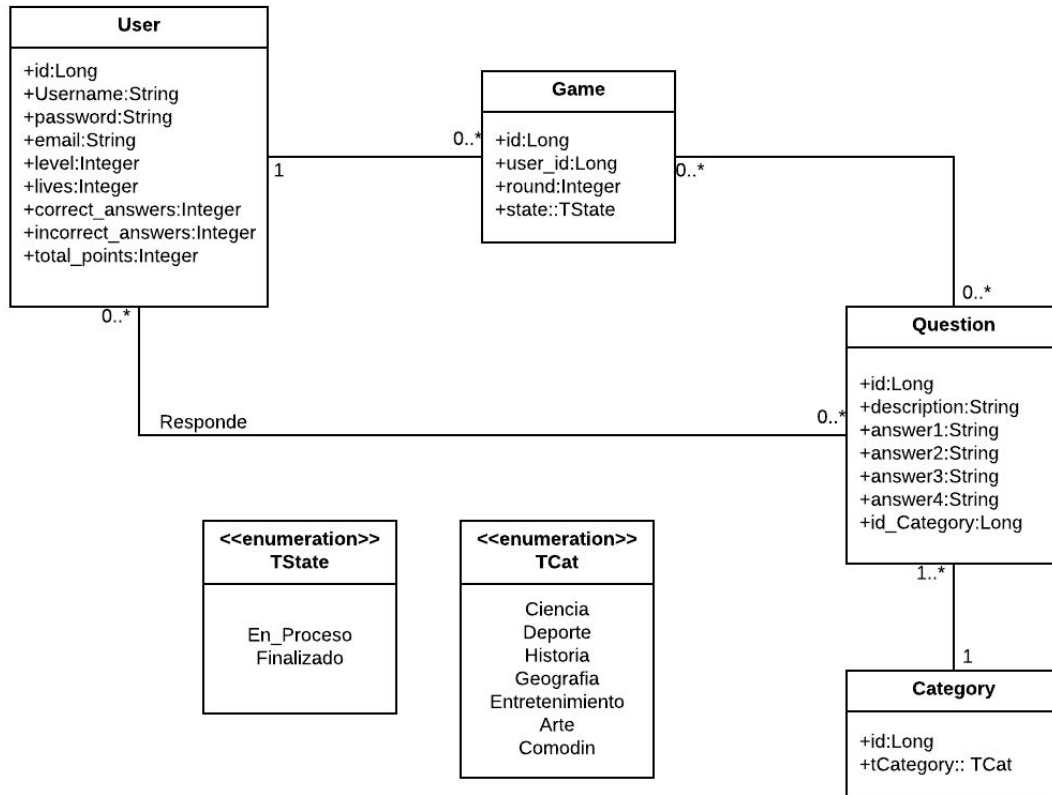


# Métricas Orientadas a Objeto - Pregunta2

## Análisis y Diseño de Sistemas



(\*) Todas las clases del diagrama heredan de Model(ActiveJDBC), y la clase App se relaciona con todas ellas.

- **APH (Árbol de Profundidad de Herencia) = 3**

Object -> Model -> Our Classes

Como la profundidad de herencia no es grande, podemos decir que es más fácil predecir el comportamiento de las clases hijas.

No tiene tanta complejidad el diseño.

- **NDD(Número de descendientes directos )= 6**

(\*) Consideramos sólo el número de descendientes directos de la clase Model

De la clase model heredan las clases **User**, **QuestionsUsers**, **Game**, **GamesQuestions**, **Question**, **Category**.

Esta métrica nos indica que al tener un número de descendientes directos elevado incrementa la reutilización, pero como aspecto negativo hay mayor número de pruebas.

Las clases **QuestionsUsers** y **GamesQuestions** podrían ser obviadas, ya que no fueron utilizadas hasta el momento.

- **ACO(Acoplamiento):**

- ACO User= 1      {Se relaciona con Game}
- ACO QuestionsUsers = 0
- ACO Question=1    {Se relaciona con Category}
- ACO Category =1    {Se relaciona con Question}
- ACO GamesQuestions= 0
- ACO Game =1      {Se relaciona con User}
- ACO App = 6      {Se relaciona con todas las anteriores}

El acoplamiento alto como el de la clase App complejiza el mantenimiento y la lógica de misma . Además que disminuye la reutilización de los módulos.

*(\*) Para esta métrica no fueron tomadas en cuenta como acoplamiento los "Import", debido a que esto para fue considerado como una relación de uso, la cual no es considerada a la hora de hacer esta métrica.*

- **TC (Tamaño de clase):**

Calculado con métodos propios:

- TC User = 13
- TC Game=9
- TC Category=4
- TC Question= 4
- TC QuerstionsUsers= 1
- TC GamesQuestion=1
- TC App= 15

Este indicador al ser alto en las clases User y App, nos determinan menor reutilización, debido a la gran responsabilidad de las clases, son complejas las implementaciones y prueba de las mismas.

- **NP-Media( Número medio de parámetros):**

- NP-Media User=  $(0+3+1+1+1+1+1+2+2+3+3+1+1)/13 = 1,54$
- NP-Media QuestionsUsers =2
- NP-Media Question =  $(0+6+1+1)/4 =2$
- NP-Media Category =  $(0+1+1+1)/4 = 0,75$
- NP-Media Game=  $(0+1+0+1+0+0+1+1+1)/9= 0,55$
- NP-Media GamesQuestions= 2
- NP-Media App=  $(2+2+(3*13))/15= 2,86$

En general creemos que la cantidad de parámetros está bien distribuida, y que no se complejiza demasiado la colaboración entre objetos, como si sucedería si esta métrica fuese demasiado elevada.

*(\*) Para calcular la métrica NP-Media, se consideró que los constructores de todas las clases, retornan Void.*

- **NOR (Número de operaciones redefinidas) =0**

Hay una jerarquía de clases fuerte, además que en este diseño el APH es bajo.

- **NOA (Número de operaciones añadidas)**

En este diseño el tamaño de clases coincide con el número de operaciones añadidas.