

# CHALLENGE - CONCURSO DE PREGUNTAS Y RESPUESTAS

Versión: 2021.08.26

## Resumen

En este reto vamos a modelar un concurso de preguntas y respuestas, la intención es diseñar una solución que permita tener un banco de preguntas con diferentes opciones para una única respuesta, además cada pregunta debe estar en una categoría o un grupos de preguntas similares del mismo nivel, por cada ronda se deberá asignar un premio a conseguir, las rondas del juego son nivel que van aumentando en la medida que el jugador gana premios.

Dentro del reto se debe considerar lo siguiente:

- Manejo de clases u objetos a nivel de modelamiento.
- Persistencia de datos o guardado de históricos.
- Manejos de listas o colecciones y ciclos de control adecuados
- Conocimiento de cualquier lenguaje de programación.
- Manejo de Git (versión de control).

Solo puedes aplicar al reto si te sientes capaz de hacerlo.

¡Buena suerte!

## Use Case/Problem

Lo que se busca en este **juego** es pre-construir una serie de preguntas con 4 **opciones** de respuesta y una de ella válida, además de una **categoría** asociada con el mismo **nivel de dificultad**. Cada pregunta debe estar categorizada y debe existir mínimo 5 preguntas por categoría donde por cada ronda se debe extraer una pregunta de esa categoría de forma aleatoria, debe existir 5 **rondas** en todo el juego, cada ronda otorga premios (puntos o dinero) cuando el jugador acierta correctamente.

El acumulado de **premios** está dentro del concurso pero si dado el caso el **jugador** pretender salir de juego puede retirarse antes de responder a la pregunta, si por el contrario el jugador pierde entonces saldría del juego sin el **acumulado** que llevaría en ese momento.

### Ejemplo:

1. Precondiciones: Debe de tener 25 preguntas (5 preguntas por categorías) para 5 rondas, cada categoría tiene una complejidad o nivel de dificultad, cada ronda debe asignarle un premio que el jugador va a ganar, el premio puede ser puntos o dinero.

2. El jugador inicia con la primera ronda, el sistema busca la categoría del primer nivel y escoge una pregunta de esa categoría.
3. El Jugador selecciona una opción de las 4 opciones que tiene, si pierde se finaliza el juego si gana continua a la siguiente ronda.
4. La siguiente ronda selecciona una pregunta de un grado de complejidad mayor según la categoría. Hace el mismo comportamiento del ítem 4.
5. Si llega a la ronda 5 y pasa, entonces gana el juego, el premio mayor debería estar en la última ronda.

## Funcionalidades

- **Configurar Juego:** Crear las preguntas y respuestas (con 3 opciones erradas y 1 valida) con sus categorías correspondientes (mínimo 25 preguntas).
- **Iniciar el juego:** se debe iniciar el juego con la primera ronda y de forma aleatoria debe seleccionar una pregunta según la categoría más baja.
- **Responder a la pregunta:** debes seleccionar una opción de 4 posibles.
- **Aumentar de nivel:** al responder de forma correcta deberás aumentar de nivel y de esa manera otorgar premios según la ronda que este. Ordena primero tus categorías y de esa manera sabrás en qué ronda estás ubicado.
- **Acomular premio:** cada vez que ganes debes tener un premio total que tienes como jugador.
- **Fin del juego voluntario o ganara ronda final:** se finaliza el juego porque el jugador deci y el acumulado pasa a l jugador (guarda los datos del jugador)
- **Fin del juego forzado:** el sistema finaliza el juego porque no selecciono una pregunta correcta. (guarda los datos del jugador)
- **Persistencia de datos:** al finalizar el juego se debe guardar los datos del jugador como histórico del juego.

## Evaluation criteria

Criteria	Percentage
Realiza un modelamiento de objetos de forma correcta, aplicando los principios de programación orientado a objetos	30.0 %
Creación de objetos de entidades; ronda, jugador, categoría, premio, pregunta, opciones, etc...	30.0 %
La lógica expuesta para el juego cumple con buenas prácticas de programación, donde se evidencia una estructura y sintaxis coherente.	30.0 %
Realiza la persistencia de los resultados obtenidos de los ganadores del juego	10.0 %

**Nota importante:** El reto se debe presentar en un sistema de control de versión como lo es [github.com](https://github.com), agregar notas para permitir una fácil ejecución del programa, si tienes rutas para la configuración de base de datos o de archivo, agregar notas pertinentes en un archivo `README.md` dentro de su proyecto.

**Tiempo estimado:** 3 días a partir de la fecha.