

# Documentación

El prototipo se divide en 8 clases:

- **AnswerChecker**, que se limita a comprobar si la respuesta es correcta o no.
- **ButtonController**, guarda los valores del botón, gestiona el prefab de los botones y avisa al GameFacade cuando ha sido pulsado.
- **ButtonsSpawner**, que se encarga de “spawnear” tantos botones como se le haya indicado en el Inspector.
- **NumberPicker**, elige aleatoriamente y devuelve un entero;
- **ScoreCounter**, suma y resta valores, después los devuelve.
- **StringConverter**, convierte y devuelve un valor numérico del tipo string a texto.
- **UIDisplay**, gestiona los elementos de la UI.
- **GameFacade**, guarda referencias del resto de clases y actúa como intermediaria.

He optado por emplear un patrón “facade” y he llamado a dicha clase GameFacade. Aún ser un proyecto pequeño, tiene funcionalidades diversas, las cuales no podía encapsular en la misma clase (por ejemplo convertir un string numérico a texto, comprobar si la respuesta es correcta, instanciar las respuestas, etc.) y me parecía más adecuado mantener las clases concisas y modulares. Por ello, necesitaba un punto de referencia y acceso a todas ellas.

En vez de crear referencias del GameFacade en cada una de las clases, he utilizado una variable pública estática de la misma a la que he llamado “instance”, de esta manera cualquier clase puede acceder a ella.

En el caso de tener más de una escena, se podría convertir esta instancia en un singleton y mantenerla a través de escenas.

También tenía pensado utilizar la clase UIDisplay como único punto de acceso a la UI, utilizarla para las animaciones de entrada y salida, las actualizaciones de texto del enunciado y los botones y los cambios de color. Al final he mantenido parte de sus funciones iniciales, pero me resultaba mas comodo delegar la asignación de los valores de los botones al propio ButtonController.

El proyecto esta preparado para tener tantas respuestas como sean necesarias, si queremos modificar el número de opciones de respuesta solo tenemos que acceder al GO o prefab ButtonSpawner y modificar la longitud de sus arrays (ButtonControllers[] y CanvasGroups[]), ambos arrays deben tener la misma longitud; no es necesario asignarles ningún valor a los elementos, pues serán instanciados.

De la misma manera, el proyecto está hecho para que los enunciados sean infinitos. Podríamos tener pausas y reanudar la acción si quisiéramos. Simplemente llamando al método `GetNewSentence()` del `GameFacade` se cargaría un nuevo enunciado con sus posibles respuestas.

Un inconveniente es que el tamaño del texto de la UI no se adapta al tamaño de sus “cajas”. Para solventar esto podríamos usar el plug-in de Unity `TMPPro` con su opción de “Auto-size”. No he querido utilizarlo, ya que especificasteis que no se podía utilizar ningún asset externo.