**03AritzXabier 🡪 Suspendido**

Control de errores en bd

* Ninguno

Control de errores en bd

* Mal planteados. Por ejemplo, pregunta if (ds == null) sin try en los Fill

Capa datos

public IReadOnlyCollection<int> DevolverTodosNiveles()// Esto no tiene ningún sentido

* Totalmente absurdo, devuelve una lista de niveles de todas las preguntas (ordenados por nivel) y que no se necesitaba. Si se controlaba que había preguntas en todos los niveles del 1 al máximo (sería lo lógico) con conocer el máximo vale

public List<Pregunta> DevolverPreguntasPorNivel(int Nivel, out string msg)

* If (ds==null) sin sentido
* Control de no superar nivel máximo
* Aunque tenga la pregunta busca entre TODAS las respuestas las que coincidan con la pregunta

Capa de negocio

public IReadOnlyCollection<int> DevolverNivel()

* Nombre ilógico ya que devuelve todos
* if (preguntas == null) // Nunca lo será, aunque no haya preguntas del nivel --> devolvería lista vacía
* Tras acceder a las preguntas del nivel, las recorre de nuevo todas buscando sus respuestas entre TODAS las respuestas

public Pregunta DevolverPreguntaPorNivel(int nivel, List<Pregunta> preguntasYaUsadas, out string msg) // TODO No tiene sentido. No controla que no se repitan ya que trabaja con variables locales y sin lógica

* Entre todas las preguntas del nivel (aunque ya haya salido alguna) un número al azar por identificador entre el mínimo y máximo y busca para ese número una pregunta

Capa de presentación

* **public Respuesta btn1Info = new Respuesta(); // TODO Así con todo........Los 12 botones, 12 copiar-pegar**
* **private void btn1\_Click(object sender, EventArgs e) // TODO Uno para cada botón y sin hacer nada común**
* Puede que saque mensajes de error cuando no son válidas las respuestas pero, aunque así fuese, luego continuaría con el resto del programa

private void FrmJuego\_Load(object sender, EventArgs e)

* Para todos los botones: btn1.Enabled = false;
* Para cargar el combo como ha recibido niveles repetidos, carga en una lista auxiliar para comprobar que no existe (para eso ya se podía usar directamente el combo)

private void btnComenzar\_Click(object sender, EventArgs e)

* De nuevo btn1.BackColor = Color.Empty;…

Comentarios generales

* En CapaDatos el método public List<Pregunta> DevolverPreguntasPorNivel(int Nivel, out string msg) devuelve varios mensajes y luego en capa de negocio pone estas dos instrucciones:

List<Pregunta> todasLasPreguntas = nuevoAcceso.DevolverPreguntasPorNivel(nivel, out msg);

if (msg == "El nivel seleccionado sobrepasa el nivel de las preguntas.")

* Para cada pregunta, recorridos por todos lados, para sacar al azar en el nivel, …
* Faltas de ortografía en mensajes (tildes)
* No control de lugar en capas: preguntasYaUsadas no debía ser problema de la capa de presentación.
* Preguntar por null tras consulta que devuelve SIEMPRE lista, aunque sea sin elementos, …
* Nombres ilógicos: dsPregunta para una lista de preguntas