**Reunião de PI Zoom 29/03/21, 14h-16h30**

**(1ª parte - Prof. André)**

Levantamento de funcionalidades a implementar:

* **Jogos**: listar, pesquisar, visualizar e recuperar estado
* **Resultados**: gerar e ver resultados (página HTML ou outro método)
* **Newsletter/Mailing**: permitir envio de emails quer para toda comunidade, quer para grupos, ex: escola, professor, turma, aluno específico
* **Utilizadores**: Atualizar conta
* **Admin**: Associar Prof a uma escola, registar escola, listar escola, associar projeto, pré-inscrição de escolas em competições
* Mapa para saber localização de escolas DGEST (manter?)
* **Provas**: listar, criar e copiar
* **Validar**: aluno - permite saber se transitou de ano ou não;
* professor -> verifica se mudou de escola ou não
* **Professor**: ver os seus alunos

Extra (após restantes funcionalidades implementadas)

* **Modelos:** criação de modelos de testes com as perguntas disponíveis. De que modo? Evento -> lista provas-> listas modelos

**(2ª parte – Prof. Joaquim)**

Nesta parte da reunião explorámos juntamente com o professor Joaquim Sousa Pinto as novas ligações entre as tabelas da BD e como convertê-las para objetos ORM em .NET Core de modo a poder utilizar no projeto final.

Pudemos ainda ver como implementar paginação sobre os resultados recebidos das querys à BD ([[1]](#footnote-1) importando x.PagedList.mvc.core). No ficheiro AppSettings define-se o número de elementos por página.

Para filtrar os dados a apresentar por entidade foi nos aconselhado após o select à BD criar um novo objeto só com os campos que precisamos.

Recomendaram-nos ainda o uso da tecnologia AJAX para fazer autocomplete nas pesquisas, de forma a torna-las eficientes para a grande quantidade de dados com que teremos que interagir.

O código base apresentado nesta reunião foi-nos facultado pelo professor Joaquim para podermos explorar e aplicar no nosso projeto de acordo com as necessidades.

Por fim, para implementar a funcionalidade de envio de emails foi-nos sugerido seguir o tutorial deste link: <http://www.macoratti.net/18/04/aspcoremvc_email1.htm>

**(Trabalho da equipa após a reunião)**

**Requisitos Não funcionais:**

**- Implementação:**

* Sistema deve ser desenvolvido em C# e HTML/CSS/JS com recurso à Framework .NET Core MVC;
* Sistema deve ser integrado com a base de dados SQL existente;

**- Eficiência:**

* Carregar os dados na interface deve ser um processo rápido;
* Atualização dos dados nas várias tabelas da BD deve ser instantâneo;

**- Portabilidade:**

* Sistema deve conseguir executar em diversas plataformas;
* Sistema deve permitir autenticação através de vários IDP’s, por exemplo, Google, UA, Facebook;

**- Segurança:**

* Sistema deve garantir a encriptação das *passwords* dos utilizadores armazenadas na BD;
* Sistema deve possibilitar a configuração de autenticação *Two-Factor*;
* Sistema deve ter diferentes permissões para diferentes tipos de utilizadores

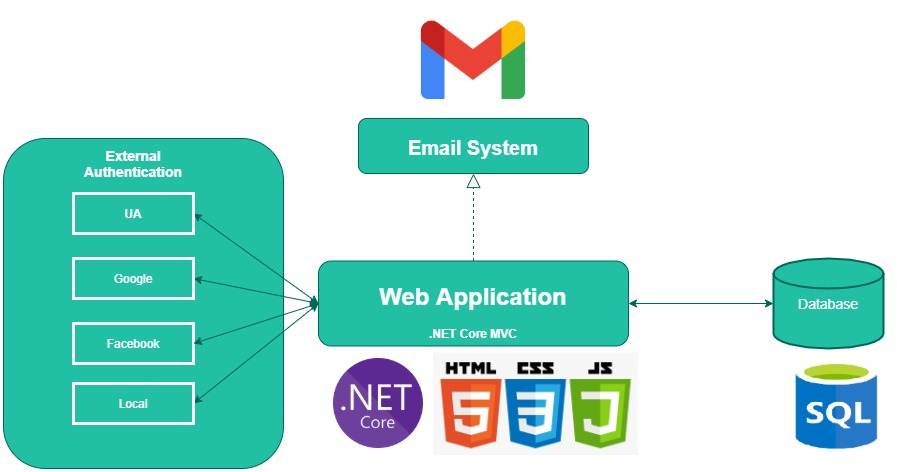
**- Usabilidade:**

* A barra de navegação deve ser simples de usar e com menus intuitivos;
* A apresentação de resultados deve ser paginada para múltiplos registos facilitando a leitura ( e rapidez de loading ) dos mesmos;

**- Legais:**

* O utilizador pode eliminar os seus dados pessoais quando quiser;
* O histórico de dados de um utilizador é mantido, mas as informações mais pessoais podem ser ocultadas para garantir anonimato.

**Diagrama de Arquitetura:**



* **Nota final:** Todos os membros da equipa estiveram presentes na reunião.

1. Usar o método toPagedListAsync(pageNumber,pageSize) [↑](#footnote-ref-1)