**TESTE TÉCNICO**

O teste consiste em criar uma aplicação para adicionar, alterar, listar e excluir usuários. A aplicação deve ter um Frontend para o usuário fazer a interação e uma API que será responsável pelas regras de negócio e persistência dos dados.

**TECNOLOGIAS**

As tecnologias que devem ser utilizadas são:

* Microsoft .NET 6+
* SQL Server
* Angular 12+

**REQUISITOS**

**Tabelas de banco de dados**

* **Escolaridade**
  + IdEscolaridade: INTEIRO (chave primária)
  + Escolaridade: TEXTO (40 caracteres)
* **HistoricoEscolar**
  + IdHistoricoEscolar: INTEIRO
  + Formato: TEXTO (4 caracteres)
  + Nome: TEXTO (200 caracteres)
* **Usuario**
  + IdUsuario: INTEIRO (chave primária)
  + Nome: TEXTO (20 caracteres)
  + Sobrenome: TEXTO (100 caracteres)
  + Email: TEXTO (50 caracteres)
  + DataNascimento: DATA
  + IdEscolaridade: INTEIRO (chave estrangeira para a tabela Escolaridade)
* **UsuarioHistoricoEscolar**
  + IdUsuarioHistoricoEscolar: INTEIRO (chave primária)
  + IdUsuario: INTEIRO (chave estrangeira para a tabela Usuario)
  + IdHistoricoEscolar: INTEIRO (chave estrangeira para a tabela HistoricoEscolar)
* Podem ser adicionados outros campos nas tabelas, caso ache necessário.

**Regras de negócio**

* Todos os campos da tabela **Usuario** são de preenchimento obrigatório.
* A quantidade de caracteres aceitos nos campos texto não poderão ultrapassar a capacidade no banco de dados.
* O e-mail deve ser validado.
* A idade do usuário não pode ser menor que 15 anos na data do cadastro.
* A tabela escolaridade deverá conter os valores (Infantil, Fundamental, Médio e Superior).
* A seleção da escolaridade deverá ser feita pela tabela de domínio **Escolaridade.**
* O histórico escolar deve ser um arquivo no formato PDF ou DOCX e podem ser incluídos vários arquivos para um mesmo usuário.
* Deve ser possível fazer o upload/download do histórico escolar.

**SQL Server**

* Criar script DDL das tabelas **Usuario,** **Escolaridade**, **HistoricoEscolar** e **UsuarioHistoricoEscolar.**
* Criar script de insert na tabela **Escolaridade** para domínio da aplicação**.**
* Criar um comando Select que traga os usuários (nome e email) agrupados por faixa etária (até 20 anos, de 20 até 50 anos e acima de 50 anos) cuja escolaridade seja igual a “Superior”.

**Backend**

* Criar uma **API** que permita adicionar, alterar, listar e excluir usuário.
* Os dados devem ser persistidos e lidos usando ORM Entity Framework ou DAPPER com banco de dados SQL Server.

**Frontend**

* Criar uma aplicação **Angular** que permita consultar, adicionar, alterar e excluir usuários (CRUD básico).
* A arquitetura e layout do frontend fica à escolha, podendo ser uma ou mais telas, disposição de elementos, etc.

**O QUE SERÁ AVALIADO**

* Funcionamento conforme especificado;
* Arquitetura e estruturação do Backend e do Frontend;
* Utilização de padrões de arquitetura DDD ou CQRS;
* Nomenclaturas de arquivos, variáveis e métodos;
* Validação de dados;
* Boas práticas e organização;
* Performance.