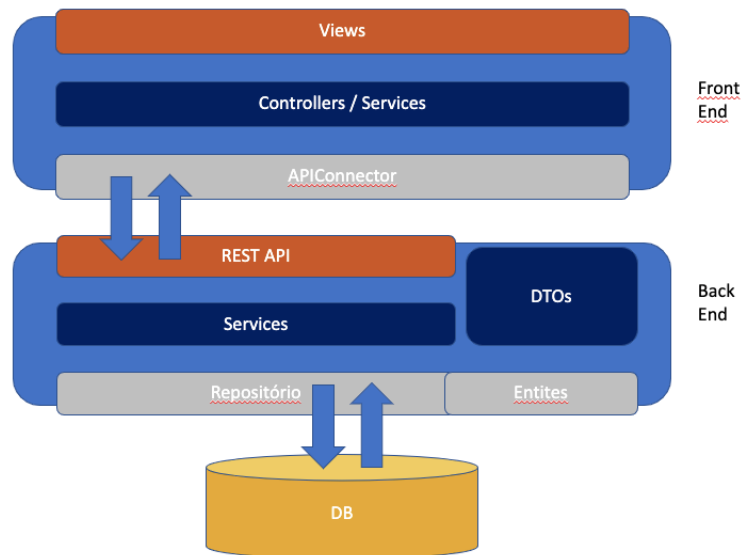


Desafio TechDEV EloGroup

Este desafio tem com objetivo provar as habilidades em desenvolvimento de aplicações.

A aplicação que será desenvolvida, tem como objetivo fazer a manutenção de Leads, esta aplicação deverá implementar a seguinte arquitetura:



Caso o seu desafio seja para a posição de desenvolvedor Backend, você deverá fazer a implementação dos componentes da camada de Backend + Banco de Dados;

Caso o seu desafio seja para a posição de desenvolvedor Frontend, você deverá fazer a implementação dos componentes da camada de FrontEnd;

Caso o seu desafio seja para a posição de desenvolvedor FullStack, você deverá implementar os componentes das camadas de FrontEnd + BackEnd + Banco de Dados;

Considerações importantes sobre os componentes

- Cada componente deverá ter suas responsabilidades bem definidas, um critério que será avaliado é a coesão na separação de responsabilidades dos componentes
- A camada de API deverá implementar algum tipo de autenticação, que seja no mínimo um mecanismo de autenticação básica sobre HTTP (Basic HTTP)
- Implementar os componentes de apresentação gráfica (GUI) sempre em arquivos específicos, evitando misturar por exemplo Controladores e Views, Services e Views em um mesmo arquivo
- A API deverá trafegar os dados em formato JSON
- Utilizar sempre o verbo correto para cada natureza de operação (GET, PUT, POST e DELETE)
- A API deverá conter um envelope padrão de resposta, evitando incluir o JSON de resposta diretamente no corpo da resposta da requisição. Exemplo:
Retornar assim:

```
{
  "metodo": "salvarLead",
  "resultado": "SUCESSO",
  "payload": {
    "id": 1,
    "nome": "Cliente EloGroup"
  }
}
```

Evitar o tipo de retorno abaixo:

```
{
  "id": 1,
  "nome": "Cliente Elogroup"
}
```

- A API não pode sob nenhuma hipótese trafegar Entidades que façam o proxy para acesso a dados (Entities por exemplo), sempre criando um Data Transfer Object (DTO) para cada finalidade
- Caso as Classes de entidades possuam anotações específicas para serialização JSON, o desafio será considerado INVÁLIDO

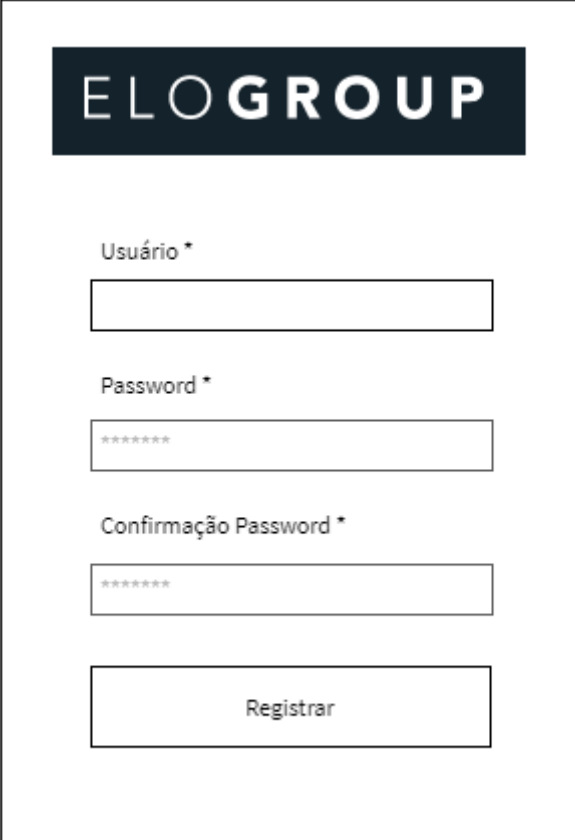
Desenvolvedor FrontEnd

Para a implementação dos componentes da camada de FrontEnd, as seguintes Frameworks são permitidas:

- Angular
- VueJs
- AngularJS
- React

Importante observar que o desenho arquitetural acima é um “guideline” para a implementação, delimitando os componentes que devem ser implementados. Estes componentes podem ter suas nomenclaturas variadas de acordo com o Framework escolhido, mas a responsabilidade de cada um deles deve ser respeitada, por exemplo: Precisamos ter um componente que entregue a camada de visualização, um componente que implemente as regras de apresentação e ou fluxo de comportamento de resposta às ações do usuário (representados em Controllers ou Classes com os devidos fins) e um componente que seja responsável por montar e encapsular as chamadas à API do Backend (representados por Services ou Classes com os devidos fins).

Abaixo alguns protótipos de tela (os itens destacados em amarelo fazem parte somente do desafio FullStack, os demais itens fazem parte dos desafios Front End e Fullstack):



The image shows a login form prototype for a system named "ELOGROUP". The form is contained within a rectangular frame. At the top, the logo "ELOGROUP" is displayed in white capital letters on a dark blue rectangular background. Below the logo, there are three input fields, each preceded by a label and an asterisk indicating it is required. The first field is labeled "Usuário *". The second field is labeled "Password *" and contains eight asterisks as a placeholder. The third field is labeled "Confirmação Password *" and also contains eight asterisks as a placeholder. Below these three fields is a single rectangular button with the text "Registrar" centered inside it.

- Todos os campos obrigatórios
- Password deve possuir ao menos 8 caracteres, contendo ao menos, um caracter especial, um caracter numérico, um caracter alfanumérico
- Password e confirmação devem ser iguais

Novo Lead (+)

Cliente em Potencial	Dados Confirmados	Reunião Agendada
Org. Internacionais		
	Ind. Farm. LTDA	
Musc. Sound Live Cmp		

- Tabela para exibição dos Leads por Status
 - Cliente em Potencial
 - Dados Confirmados
 - Reunião Agendada
- Cada Lead poderá ser “arrastado” (tendo assim seu status interno alterado) na seguinte sequencia:
 - Cliente em Potencial >> Dados Confirmados
 - Dados Confirmados >> Reunião Agendada
- Não poderão ser “arrastados” na sequencia inversa, nem tampouco de Cliente em Potencial diretamente para Reunião Agendada
- Para que um Lead seja incluído no “pipe”, deverá ser utilizada a opção Novo Lead, e uma tela (exemplo abaixo) para inclusão do Lead deverá ser exibida
- Ao se arrastar um Lead da etapa de Cliente em Potencial para Dados Confirmados, uma entidade deverá ser criada na tabela cliente (Somente Backend)
- Ao se arrastar um Lead da etapa Dados Confirmados para Reunião agendada, deverá ser exibido um campo de Data para informar a data de agendamento (esta data deverá ser concatenada ao final do campo descrição da cada item de oportunidade do Lead, da seguinte maneira [Agendado: dd/MM/AAAA HH:mi]

ELOGROUP

Novo Lead

Nome *

Telefone *

Email *

Oportunidades *

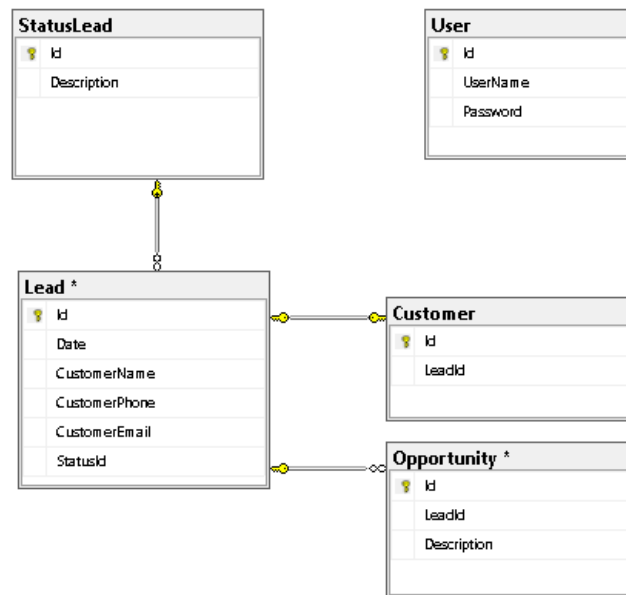
<input type="checkbox"/>	
<input checked="" type="checkbox"/>	RPA
<input checked="" type="checkbox"/>	Produto Digital
<input checked="" type="checkbox"/>	Analytics
<input checked="" type="checkbox"/>	BPM

Salvar

- Todos os campos obrigatórios
- As oportunidades podem ser marcadas / desmarcadas individualmente (utilizar CheckBox)
- Ao clicar no CheckBox no cabeçalho da coluna esquerda na grade de oportunidades, este deverá Marcar ou Desmarcar todos
- Ao clicar no Salvar exibir mensagem de confirmação de Lead incluído com sucesso
- O Lead deverá ser incluído com o Status "Cliente em Potencial"
- Ao salvar um lead, a data/hora atual deverá ser gravada automaticamente

Desenvolvedor Full Stack

- Modelo de Dados



- Script para criação das Tabelas (Exemplo para SQLServer)

```
CREATE TABLE [dbo].[User](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [UserName] [varchar](120) NOT NULL,
    [Password] [varchar](50) NOT NULL,
    CONSTRAINT [PK_User] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

```
CREATE TABLE [dbo].[Customer](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [LeadId] [int] NOT NULL,
    CONSTRAINT [PK_Customer] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

```
CREATE TABLE [dbo].[Lead](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [Date] [datetime] NOT NULL,
    [CustomerName] [varchar](120) NOT NULL,
    [CustomerPhone] [varchar](12) NULL,
    [CustomerEmail] [varchar](255) NULL,
    [StatusId] [int] NULL,
    CONSTRAINT [PK_Lead] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

```

CREATE TABLE [dbo].[Opportunity](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [LeadId] [int] NOT NULL,
    [Description] [varchar](max) NOT NULL,
    CONSTRAINT [PK_Opportunity] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

```

```

CREATE TABLE [dbo].[StatusLead](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [Description] [varchar](120) NULL,
    CONSTRAINT [PK_StatusLead] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

Criação das chaves estrangeiras

```

ALTER TABLE [dbo].[Lead] WITH CHECK ADD CONSTRAINT [FK_Lead_Customer] FOREIGN
KEY([Id])
REFERENCES [dbo].[Customer] ([Id])
GO
ALTER TABLE [dbo].[Lead] CHECK CONSTRAINT [FK_Lead_Customer]
GO
ALTER TABLE [dbo].[Lead] WITH CHECK ADD CONSTRAINT [FK_Lead_StatusLead] FOREIGN
KEY([StatusId])
REFERENCES [dbo].[StatusLead] ([Id])
GO
ALTER TABLE [dbo].[Lead] CHECK CONSTRAINT [FK_Lead_StatusLead]
GO

```


- Backend

- Desenvolver uma aplicação que irá fazer a manutenção de leads.
- Esta aplicação deverá implementar a seguinte Arquitetura

