

Candidato: SERGIO DA SILVA FARIAS

Abaixo os links do GITHUB para baixar os projetos:

Backend disponível em <https://github.com/sergfarias/BackendToro>. Requisitos .NET 5.

FrontEnd disponível em <https://github.com/sergfarias/FrontendToro> . Requisitos Angular 9.

Obs.: Acredito que o superior irá funcionar.

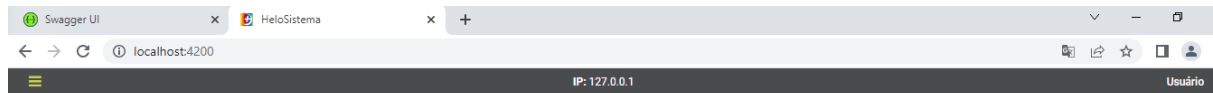
Desafio Toro Desenvolvedor Full-Stack e Backend

Histórias de Usuário

Considere as seguintes User Stories:

TORO-001 - Eu, como investidor, gostaria de acessar a plataforma Toro usando minhas credenciais de usuário e senha, para que eu possa aprender mais, investir mais e acompanhar meus investimentos.

FEITO:

A mockup of a login form. It features a dark blue header bar. Below it, there are two input fields: the first is labeled 'CPF *' and the second is labeled 'Senha *'. At the bottom of the form is a button labeled 'ENTRAR'.

☰

IP: 127.0.0.1

↑

Correntista: sergio farias || CPF: 06952688722

Saldo atual conta corrente: R\$83.857

Meus Ativos

Descrição	Quantidade	Preço
PETR4	2	28.44
SANB11	3	40.77

Saldo consolidado: R\$368.047

Duplo click na lista para comprar um dos 5 títulos mais negociados nos últimos 7 dias.

Comprar Ativos

Descrição	Preço
AZUL4	22.23
BBDC3	17.5


Ao criar o BD faço uma carga de usuário fictícios, onde a quarta coluna seria cpf e a senha é "1234".


```
modelBuilder.Entity<Usuario>().HasData(
    new Usuario(2, "000002", "Teste2", "12345678902", "email@gmail.com", "1234", 100, DateTime.Now, null),
    new Usuario(3, "000003", "Teste3", "12345678903", "email@gmail.com", "1234", 100, DateTime.Now, null),
    new Usuario(4, "000004", "Teste4", "12345678904", "email@gmail.com", "1234", 100, DateTime.Now, null),
    new Usuario(5, "000005", "Teste5", "12345678905", "email@gmail.com", "1234", 100, DateTime.Now, null),
    new Usuario(6, "000006", "Teste6", "12345678906", "email@gmail.com", "1234", 100, DateTime.Now, null),
    new Usuario(7, "000007", "Teste7", "12345678907", "email@gmail.com", "1234", 100, DateTime.Now, null),
    new Usuario(8, "000008", "Teste8", "12345678908", "email@gmail.com", "1234", 100, DateTime.Now, null),
    new Usuario(9, "000009", "Teste9", "12345678909", "email@gmail.com", "1234", 100, DateTime.Now, null)
);
```

TORO-002 - Eu, como investidor, gostaria de visualizar meu saldo, meus investimentos e meu patrimônio total na Toro.

- Restrições:
 - Patrimônio do usuário deve conter as seguintes informações
 - Saldo atualmente em conta corrente
 - Lista de ativos (ações) pertencentes ao usuário, com quantidade de cada ativo e valor individual atual de cada um. (Ex: 10 ações PETR4, valor individual R\$25,00)
 - Patrimônio sumarizado (Saldo + Valor totalizado dos ativos)

FEITO:

 IP: 127.0.0.1



Correntista: sergio farias || CPF: 06952688722

Saldo atual conta corrente: R\$83.857

Meus Ativos

Descrição	Quantidade	Preço
PETR4	2	28.44
SANB11	3	40.77

Saldo consolidado: R\$368.047

Duplo click na lista para comprar um dos 5 títulos mais negociados nos últimos 7 dias.

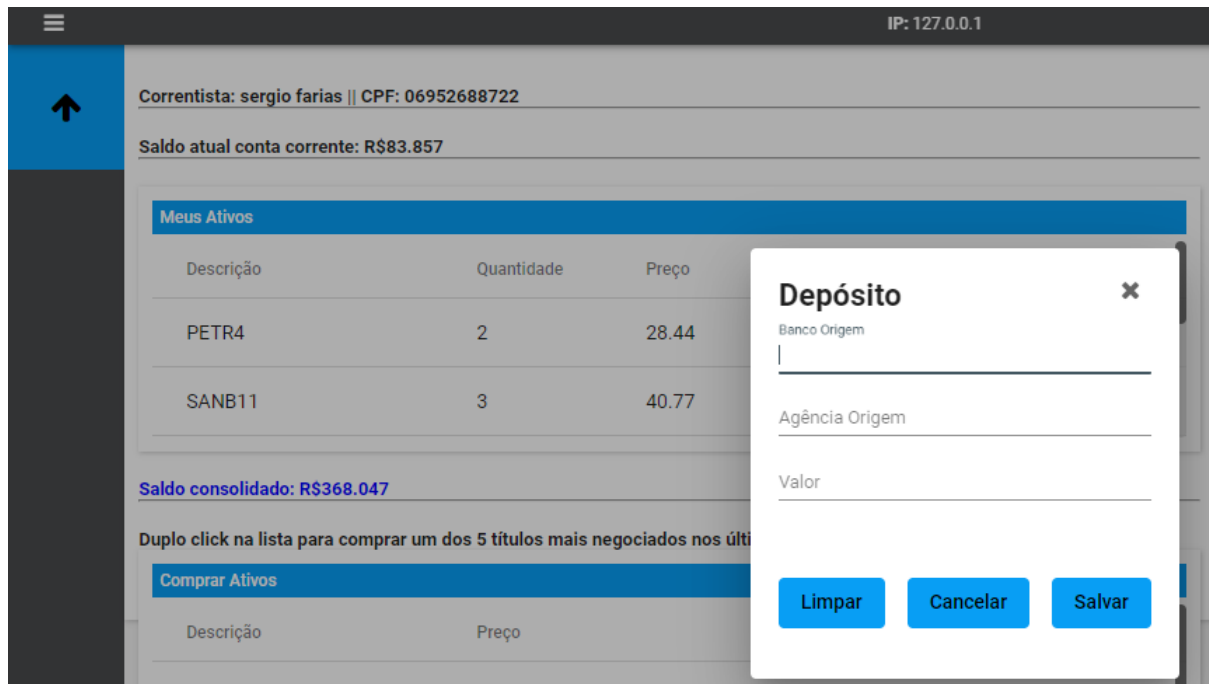
Comprar Ativos

Descrição	Preço
AZUL4	22.23
BBDC3	17.5

TORO-003 - Eu, como investidor, gostaria de poder depositar um valor na minha conta Toro, através de PiX ou TED bancária, para que eu possa realizar investimentos.

FEITO:

Ao clicar na seta à esquerda da tela, abre para fazer o depósito.



- Restrições:
 - A Toro já participa do Sistema Brasileiro de Pagamentos (SPB) do Banco Central, e está integrado a ele. Isto significa que a Toro tem um número de banco (352), cada cliente tem um número único de conta na Toro, e que toda transferência entre bancos passa pelo SBP do Banco Central, e quando a transferência é identificada como tendo o destino o banco Toro (352), uma requisição HTTP é enviada pelo Banco Central notificando tal evento. O formato desta notificação segue o padrão REST + JSON a seguir (hipotético para efeito de simplificação do desafio):

```
POST <apiBaseUrl>/spb/events
```

```
{
  "event": "TRANSFER",
  "target": {
    "bank": "352", // Banco Toro
    "branch": "0001", // Única agenda, sempre 0001
    "account": "300123", // Conta do usuário na Toro (unica por usuário)
  },
  "origin": {
    "bank": "033", // Banco de origem
    "branch": "03312", // Agencia de origem
  }
}
```

```
        "cpf": "45358996060" // CPF do remetente
    },
    "amount": 1000, // R$ 1000,00 reais
```

Ao fazer o depósito criei um objeto dentro do solicitado onde poderíamos esta enviando para o Sistema Brasileiro de Pagamento.

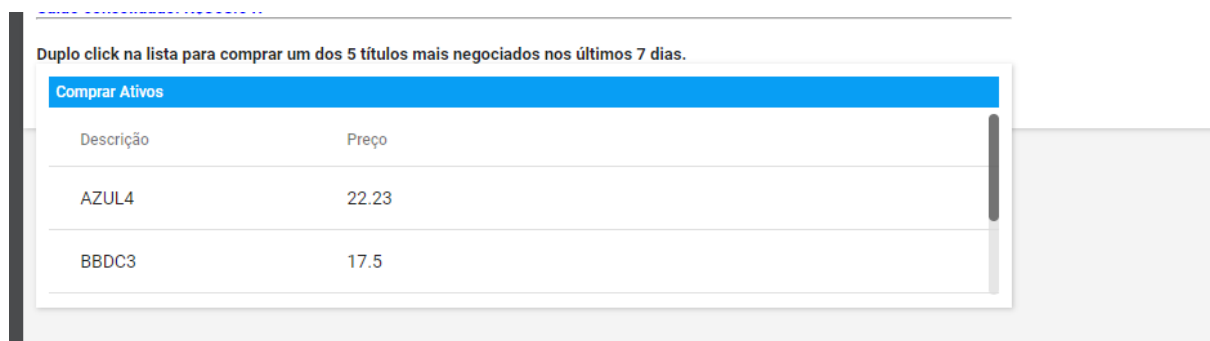
```
public class MovimentoViewModel
{
    public string evento { get; set; }
    public MovimentoTargetViewModel target { get; set; }
    public MovimentoOriginViewModel origin { get; set; }
    public decimal amount { get; set; }
    public int usuarioid { get; set; }
}
```

- - Outra restrição é que a origem da transferência deve sempre ser do mesmo CPF do usuário na Toro.

Sim, sempre será do usuário Toro pois só faz transferência quem estiver logado...

TORO-004 - Eu, como investidor, gostaria de ter acesso a uma lista de 5 ações mais negociadas nos últimos 7 dias, com seus respectivos preços, para que eu possa escolher uma delas e comprar a quantidade que eu escolher, respeitando o limite de saldo disponível na minha conta Toro, para que assim eu consiga possa montar minha carteira de investimentos.

FEITO:



- Restrições:
 - Para efeito de simplificação do desafio, as 5 ações mais negociadas nos últimos 7 dias e seus respectivos preços não precisa ser "real",

pode ser definida utilizando algum recurso pre-definido no backend (uma coleção predefinida no banco de dados ou arquivo JSON).

ao criar o BD faça uma carga simulando os ativos vendidos:

```
modelBuilder.Entity<UsuarioAtivo>().HasData(  
    new UsuarioAtivo(3, 1, 2, 3, null, null, System.Convert.ToDateTime("19/03/2022")),  
    new UsuarioAtivo(4, 3, 2, 2, null, null, System.Convert.ToDateTime("20/03/2022")),  
    new UsuarioAtivo(5, 1, 3, 3, null, null, System.Convert.ToDateTime("23/03/2022")),  
    new UsuarioAtivo(6, 3, 3, 2, null, null, System.Convert.ToDateTime("23/03/2022")),  
    new UsuarioAtivo(7, 5, 4, 3, null, null, System.Convert.ToDateTime("23/03/2022")),  
    new UsuarioAtivo(8, 6, 4, 2, null, null, System.Convert.ToDateTime("24/03/2022")),  
    new UsuarioAtivo(9, 7, 5, 3, null, null, System.Convert.ToDateTime("24/03/2022")),  
    new UsuarioAtivo(10, 3, 5, 2, null, null, System.Convert.ToDateTime("22/03/2022")),  
    new UsuarioAtivo(11, 5, 6, 3, null, null, System.Convert.ToDateTime("22/03/2022")),  
    new UsuarioAtivo(12, 6, 6, 2, null, null, System.Convert.ToDateTime("23/03/2022")),  
    new UsuarioAtivo(13, 7, 7, 3, null, null, System.Convert.ToDateTime("24/03/2022")),  
    new UsuarioAtivo(14, 8, 7, 2, null, null, System.Convert.ToDateTime("24/03/2022")),  
    new UsuarioAtivo(15, 7, 8, 3, null, null, System.Convert.ToDateTime("24/03/2022")),  
    new UsuarioAtivo(16, 8, 8, 2, null, null, System.Convert.ToDateTime("22/03/2022")),  
    new UsuarioAtivo(17, 8, 9, 2, null, null, DateTime.Now),  
    new UsuarioAtivo(18, 7, 9, 3, null, null, DateTime.Now),  
    new UsuarioAtivo(19, 8, 9, 2, null, null, DateTime.Now)  
);
```

Entregue por:

```
public class TrendsViewModel
```

```
{  
    public string symbol { get; set; }  
    public decimal currentPrice { get; set; }  
}
```

Etapa:

O Desafio Técnico da Toro Investimentos deve ser feito no tempo que você precisar.

Você está vivendo um dia a dia real nosso, as histórias acima são inspiradas em histórias reais elaboradas por nossos Product Managers (PM ou PO). Você, no papel de Time de Desenvolvimento, tem a liberdade para propor, discutir e implementar da melhor forma possível, buscando entregar o maior valor ao usuário no menor tempo, sem comprometer os requisitos, inclusive de qualidade.

O desafio consiste em você escolher uma das histórias acima e implementá-la. Prepare seu ambiente, crie um novo projeto, e implemente uma das USs. Este é o momento para considerar:

- Qual padrão arquitetural será adotado MVC? Clean Architecture? APIs Restful?
- Como irei automatizar os testes? Em quais níveis irei implementar testes? Unitário? Integração? E2E?
- Adotarei algum framework? Quais?

Então é isso. Escolha uma as 4 Histórias de usuário acima e Mãos à obra! Te esperamos na segunda etapa!

Iremos também fazer algumas perguntas sobre algumas das decisões tomadas por você em cima do seu projeto. Enfim, iremos conhecer melhor seu skill técnico.

1 - Tentei fazer 4 histórias.

2 - Padrão arquitetural adotado foi CQRS entregando APIs para consumo do front, fiz em CQRS devido achar mais organizado no quesito de termos pesquisas separadas de comandos no BD. Usei como exemplo o projeto Equinox do Eduardo Pires.

3 - No Front usei angular 9. fui “obrigado” usar esta versão pois é a versão da máquina do trabalho, não atualizamos para não ter risco de impacto.

4 - Usei entity framework com migrations.

Add-Migration InitialCreate -c EquinoxContext

Update-Database -c EquinoxContext

Remove-Migration -c EquinoxContext

5 - Usei IOC.

6 - Não usei testes.

7 - Banco de dados usado foi SQL Server Express. Podendo ser mudado o caminho no arquivo “appsetting”: Services - Equinox.Services.Api

```
"ConnectionStrings": {  
  "DefaultConnection": "Server=localhost\\SQLEXPRESS;Database=Toro;Trusted_Connection=True;MultipleActiveResultSets=true"
```

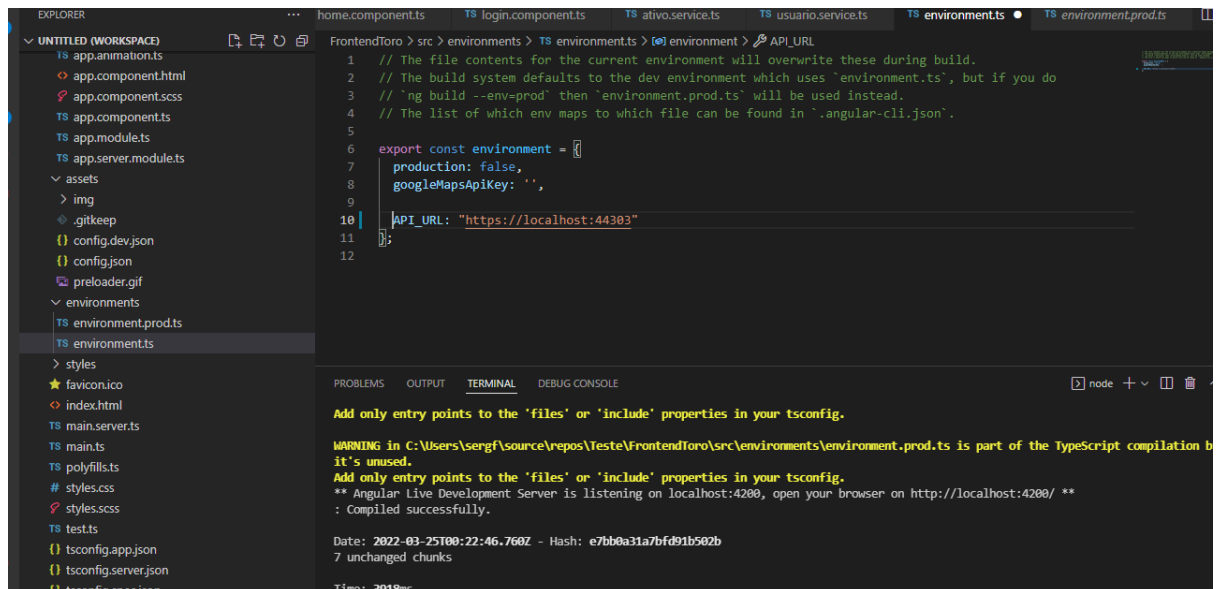
```
"ConnectionStrings": {
```

```
  "DefaultConnection":
```

```
"Server=localhost\\SQLEXPRESS;Database=Toro;Trusted_Connection=True;MultipleActive  
ResultSets=true"
```

```
}
```

8 - Para mudar o caminho do front para conectar ao back. Vá
FrontendToro\\src\\environments\\environment.ts



9 - Abaixo a arquitetura.

