DEVinHouse

Módulo 1 - Projeto Avaliativo 2

SUMÁRIO

1 INTRODUÇÃO

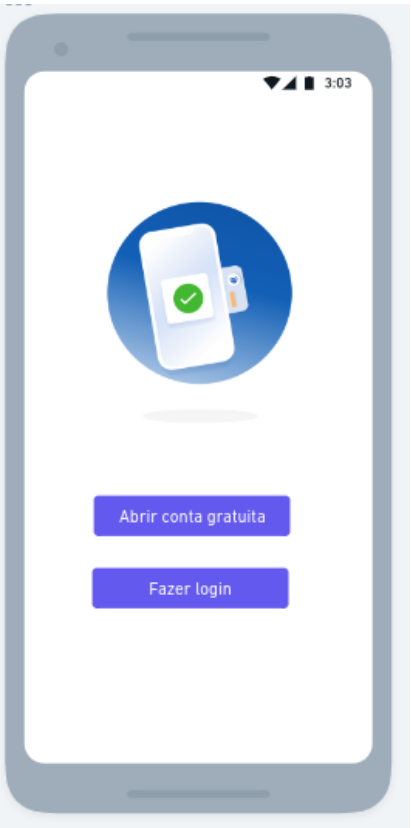
A **PAY PAGAMENTOS,** empresa líder no segmento Fintech, está com um projeto novo intitulado **PAY**, um aplicativo audacioso focado no pagamento de boletos e ganho de cashback. Com esse novo app, os usuários poderão pagar seus boletos sem sair de casa e além de acumular um bom valor de cashback($). Com os pay points adquiridos(cashback), o usuário poderá trocar por cupons e serviços disponibilizados pelos mais variados estabelecimentos.

Para desenvolver o projeto, você irá utilizar programação mobile utilizando **React Native** e uma Fake API com json-server.

2 REQUISITOS E ROTEIRO DA APLICAÇÃO

A aplicação que deverá ser realizada **individualmente** deve contemplar os seguintes requisitos:

Requisito 1 - Tela inicial - 0,5 ponto
Uma tela inicial contendo **uma animação** relacionada ao tema do projeto, um **botão de abrir conta gratuita** e um **botão de fazer login**.



Material auxiliar: https://lottiefiles.com/search?q=scan&category=animations
Docs Lottie: https://docs.expo.dev/versions/latest/sdk/lottie/

Requisito 2 - Tela de login - 1 ponto
Uma tela de login contendo **uma logo**, um **input de CPF,** um **input de senha**, um **botão(logar)** e um **elemento textual(abrir conta gratuita)**. Ao clicar no botão de logar, validar os dados e realizar uma requisição GET para o endpoint **/users** (somente se os dados forem validados). **CPF: (Aplicar propriedades de acessibilidade)** Preenchimento: Obrigatório Tipo de campo: Input **Password (Aplicar propriedades de acessibilidade)** Preenchimento: Obrigatório Tipo de campo: Input/Senha

Material auxiliar: https://reactnative.dev/docs/textinput

Requisito 3 - Tela de cadastro (etapa dados pessoais) - 1 ponto

Uma tela de cadastro de nova conta contendo um input de nome completo, telefone, e-mail, número do RG, CPF e senha. Ao clicar no **botão voltar:** retornar para a tela inicial do app. Ao clicar no **botão continuar**, avançar para a tela de endereço do usuário. Não permitir avançar para a **próxima tela** até todos os dados forem validados conforme especificado.

Nome completo: (Aplicar propriedades de acessibilidade)

Preenchimento: Obrigatório

Tipo de campo: Input

Mínimo de caracteres: 8

Máximo de caracteres: 120

Telefone (Aplicar propriedades de acessibilidade)

Preenchimento: Obrigatório

Tipo de campo: Input

Email (Aplicar propriedades de acessibilidade)

Preenchimento: Obrigatório

Tipo de campo: Input

Nº do RG (Aplicar propriedades de acessibilidade)

Preenchimento: Obrigatório

Tipo de campo: Input

CPF: (Aplicar propriedades de acessibilidade)

Preenchimento: Obrigatório

Tipo de campo: Input

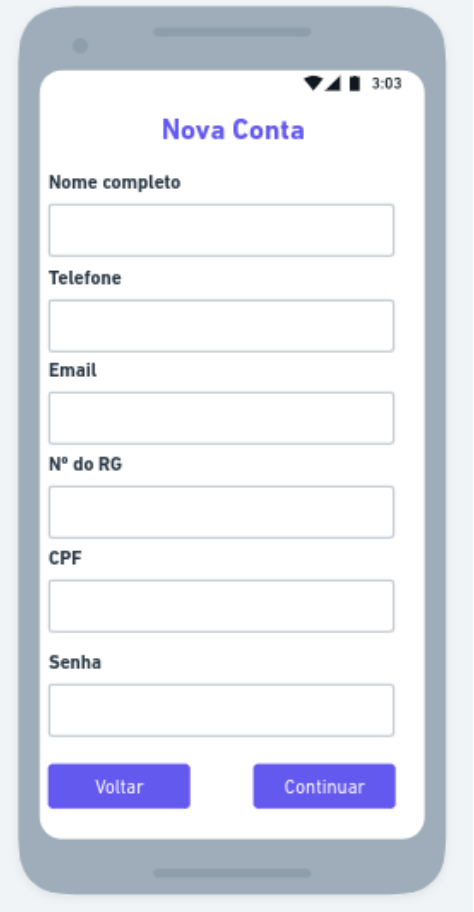
Password (Aplicar propriedades de acessibilidade)

Preenchimento: Obrigatório

Tipo de campo: Senha

Mínimo de caracteres: 8

Máximo de caracteres: 16



Requisito 4 - Tela de cadastro (etapa de endereço) - 1 ponto
Uma tela de cadastro de endereço contendo um input cep, rua, cidade, estado, bairro, número da residência e complemento. Ao clicar no **botão voltar:** retornar para a tela anterior do app. Ao clicar no **botão continuar**, avançar para a tela de data da cobrança do usuário. Não permitir avançar para a **próxima tela** até todos os dados forem validados conforme especificado. Além disso, buscar os dados de CEP da API do via cep com auxílio de um useEffect(). Após encontrar o cep, atualize o estado com as informações de **logradouro, complemento, localidade, bairro e uf. (Opcional)** ViaCep: https://viacep.com.br/ **CEP: (Aplicar propriedades de acessibilidade)**

Preenchimento: Obrigatório
Tipo de campo: Input

Rua: (Aplicar propriedades de acessibilidade)
Preenchimento: Obrigatório
Tipo de campo: Input

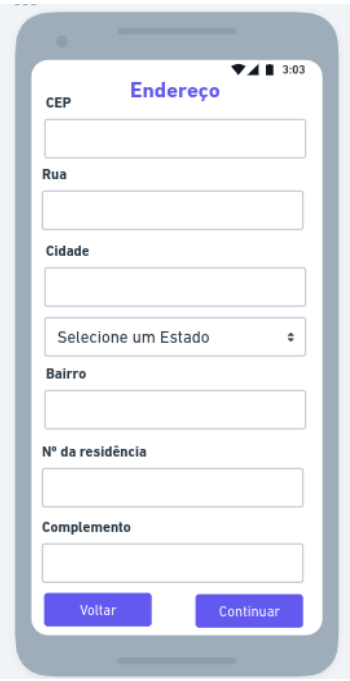
Cidade: (Aplicar propriedades de acessibilidade)
Preenchimento: Obrigatório
Tipo de campo: Input

Estado (Aplicar propriedades de acessibilidade)
Preenchimento: Obrigatório
Tipo de campo: Picker
Pré-cadastrado: Todos os Estados do Brasil

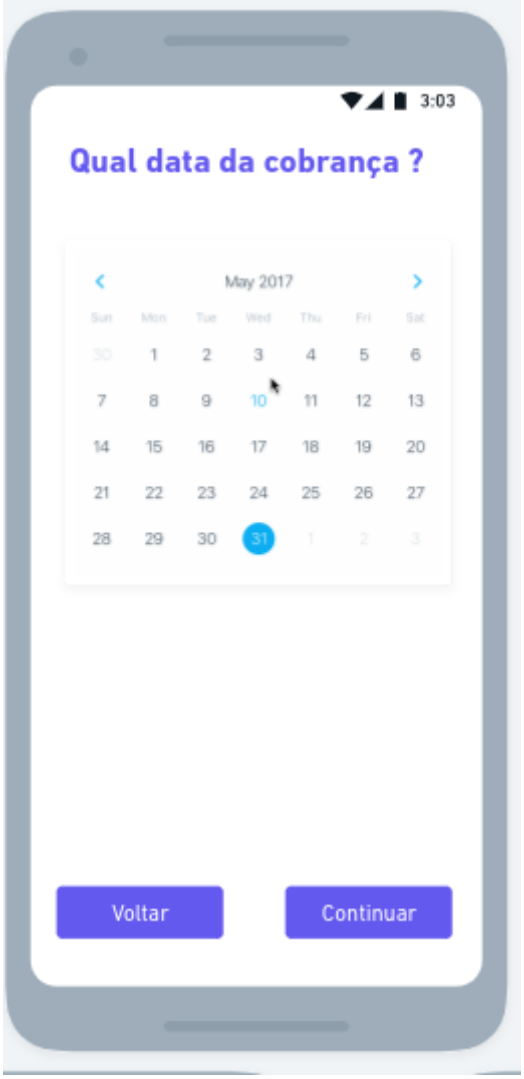
Bairro (Aplicar propriedades de acessibilidade)
Preenchimento: Obrigatório
Tipo de campo: Input

Nº da residência (Aplicar propriedades de acessibilidade)
Preenchimento: Obrigatório
Tipo de campo: Input

Complemento (Aplicar propriedades de acessibilidade)
Preenchimento: Opcional
Tipo de campo: Input



Requisito 5 - Tela de data da cobrança - 1 ponto
Uma tela de data da cobrança com um componente de calendário permitindo selecionar a data da primeira cobrança. Ao clicar no **botão voltar:** retornar para a tela anterior do app. Ao clicar no **botão continuar**, avançar para a tela de **termos de usos**. Não permitir avançar para a **próxima tela** até todos os dados forem validados conforme especificado. **Data da cobrança: (Aplicar propriedades de acessibilidade)** Preenchimento: Obrigatório Tipo de campo: Calendar



Requisito 6 -Tela de termos de uso - 1 ponto
Uma tela de termos de uso contendo um scrollview com um texto “Lorem Ipsum” e um componente de switch com estado booleano. Não permite cadastrar o usuário, caso não aceite os termos de uso. Ao clicar no **botão voltar:** retornar para a anterior do app. Ao clicar no **botão finalizar**, realiza uma requisição do tipo **POST** enviando como um **objeto de usuário** no body da requisição.

BODY da requisição:

{

"**fullname**": "nome completo do usuário",

"**contact**: "Telefone do usuário",

"**email**": "Email do usuário",

"**number_rg**": "Nº do rg do usuário",

"**cpf**": "Cpf do usuário",

"**password**": "Senha do usuário",

"**address**": {

"**cep**": "Cep do usuário",

"**street**": "Rua do usuário"

"**city**": "Cidade do usuário",

"**state**: "Estado do usuário",

"**region**": "Bairro do usuário",

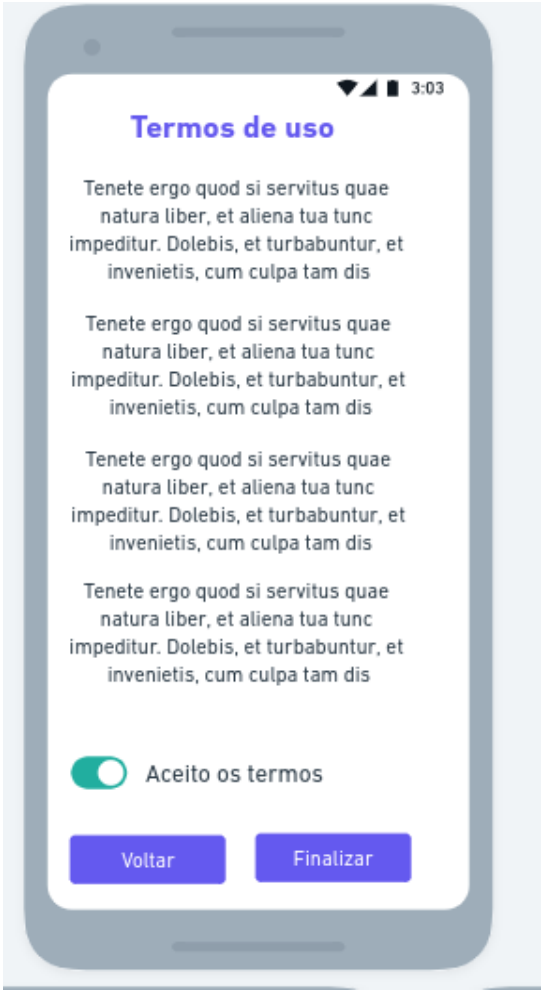
"**number**" "Número da casa do usuário",

“**complement**”: “Complemento do usuário”

},

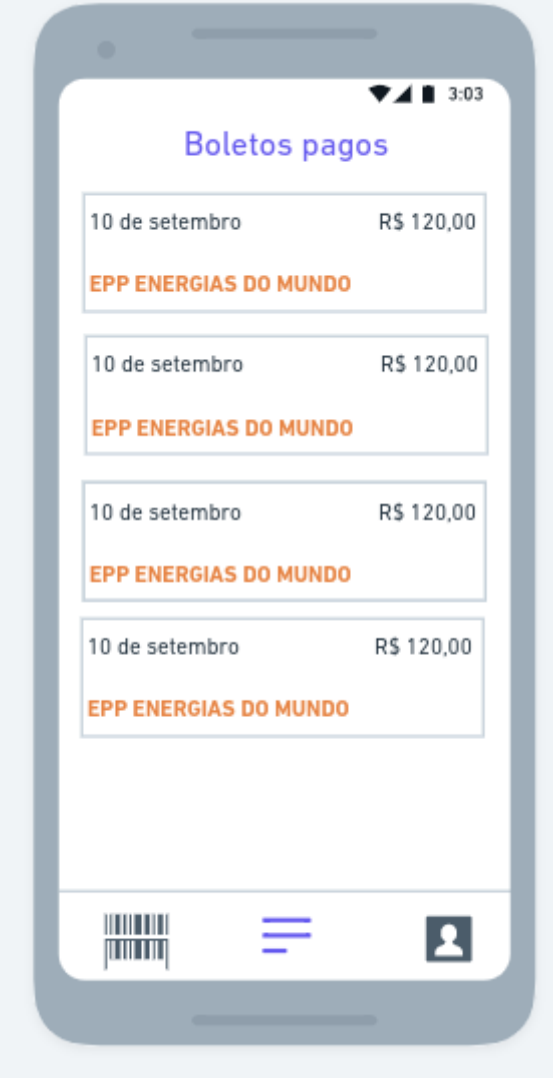
"**billing_day**": "Data da cobrança no formato ano-mes-dia"

}



Requisito 7 - Tela de dados da conta -1 ponto
Uma aba listando todos os dados do cliente(**somente textual**) e botão de **sair do app**. Busca os dados do cliente fazendo uma requisição do tipo **GET** enviando o id da conta do cliente como parâmetro. **Ex: {URL}/users/432**

Requisito 8 - Tela de boletos pagos - 1,0 ponto
Uma aba listando **todos os boletos pagos** pelo app. Ao entrar na tela, fazer uma requisição do tipo **GET** para **/invoices** listando todos os boletos pagos do **USUÁRIO LOGADO**. Cada Card contém o nome do recebedor do pagamento, o valor pago e a data do pagamento. Cada vez que a tab de **boleto pagos** for focada, fazer uma requisição para buscar a lista atualizada de boletos (Use o recurso isFocused do react-navigation).



Requisito 9 - Tela de pagamento de boleto - 1,5 pontos
Uma aba com **botão de scannear novo boleto**. Ao clicar sobre o botão, habilita o componente BarCodeScanner. Ao **identificar um boleto**, redirecionar o usuário para a tela de Detalhes do boleto. Além disso, implemente o pedido de permissão de câmera para usuário. Para esse projeto, utilize o tipo de boleto **code39.** Exibir no topo da tela o nome do usuário logado. Ao sair da Tab de scanner, desmontar o componente de BarCodeScanner.

Requisito 10 - Tela de detalhes do boleto - 1,0 pontos

Uma tela de detalhes do boleto listando o destinatário, o valor do boleto, código do boleto e valor de cashback (10%). Ao clicar no **botão pagar**, realize uma requisição do tipo **POST** para o endpoint **/invoices** enviando como Body o seguinte objeto.
{
“recipient”: “Nome do recebedor do pagamento”,
“amount”: “Valor do boleto”
“date: “Data do pagamento realizado no formato dia/mes/ano horas:minutos”
“code”: “O código do boleto”,
“user_id”: “ID do usuário que fez o pagamento”,
“cashback” “Valor de 10% do valor do boleto”
}

Modelos de boletos:

1234534

Contas pré-cadastradas:

```
"debts": [
   {
     "id": 1234534,
     "amount": 230.23,
     "recipient": "EPP Energias do Mundo"
   },
   {
     "id": 545434326,
     "amount": 30.23,
     "recipient": "DevInHouse Ltda"
   }
 ]
```

4 CRITÉRIOS DE AVALIAÇÃO

A tabela abaixo apresenta os critérios que serão avaliados durante a correção do projeto. O mesmo possui variação de nota de 0 (zero) a 10 (dez) como nota mínima e máxima, e possui peso de **45% sobre a avaliação do módulo**.

Serão **desconsiderados e atribuída a nota 0 (zero)** os projetos que apresentarem plágio de soluções encontradas na internet ou de outros colegas. Lembre-se: Você está livre para utilizar outras soluções como base, mas **não é permitida** a cópia.

Nº	Critério de Avaliação	0	0,15 a 0,25	0,35 a 0,5
1	O aluno desenvolveu a tela inicial ?	O aluno não desenvolveu a tela inicial.	O aluno inseriu a animação e os botões, porém os botões não navegavam para as referidas telas.	O aluno inseriu todos elementos visuais da tela e implementou a navegação ao clicar nos botões. Além disso, desenvolveu um layout agradável com reuso de código e seguindo as boas práticas de programação.
Nº	**Critério de Avaliação**	**0**	**0,15 a 0,5**	**0,65 a 1**
2	O aluno desenvolveu a tela de login ?	O aluno não desenvolveu a tela de login.	O aluno desenvolveu a tela de login visualmente, porém não validou os campos de input e não redirecionou corretamente o usuário.	O aluno desenvolveu a tela de login, validou os campos de input e redirecionou corretamente o usuário. Além disso, desenvolveu um layout agradável com reuso de código e seguindo as boas práticas de programação.
3	O aluno desenvolveu a tela de cadastro de usuário (etapa dados pessoais) ?	O aluno não desenvolveu a tela de cadastro de usuário (etapa dados pessoais).	O aluno inseriu os campos de input, porém não validou o formulário conforme especificado e não navegou para a próxima tela ao clicar no botão continuar.	O aluno inseriu os campos de input corretamente, validou o formulário e navegou para a próxima tela corretamente. Além disso, desenvolveu um layout agradável com reuso de código e seguindo as boas práticas de programação.
4	O aluno desenvolveu a tela de cadastro de usuário (etapa de endereço) ?	O aluno não desenvolveu a tela de cadastro de usuário (etapa de endereço).	O aluno inseriu os campos de input, porém não validou o formulário conforme especificado e não navegou para a próxima tela ao clicar no botão continuar.	O aluno inseriu os campos de input corretamente, validou o formulário e navegou para a próxima tela corretamente. Além disso, desenvolveu um layout agradável

				com reuso de código e seguindo as boas práticas de programação.
5	O aluno desenvolveu a tela de cadastro de usuário (etapa da data da cobrança) ?	O aluno não desenvolveu a tela de cadastro de usuário (etapa da data da cobrança) .	O aluno inseriu o calendário, porém não gerenciou o estado corretamente e não navegou para a próxima tela ao clicar no botão de continuar.	O aluno inseriu o calendário com o estado corretamente e navegou para a próxima tela ao clicar no botão de continuar. Além disso, desenvolveu um layout agradável com reuso de código e seguindo as boas práticas de programação.
6	O aluno desenvolveu a tela de cadastro de usuário (etapa de termos de uso)?	O aluno não desenvolveu a tela de cadastro de usuário (etapa de termos de uso)?	O aluno desenvolveu os elementos textuais e o switch na tela, porém não implementou a chamada POST conforme especificado.	O aluno desenvolveu todos os elementos da tela de termos de uso e além disso tratou a requisição de cadastro do usuário e enviou o body conforme especificado. Além disso, desenvolveu um layout agradável com reuso de código e seguindo as boas práticas de programação.
7	O aluno desenvolveu a aba de dados da conta do usuário ?	O aluno não desenvolveu a aba de dados da conta do usuário .	O aluno inseriu as informações, porém não buscou os dados da API usando uma requisição GET.	O aluno inseriu as informações dinamicamente da API e todos elementos visuais da tela. Além disso, desenvolveu um layout agradável com reuso de código e seguindo as boas práticas de programação.
8	O aluno desenvolveu a aba de boletos pagos ?	O aluno não desenvolveu a aba de boletos pagos	O aluno listou cada boleto pago conforme o mockup,	O aluno listou os boletos pagos dinamicamente com a

			porém não buscou os dados pela API.	API e listou os cartões conforme especificado. Além disso, desenvolveu um layout agradável com reuso de código e seguindo as boas práticas de programação.
Nº	**Critério de Avaliação**	**0**	**0,15 a 0,75**	**0,85 a 1,5**
9	O aluno desenvolveu a aba de Scannear boletos ?	O aluno desenvolveu a aba de Scannear boletos ?	O aluno inseriu o componente de câmera, porém não pediu permissão ao usuário e não implementou a função de identificação do boleto.	O aluno implementou o componente de BarCodeScanner com todas suas variações de estado e a função de identificação do boleto. Além disso, desenvolveu um layout agradável com reuso de código e seguindo as boas práticas de programação.
Nº	**Critério de Avaliação**	**0**	**0,15 a 0,50**	**0,65 a 1**
10	aluno desenvolveu a tela de detalhes do boleto ?	O aluno não desenvolveu a tela de detalhes do boleto.	O aluno inseriu os dados do boleto scanneado na tela, porém não implementou a ação de pagar o boleto conforme especificado.	O aluno inseriu os dados do boleto scanneado e implementou o cadastro do boleto conforme especificado. Além disso, desenvolveu um layout agradável com reuso de código e seguindo as boas práticas de programação.
Nº	**Ponto extra**	**0**	**0,15 a 0,50**	**0,65 a 1**
11	O aluno fez alguma funcionalidade extra ?	O aluno não fez nenhuma funcionalidade extra.	O aluno fez a funcionalidade, porém a implementação foi muito simples ou não agregou valor significativo ao projeto.	O aluno desenvolveu uma funcionalidade que agregou valor ao projeto e uso do app.

5 ENTREGA

O código desenvolvido deverá ser submetido no **Bitbucket**, e o link deverá ser disponibilizado na tarefa **Módulo 1 - Projeto Avaliativo 2**, presente na semana 12 do AVA até o dia **25/09/2022** às **23h55**.

O repositório deverá ser privado, com as seguintes pessoas adicionadas:

- Fernando Puntel **- fernando.puntel@edu.sc.senai.br**
- Operação DEVinHouse - **devinhouse-operacao**
- Henrique Douglas **- henrique.cavalcante@edu.sc.senai.br**

Não serão aceitos projetos submetidos após a data limite da atividade, e, ou alterados depois de entregues.

Importante:

1. Será considerado como data final de entrega a **última atualização** no repositório do projeto no Bitbucket. Lembre-se de não modificar o código até receber sua nota.
2. **Não esqueça de submeter submeter o link no AVA.**

6 PLANO DE PROJETO

Ao construir a aplicação proposta, o aluno estará colocando em prática os aprendizados em:

- **Componentes react-native**: Text, View, SafeAreaView, ScrollView, TextInput, Switch, Button, TouchableOpacity, estados, useState, useEffect

- **Navegação:** Navigator Container, Stack Navigator, Tab Navigator

- **Consumo de apis:** fetch, método POST, método GET, método DELETE, método PATCH, método PUT, body, headers, query params

- **Componentes expo**: BarCodeScanner, Lottie, StatusBar, Picker, Calendar