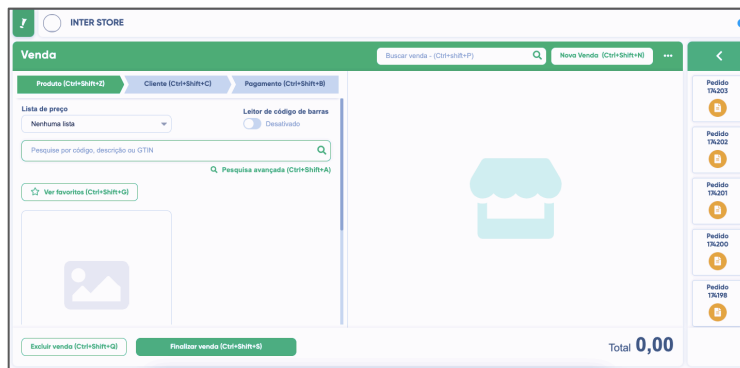


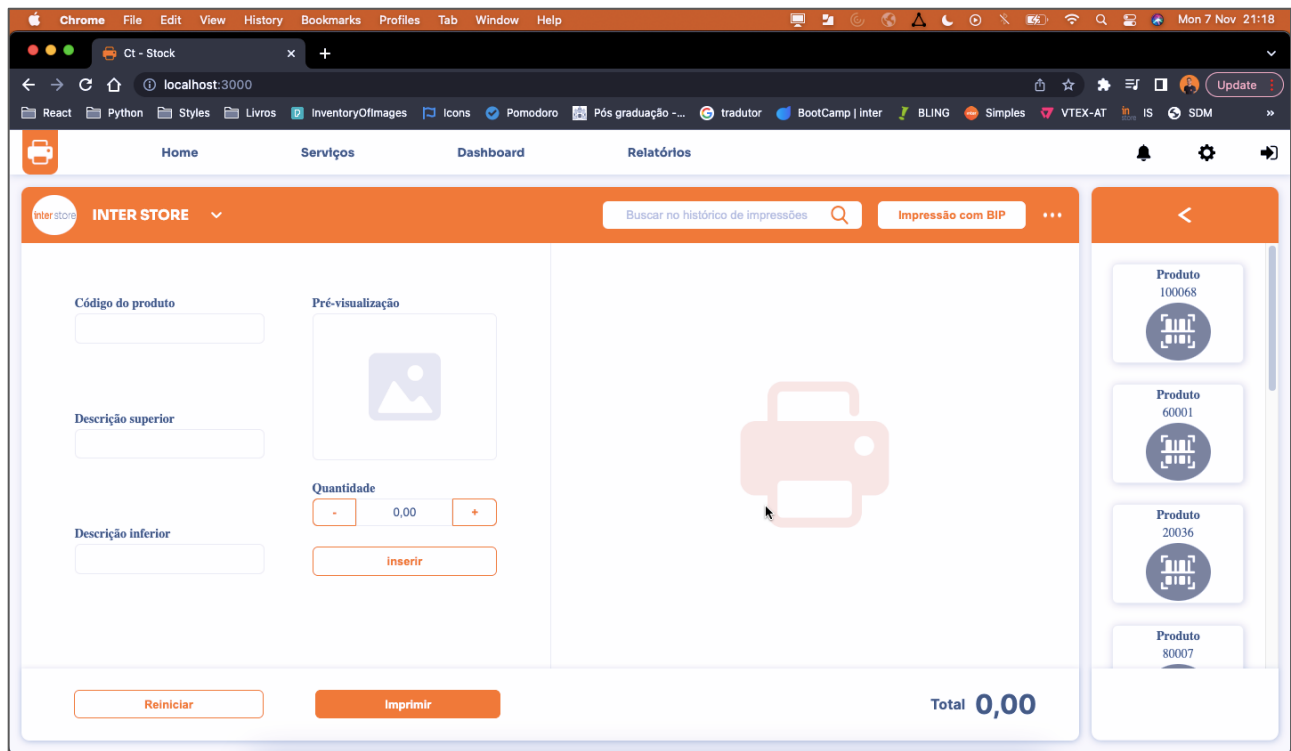
# Tiago Hebert Paz da Silva

## Sistema 1: CT – Stock

- **Funcionalidade:** Sistema de gestão de cadastro de etiquetas integrado com impressora Zebra e leitor de código de barras.
- **Necessidade do usuário:**
  - Fazer a verificação dos produtos via código de barras.
  - Ter uma etiqueta com os detalhes do produto.
  - Ter um sistema parecido com o ERP já usado no dia a dia Bling (print abaixo).



- **Especificação técnica:** Front-end em **React**, back-end em **Node.js**, usando o **mongoDB** e layout da etiqueta na linguagem de configuração **Zebra**.
- **Tela inicial:**



- Após o usuário digitar o código do produto e inserir a etiqueta, aparece uma **pré-visualização**: Foram utilizados **Hooks** usados para alterar: imagem de pré-visualização, componente do lado direito com a lista de etiquetas digitadas, quantidade total e histórico do lado direito.

The screenshot displays the 'INTER STORE' application interface. The top navigation bar includes links for Home, Serviços, Dashboard, and Relatórios. The main content area is divided into several sections:

- Form Section:** Contains input fields for 'Código do produto' (07112022), 'Descrição superior' (Pantufa InterPig), and 'Descrição inferior' (tamanho M). It also features a 'Pre-visualização' of a barcode and a 'Quantidade' input set to 3,00 with a 'Inserir' button.
- Table Section:** A table with columns 'Produto', 'Quantidade', and 'Ações'. It lists '07112022 - Pantufa InterPig -' with a quantity of 3,00.
- Sidebar:** A vertical list of product entries with icons, including 'Produto 07112022', 'Produto 60001', 'Produto 20036', and 'Produto 80007'.
- Footer:** Includes 'Reiniciar' and 'Imprimir' buttons, and a 'Total 3,00' display.

#### Ao clicar em imprimir o sistema:

- Envia os dados das etiquetas para o back-end, fazendo post em duas rotas: histórico de impressão (nome do usuário, data e hora, código da etiqueta e descrição) e na rota de cadastros únicos (configurada para receber **posts** e **puts**).
- Habilita o **get** e preenchimento da descrição após o usuário digitar o código.
- Com o botão de impressão por BIP ativado, o usuário consegue adicionar etiquetas no stage de impressão com somente um BIP do leitor de código onde é feito um **get** da descrição da etiqueta.
- Aciona a função fs do back-end que faz a leitura do arquivo TXT já com a linguagem zebra no layout da etiqueta. Modifica as variáveis para impressão local na impressora específica de etiquetas.

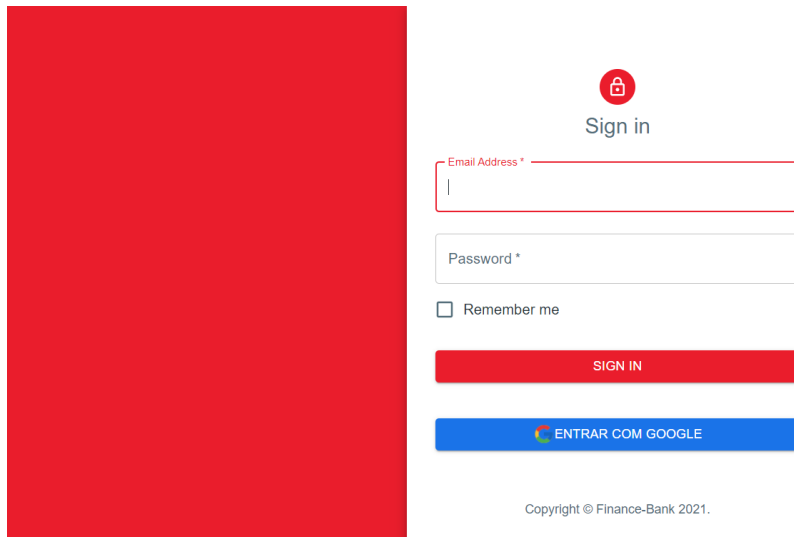
**Observação:** Por se tratar de uma empresa do setor bancário. Não estou compartilhando o código em link público.

## Sistema 2: Finance-Bank

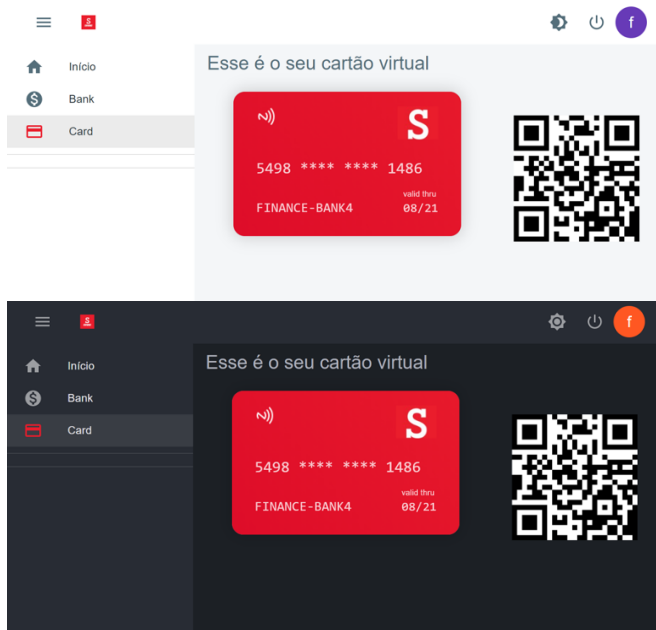
Código: <https://github.com/tiagopazhs/finance>

**Descrição:** Sistema criado para simular transações bancárias.  
Transações disponíveis: depósito, transferência e pagamento.

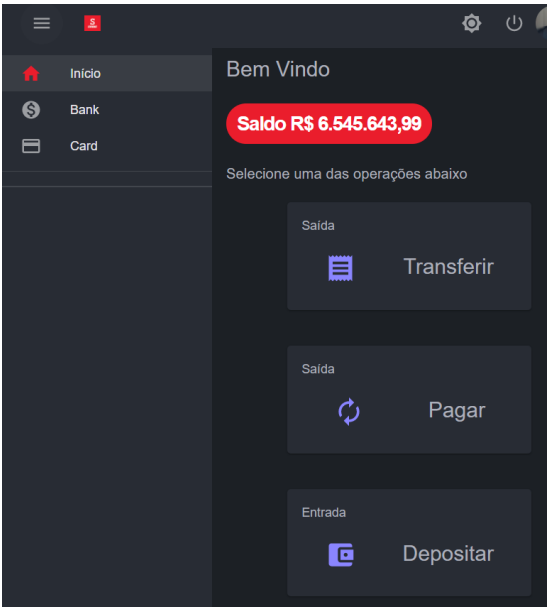
**Tela de login:** Autenticação com o google utilizando o firebase.



Dados do cartão modo normal e modo escuro:



Tela de início:



Tela de transações:

↔ Transações de Outros Usuários  
▼ Saídas da sua Conta  
▲ Entradas na sua Conta

Pesquisa Geral

Data	Transação	Tipo	Categoria	Destinatário	Remetente	Valor	Opções
	<div>Todos</div>	<div>Todos</div>	<div>Todos</div>	<div>BancoF</div>	<div>sense2</div>		
11/07/2021	Pagamento	↔	Conta de Luz	B	s	R\$ 4565,46	
11/07/2021	Pagamento	↔	Conta de Luz	B	s	R\$ 465465,44	
11/07/2021	Pagamento	↔	Pagamento Boleto	B	s	R\$ 4,55	
11/07/2021	Pagamento	↔	Conta de Luz	B	s	R\$ 654,65	
11/07/2021	Pagamento	↔	Pagamento Boleto	B	s	R\$ 87879,87	
11/07/2021	Pagamento	↔	Pagamento Boleto	B	s	R\$ 466,45	

## Sistema 3: Chat-Real-Time

Código: <https://github.com/tiagopazhs/chat-online>

**Descrição:** Implementa um sistema de Chat online via conexão real-time com web-socket utilizando React JS + Socket.IO.

**Tela com a versão de quem está enviando/ recebendo a mensagem:**

