

**Data:23/08/2023**

## **RELATÓRIO NO PAPEL DE DESENVOLVEDOR**

### **A. INTEGRANTES (apenas incluir aqueles que participaram ativamente da etapa)**

Nome Completo: Artur Ruiz de Souza  
E-mail: arturruizs@gmail.com

Nome Completo: Douglas Ardenghi Schlatter  
E-mail: douglas.ardenghi18@gmail.com

Nome Completo: Henry Ribeiro Piceni  
E-mail: henryribe18@gmail.com

Nome Completo: Luana Taynara da Costa Hahn  
E-mail: luanatc.hahn@gmail.com

### **B. RESUMO DO NEGÓCIO RECEBIDO DO GRUPO DE ANALISTAS**

RU++ é um sistema que busca trazer melhorias e novas funcionalidades para o já existente sistema de compras de tickets, para os usuários do restaurante universitário, buscando trazer mais dinâmica para a compra das refeições. A principal novidade é um sistema de compra compartilhada, onde os membros de um grupo pré-definido podem comprar e utilizar tíquetes em comum. Além disso, há também a possibilidade de transferir tíquetes pessoais para outras pessoas.

Buscando mais interação entre os administradores do RU e os membros da comunidade acadêmica, o sistema prevê uma funcionalidade de enquetes, onde o administrador poderá criar enquetes a respeito dos alimentos que serão servidos no restaurante aquele dia, o formato das enquetes é dinâmico e poderá ser alterado conforme o criador quiser. Os usuários do RU poderão votar em, por exemplo, quais saladas serão servidas no dia, qual opção de proteína preferem, etc.

### **C. ARQUITETURA ESCOLHIDA (INDICAR O PADRÃO ARQUITETURAL) E JUSTIFICATIVA PARA SUA ESCOLHA**

Conforme a tabela abaixo, a arquitetura mais apropriada para o desenvolvimento do software RU++ é a MVC.

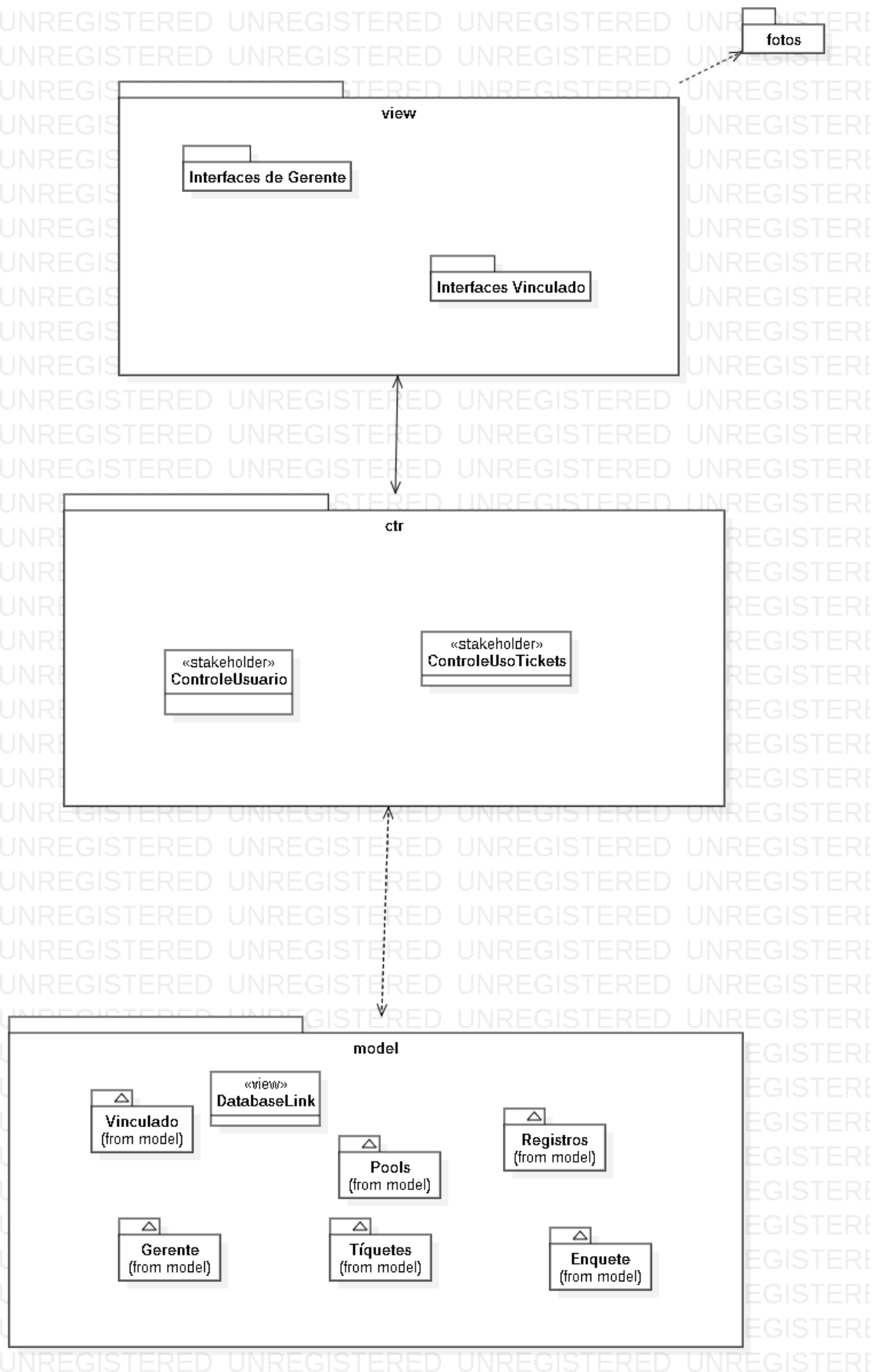
De acordo com os analistas do projeto, as características mais importantes a serem levadas em conta no desenvolvimento são: confiabilidade, disponibilidade, usabilidade, e robustez. Além disso, a arquitetura escolhida deve permitir uma implementação para Web.

A partir desses critérios, as arquiteturas MVC e Camadas estariam empatadas. Como critério de desempate, escolhemos o conhecimento técnico mais edificante para os desenvolvedores, ou seja, MVC, mais comum e aplicações do dia-a-dia.

Objetivo/Arquitetura	MVC	Cliente Servidor	Camadas	Black Board
Web	X		X	
Confiabilidade		X	X	
Disponibilidade	X	X		X
Usabilidade	X			
Robustez			X	
Conhecimento Técnico	X			

**Arquitetura não alterada desde a etapa 2.**

**D. DIAGRAMA DE PACOTES DA ARQUITETURA DO SISTEMA (SISTEMA NA ÍNTEGRA)**  
**(Explicar textualmente, além de incluir as imagens)**



Arquitetura Model, View, Control define três grandes pacotes do sistema.

Model: dentro do pacote Model localizam-se principalmente os bancos de dados do sistema. Estimamos que haverá pelo menos cinco tabelas no banco de dados desta aplicação, sendo uma de informações de usuários do RU, uma com informações sobre os tickets, outra com as informações do gerente do RU, uma com informações básicas de registros e por último a tabela de pools de tickets.

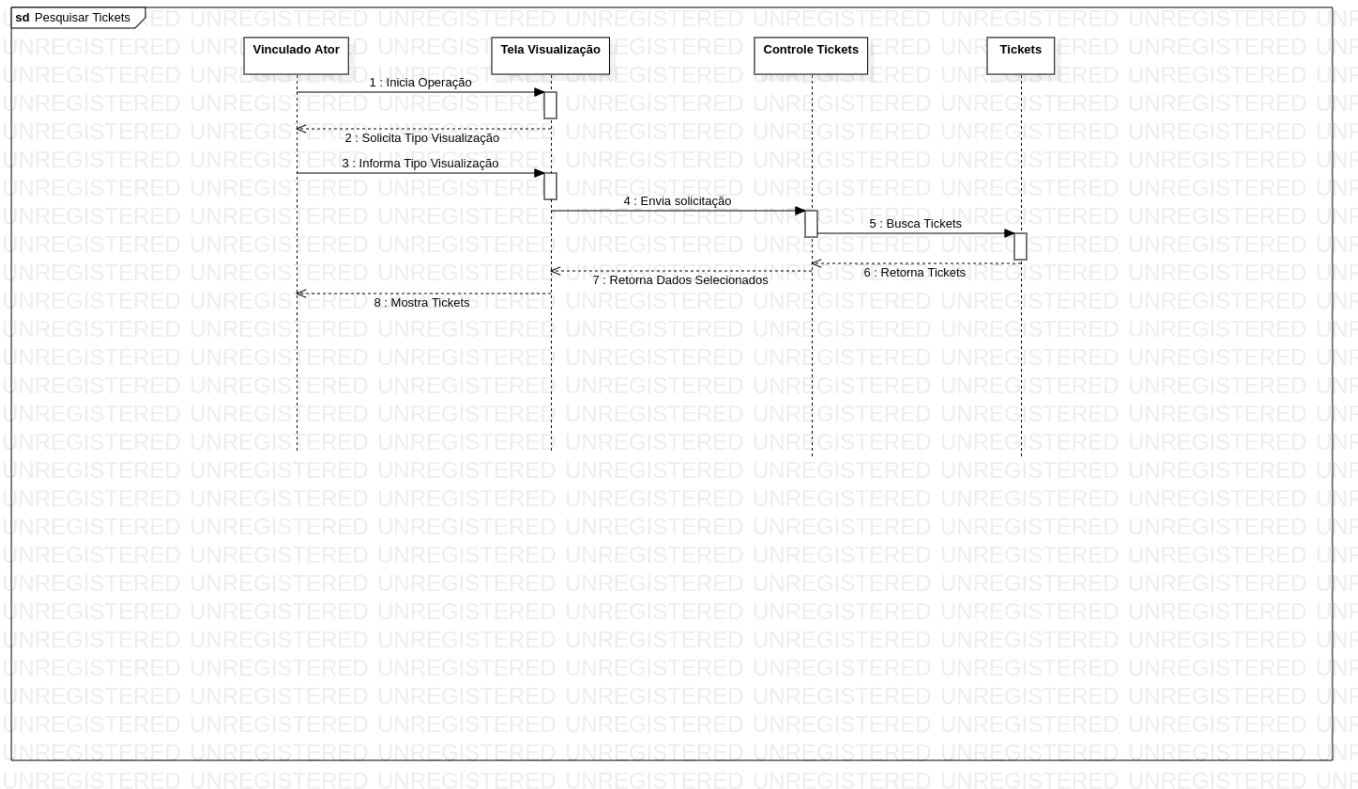
View: todas as interfaces que interagem diretamente com o usuário estão nesta pasta, telas de login, visualização de tickets, e etc. Tais interfaces estão divididas em dois grandes pacotes, que englobam a visualização completa dos Gerentes e dos Vinculados, permitindo uma fácil personalização dos dados dependendo do tipo de usuário que os acessa.

Control: divide-se em vários subsistemas (login, pesquisa, compra, etc) responsáveis por receber os comandos dos usuários nas interfaces, solicitar e converter os dados do Model e devolver para a View, a fim de serem exibidos para o usuário do sistema. Esses sistemas são englobados, principalmente, pelos casos de uso anteriormente definidos pelas especificações do software.

Na arquitetura MVC, o caso de uso *Visualizar Ticket*, por exemplo, funcionará da seguinte forma:

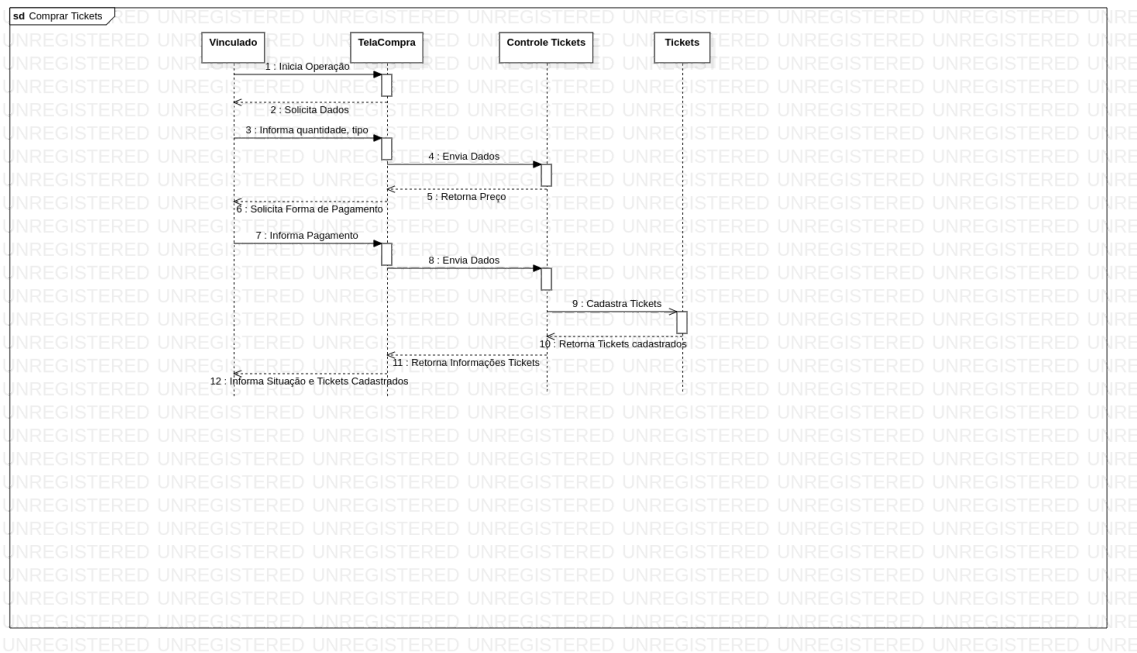
O Sistema de Visualização de Tickets, como um pacote de controle do sistema, acessa, no pacote Model, as informações referentes ao vinculado ativo e seus respectivos tickets, retornando para o grupo de interfaces *InterfacesVinculado*, do pacote view, mostrando as informações do usuário.

#### **E. DIAGRAMA DE INTERAÇÃO (Explicar textualmente, além de incluir as imagens)**



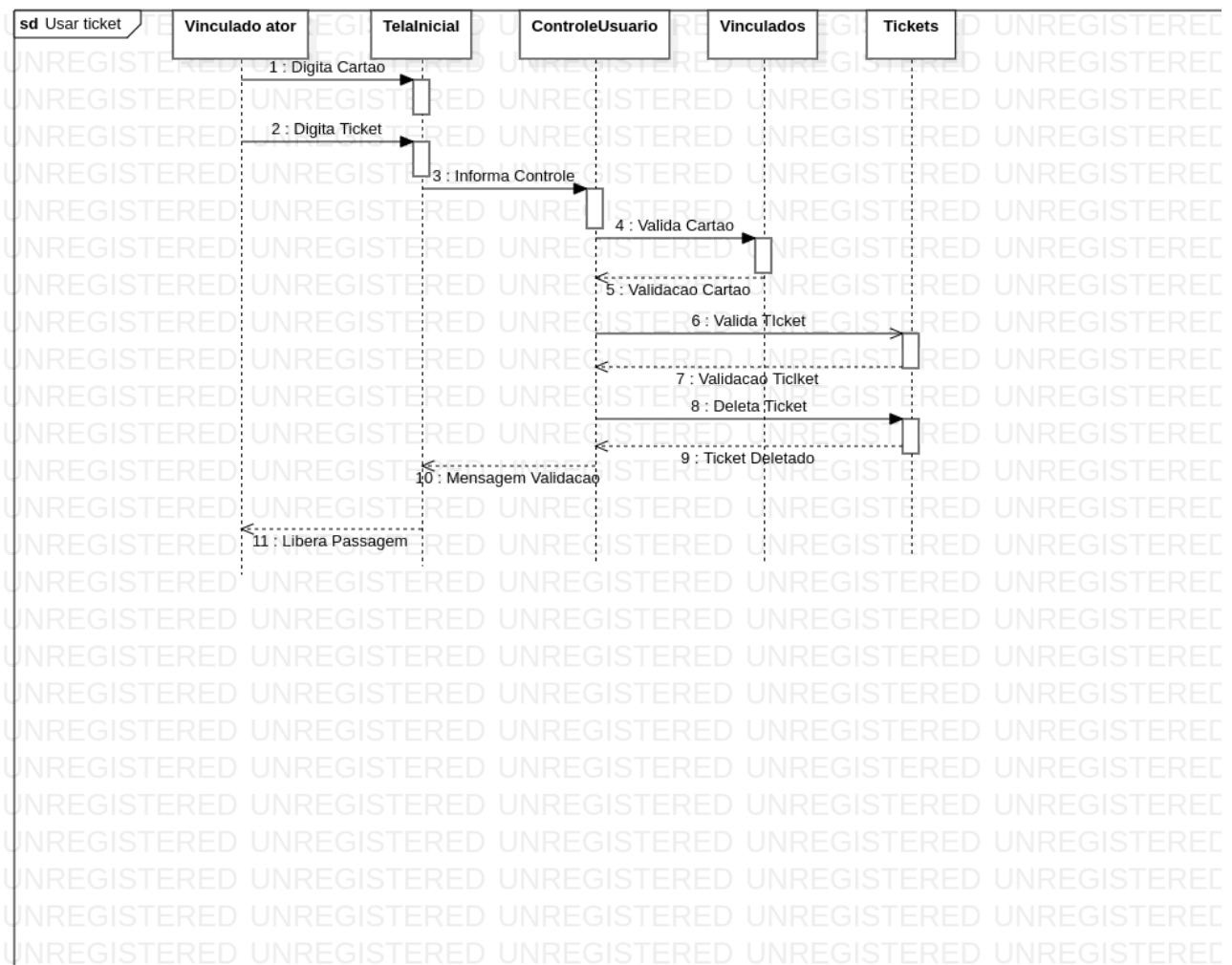
**Diagrama de sequência Pesquisar Tickets, na categoria pesquisar recursos. Representação da ordem de eventos quando o Vinculado seleciona essa opção.**

A interface solicita as informações por meio de um formulário, o vinculado informa o tipo de visualização que deseja à interface. Os dados obtidos na interface são enviados ao controle que busca os dados na tabela de tickets de acordo com o tipo de visualização selecionado. Após verificar a validade e disponibilidade de dados no banco, o controle retorna a resposta que será impressa na tela de visualização.



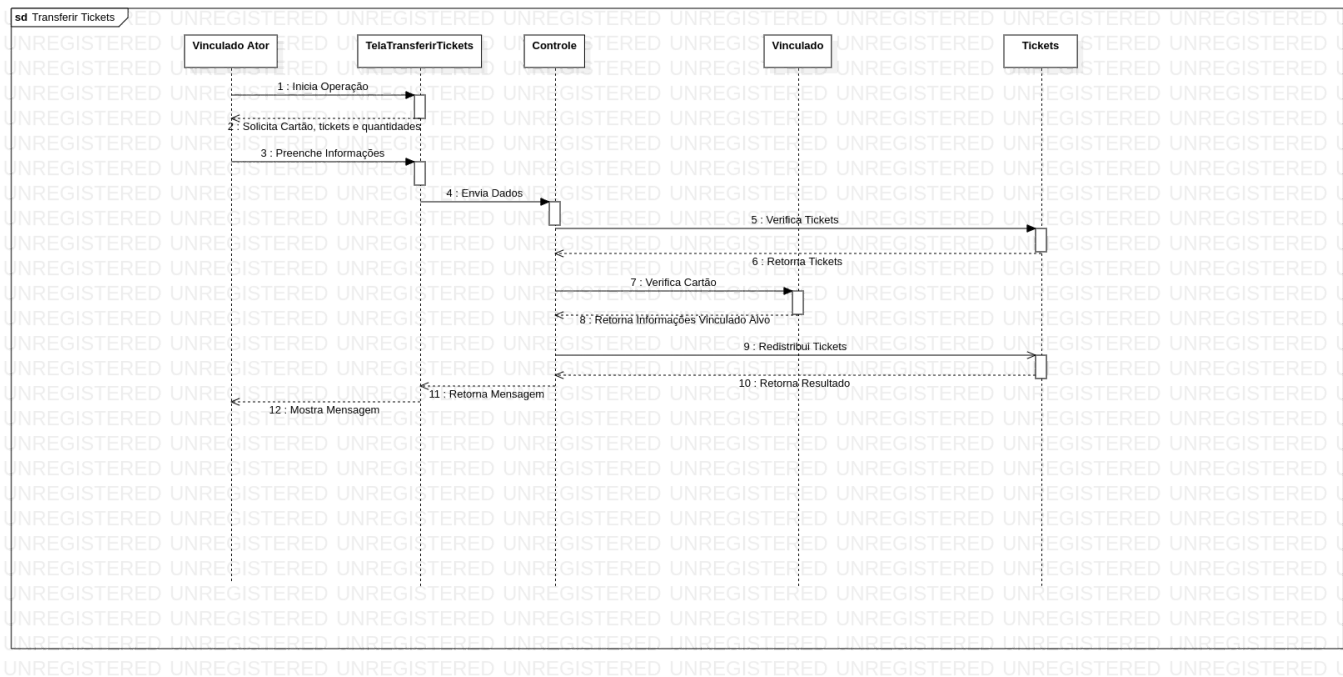
**Diagrama de sequência Comprar Tickets, na categoria “cadastrar” recursos.**  
**Representação da ordem de eventos quando o Vinculado seleciona esta opção.**

O sistema solicita ao usuário por meio de formulário, as informações para realizar a compra de tíquetes. O usuário envia estas informações para a interface. O sistema então envia estes dados para o controle de tíquetes, que retorna o valor solicitando a forma de pagamento. Após confirmar, o sistema envia os dados para o controle de tíquetes, que cadastra os tíquetes comprados no banco de dados do Vinculado. O usuário então recebe uma confirmação de que a compra foi bem sucedida e que os tíquetes já estão disponíveis na sua conta.



**Diagrama de Sequência Usar ticket na categoria selecionar um recurso para transação. Representação da ordem de eventos quando o Vinculado seleciona essa opção.**

Vinculado informa seu cartão e Ticket para a interface que envia os dados para o controle. Controle verifica se os dados são válidos e condizentes com o banco de dados, e então, monta uma mensagem de acordo com o resultado da verificação. A tela exibe a liberação ou não da passagem, solicitando outra vez os dados, se necessário.

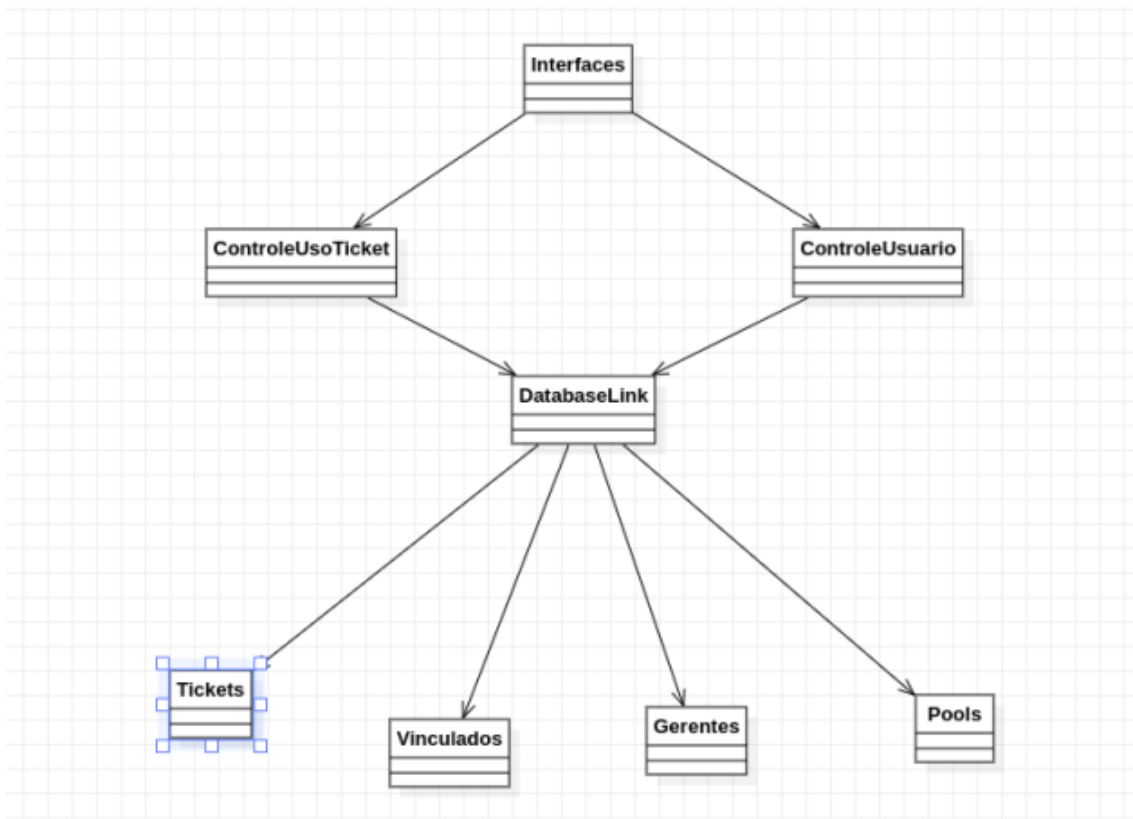


**Diagrama de Sequência Transferir tickets na categoria realizar transação de dois sentidos. Representação da ordem de eventos quando o Vinculado seleciona essa opção.**

O vinculado logado informa o ticket e a quantidade que deseja transferir, bem como o cartão do vinculado que receberá os tickets. A interface envia esses dados para o controle, que faz a validação se o vinculado possui tickets suficientes para transferir e se o vinculado receptor existe e está apto a receber (vinculados PRAE não podem receber ou transferir tickets, já que os recebem gratuitamente). Após as verificações, o controle faz as alterações necessárias no banco de dados, apaga os tickets de um vinculado e envia para o outro, gerando uma mensagem com os resultados para ser mostrada na Interface.

#### **F. DIAGRAMA DE CLASSES (Explicar textualmente, além de incluir as imagens)**





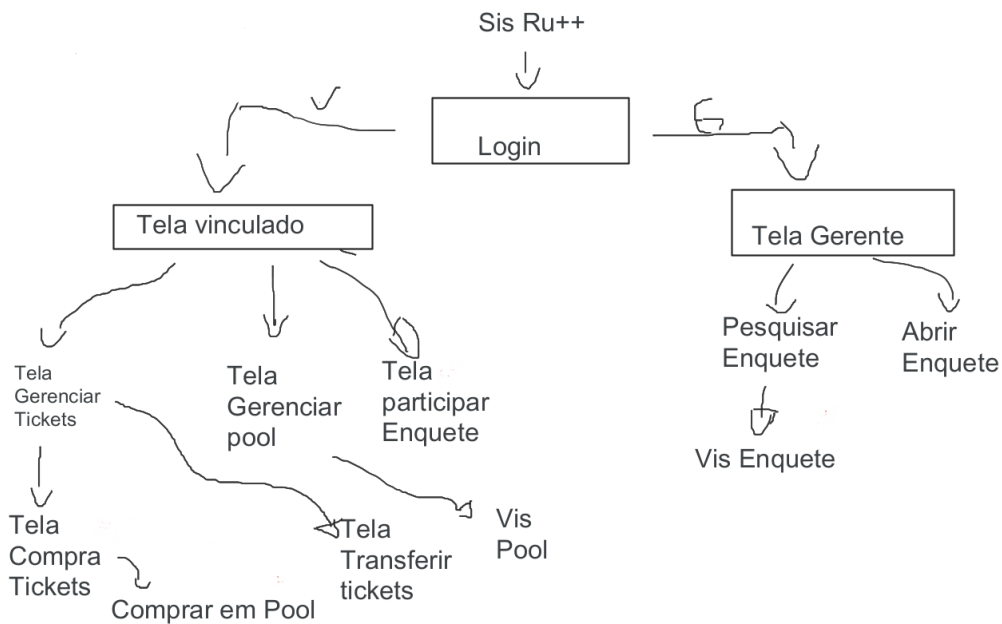
O diagrama descreve superficialmente as classes do sistema, as relações entre elas e suas dependências.

#### **G. LINK PARA GITHUB DO CÓDIGO FONTE**

<https://github.com/Douglas-Schlatter/EngSoft>

#### **PROJETO DE INTERFACE**

Mockups das telas em fase de projeto



*Imagem 1 - Planejamento geral do sistema*

Tela Sistema Ru++

Logo

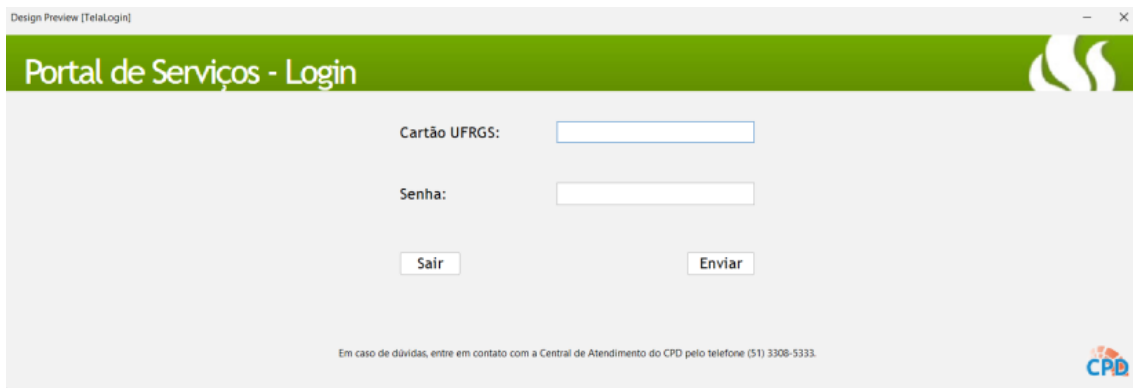
Matricula

Ticket

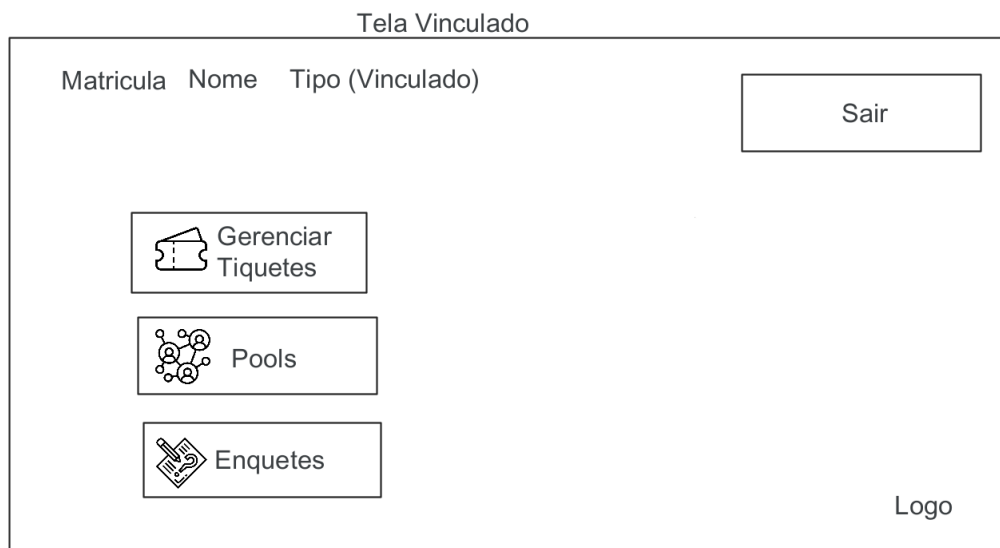
🔍 Sair

Logar

*Imagem 2 - Tela inicial*

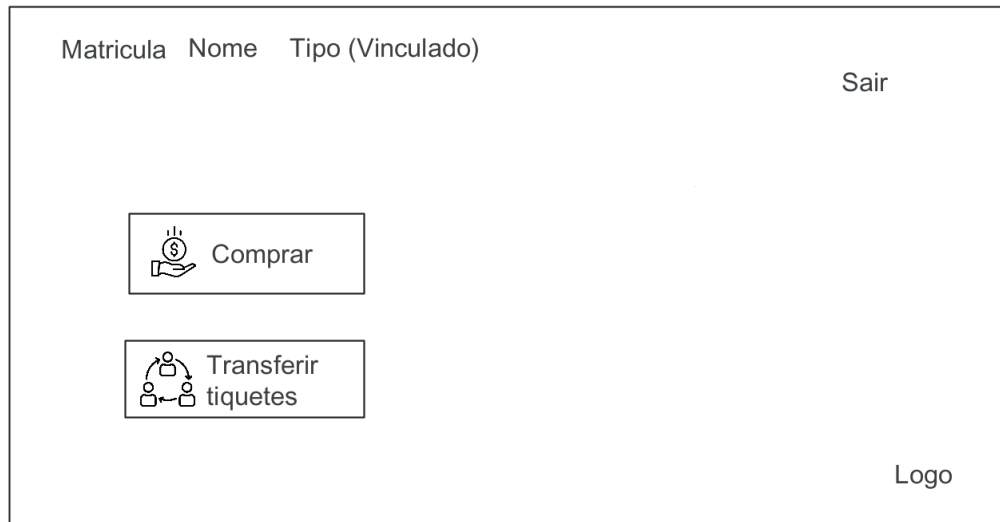


*Imagem 3 - Tela de login (já desenvolvida na etapa 2)*



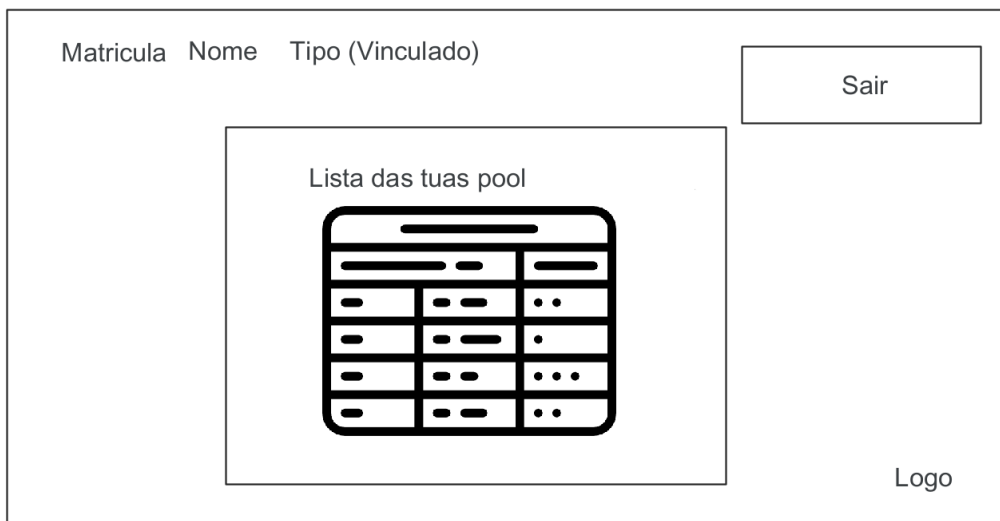
*Imagem 4 - Tela inicial do Vinculado*

Tela Gerenciar Tickets



*Imagem 5 - Tela Gerenciar Tickets*

Tela Gerenciar Pool



*Imagem 6 - Tela Gerenciar Pools*

Tela Vis Pool

Matricula	Nome	Tipo (Vinculado)
Pool Tal		
Participante, tickets restantes		

Sair

Logo

*Imagem 7 - Tela Visualizar Pools*

Tela Compra tickets

Matricula	Nome	Tipo (Vinculado)
Comprar Tickets		
Comprar Em Pools		

Sair

Logo

*Imagem 8 - Tela Comprar Tickets*

Tela Gerente

Matricula	Nome	Tipo(Gerente)
<div>Criar Enquete</div> <div>Pesquisar Enquetes</div> <div>Sair</div>		

*Imagem 9 - Tela Gerente*

## H. ATAS DE REUNIÕES

### **Simplificações:**

O sistema será implementado em uma aplicação Java, ao invés de Web.

Os casos de uso relacionados as enquetes não serão implementados.

---

## Reunião do dia 12/08/2023

### Participantes

Artur Ruiz de Souza  
Douglas Schlatter  
Henry Ribeiro Piceni  
Luana Taynara da Costa Hahn

### Pautas

1. Definições iniciais do trabalho
2. Divisão de tarefas

### Meio

Reunião em chamada via Discord. Elaboração dos documentos compartilhados via Google Docs.

### Atividades realizadas

Definições sobre atualização do código fonte atual. Discussões sobre versões da ferramenta Postgres. Análise inicial das tarefas a serem desenvolvidas e do progresso do trabalho.

---

## Reunião do dia 18/08/2023

---

### Participantes

Luana Taynara da Costa Hahn  
Henry Ribeiro Piceni  
Artur Ruiz de Souza  
Douglas Schlatter

### Pautas

1. Resolução de bugs
2. Elaboração de sketches das interfaces do sistema

### Atividades realizadas

decidimos como funcionará a diferenciação entre pool e tickets.

Tickets: começarão com identificador 0 + numero random de 7 caracteres

Pool: começarão com identificador 1 + número random de 7 caracteres

Isso é feito para no banco de dados ter na pool:

Lista de vinculados | cod da pool | usos restantes

mockups das interfaces desenhado

---



---

## Reunião do dia 23/08/2023

### Participantes

Artur Ruiz de Souza  
Douglas Schlatter  
Henry Ribeiro Piceni  
Luana Taynara da Costa Hahn

### Pautas

1. Finalização dos diagramas de sequência
2. Finalização da Implementação
3. Reunião dos documentos para relatório final

### Meio

Reunião em chamada via Discord. Elaboração dos documentos compartilhados via Google Docs.

### Atividades realizadas

---

## Reunião do dia 17/08/2023

---

### Participantes

Luana Taynara da Costa Hahn  
Henry Ribeiro Piceni  
Artur Ruiz de Souza  
Douglas Schlatter

### Atividades realizadas

Ajustamos os ambientes de programação nos computadores de todos os membros do grupo. Resolução de eventuais bugs.

---

## Reunião do dia 20/08/2023

---

### Participantes

Luana Taynara da Costa Hahn  
Artur Ruiz de Souza

### Atividades realizadas

Desenvolvidas algumas telas de interface do Vinculado e do Gerente

---