

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ – UTFPR**  
DEPARTAMENTO ACADÊMICO DE COMPUTAÇÃO  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

ANA LUIZA SENGER MACOWSKI  
BEATRIZ CAROLINE MOGGIO  
FELIPE EVANGELISTA GOMES  
ISABELA CAMILLE BAROTTO JANGUAS  
PAULO HENRIQUE SALVADORI JUNIOR

**ANÁLISE E PROJETO DE SOFTWARE**  
CHECK-MOGGIO

CAMPO MOURÃO, PARANÁ  
2025

ANA LUIZA SENGER MACOWSKI  
BEATRIZ CAROLINE MOGGIO  
FELIPE EVANGELISTA GOMES  
ISABELA CAMILLE BAROTTO JANGUAS  
PAULO HENRIQUE SALVADORI JUNIOR

**ANÁLISE E PROJETO DE SOFTWARE**  
**CHECK-MOGGIO**

Trabalho desenvolvido durante a disciplina de  
Análise e Projeto de Software, sob orientação do  
Prof. Walter Nakamura.

CAMPO MOURÃO, PARANÁ  
2025

## *Sumário*

<b>Histórico de revisões</b>	<b>6</b>
<b>1. Introdução</b>	<b>9</b>
<b>2. Descrição do Sistema</b>	<b>10</b>
2.1 Informações Gerais	10
2.3 Usuários e stakeholders	12
2.4 Plataformas e Tecnologias	13
2.5 Integrações	13
2.6 Requisitos de Dados	13
2.7 Interface do Usuário (UI)	14
2.8 Cronograma	14
2.9 Orçamento	15
2.10 Requisitos de Testes e Qualidade	15
2.11 Manutenção e Suporte	16
2.12 Outras considerações	16
<b>3. Diagrama de Casos de Uso (DCU)</b>	<b>18</b>
<b>4. Descrição Textual dos Casos de Uso</b>	<b>19</b>
4.1 Formato Resumido	19
4.1.1 Caso de uso: <<CRUD>> Cadastro de Alunos	19
4.1.2 Caso de uso: <<CRUD>> Instituições de Ensino	19
4.1.3 Caso de uso: <<CRUD>> Horários de Embarque	19
4.2 Formato Completo Abstrato	20
4.2.1 Caso de Uso: Aceitar cadastros	20
4.2.2 Caso de Uso: Realizar login	21
4.2.3 Caso de Uso: Realizar check-in	22
4.2.4 Caso de Uso: Selecionar horários de embarque próprios	23
4.3 Formato Completo Concreto	26
4.3.1 Caso de Uso: Listar alunos para ida	26
4.3.2 Caso de Uso: Listar alunos para volta	29
<b>5. Modelo Conceitual (MC)</b>	<b>32</b>
<b>6. Diagrama de Sequência do Sistema (DSS)</b>	<b>33</b>
6.1 Avaliar cadastro:	33
6.2 Realizar login:	33
6.3 Realizar check-in:	34

6.4 Selecionar horários de embarque:	34
6.5 Atualizar horários de embarque:	35
6.6 Listar alunos para ida:	35
6.7 Listar alunos para volta:	36
<b>7. Contratos da Operação (CO)</b>	<b>36</b>
7.1 Operação: realizarLogin(email, senha)	36
7.2 Operação: avaliarCadastro(email)	37
7.3 Operação: selecionarHorarioIda(email, id_ida)	37
7.4 Operação: selecionarHorarioVolta(email, id_volta)	38
7.5 Operação: salvarHorarios(email, id_horario)	38
7.6 Operação: atualizarHorarioIda(email, dia_semana, id_horario)	39
7.7 Operação: atualizarHorarioVolta(email, id_horario, dia_semana)	39
7.8 Operação: selecionarDia(dia_semana)	40
7.9 Operação: listarPorHorario(id_horario)	40
7.10 Operação: realizarCheckin(email)	41
<b>8. Diagramas de Comunicação (DC)</b>	<b>41</b>
8.1 Realizar login	41
8.2 Avaliar cadastros	42
8.3 Selecionar horários de embarque	42
8.4 Atualizar horários de embarque	43
8.5 Listar alunos para ida	43
8.6 Listar alunos para volta	44
8.7 Realizar check-in	44
8.8 Salvar Login	45
8.9 Resetar horários para o preset	45
8.10 Enviar notificações perto do horário de embarque	46
<b>9. Diagrama de Classes em Tempo de Projeto (DCP)</b>	<b>47</b>

## ***Figuras***

Figura 1: DCU - Diagrama de Casos de Uso.	18
Figura 2.1: Protótipo da tela de menu do caso de uso “Listar alunos para ida”.	26
Figura 2.2: Protótipo da primeira tela do caso de uso “Listar alunos para ida”.	26
Figura 3.1: Protótipo da tela de menu do caso de uso “Listar alunos para volta”.	29
Figura 3.2: Protótipo da primeira tela do caso de uso “Listar alunos para a volta”.	29
Figura 4: Diagrama do Modelo Conceitual.	32
Figura 5.1: DSS - Avaliar cadastro.	33
Figura 5.2: DSS - Realizar login.	33
Figura 5.3: DSS - Realizar check-in.	34
Figura 5.4: DSS - Selecionar horários de embarque.	34
Figura 5.5: DSS - Atualizar horários de embarque.	35
Figura 5.6: DSS - Listar alunos para ida.	35
Figura 5.7: DSS - Listar alunos para volta.	36
Figura 6.1: DC - Realizar login.	41
Figura 6.2: DC - Avaliar cadastros.	42
Figura 6.3: DC - Selecionar horários de embarque.	42
Figura 6.4: DC - Atualizar horários de embarque.	43
Figura 6.5: DC - Listar alunos para ida.	43
Figura 6.6: DC - Listar alunos para volta.	44
Figura 6.7: DC - Realizar check-in.	44
Figura 6.8: DC - Salvar login.	45
Figura 6.9: DC - Resetar horários.	45
Figura 6.9: DC - Notificar alunos.	46
Figura 7: DCP - Diagrama de Classes em Tempo de Projeto.	47

## ***Histórico de revisões***

As abreviações presentes no histórico representam cada uma das seções do projeto:

- DS: Descrição do Sistema;
- DCU: Diagrama de Casos de Uso;
- DTCU: Descrição Textual dos Casos de uso;
- MC: Modelo Conceitual;
- DSS: Diagrama de Sequência do Sistema;
- CO: Contratos de Operação;
- DC: Diagrama de Comunicação;
- DCP: Diagrama de Classes em Tempo de Projeto.

<b>Descrição</b>	<b>Responsável</b>	<b>Data</b>
Capa e folha de rosto	Beatriz	11/04/2025
DS - Requisitos funcionais e não funcionais	Ana Luiza, Beatriz, Felipe, Isabela, Paulo	11/04/2025
DS - Cronograma e identificação dos stakeholders	Ana Luiza, Beatriz, Felipe, Isabela, Paulo	16/04/2025
DS - Definição do orçamento e requisitos de teste e qualidade	Beatriz, Isabela	16/04/2025
DS - Possível interface de usuário	Paulo	16/04/2025
<b>1ª entrega: DS - 20/04/2025</b>		
DCU - Levantamento inicial	Ana Luiza, Felipe, Paulo	16/04/2025
DCU - Desenvolvimento	Ana Luiza, Beatriz, Felipe, Isabela, Paulo	25/04/2025
DCU - Finalização	Ana Luiza, Felipe, Isabela, Paulo	26/04/2025
<b>2ª entrega: DCU - 27/04/2024</b>		
DTCU - Alinhando possíveis ideias de casos de uso textuais	Ana Luiza, Beatriz, Isabela	30/04/2025
Introdução	Beatriz	01/05/2025
DCTU - Instituições de Ensino DCTU - Listar alunos para ida	Beatriz	01/05/2025
DCTU - Realizar check-in	Isabela	03/05/2025

DCTU - Listar alunos para volta		
DCTU - Finalização das interfaces do protótipo		
DCTU - Cadastro de Alunos	Felipe	06/05/2025
DCTU - Aceitar Cadastros		
DCTU - Fazer login	Ana Luiza	06/05/2025
DCTU - Definir horários de embarque		
Revisão de ortografia e formatação 1	Ana Luiza	06/05/2025
DCTU - Horários de Embarque	Paulo	06/05/2025
DCTU - Atualizar horário de embarque		
DCTU - Revisão final	Ana Luiza e Isabela	06/05/2025
MC - Prototipação Inicial	Beatriz, Felipe e Paulo	06/05/2025
3ª entrega: DCTU - 07/05/2024		
MC - Identificação dos atributos e desenvolvimento	Beatriz e Isabela	10/05/2025
MC - Desenvolvimento, revisão e modificações	Ana Luiza, Beatriz, Isabela e Paulo	11/05/2025
4ª entrega: MC - 11/05/2024		
DSS - Desenvolvidos	Ana Luiza, Beatriz, Felipe, Isabela e Paulo	14/05/2025
DSS - Finalização	Ana Luiza, Beatriz, Felipe, Isabela e Paulo	16/05/2025
5ª entrega: DSS - 18/05/2025		
Atualizações segundo correções do professor	Ana Luiza	21/05/2025
CO - Desenvolvimento dos Contrato da Operação	Ana Luiza, Beatriz, Felipe, Isabela e Paulo	21/05/2025
DSS - Correções e atualizações	Isabela	25/05/2025
DSS - Revisão	Beatriz e Isabela	25/05/2025
CO - Revisão		
6ª entrega: CO - 25/05/2025		
MC - Conversa com professor e alterações	Ana Luiza e Paulo	28/05/2025
DC - Selecionar horários de embarque próprios	Ana Luiza	01/06/2025
DC - Atualizar horários de embarque próprios		
DC - Realizar check-In	Paulo	01/06/2025
DC - Resetar horários para preset		
DC - Avaliar cadastro	Beatriz	02/06/2025
DC - Enviar notificações perto do horário de embarque		

DC - Realizar Login DC - Salvar Login	Isabela	03/06/2025
DC - Listar Alunos para ida DC - Listar Alunos para Volta	Felipe	03/06/2025
<b>7ª entrega: DC - 08/06/2025</b>		
DC - Revisão e Correções	Beatriz e Isabela	29/06/2025
MC - Correção	Ana Luiza	29/06/2025
DC - Revisão e Correções	Felipe e Paulo	03/07/2025
DGP - Realização	Ana Luiza, Isabela, Paulo	03/07/2025
Revisão final	Ana Luiza, Beatriz, Felipe	03/07/2025
<b>8ª entrega: Projeto Completo - 04/07/2025</b>		



## ***1. Introdução***

O projeto leva o nome da empresa para a qual está sendo desenvolvido – Check-Moggio, sistema criado para atender às necessidades da Transportes Moggio. Os responsáveis pelo desenvolvimento são: Ana Luiza Senger Macowski, Beatriz Caroline Moggio, Felipe Evangelista Gomes, Isabela Camille Barotto Janguas e Paulo Henrique Salvadori Junior, alunos do segundo período de Bacharelado em Ciência da Computação na UTFPR-CM.

A empresa Transportes Moggio é natural do município de Araruna/PR e atua no transporte escolar de alunos do ensino fundamental, médio e superior, com rotas diárias entre Araruna e Campo Mourão. A empresa é administrada por Leonildo Moggio, que, juntamente com sua filha – uma das responsáveis pelo desenvolvimento do projeto –, identificou a necessidade de modernizar o controle de horários e organização dos alunos que utilizam seus serviços.

Para otimizar o controle operacional, surgiu a ideia de desenvolver um sistema mobile, multiplataforma (Android e iOS), que possibilite aos alunos cadastrarem seus horários de ida e volta, realizarem alterações conforme necessário e confirmarem a presença no ônibus por meio da funcionalidade de check-in, a qual também compõe o nome do projeto. Ao mesmo tempo, o sistema permite ao administrador visualizar, em tempo real, os alunos confirmados para o turno, organizados por instituição de ensino, com destaque para aqueles que já realizaram o check-in.

Com isso, o objetivo do sistema é automatizar processos que antes eram realizados manualmente, reduzir falhas de comunicação e proporcionar maior organização, tanto para motoristas quanto para os alunos e seus responsáveis. A aplicação será projetada com tecnologias modernas, como Flutter para o desenvolvimento da interface e Supabase para o backend, além de oferecer suporte a funcionalidades offline e envio de notificações em horários estratégicos.

Portanto, este projeto busca atender diretamente às necessidades da empresa, oferecendo uma solução tecnológica acessível, prática e eficiente para o gerenciamento do transporte escolar.

## ***2. Descrição do Sistema***

### ***2.1 Informações Gerais***

- **Nome do projeto**  
Check-Moggio.
- **Responsável pelo projeto**  
Leonildo Moggio, pai da Beatriz, uma das alunas responsáveis pelo projeto.
- **Objetivo principal do software a ser desenvolvido**  
Gerenciar horários em que os alunos clientes devem ser levados ou buscados de suas faculdades pela Transportes Moggio, empresa de transporte escolar.

### ***2.2 Escopo do projeto***

#### **1. Descrição geral do software**

Trata-se de um sistema mobile com dois tipos de acesso: administrador e clientes. A aplicação permite o cadastro dos horários de idas e voltas diárias dos alunos, com a possibilidade de realizar alterações, caso necessário. Esses horários ficam visíveis para os motoristas, facilitando a organização do transporte. Além disso, o sistema oferece a funcionalidade de confirmação de embarque, na qual o aluno pode marcar a opção “check-in”, proporcionando um controle mais eficiente por parte dos motoristas.

#### **2. Principais funcionalidades que o software deve possuir**

- Para o administrador: gerar lista de alunos que vão embarcar ou já estão embarcados no ônibus em dias e horários específicos;
- Para os alunos: gerenciar seus horários de embarque e fazer check-in ao embarcar.

### 3. Requisitos funcionais específicos

RF01	Os alunos devem poder se cadastrar utilizando e-mail, senha e seu nome completo;
RF02	O administrador deve poder aceitar ou rejeitar cadastros;
RF03	Os usuários devem poder fazer login utilizando e-mail e senha;
RF04	O aluno deve poder gerenciar seus horários de ida e volta no sistema;
RF05	Os horários alterados devem resetar aos sábados, voltando aos horários originais/padrão;
RF06	O administrador pode ver os alunos confirmados para embarque nos determinados horários/turnos;
RF07	O aluno deve poder fazer check-in do seu embarque.
RF08	O sistema deve enviar uma notificação em tempo real ao aluno, 10 minutos antes do horário de ida, informando que o ônibus está prestes a iniciar a rota, e também no horário de volta, informando que o ônibus está saindo da instituição de ensino.

### 4. Requisitos não funcionais importantes

RNF01	O sistema deve ser implementado em Flutter com banco de dados PostgreSQL;
RNF02	O sistema deve salvar as escolhas feitas sem conexão com a internet e aplicá-las automaticamente quando houver conectividade wi-fi novamente;

RNF03	O sistema deve ser responsivo para diferentes tamanhos de tela de celular;
RNF04	O sistema deve ser compatível com Android e iOS;
RNF05	O tempo de resposta máxima de uma operação no sistema para o usuário deve ser de 5 segundos;
RNF06	Em caso de falha, o sistema deve voltar a operar normalmente em até 2 minutos;
RNF07	Após o cadastro ter sido aceito pelo administrador, a partir do próximo login, os dados de login devem poder ser salvos em uma opção “Lembrar login”;
RNF08	O tamanho das fontes dentro do aplicativo deve seguir a configuração de tamanho do próprio dispositivo.

### ***2.3 Usuários e stakeholders***

- **Stakeholders principais: usuários do sistema**
  - Leonildo Moggio, responsável pela empresa de transporte;
  - Motoristas, que utilizarão o aplicativo para visualizar os horários programados dos alunos e verificar se todos realizaram o check-in;
  - Alunos que utilizarão o aplicativo para cadastrar e gerenciar seus horários de ida e volta, além de confirmar o embarque por meio da funcionalidade de check-in.
  
- **Stakeholders secundários: demais partes interessadas**
  - Responsáveis pelos alunos: embora não sejam usuários do sistema, podem se tornar partes interessadas no projeto, uma vez que podem acompanhar o estado de embarque dos alunos de quem cuidam.
  - Google Play e Apple Store: plataformas responsáveis pela distribuição do aplicativo.

## ***2.4 Plataformas e Tecnologias***

- **Plataformas específicas na qual o software deve ser executado**

O software será desenvolvido para a plataforma mobile, estando disponível para os sistemas operacionais Android e iOS.

- **Tecnologia ou linguagem de programação preferencial**

O aplicativo será criado usando Flutter, um framework ideal para desenvolver a interface mobile e garantir que o aplicativo funcione tanto no Android quanto no iOS. Para o backend, usaremos o Supabase, uma plataforma que oferece banco de dados em tempo real.

## ***2.5 Integrações***

- **Integração com outros sistemas e requisitos específicos de protocolos**

- Será realizada uma integração de sistemas com o Pusher, uma API que permite o envio de notificações em tempo real para os estudantes. As notificações serão enviadas em dois momentos específicos:
- Às 5h50, 11h10 e 18h10, um aviso lembrando os estudantes de que a rota está prestes a começar, para que não se atrasem e não percam o transporte.
- Às 11h55, 17h25 e 23h45, uma notificação informando que o ônibus está saindo da faculdade/colégio para retornar a Araruna.

## ***2.6 Requisitos de Dados***

- **Tipos de dados que o software precisará armazenar ou manipular**

- Informações dos usuários: nome completo, e-mail e senha dos alunos para autenticação e gerenciamento de contas;
- Horários de embarque: horários de ida e volta dos alunos, que serão definidos por eles mesmos, além de possíveis alterações feitas ao longo do tempo.

- **Requisitos de segurança ou privacidade dos dados**

- Controle de acesso: autenticação segura com e-mail e senha;

- Aprovação do cadastro: todo cadastro de aluno ou usuário será submetido à aprovação do administrador, garantindo um controle adicional sobre os dados e o acesso ao sistema.

## ***2.7 Interface do Usuário (UI)***

- **Design de interface ou estilo visual específico a ser seguido**

O sistema adotará um design responsivo e acessível. Além disso, deverá oferecer as opções de modo claro e escuro, permitindo que o usuário escolha a interface que melhor se adapta às suas preferências e ao ambiente de uso.

- **Principais fluxos de interação do usuário**

- Administrador: cadastro, login, listar alunos por dia e horário, aceitar solicitação de cadastro.
- Cliente: cadastro, login, adicionar grade de horários, alterar grade de horários, atualizar horários, confirmar embarque.

## ***2.8 Cronograma***

- **Prazo estimado para o desenvolvimento e lançamento do software**

- Prazo para desenvolvimento: 16/10/2025.
- Prazo para lançamento: 16/12/2025.

- **Marcos importantes ou prazos intermediários**

- 27/06/2025: entrega do documento de Análise e Projeto de Software, consolidando os requisitos e diretrizes para o desenvolvimento do sistema.

## ***2.9 Orçamento***

- **Remuneração da equipe:** pagamento pelos serviços de desenvolvimento de software e design de interface do usuário;
- **Publicação do aplicativo:** taxas referentes à publicação do aplicativo nas lojas Google Play (taxa única de US\$ 25, aproximadamente R\$ 145,00 em 07/05/25) e Apple Store (anuidade de US\$ 99, aproximadamente R\$ 570,00 em 07/05/25).

Considerando os custos operacionais, o tempo de desenvolvimento e a complexidade do sistema, o valor total cobrado pelo projeto será de R\$ 40.000,00.

## ***2.10 Requisitos de Testes e Qualidade***

- **Expectativas em relação à qualidade do software**

O software deve ser responsivo, intuitivo e possuir um alto desempenho com respostas rápidas, garantindo uma experiência de uso fluida. Além disso, é importante que ele seja estável mesmo em condições de baixa conectividade.

- **Cenários de teste críticos**

1. Cadastro de usuário e login:
  - a. Testar o fluxo de cadastro de um novo usuário;
  - b. Testar a validação de dados no cadastro (verificar se o e-mail já está registrado);
  - c. Testar as funcionalidades de login (e-mail e senha corretos);
  - d. Testar a recuperação de senha.
2. Acesso do administrador:
  - a. Testar a visualização da lista de alunos confirmados para embarque;
  - b. Testar a permissão de acesso dos usuários ao aplicativo.
3. Gerenciamento de horários de embarque:
  - a. Testar as funções do aluno de adicionar, alterar, salvar e excluir seus horários de embarque;
  - b. Testar se os horários são restaurados aos sábados, após as alterações.
4. Check-in de embarque:
  - a. Testar que o aluno consiga marcar sua presença de embarque;
  - b. Testar se o motorista pode visualizar os alunos embarcados.
5. Funcionamento offline:
  - a. Testar se as alterações são sincronizadas quando a conexão for restabelecida.

## ***2.11 Manutenção e Suporte***

- **Suporte contínuo e a manutenção do software após o lançamento**
  - Suporte durante horário comercial por aplicativo de mensagens;
  - De 6 em 6 meses, a equipe de suporte faz uma limpeza no banco de dados, excluindo informações obsoletas.

## ***2.12 Outras considerações***

- **Regras de Negócio:**
  - RN01: Apenas cadastros com todos os campos obrigatórios preenchidos podem ser aprovados;
  - RN02: Uma vez aprovado, o cadastro do aluno não poderá ser editado;
  - RN03: A função “Realizar check-in” presente no aplicativo somente estará habilitada para o aluno registrar sua presença dentro do ônibus 15 minutos antes do horário de retorno;
  - RN04: Os horários atualizados valem apenas para os dias modificados da semana em questão;
  - RN05: No fim de semana, o sistema restaura automaticamente os horários padrão do aluno para a próxima semana.
  - RN06: Os horários de ida são exibidos conforme cadastrados ou alterados temporariamente;
  - RN07: A lista de alunos é resetada aos sábados, retornando aos horários padrão;
  - RN08: O sistema deve funcionar offline, sincronizando dados quando a conexão for restabelecida;
  - RN09: O sistema atualiza automaticamente a lista quando um aluno altera seu horário, mantendo o foco na aba “IDA” e no dia da semana selecionado;
  - RN10: Os horários de volta são exibidos conforme cadastrados ou alterados temporariamente;
  - RN11: A lista de alunos é resetada aos sábados, retornando aos horários padrões;
  - RN12: O sistema deve funcionar offline, salvando os dados localmente e os sincronizando quando a conexão for restabelecida.



### 3. Diagrama de Casos de Uso (DCU)

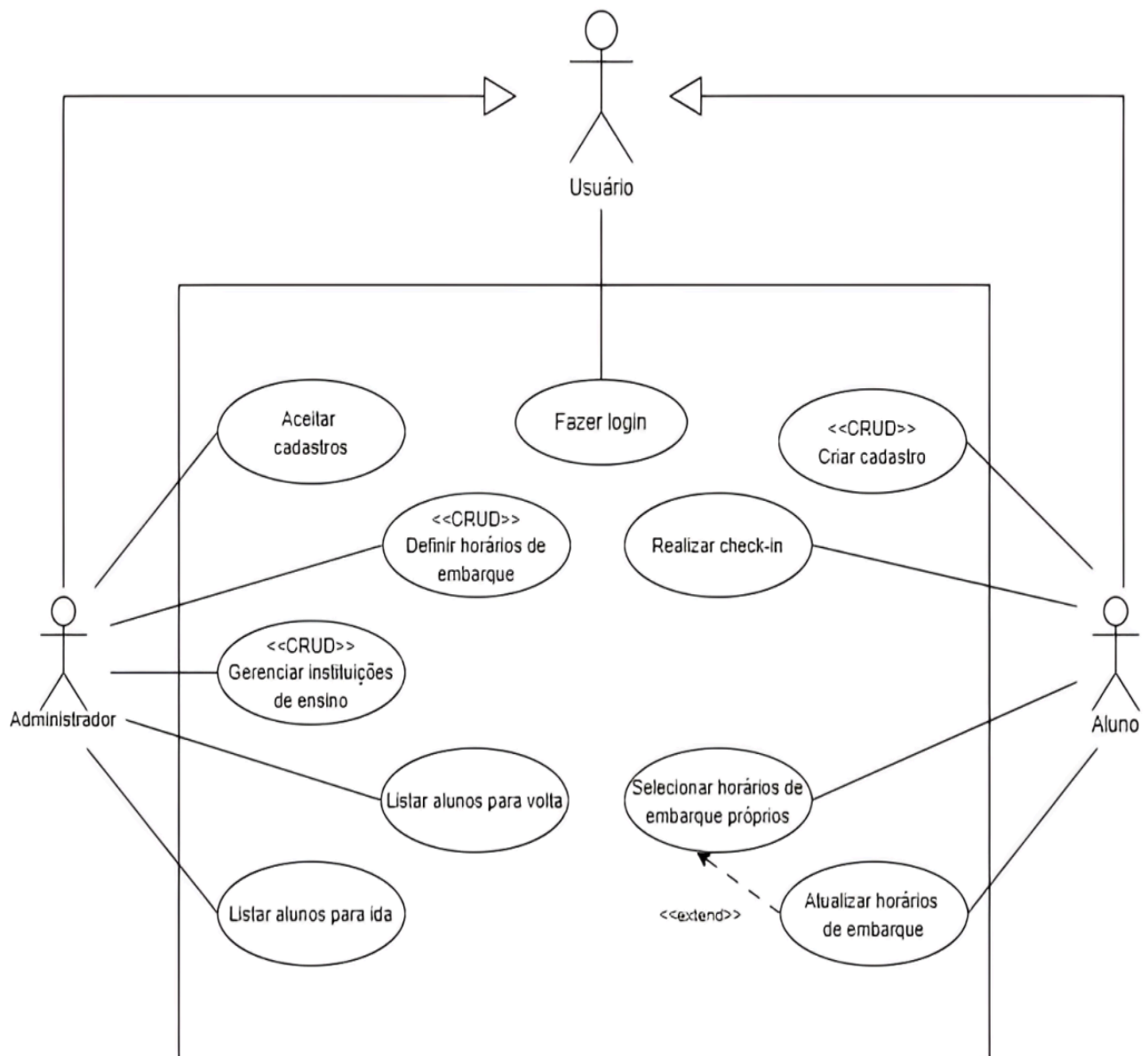


Figura 1: DCU - Diagrama de Casos de Uso.

## ***4. Descrição Textual dos Casos de Uso***

### ***4.1 Formato Resumido***

#### ***4.1.1 Caso de uso: <<CRUD>> Cadastro de Alunos***

O aluno acessa o sistema e preenche um cadastro informando os seguintes dados solicitados: nome completo, e-mail válido, senha para a conta e escolhe a opção que aparece o nome da instituição que ele frequenta, dentre as cadastradas no sistema previamente. Após o envio das informações, o sistema irá enviar o cadastro para o administrador, que deverá validar o cadastro dentro do próprio aplicativo para que o aluno possa logar. As opções de alterar, visualizar e remover cadastros são liberadas apenas para o administrador.

#### ***4.1.2 Caso de uso: <<CRUD>> Instituições de Ensino***

O administrador acessa a funcionalidade de cadastro de instituições e insere os dados solicitados: nome da instituição, endereço completo, telefone de contato e o(s) nível(is) de ensino atendido(s) (fundamental, médio e superior). Após preencher os dados, solicita que o sistema salve o registro. O sistema deve permitir alterar, visualizar e remover as instituições previamente cadastradas. As instituições cadastradas serão utilizadas posteriormente para a organização da visualização dos alunos por local de estudo.

#### ***4.1.3 Caso de uso: <<CRUD>> Horários de Embarque***

O administrador acessa a funcionalidade de definir horários de embarque, na qual ele insere os horários de saída do ônibus para cada dia da semana. Esses horários serão colocados na tela principal, e serão usados pelo aluno, para selecionar seus horários próprios de embarque, e pelo próprio motorista, que conseguirá ver os alunos confirmados por horário.

## ***4.2 Formato Completo Abstrato***

### ***4.2.1 Caso de Uso: Aceitar cadastros***

**Descrição:** O administrador realiza uma verificação e aceita os cadastros dos usuários que enviaram um pedido de cadastro no aplicativo.

**Ator principal:** Administrador.

#### **Interessados e interesses:**

- **Administrador:** deseja gerenciar os cadastros dos alunos, garantindo que apenas usuários válidos e com informações corretas tenham acesso ao sistema;
- **Alunos:** têm interesse em ter seus cadastros aprovados para poder acessar os serviços disponíveis.

#### **Pré-condições:**

- O administrador deve estar autenticado no sistema;
- Deve haver cadastros pendentes aguardando análise.

#### **Pós-Condições:**

- Os cadastros aprovados são ativados, liberando o acesso dos alunos ao sistema;
- Os cadastros rejeitados são excluídos.

#### **Fluxo Principal:**

1. O sistema exibe a tela principal com as seguintes seções:
  - (A) Menu lateral: permite navegar entre os módulos do sistema;
  - (B) Aba “Solicitações de acesso”: exibe a lista de alunos que aguardam validação.
2. O administrador acessa a aba “Solicitações de acesso” (B);
3. O sistema exibe os nomes dos cadastros aguardando aprovação;
4. O administrador seleciona um cadastro da lista para visualizar os detalhes completos;
5. O administrador escolhe entre aprovar ou rejeitar o cadastro;
6. O sistema registra a decisão e atualiza o status do cadastro.

#### **Fluxo Alternativo:**

3. Sem cadastros pendentes:

3.1. Se não houver cadastros pendentes, o sistema exibe a mensagem: “Nenhum cadastro pendente no momento”, e finaliza o fluxo.

**Regras de negócio:**

1. RN01: Apenas cadastros com todos os campos obrigatórios preenchidos podem ser aprovados;
2. RN02: Uma vez aprovado, o cadastro do aluno não poderá ser editado sem nova análise.

**4.2.2 Caso de Uso: Realizar login**

**Descrição:** Os usuários acessam o sistema por meio de login.

**Atores principais:** Administrador e aluno.

**Interessados e interesses:**

- **Aluno:** Acessar o sistema para utilizar os serviços;
- **Motoristas:** Acessar o sistema para utilizar os serviços.

**Pré-condições:**

- O motorista deve saber o nome de usuário e a senha de administrador, os quais são passados diretamente à empresa pela equipe de suporte;
- O aluno deve ter criado seu cadastro com seu e-mail e senha;
- O cadastro do aluno deve ter sido aceito pelo administrador.

**Pós-Condições:**

- O administrador recebe acesso às funcionalidades designadas a ele;
- O aluno recebe acesso às funcionalidades disponibilizadas a ele.

**Fluxo Principal:**

1. O usuário acessa a seção de login;
2. O sistema mostra os campos e-mail, senha e um botão toggle de “Lembrar login”;
3. O usuário preenche os campos (se tiver selecionado “Lembrar login”, já estarão preenchidos) e clica em “Entrar”;
4. O acesso ao sistema é permitido ao usuário;
5. As telas de ação são mostradas ao usuário, a depender do seu tipo de acesso: aluno ou administrador.

**Fluxo Alternativo 1:**

5. As informações de login não foram encontradas na base de dados:
  - 5.1. O sistema exibe a mensagem de erro: “Usuário ou senha incorretos!”;
  - 5.2. Volta ao passo 2.

**Fluxo Alternativo 2:**

2. O usuário ativou, em uma entrada anterior no sistema, a opção “Lembrar login”:
  - 2.1. O sistema pula a etapa 2, preenchendo automaticamente seu nome de usuário e senha, que fica ocultada da visualização direta;
  - 2.2. Segue o fluxo principal normalmente, a partir da etapa 3.

**4.2.3 Caso de Uso: Realizar check-in**

**Descrição:** O aluno realiza o check-in por meio do aplicativo.

**Ator principal:** Aluno.

**Interessados e interesses:**

- **Aluno:** Indica que está presente no ônibus;
- **Motoristas:** Visualizar quais alunos efetuaram o check-in.

**Pré-condições:**

- O aluno deve estar cadastrado no sistema;
- O cadastro do aluno deve ter sido aceito pelo administrador;
- O aluno deve estar logado no sistema no momento do check-in;
- Estar na lista de alunos que voltarão neste horário.

**Pós-Condições:**

- O check-in é registrado no sistema;
- A lista de alunos que embarcaram é atualizada para visibilidade do motorista.

**Fluxo Principal:**

1. Aluno abre o aplicativo;
2. O aluno seleciona a aba de menu lateral;
3. No menu será selecionado a opção “realizar check-in”;
4. O sistema apresenta a tela com a opção de realizar check-in;

5. O aluno clica no botão denominado “check-in”;
6. O sistema registra o embarque do aluno;
7. O sistema atualiza a lista de alunos embarcados.

#### **Fluxo Alternativo 1:**

6. O sistema verifica a ausência de conexão:
  - 6.1. O check-in é salvo localmente;
  - 6.2. No momento que a conexão for restabelecida, retorna ao passo 6.

#### **Regras de negócio:**

1. RN03: A função “Realizar check-in” presente no aplicativo somente estará habilitada para o aluno registrar sua presença dentro do ônibus 15 minutos antes do horário de retorno.

#### **4.2.4 Caso de Uso: Selecionar horários de embarque próprios**

**Descrição:** O aluno informa seus horários de embarque no ônibus de segunda a sexta-feira.

**Atores principais:** Aluno.

#### **Interessados e interesses:**

- **Aluno:** Selecionar, dentre os horários disponíveis, seus horários de ida e volta para a faculdade ou escola com a Transportes Moggio, para que esteja dentro da lista de embarque dos motoristas;
- **Motoristas:** Saber os horários de embarque de cada aluno.

#### **Pré-condições:**

- O aluno deve estar devidamente logado no sistema;
- O administrador deve ter inserido as opções de horários de embarque no sistema.

#### **Pós-Condições:**

- Após a confirmação, os horários selecionados pelo aluno ficam salvos e valendo até que sejam alterados;
- O aluno entra para a lista de embarque dos horários selecionados.

#### **Fluxo Principal:**

1. O aluno escolhe a opção “Selecionar horários de embarque”;

2. A grade com todas as opções de horários para cada dia é aberta;
3. O aluno seleciona seus horários de ida e volta para cada um dos dias da semana;
4. O aluno clica em “Salvar alterações”;
5. A grade de horários do aluno fica salva e ele sai do modo de edição automaticamente.

#### ***4.2.5 Caso de Uso: Atualizar horário de embarque***

**Descrição:** O aluno pode modificar os horários de ida e volta para dias específicos da semana atual, sem alterar seu horário padrão previamente definido.

**Atores principais:** Aluno.

**Interessados e interesses:**

- Aluno: Ajustar seus horários de embarque para dias específicos, conforme necessidades eventuais da semana;
- Motoristas: Visualizar corretamente os horários atualizados para organização do transporte da semana.

**Pré-condições:**

- O aluno deve estar logado no sistema;
- O aluno deve ter horários padrão previamente cadastrados;
- A alteração pode ser feita até 10 minutos antes da partida do ônibus.

**Pós-condições:**

- Os horários do aluno são atualizados;

**Fluxo Principal:**

1. O aluno acessa a página principal;
2. O sistema exibe a grade da semana com os dias úteis (segunda a sexta-feira);
3. O aluno seleciona o dia desejado;
4. O aluno escolhe o novo horário de ida e/ou volta para aquele dia;
5. O aluno clica em “Salvar alteração”;
6. O sistema registra a alteração apenas para aquele dia da semana corrente;
7. Ao final da semana (sábado), o sistema restaura automaticamente os horários padrão para a semana seguinte.

**Regras de negócio:**

1. RN04: Os horários atualizados valem apenas para os dias modificados da semana em questão;
2. RN05: No fim de semana, o sistema restaura automaticamente os horários padrão do aluno para a próxima semana.



### 4.3 Formato Completo Concreto

#### 4.3.1 Caso de Uso: Listar alunos para ida

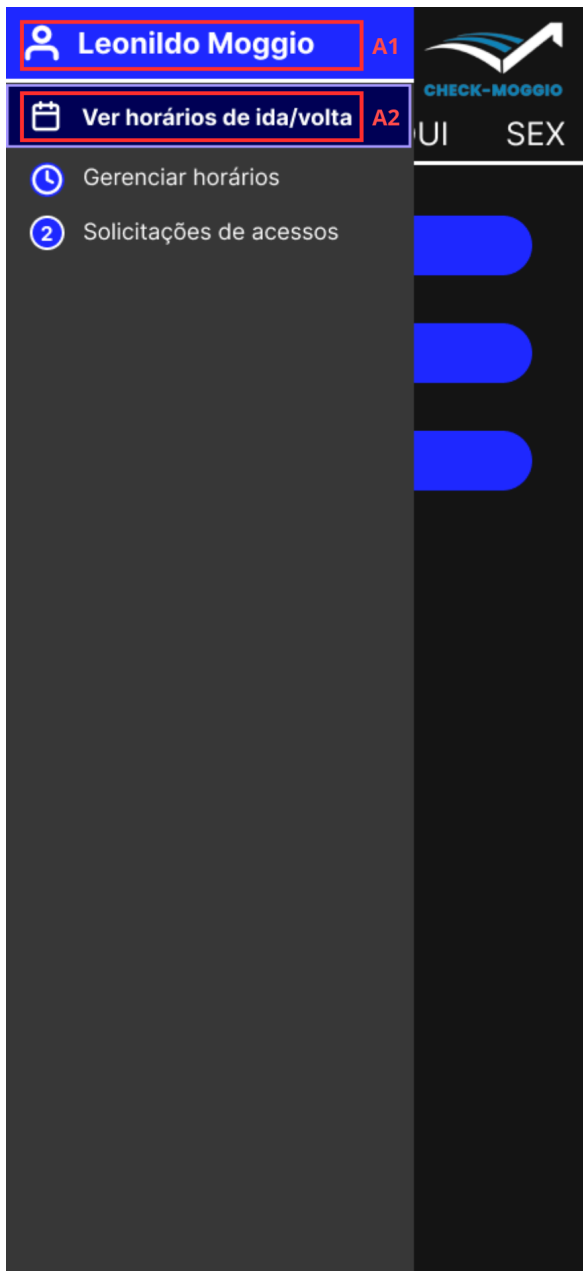


Figura 2.1: Protótipo da tela de menu do caso de uso “Listar alunos para ida”.



Figura 2.2: Protótipo da primeira tela do caso de uso “Listar alunos para ida”.

**Descrição:** O motorista visualiza, por dia da semana, os horários de ida e a lista de alunos confirmados para o embarque em um horário específico.

**Ator Principal:** Administrador/Motorista.

**Interessados e Interesses:**

- **Administrador/Motorista:** deseja visualizar a lista de alunos confirmados para o embarque em um horário específico, organizando o transporte de forma eficiente;
- **Alunos:** têm interesse em ter seu embarque registrado corretamente no sistema.

**Pré-Condições:**

- O administrador/motorista deve estar devidamente autenticado no sistema;
- Os alunos já estarem com seus horários de ida cadastrados ou realizado alterações temporárias.

**Pós-Condições:**

- O sistema exibe a lista de alunos confirmados para o horário de ida selecionado;
- Os dados exibidos são atualizados de acordo com as últimas alterações cadastradas (horários fixos ou temporários).

**Fluxo Principal:**

1. O administrador/motorista acessa o sistema e faz o login;
2. O sistema exibe a tela principal com as seguintes seções marcadas na Figura 1.2:
  - (A) Menu lateral: permite acessar recursos como perfil, ida, volta, gerenciar horários e solicitações de acesso;
  - (B) Aba “IDA”: opção para visualizar apenas os horários de ida;
  - (C) Dias da semana (SEG, TER, QUA, QUI, SEX): permite filtrar alunos por dia específico;
  - (D) Horários de embarque (6:00, 11:20, 18:20): lista os horários disponíveis para o dia selecionado;
3. O administrador seleciona:
  - (B) “IDA” para visualizar apenas os embarques de ida;
  - (C) Um dia da semana (ex: TER - terça-feira);
4. O sistema exibe os horários de ida (D) para o dia selecionado (ex: 6:00, 11:20, 18:20);
5. O administrador escolhe um horário específico (ex: 11:20);

6. O sistema lista os alunos confirmados para o horário selecionado, conforme destacado na imagem.

#### **Fluxo Alternativo 1:**

6. Nenhum aluno confirmado para o horário:
  - 6.1. Se não houver alunos confirmados para o horário selecionado, o sistema exibe: “Nenhum aluno confirmado para este horário” e encerra o fluxo.

#### **Fluxo Alternativo 2:**

6. O sistema verifica a ausência de conexão:
  - 6.1. O sistema exibe a última lista de alunos sincronizada e um aviso: “Sem conexão. Exibindo os últimos dados salvos” e encerra o fluxo.

#### **Regras de Negócio:**

1. RN05: os horários de ida (D) são exibidos conforme cadastrados ou alterados temporariamente;
2. RN06: a lista de alunos é resetada aos sábados, retornando aos horários padrão;
3. RN07: o sistema deve funcionar offline, sincronizando dados quando a conexão for restabelecida;
4. RN08: O sistema atualiza automaticamente a lista quando um aluno altera seu horário, mantendo o foco na aba “IDA” (B) e no dia da semana selecionado (C).

#### 4.3.2 Caso de Uso: Listar alunos para volta

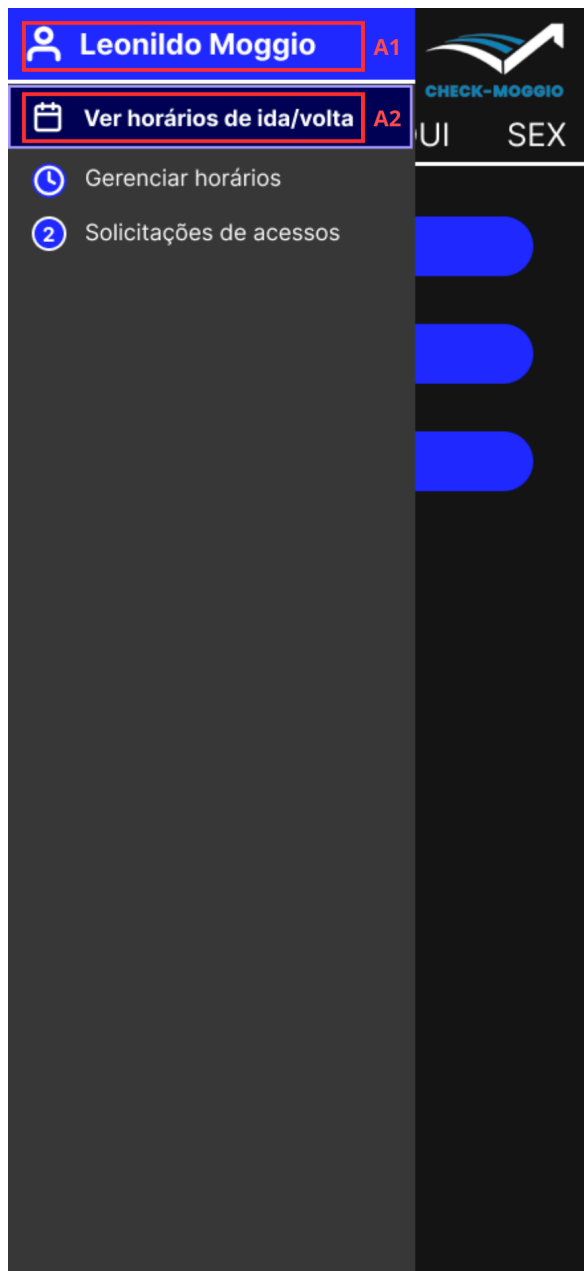


Figura 3.1: Protótipo da tela de menu do caso de uso "Listar alunos para volta".



Figura 3.2: Protótipo da primeira tela do caso de uso "Listar alunos para a volta".

**Descrição:** O motorista vê os alunos, com suas respectivas instituições de ensino, que irão voltar e os que estão presentes em um determinado horário.

**Ator principal:** Administradores/motoristas.

**Interessados e interesses:**

- **Administradores/Motoristas:** Deseja visualizar a lista dos alunos que irão retornar em um horário específico, juntamente com o respectivo colégio no qual o aluno pertence. Além disso, o motorista tem o interesse em visualizar os alunos que já realizaram check-in e estão presentes no ônibus;
- **Alunos:** Têm o interesse de ter o seu retorno registrado e realizar o check-in para confirmar sua presença no ônibus.

**Pré-condições:**

- O administrador/motorista deve estar devidamente autenticado no sistema;
- Os alunos já estarem com seus horários de volta cadastrados ou realizado alterações temporárias.

**Pós-Condições:**

- O sistema exibe a lista de alunos confirmados para o horário de volta selecionado, separado por colégios e indicando quais deles estão presentes no veículo;
- Os dados exibidos são atualizados de acordo com as últimas alterações cadastradas (caso haja mudanças de horários).

**Fluxo Principal:**

1. O administrador/motorista acessa o aplicativo e faz o login;
2. O sistema exibe a tela principal apresentando os horários de volta como mostrado na Figura 2.2;
3. O administrador/motorista clica no botão “volta” no lado inferior direito da tela;
4. O sistema exibe a tela de volta com as seguintes seções marcadas na Figura 2.2:
  - (A) Menu lateral: permite acessar recursos como perfil, ida e volta, gerenciar horários e solicitações de acesso;
  - (B) Funcionalidade “VOLTA”: Opções para visualizar os horários de volta;
  - (C) Dias da semana (SEG, TER, QUA, QUI, SEX): permite filtrar alunos por dia específico;

- (D) Horários de volta (12:00, 17:30, 23:00): lista os horários disponíveis para o dia selecionado;
5. O administrador/motorista seleciona:  
(C) Um dia da semana (ex: SEG - segunda-feira);
  6. O sistema exibe os horários de volta (D) para o dia selecionado (ex: 12:00, 17:30, 23:00);
  7. O administrador/motorista escolhe um horário específico (ex: 23:00);
  8. O sistema lista os alunos que irão voltar no horário selecionado com seus respectivos colégios e uma marcação em verde para os alunos que já estão presentes no veículo e em vermelho para os que não estão.

#### **Fluxo Alternativo 1:**

7. O sistema verifica que não há nenhum aluno na lista para retornar no horário selecionado:
  - 7.1 O sistema exibe a mensagem: “Nenhum aluno confirmado para este horário” e encerra o fluxo.

#### **Fluxo Alternativo 2:**

8. O sistema verifica a ausência de conexão:
  - 8.1. O sistema exibe a última lista de alunos sincronizada e um aviso: “Sem conexão. Exibindo os últimos dados salvos” e encerra o fluxo.

#### **Regras de negócio:**

1. RN09: Os horários de volta (D) são exibidos conforme cadastrados ou alterados temporariamente;
2. RN10: A lista de alunos é resetada aos sábados, retornando aos horários padrões;
3. RN11: O sistema deve funcionar offline, salvando os dados localmente e os sincronizando quando a conexão for restabelecida.

## 5. Modelo Conceitual (MC)

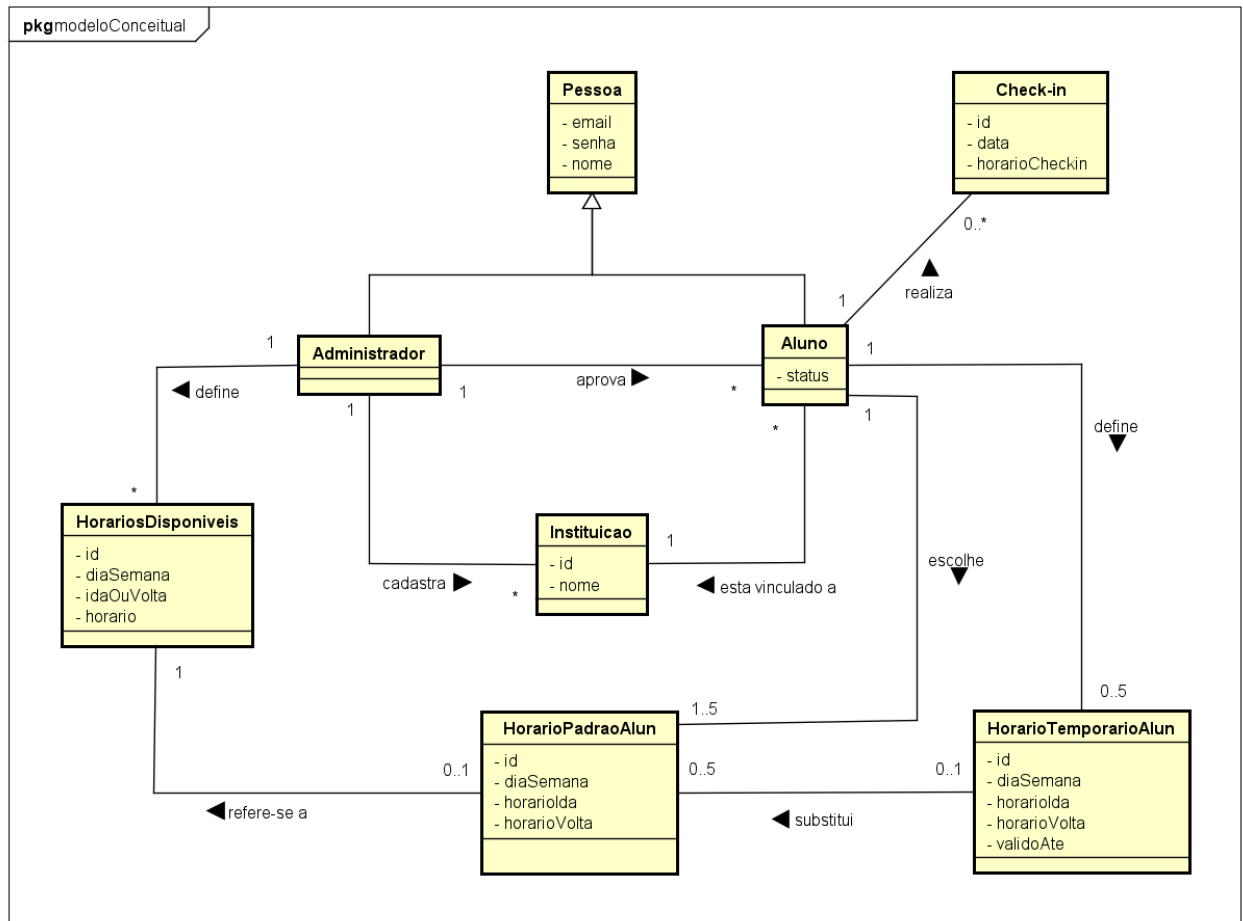


Figura 4: Diagrama do Modelo Conceitual.

## 6. Diagrama de Sequência do Sistema (DSS)

### 6.1 Avaliar cadastro:

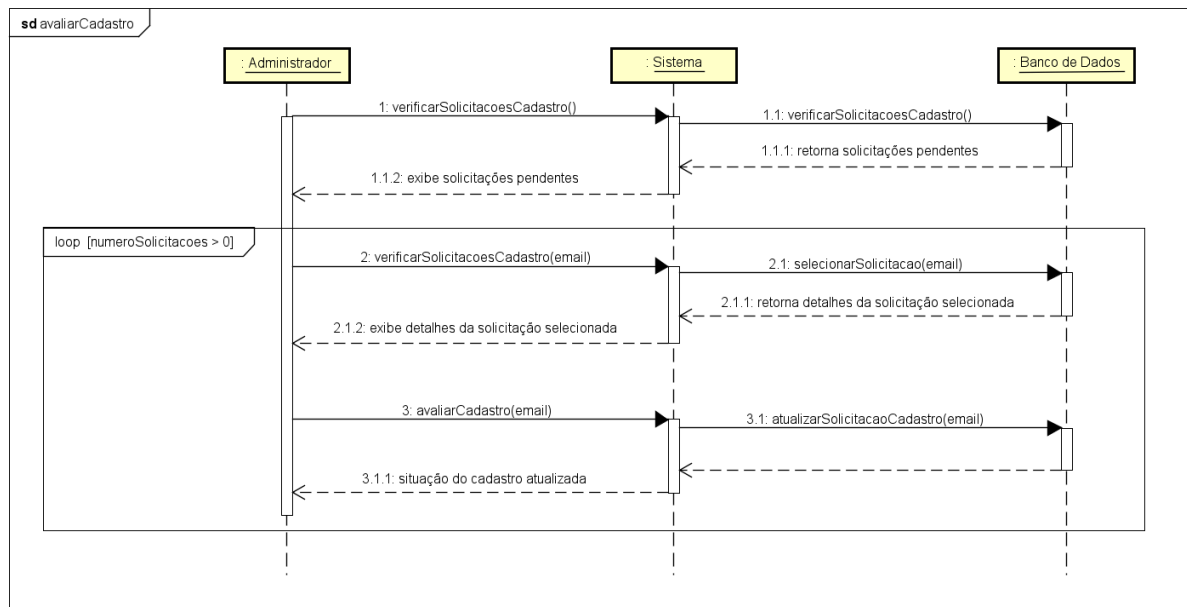


Figura 5.1: DSS - Avaliar cadastro.

### 6.2 Realizar login:

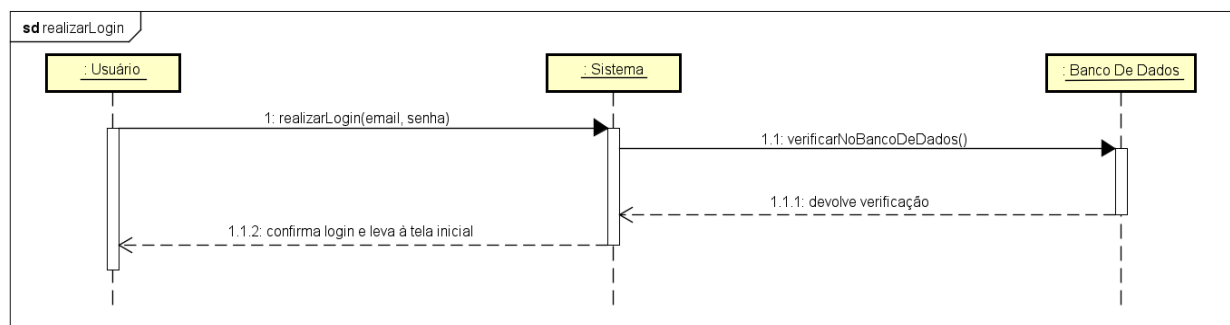


Figura 5.2: DSS - Realizar login.



### 6.3 Realizar check-in:

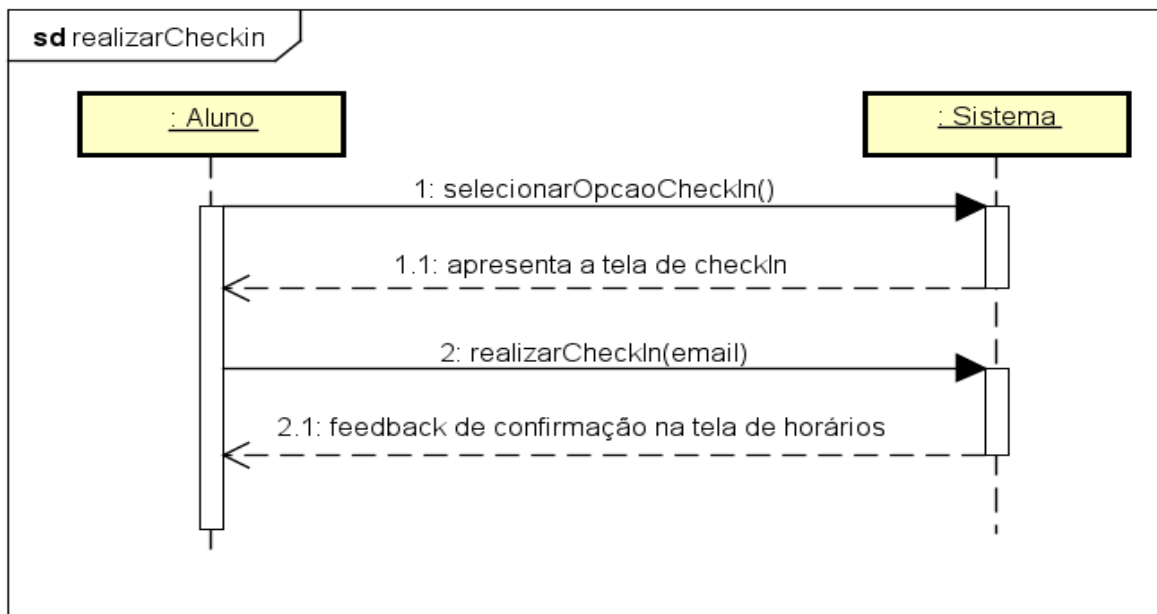


Figura 5.3: DSS - Realizar check-in.

### 6.4 Selecionar horários de embarque:

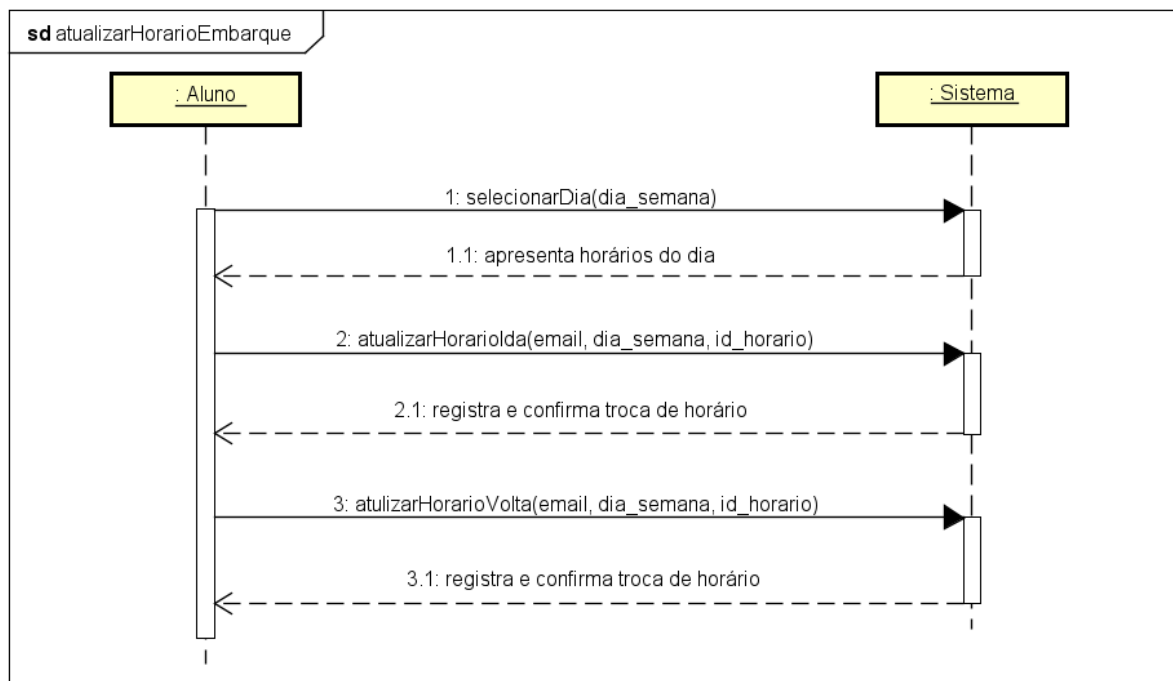


Figura 5.4: DSS - Atualizar horários de embarque.

## 6.5 Atualizar horários de embarque:

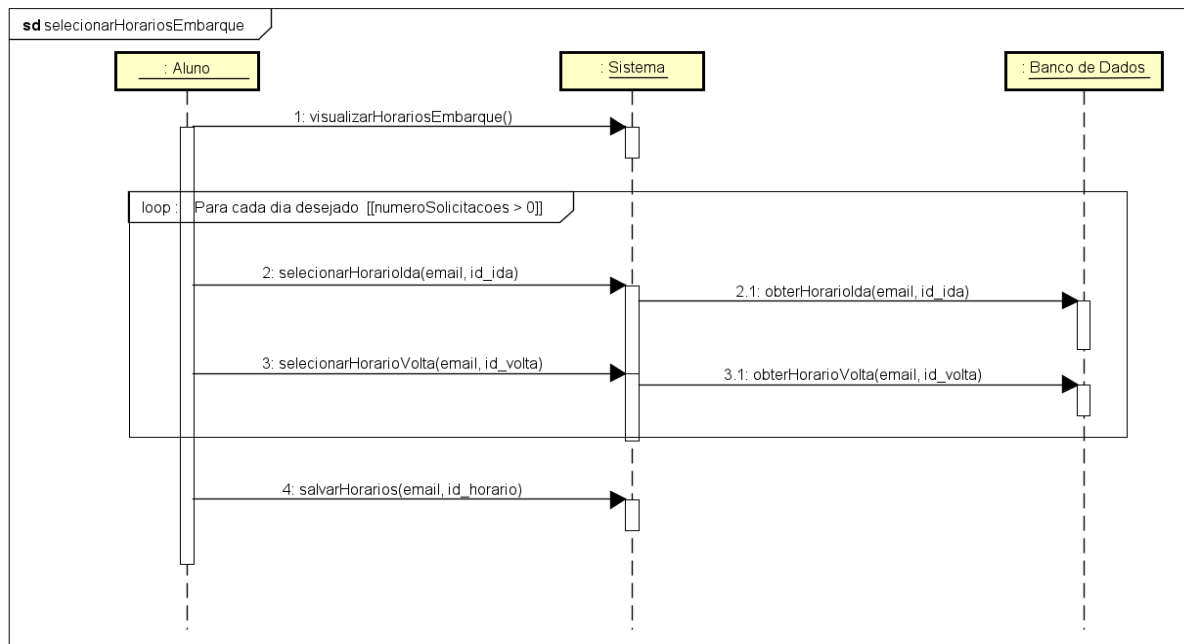


Figura 5.5: DSS - Atualizar horários de embarque.

## 6.6 Listar alunos para ida:

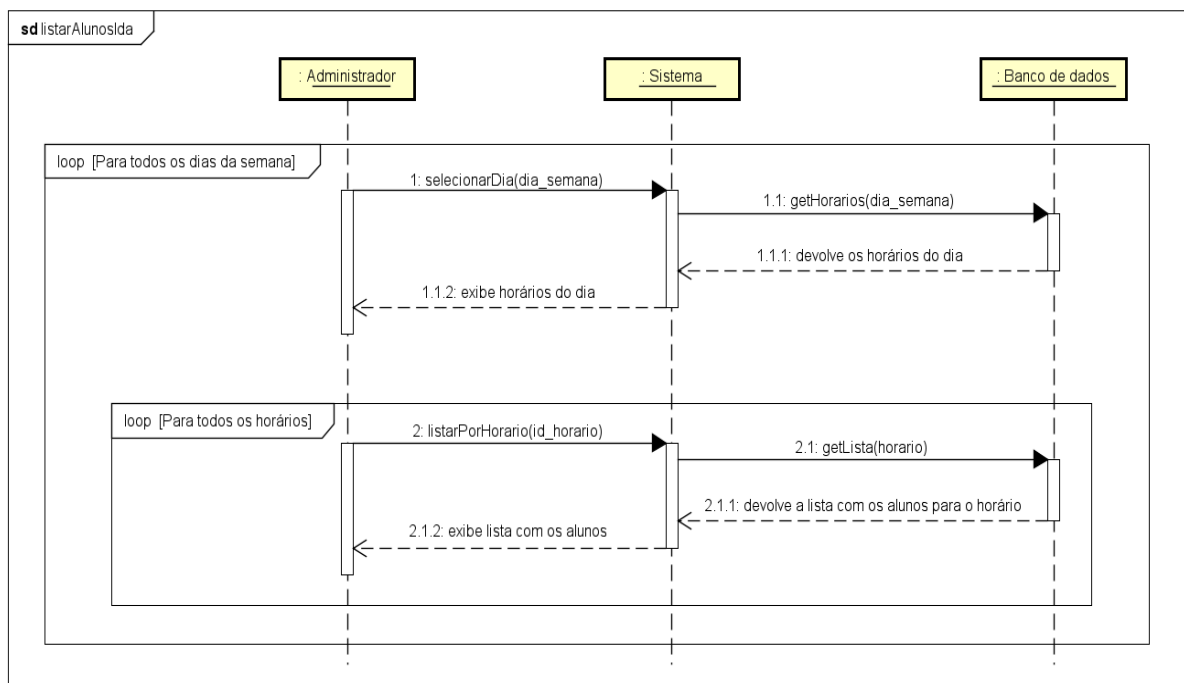


Figura 5.6: DSS - Listar alunos para ida.

## 6.7 Listar alunos para volta:

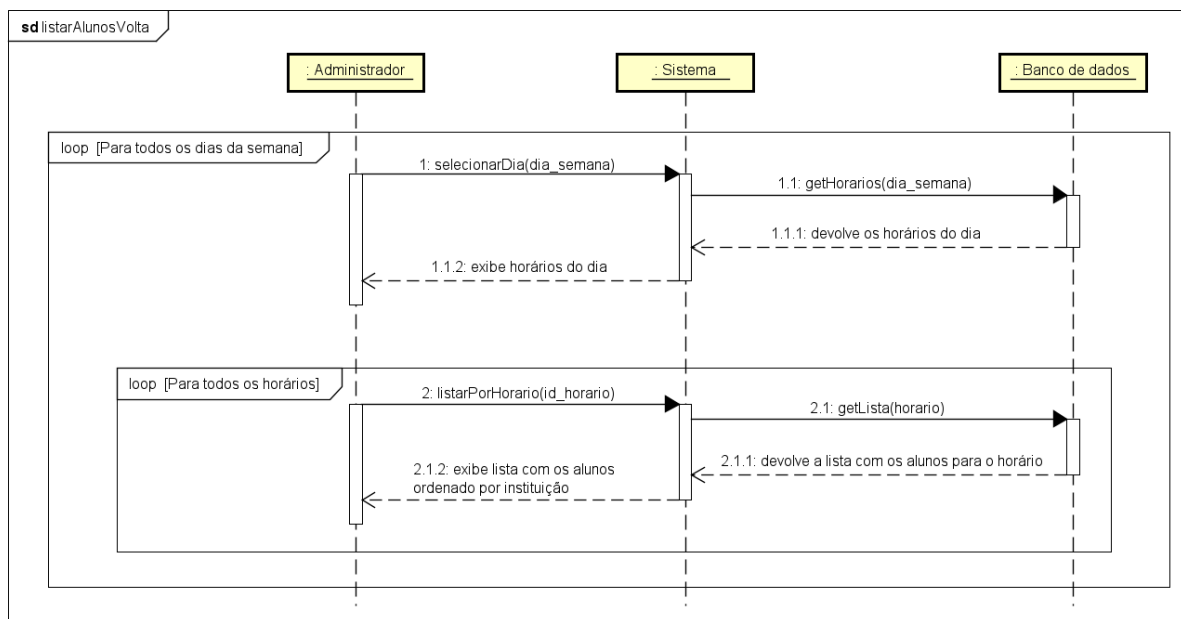


Figura 5.7: DSS - Listar alunos para volta.

## 7. Contratos da Operação (CO)

### 7.1 Operação: realizarLogin(email, senha)

- **Parâmetros:**
  - <email>: identificação do aluno;
  - <senha>: senha de acesso do aluno.
- **Referências Cruzadas:**
  - Caso de uso: “Realizar Login”.
- **Pós-Condições:**
  1. Uma instância de login é criada.

### ***7.2 Operação: avaliarCadastro(email)***

- **Parâmetros:**

<email>: identificação do aluno que está sendo avaliado.

- **Referências Cruzadas:**

Caso de uso: “Aceitar cadastro”.

- **Pré-Condições:**

1. Deve existir uma instância Aluno.

- **Pós-Condições:**

1. Os cadastros aprovados são ativados, liberando o acesso dos alunos ao sistema;
2. Os cadastros rejeitados têm a instância Aluno excluída do sistema.

### ***7.3 Operação: selecionarHorarioIda(email, id\_ida)***

- **Parâmetros:**

<email>: identificação do aluno pelo e-mail;

<id\_ida>: identificação do horário de ida do dia.

- **Referências Cruzadas:**

Caso de uso: “Selecionar horários de embarque próprios”.

- **Pré-Condições:**

1. O aluno correspondente ao e-mail deve estar logado;
2. Os horários de ida devem estar previamente cadastrados no sistema.

- **Pós-Condições:**

1. Os horários de ida identificados por id\_ida são associados à instância Aluno referente ao e-mail informado;
2. Mantém os horários selecionados.

#### **7.4 Operação: *selecionarHorarioVolta(email, id\_volta)***

- **Parâmetros:**

<email>: identificação do aluno pelo e-mail;

<id\_volta>: identificação do horário de volta do dia.

- **Referências Cruzadas:**

Caso de uso: “Selecionar horários de embarque próprios”.

- **Pré-Condições:**

1. O aluno correspondente ao e-mail deve estar logado;
2. Os horários de volta devem estar previamente cadastrados no sistema.

- **Pós-Condições:**

1. Os horários de volta identificados por id\_volta são associados à instância Aluno referente ao e-mail informado;
2. Mantém os horários selecionados.

#### **7.5 Operação: *salvarHorarios(email, id\_horario)***

- **Parâmetros:**

<email>: identificação do aluno pelo e-mail;

<id\_horario>: identifica os horários do aluno para cada dia;

- **Referências Cruzadas:**

Caso de uso: “Selecionar horários de embarque próprios”.

- **Pré-Condições:**

1. Aluno ter selecionado seus horários de ida e volta de cada dia.

- **Pós-Condições:**

1. Salva os horários do aluno na tabela “preset\_horario”, associados à instância Aluno correspondente;
2. Cria uma cópia dos horários do preset na tabela “Horario\_atualizado”.

### ***7.6 Operação: atualizarHorarioIda(email, dia\_semana, id\_horario)***

- **Parâmetros:**

<email>: identificação do aluno pelo e-mail;

<id\_horario>: identifica os horários do aluno para cada dia.

- **Referências Cruzadas:**

Caso de uso: “Atualizar horários de embarque” - ida.

- **Pré-Condições:**

1. O aluno deve estar logado no sistema;
2. A instância Aluno deve possuir os horários de ida definidos previamente;

- **Pós-Condições:**

1. O horário de ida do dia especificado é atualizado na instância Horario\_atualizado do aluno, substituindo a anterior.

### ***7.7 Operação: atualizarHorarioVolta(email, id\_horario, dia\_semana)***

- **Parâmetros:**

<email>: identificação do aluno pelo e-mail;

<id\_horario>: identifica os horários do aluno para cada dia.

- **Referências Cruzadas:**

Caso de uso: “Atualizar horários de embarque” - volta.

- **Pré-Condições:**

1. O aluno deve estar logado no sistema;
2. A instância Aluno deve possuir os horários de volta definidos previamente;

- **Pós-Condições:**

1. O horário de ida do dia informado é atualizado na instância Horario\_atualizado do aluno, substituindo a anterior.

### **7.8 Operação: *selecionarDia(dia\_semana)***

- **Parâmetros:**

<dia\_semana>: identificação do dia da semana selecionado.

- **Referências Cruzadas:**

Caso de uso: “Listar alunos para ida”;

Caso de uso: “Listar alunos para volta”.

- **Pré-Condições:**

1. O administrador/motorista, ou o aluno, deve estar devidamente autenticado no sistema;
2. Os horários de embarque de ida/volta devem estar cadastrados em seus respectivos dias da semana.

- **Pós-Condições:**

- O sistema exibe a lista de horários de ida/volta do dia selecionado;

### **7.9 Operação: *listarPorHorario(id\_horario)***

- **Parâmetros:**

<id\_horario>: identifica os horários de ida do dia em questão;

- **Referências Cruzadas:**

Caso de uso: “Listar alunos para ida”;

Caso de uso: “Listar alunos para volta”.

- **Pré-Condições:**

1. O administrador/motorista deve estar autenticado no sistema;
2. Os alunos já estarem com seus horários de ida cadastrados ou realizado alterações temporárias;
3. O dia da semana para listagem deve ter sido selecionado.

- **Pós-Condições:**

1. O sistema exibe a lista de alunos associado ao horário de ida ou volta informado.

### 7.10 Operação: *realizarCheckin(email)*

- **Parâmetros:**

<email>: Identificação do aluno por meio do e-mail.

- **Referências Cruzadas:**

Caso de uso: “Realizar check-in”

- **Pré-Condições:**

1. A instância do Aluno deve estar cadastrada e ativa;
2. O aluno deve estar logado no sistema;
3. O cadastro do aluno deve ter sido aceito pelo administrador;
4. Estar na lista de alunos que voltarão neste horário.

- **Pós-Condições:**

1. O check-in é registrado no sistema;
2. A lista de alunos que embarcaram é atualizada para visibilidade do motorista.

## 8. Diagramas de Comunicação (DC)

### 8.1 Realizar login

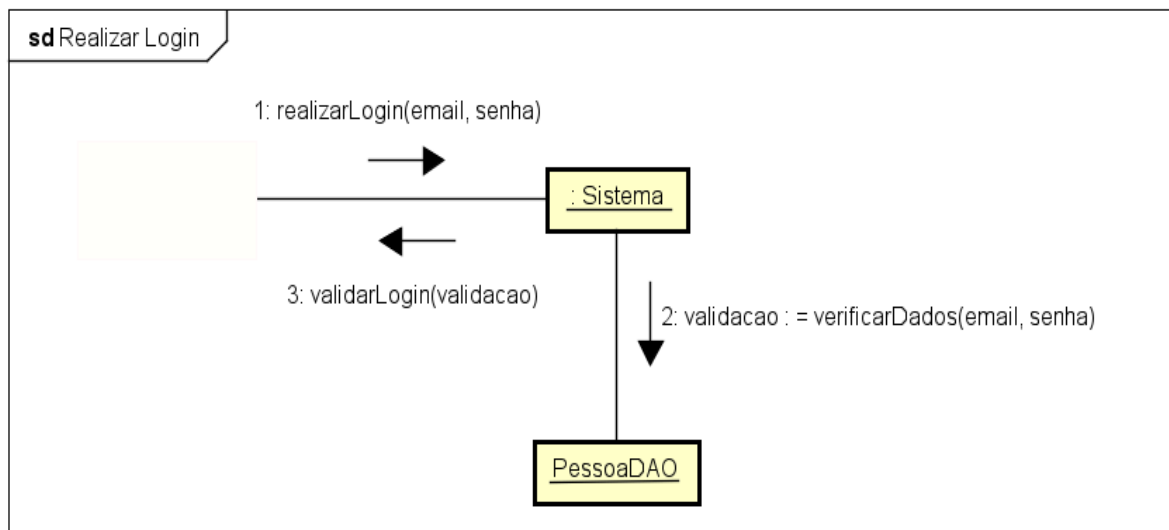


Figura 6.1: DC - Realizar login.



## 8.2 Avaliar cadastros

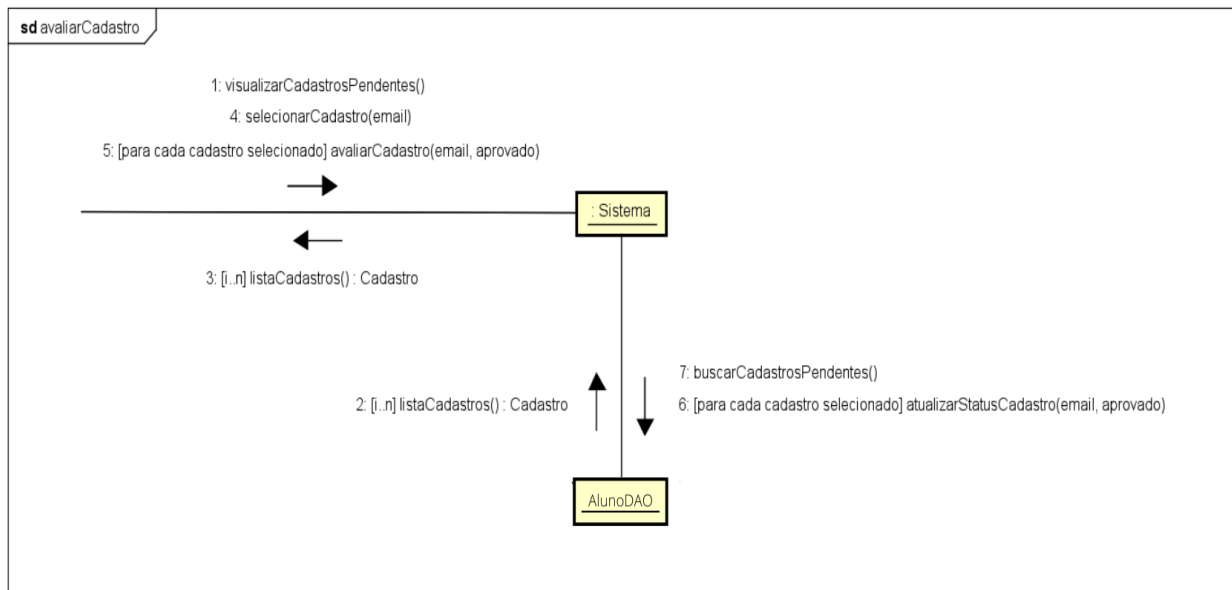


Figura 6.2: DC - Avaliar cadastros.

## 8.3 Selecionar horários de embarque

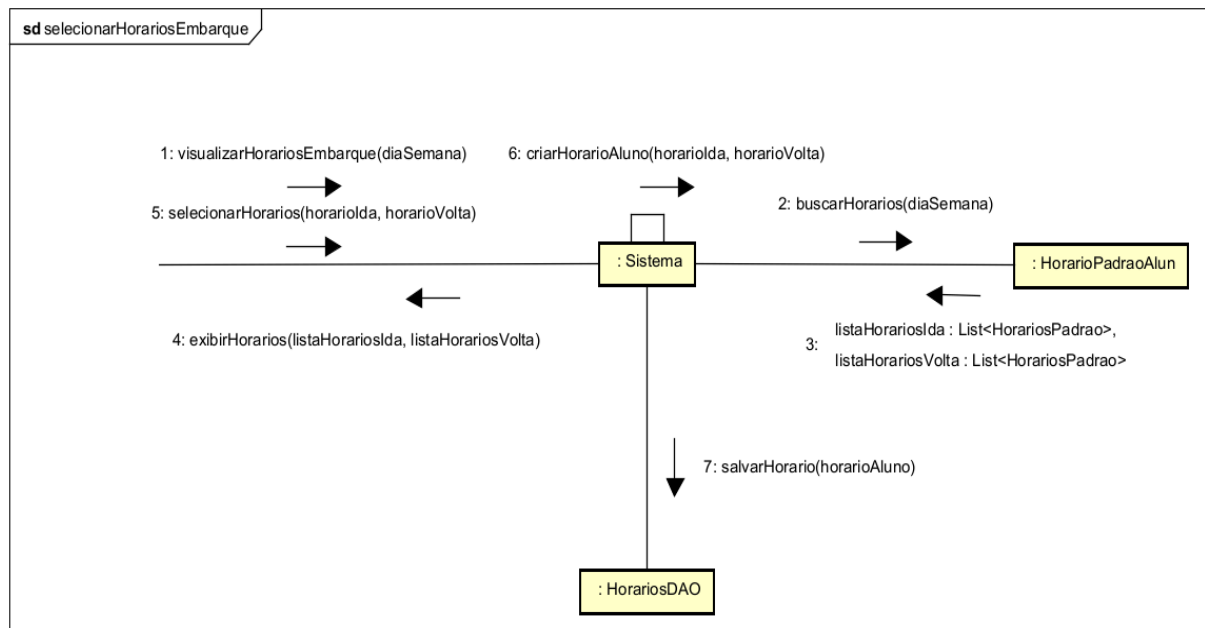


Figura 6.3: DC - Selecionar horários de embarque.

## 8.4 Atualizar horários de embarque

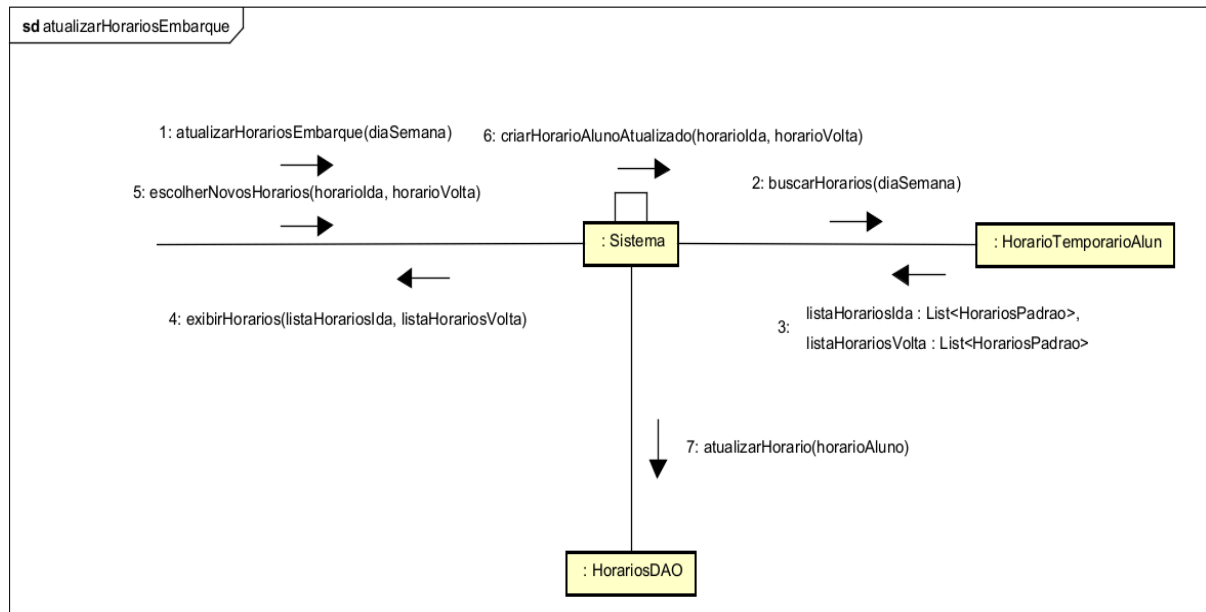


Figura 6.4: DC - Atualizar horários de embarque.

## 8.5 Listar alunos para ida

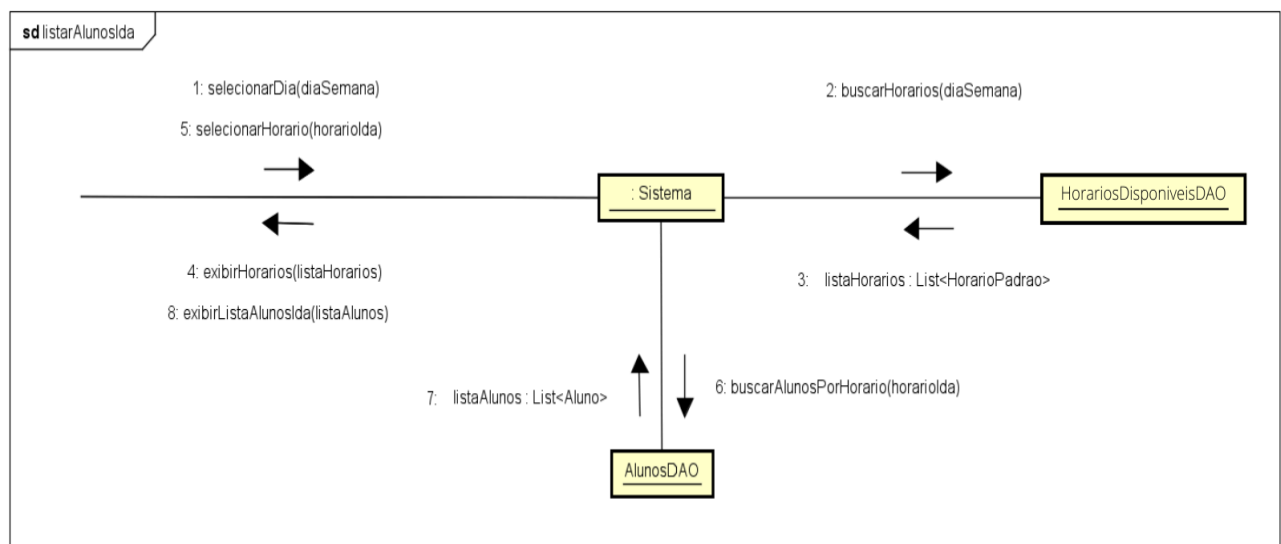


Figura 6.5: DC - Listar alunos para ida.

## 8.6 Listar alunos para volta

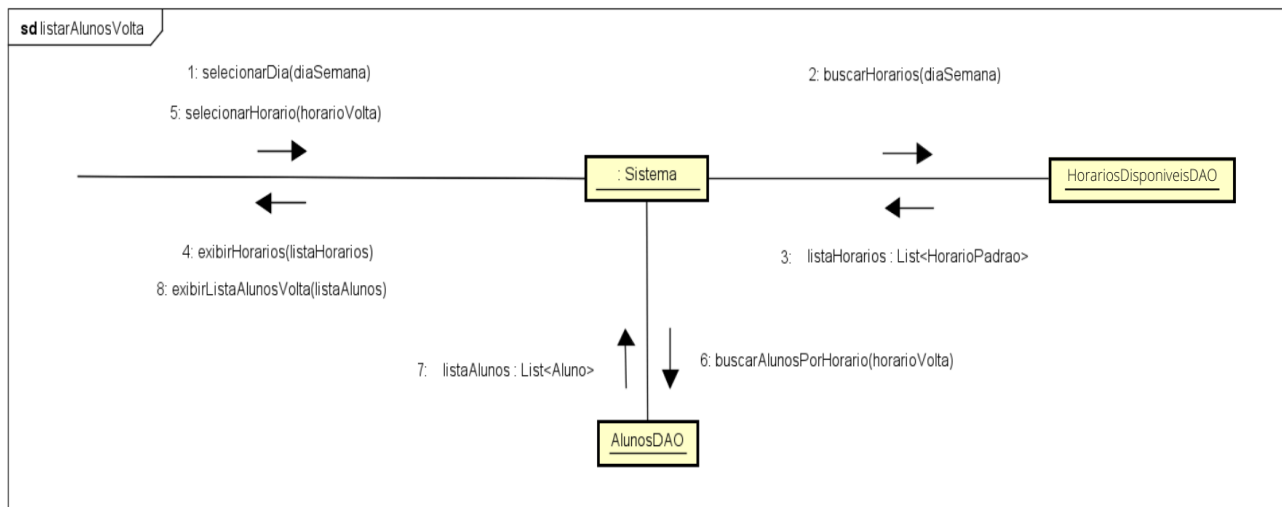


Figura 6.6: DC - Listar alunos para volta.

## 8.7 Realizar check-in

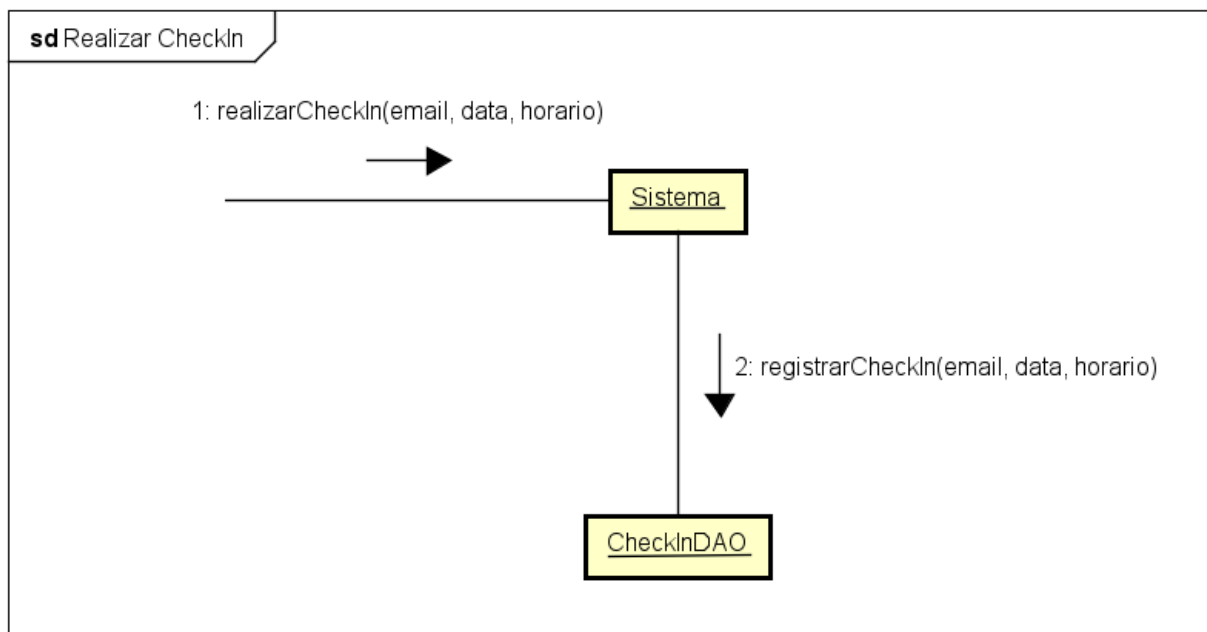


Figura 6.7: DC - Realizar check-in.

## 8.8 Salvar Login

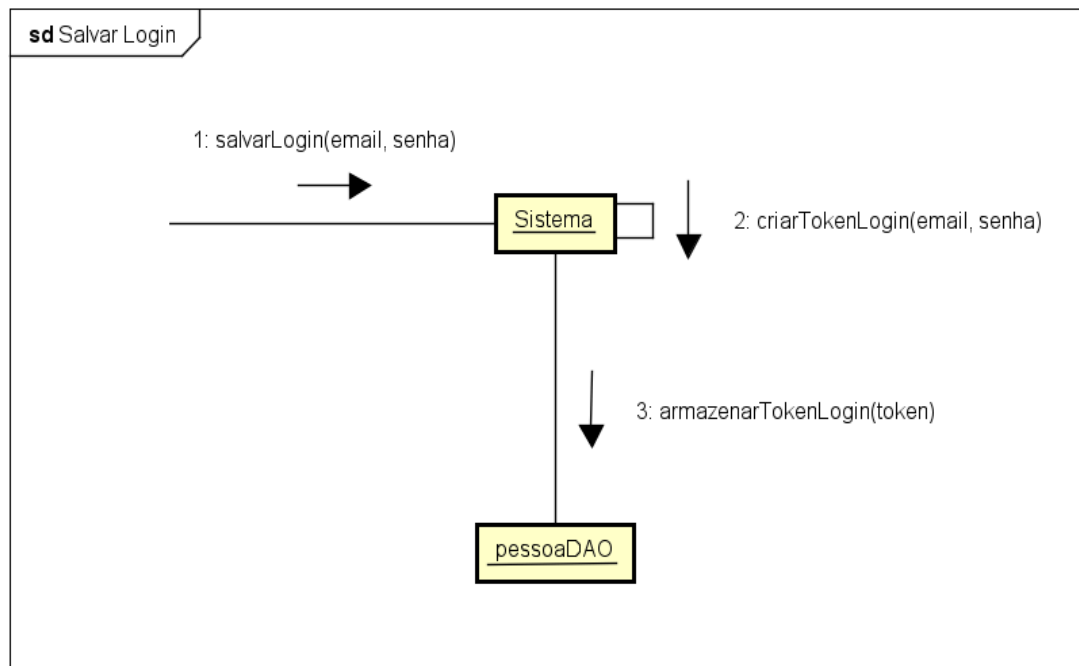


Figura 6.8: DC - Salvar login.

## 8.9 Resetar horários para o preset

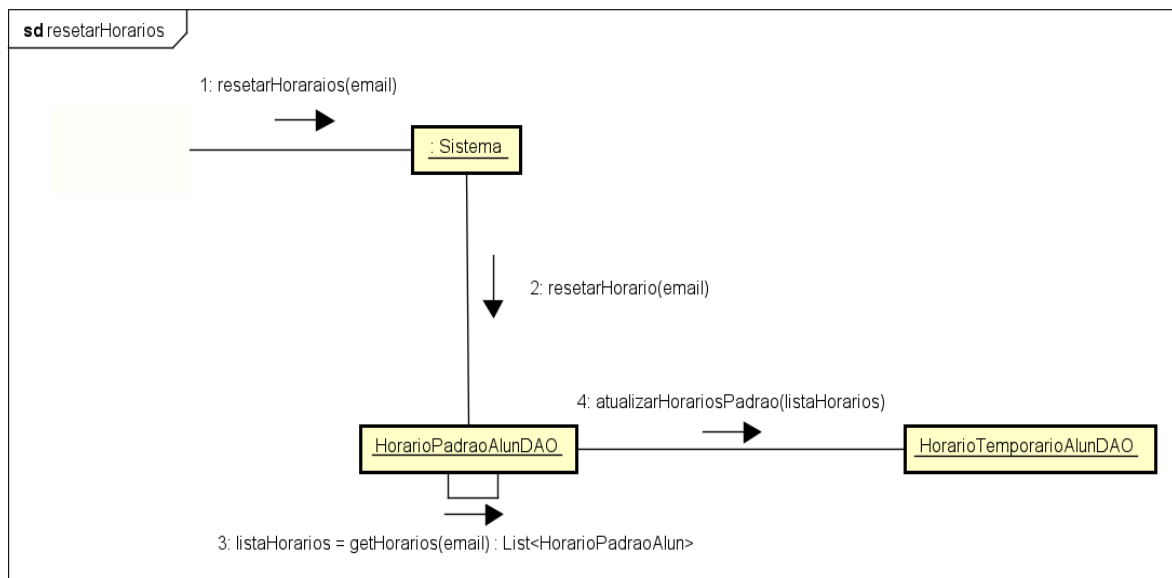


Figura 6.9: DC - Resetar horários.

### 8.10 Enviar notificações perto do horário de embarque

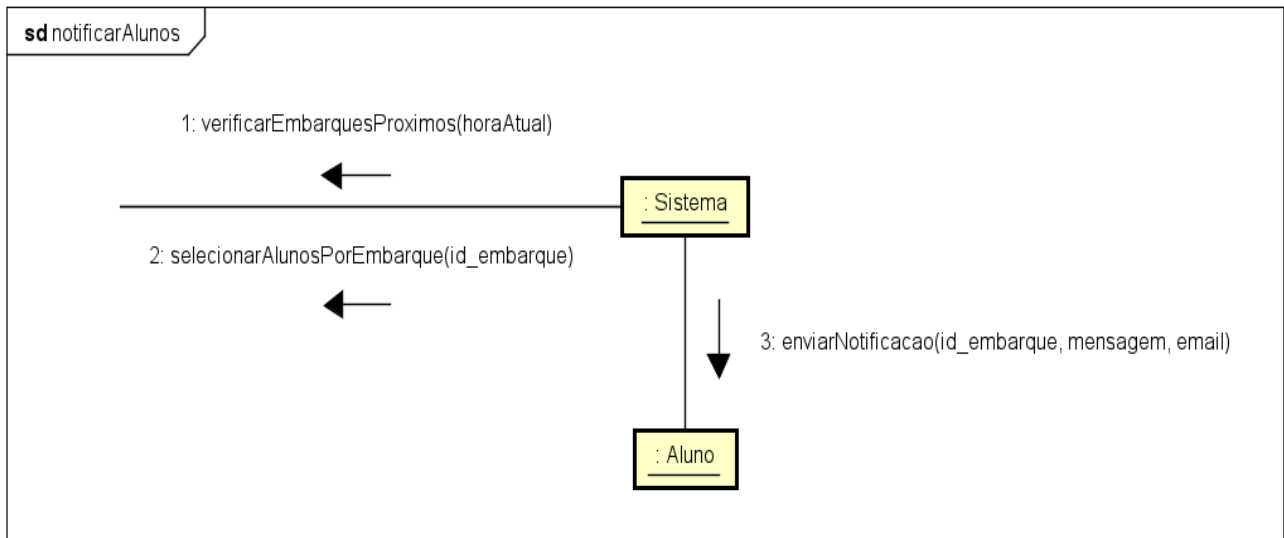


Figura 6.9: DC - Notificar alunos.

## 9. Diagrama de Classes em Tempo de Projeto (DCP)

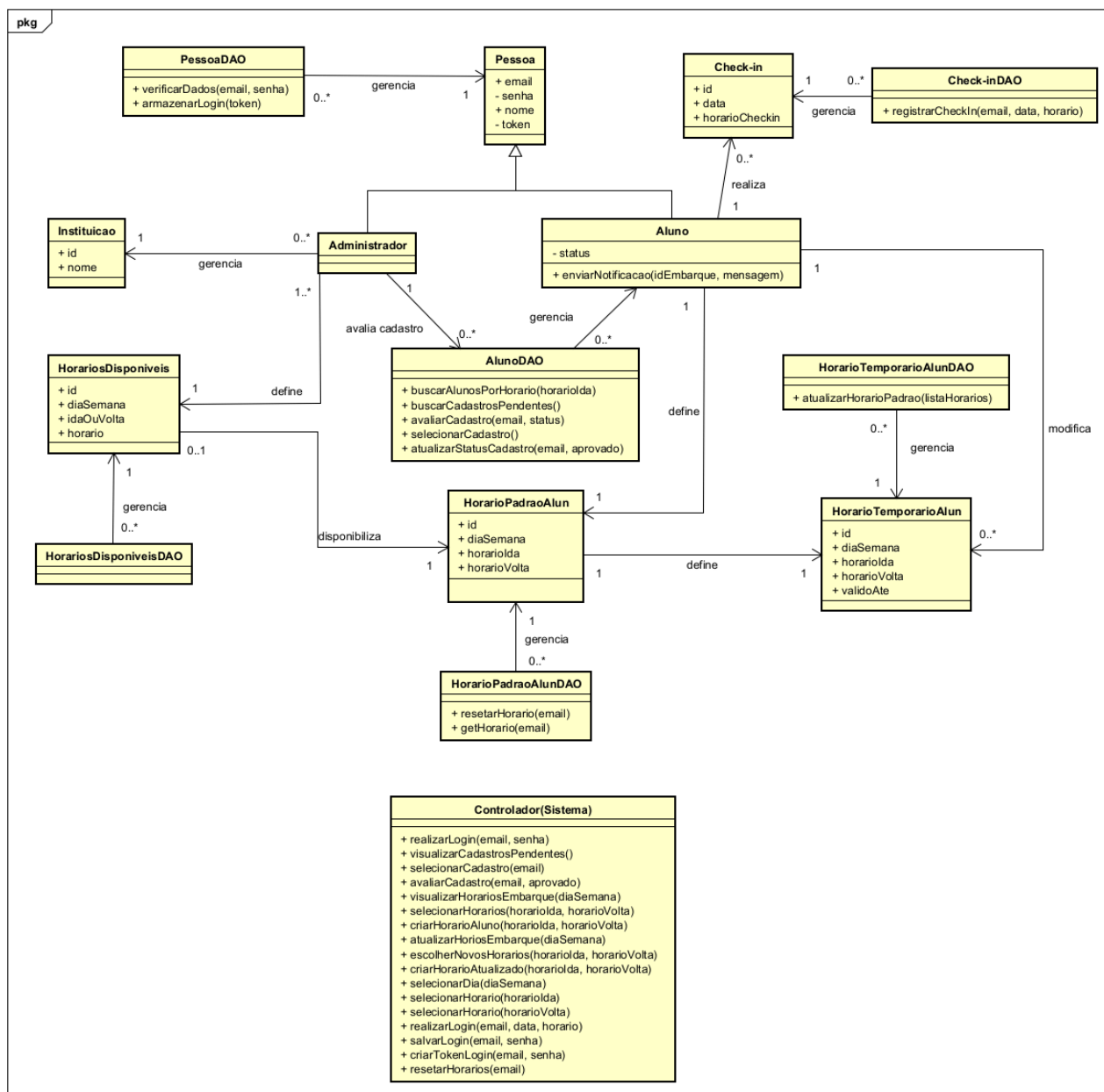


Figura 7: DCP - Diagrama de Classes em Tempo de Projeto.