

Universidade do Minho

Escola de Engenharia

Licenciatura em Engenharia Informática

Gestão de Turnos

Desenvolvimento de Sistemas de Software

Trabalho Prático - Fase 1

Grupo 11

Link GitHub - <https://github.com/LEI-DSS/DSS2425-Grupo-11.git>



1 Introdução	3
2 Diagramas Estruturais	4
2.1 Modelo de Domínio	4
3.2 Diagrama Use Case	6
3 Conclusões	13

1 Introdução

Este documento tem como objetivo apresentar o desenvolvimento de um sistema de gestão de turnos práticos para a Unidade Curricular ***Desenvolvimento de Sistemas Software*** no contexto do ano letivo 2024/2025. O projeto foca-se em resolver o problema da alocação de alunos, de forma automatizada, em turnos de diferentes disciplinas de uma maneira eficiente e sem quaisquer sobreposições. Além disso, quando necessário, também deve ser possível realizar ajustes manualmente.

Em cursos de grande dimensão, como é o caso de Engenharia Informática, a distribuição dos alunos pelos diversos turnos é uma tarefa complexa e crítica para o bom funcionamento das aulas teóricas e especialmente das práticas. Sendo assim, a gestão manual da alocação de turnos, torna-se um processo bastante demorado e exaustivo devido a possíveis sobreposições de turnos que se deve ter em atenção.

A nossa solução propõe a centralização da criação e gestão dos turnos sob a responsabilidade do Diretor de Curso. Para isso, fazemos uso de um sistema que automatiza o processo de alocação dos alunos, com base em listas de inscritos e preferências definidas para cada UC através do docente responsável pela mesma. O sistema será capaz de lidar com diferentes algoritmos de priorização de alunos, incluindo, por exemplo, a priorização de trabalhadores-estudantes ou de alunos que apresentem o estatuto de atleta. Dito isto, o mesmo terá de ser robusto e escalável para lidar com um grande número de alunos e UCs .

2 Diagramas Estruturais

2.1 Modelo de Domínio

Aluno:

- **Descrição:** Representa os estudantes matriculados num curso. Inicialmente não estão associados a turnos.
- **Relações:**
 - Está inscrito em um ou mais Turnos.
 - Pode possuir ou não um Estatuto (Trabalhador-Estudante ou Atleta), o que influencia na alocação dos turnos.
 - Cada aluno tem uma Média. No entanto vários alunos podem ter a mesma média
 - Um aluno pertence a um e apenas um Curso.

Turno:

- **Descrição:** Representa uma aula, de uma UC, que pode ser Teórica ou Prática.
- **Relações:**
 - Está associado a uma UC, que pode ser Obrigatória ou Opcional.
 - Um conjunto de turnos forma um Horário.
 - Ocorre num determinado Dia da Semana.
 - Decorre numa Sala, que tem uma Capacidade máxima.
 - Respeita Restrições solicitadas pelos docentes das UCs e que foram definidas no sistema pelo diretor do curso.

Horário:

- **Descrição:** Objeto que o aluno possui e que é composto por um conjunto de turnos.
- **Relações:**
 - Um Horário é um conjunto de turnos.
 - Um aluno possui um Horário.
 - Vários alunos podem ter o mesmo Horário.

UC (Unidade Curricular):

- **Descrição:** Representa uma disciplina ou matéria lecionada num curso. Pode ser Obrigatória ou Opcional.
- **Relações:**
 - Cada UC é administrada por um Curso e pode ter vários Turnos (Teóricos e Práticos).
 - Está associada a um Ano e Semestre em que é lecionada.

Curso:

- **Descrição:** Representa o programa de estudo ao qual o Aluno pertence e é constituído por um conjunto de UCs.
- **Relações:**
 - Um Curso tem várias UCs.
 - Cada curso tem um Diretor, responsável por gerir e definir as regras de alocação dos turnos.

Sala:

- **Descrição:** Representa o local onde um Turno é realizado.
- **Relações:**
 - Cada Sala tem uma Capacidade máxima que define o número de alunos que pode acomodar.

Restrições:

- **Descrição:** Representa as condições aplicadas à alocação de turnos.
- **Relações:**
 - Uma mesma regra pode ser aplicada em diversos turnos, no entanto um turno pode também ter várias restrições definidas.

Estatuto:

- **Descrição:** Representa o estado especial de um aluno, como Trabalhador-Estudante ou Atleta, que pode influenciar a prioridade na alocação de turnos. Um aluno pode não ter um estatuto especial.
- **Relações:**
 - Um Aluno pode ou não ter um estatuto especial que impacta a alocação de turnos.

Média:

- **Descrição:** Refere-se à média acadêmica do aluno, que pode ser utilizada como priorização na escolha de turnos.
- **Relações:**
 - Cada Aluno possui apenas uma média associada, que pode ser considerada nas alocações.

Diretor:

- **Descrição:** Representa o responsável pelo Curso, que define as regras de alocação de turnos e gere o processo de geração de horários.
- **Relações:**
 - O Diretor é apenas responsável por um curso.

Ano e Semestre:

- **Descrição:** Usados para definir em que período as UCs são administradas.
- **Relações:**
 - As UCs estão associadas a um Ano e Semestre específicos, definindo quando elas são lecionadas.

2.1 Arquitetura Geral

O diagrama apresentado descreve a **arquitetura geral** do sistema de gestão de horários, ilustrando as suas principais camadas e componentes. Essa arquitetura é baseada no padrão “**Camadas**” (Layered Architecture), que organiza os elementos do sistema em camadas com responsabilidades bem definidas. Cada camada interage apenas com as adjacentes, promovendo o desacoplamento e facilitando a manutenção e evolução do sistema.

Estrutura do Diagrama

1. Camada ui (Interface do Utilizador)

Esta camada é responsável por gerir a interação com o utilizador, servindo como ponto de entrada do sistema. No diagrama, ela contém dois componentes principais:

- Menu:
 - Um componente genérico responsável por gerenciar as opções disponíveis ao utilizador.
 - Atua como a entrada inicial, onde são direcionadas ações específicas para alunos ou diretores de curso.
- GestTurnosUI:
 - Este componente é especializado em lidar com as operações relacionadas à gestão de turnos.
 - Interage diretamente com a camada de **negócio** (business) por meio das interfaces definidas, como ISSUtilizadores, ISSUC, e ISSHorario.

2. Camada business (Lógica de Negócio)

A camada de **lógica de negócio** contém as principais funcionalidades do sistema e é composta por três subsistemas bem definidos:

- SSUtilizadores (**Subsistema de Utilizadores**):
 - Responsável pela gestão de utilizadores (alunos e diretores).
 - Apresenta uma **interface** (ISSUtilizadores) que define as operações principais, como:
 - importarAlunos(String filepath): Importa uma lista de alunos a partir de um ficheiro.
 - login(String id, String pass): Permite autenticar um utilizador no sistema.
 - mostrarAlunos(): Retorna uma lista de alunos registados.
 - registarAluno(): Regista novos alunos no sistema.
 - O componente UtilizadoresFacade implementa a interface, garantindo que todas as operações relacionadas aos utilizadores sejam centralizadas. Este design facilita a extensão e substituição da implementação no futuro.
- SSUC (**Subsistema de Unidades Curriculares**):
 - Trata da gestão das Unidades Curriculares (UCs) e seus turnos.
 - A interface ISSUC define operações para:
 - Criar UCs e seus turnos.
 - Definir restrições para turnos.

- O componente UCFacade implementa esta interface, gerindo os dados relacionados às UCs e turnos.
- SSHorario (**Subsistema de Horários**):
 - Focado na geração e consulta de horários para alunos e cursos.
 - A interface ISSHorario oferece métodos como:
 - `alocarManualmente(String cursoId, String idAluno)`: Aloca alunos manualmente a turnos.
 - `gerarHorarios(String cursoId)`: Gera automaticamente os horários de um curso.
 - `consultarHorario(String idAluno)`: Retorna o horário de um aluno específico.
- O componente HorarioFacade implementa essa interface, garantindo a centralização das operações relacionadas aos horários.

3. Camada data (Persistência de Dados)

Esta camada é responsável pelo acesso e manipulação dos dados persistidos, servindo de abstração para o armazenamento. O diagrama inclui:

- ITurnoData:
 - Uma interface que define as operações relacionadas aos dados dos turnos.
 - Apesar de apenas este DAO ser representado no diagrama, outros DAOs podem existir para entidades como Aluno, UC, e Horario.

A camada data é projetada para isolar a lógica de negócio das particularidades de implementação do armazenamento (por exemplo, MySQL, PostgreSQL), permitindo futuras alterações na tecnologia de persistência sem impacto direto na camada de negócio.

2.2 Diagramas de classe

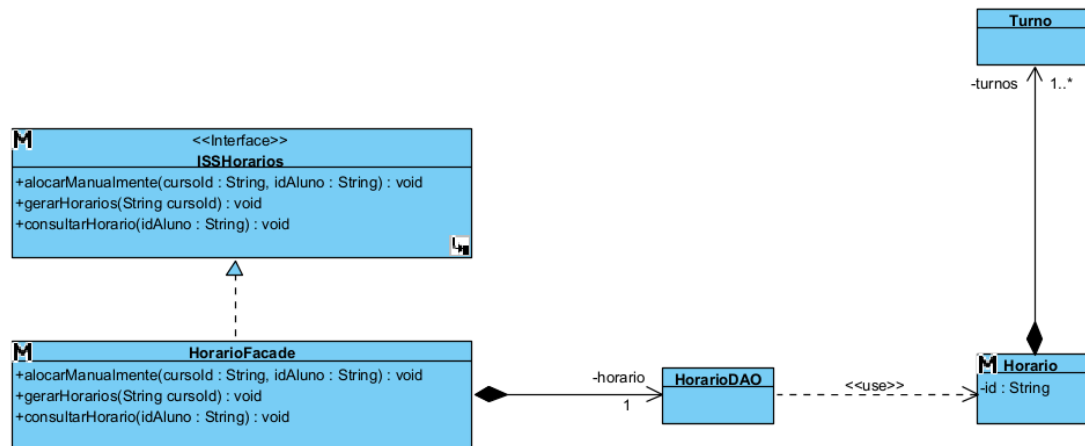


Figura - Diagrama de Classes para SSHorario

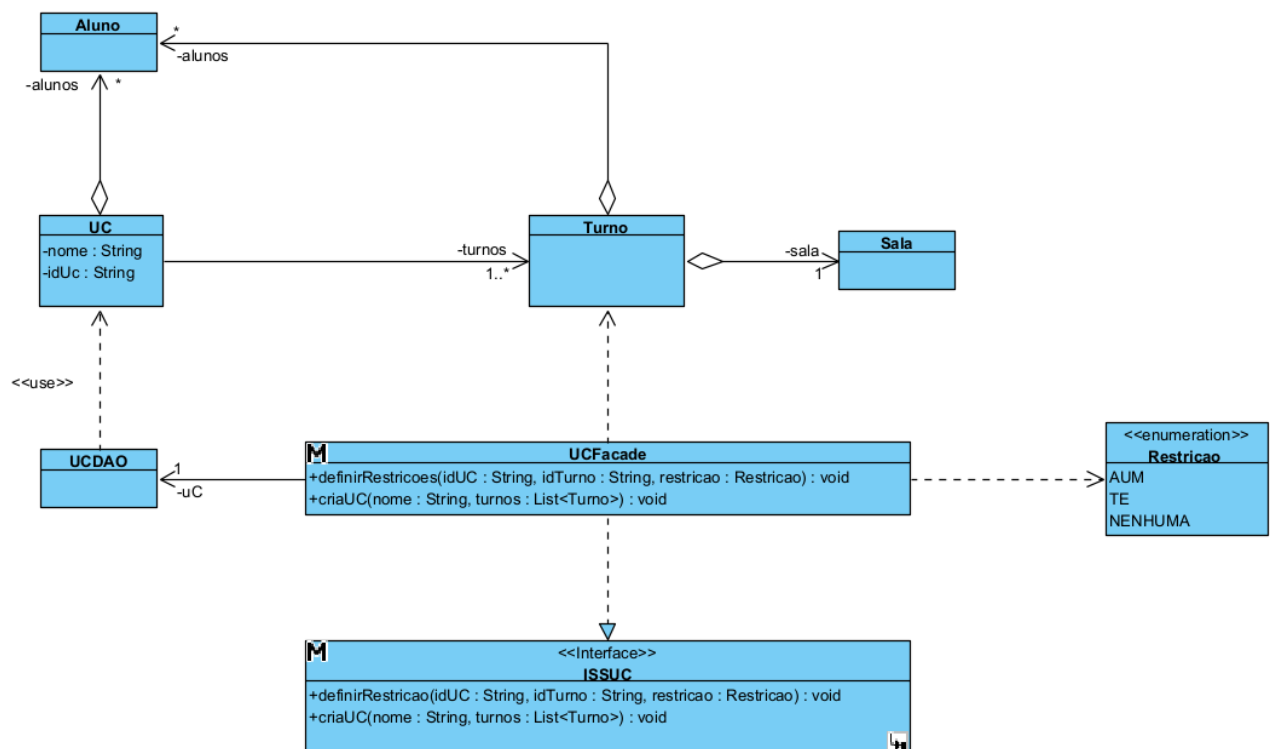


Figura - Diagrama de Classes para SSUC

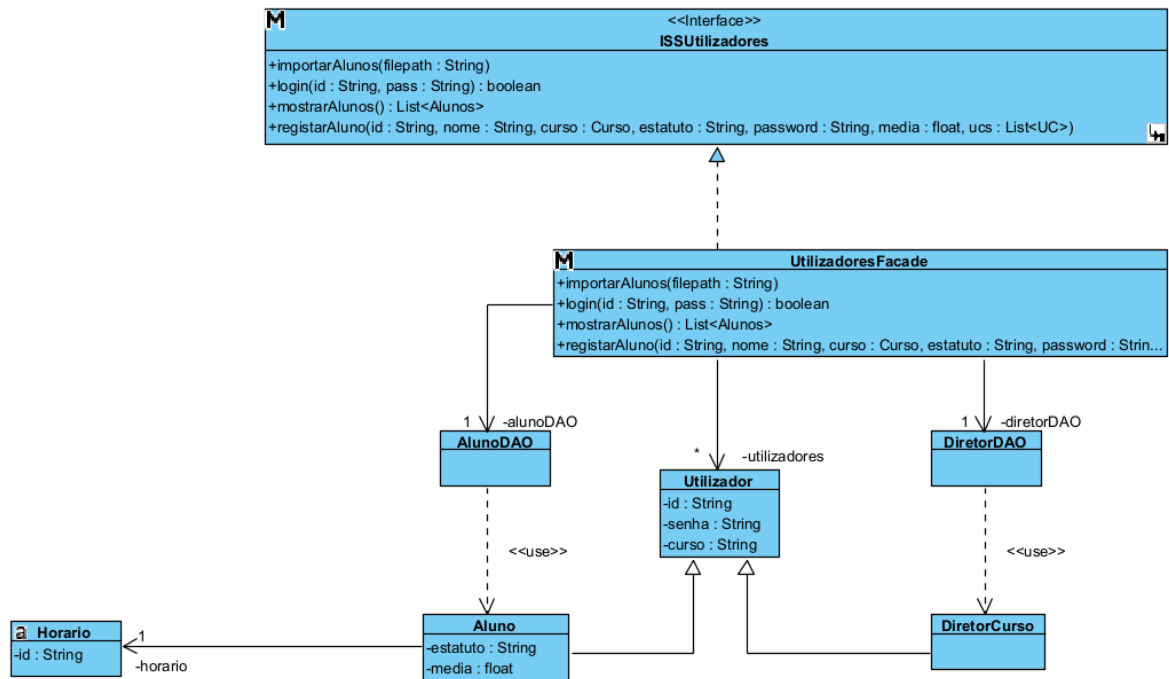


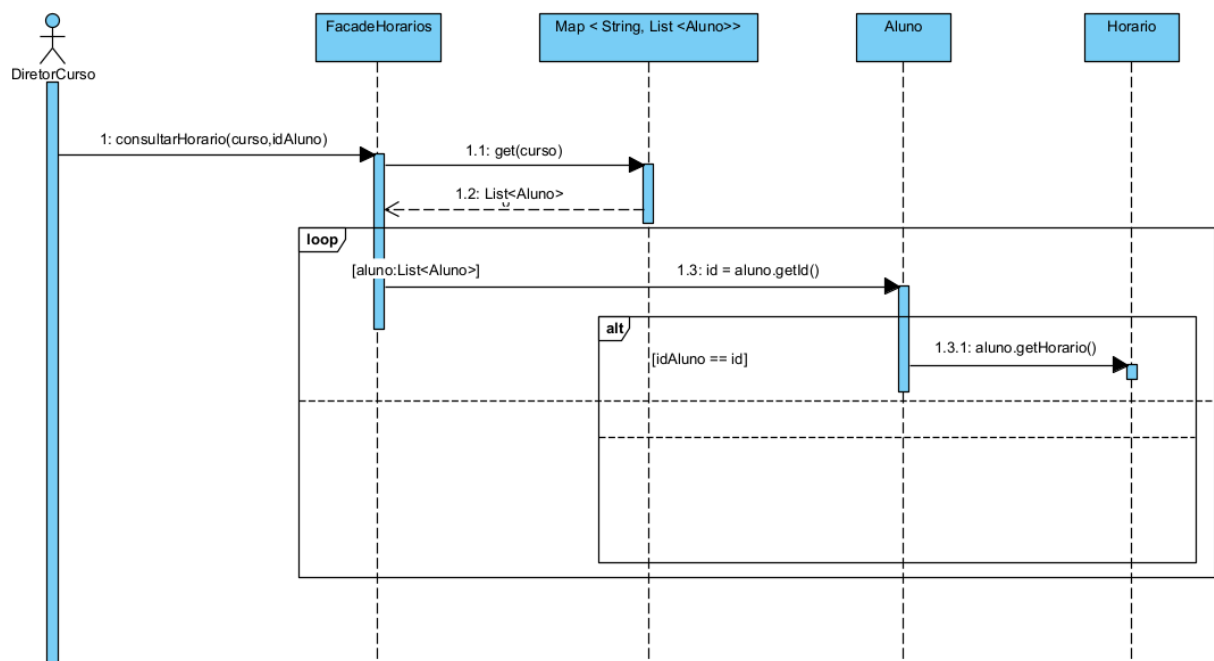
Figura - Diagrama de Classes para SSUtilizadores

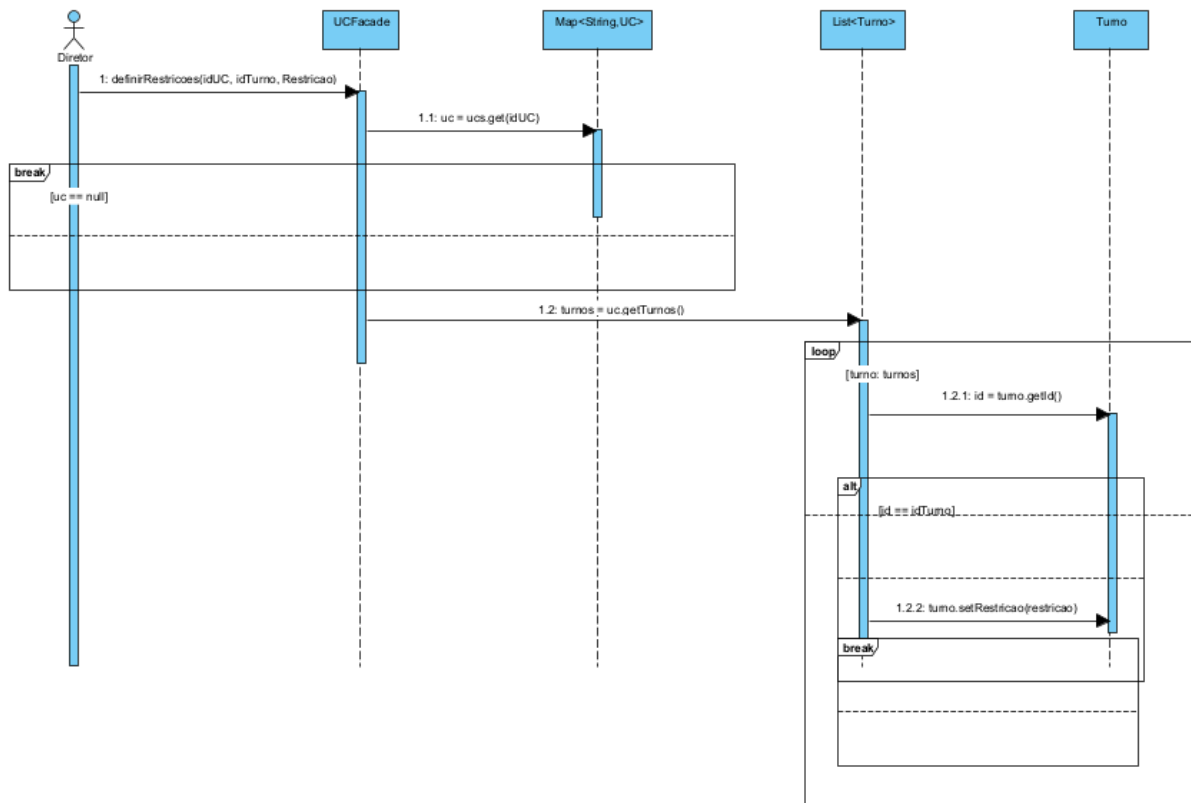
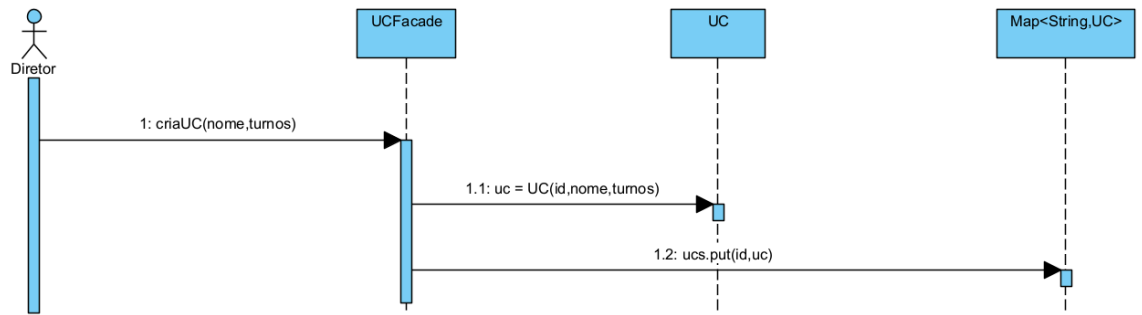
2.2 Diagrama de Componentes

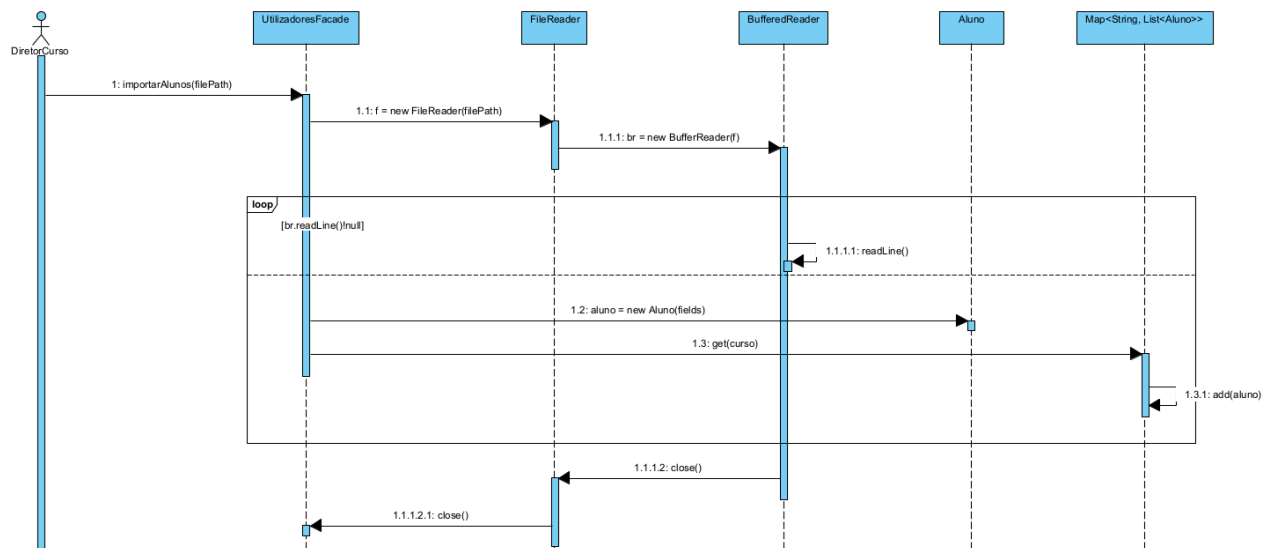
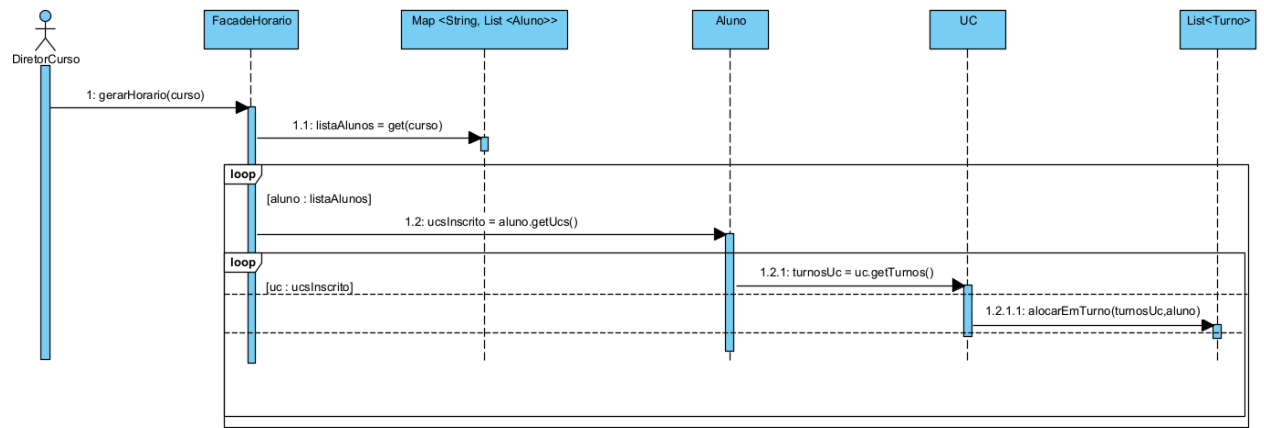
3 Diagramas Comportamentais

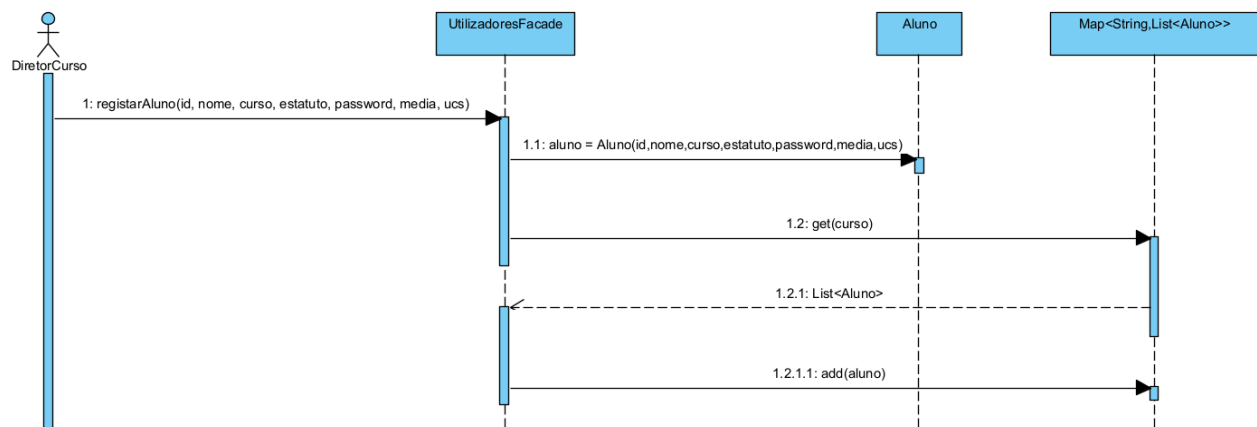
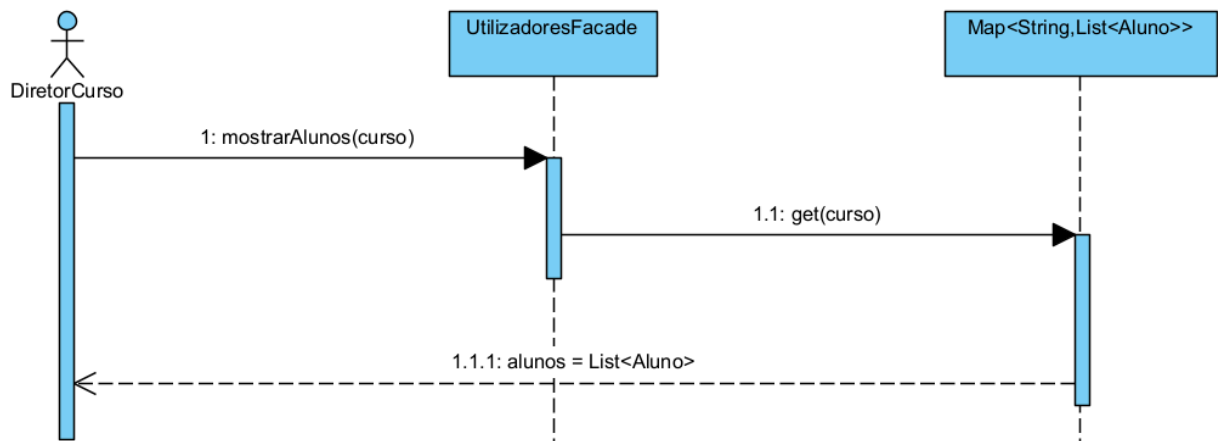
3.1 Diagrama Use Case

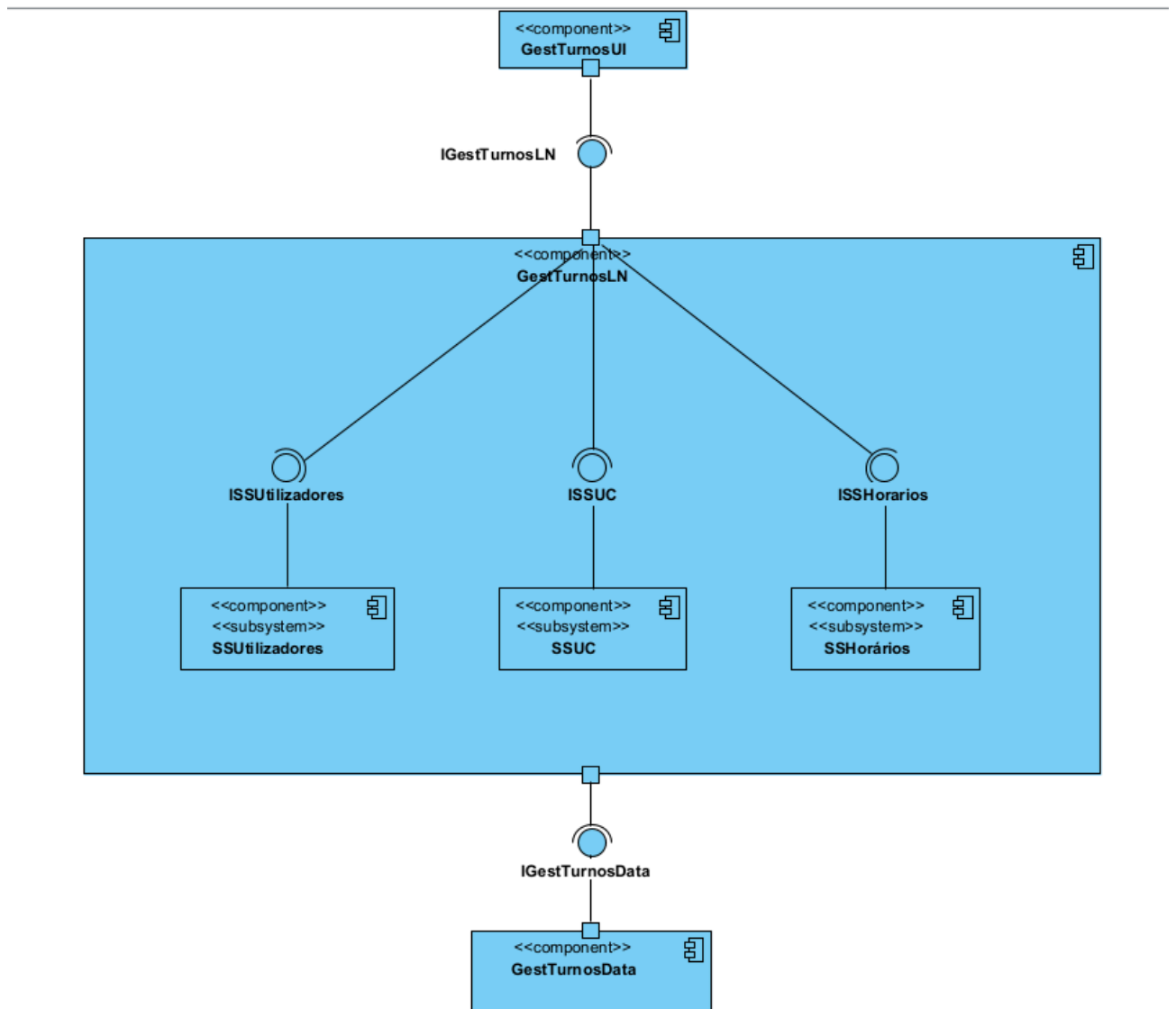
3.2 Diagramas de Sequência

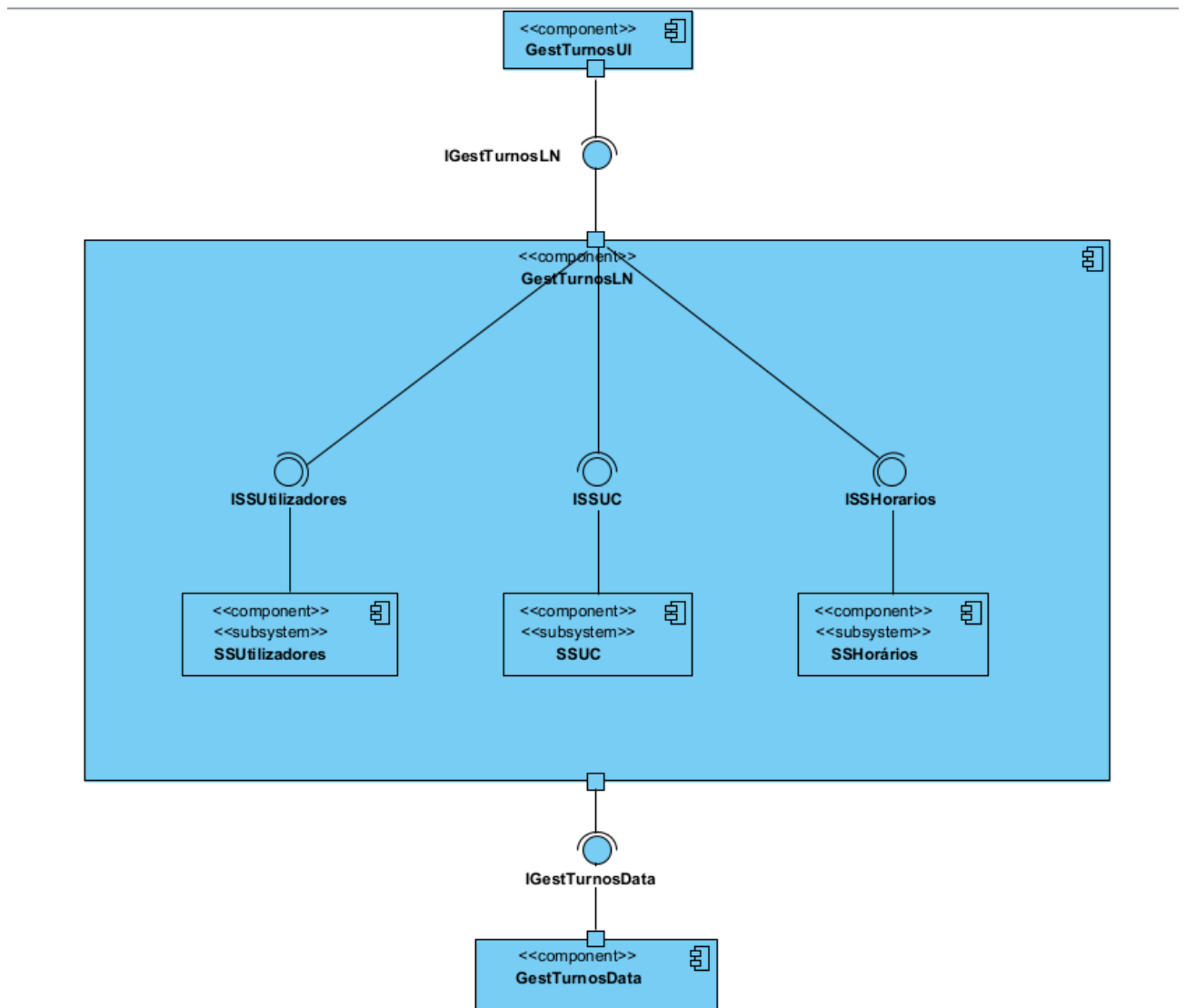












2 Conclusões

Nesta primeira fase do projeto, foram cumpridos os objetivos estabelecidos, que incluíam a definição de um **Modelo de Domínio** claro e abrangente e o desenvolvimento de um **Diagrama de Use Case** que reflete as interações críticas entre os atores e o sistema. A modelação realizada até agora oferece uma base sólida para a próxima fase, onde o foco será a implementação prática do sistema juntamente com as funcionalidades descritas.

O modelo de domínio já contempla a complexidade da gestão de turnos, tendo em consideração a diversidade de alunos, tipos de turnos e a limitação de recursos como salas e horários. Isto permite que o sistema seja escalável e flexível, e esteja pronto para lidar com diferentes cenários de alocação, evitando conflitos e otimizando a alocação de recursos.

Na próxima fase, a implementação deste modelo em código criará um sistema capaz de lidar com a atribuição automática de horários e a alocação eficiente de alunos, consolidando as funcionalidades descritas nesta fase de análise.