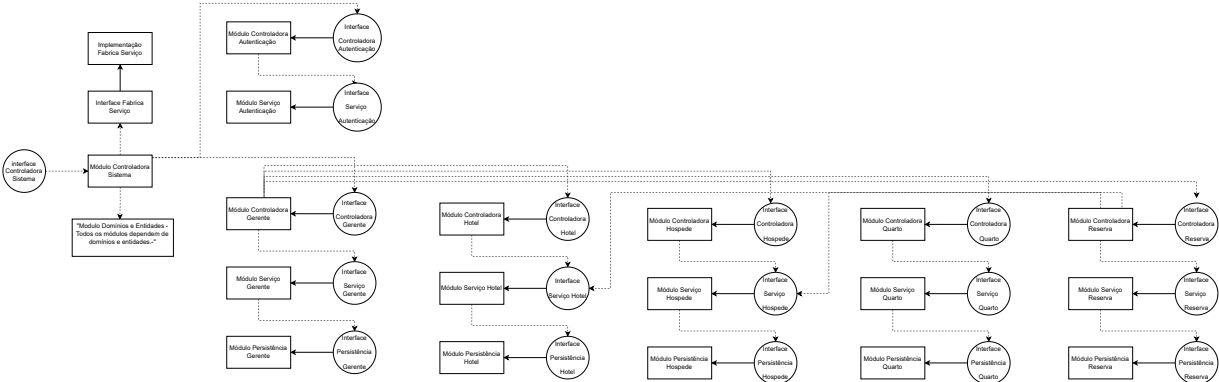


Diagrama UML Hotel



Módulo Controladora Sistema

**Responsabilidade Primária:** Gerenciar o ciclo de vida completo da aplicação, desde o início (main.cpp) até o encerramento. É o disparador principal que concentra a navegação entre os módulos de apresentação de nível superior.

**Funções Chave:** Inicia o fluxo de autenticação (ControladoraAutenticacao) e, após o login bem-sucedido, gerencia a navegação para o Menu Principal (ControladoraGerente).

**Dependências Chave:** Depende da 'Fábrica de Serviços' para obter as sub-controladoras e das interfaces ControladoraAutenticacao e ControladoraGerente para gerenciar o fluxo de navegação.

Módulo Fábrica de Serviço Impl

**Responsabilidade Primária:** Centralizar a lógica de criação de objetos. Esta classe é a responsável por "montar" toda a Camada de Serviço a Persistência, aplicando o padrão Abstract Factory.

**Funções Chave:** Implementa o método `criarServico<T>()` para cada serviço necessário nas aplicações.

**Lógica de Negócio:** Determina quais classes de Persistência (DAO) serão utilizadas (ex: `Memoria do SQL`) e **regista** essa dependência (injeção de dependência) nas classes de Serviço correspondentes.

**Dependências:** Depende diretamente de **todas** as classes concretas da Camada de Serviço (`ServicoAutenticacao`, `ServicoGerente`, etc.) e **de todas** as classes de Persistência (`PersistenciaGerente`, etc.).

Módulo Controladora Quarto

**Responsabilidade Primária:** Gerenciar o fluxo de interações do usuário (via console) para as operações **CRUD** (Criar, Ler, Editar, Excluir) e **Listagem** da entidade **Quarto**.

**Funções Chave:** Exibir o menu de gestão de quartos, coletar dados de entrada, e chamar os métodos do `ServicoQuarto`.

**Lógica de Comunicação:** Atua como a **interface direta** entre o usuário e a Camada de Serviço. É responsável por capturar a entrada do usuário e criar os objetos de domínio (ex: `Quarto`, `Reserva`, `Reservacao`) antes de enviá-los ao serviço.

**Dependências:** Depende do `ServicoQuarto` (o contrato de serviço) para realizar as ações de negócio.

Módulo Serviço Quarto

**Responsabilidade Primária:** Gerenciar o **CRUD** (Criar, Ler, Editar, Excluir) da entidade **Quarto** e garantir a **integridade referencial** com as entidades **Reserva** e **Reservacao**.

**Funções Chave:** Implementa os métodos do contrato `ServicoQuarto` (ex: `cadastroQuarto`, `consultarQuarto`).

**Lógica de Negócio:** Aplica duas regras de consistência: 1. Verifica se o `quarto` já existe no banco de dados antes de permitir a criação de uma nova reserva. 2. Verifica se há reservas ativas antes de excluir o `quarto`.

**Dependências:** Depende da `PersistenciaGerente` (para operações primárias) e da `PersistenciaReserva` (para checagem de reservas).

Módulo Persistência Quarto

**Responsabilidade Primária:** Realizar as operações de **CRUD** (Criar, Ler, Editar, Excluir) e **Listagem** da entidade **Quarto** no repositório de dados (SQLite).

**Funções Chave:** Implementa o contrato `PersistenciaGerente`, manipulando comandos SQL para salvar, buscar e atualizar registros, utilizando a chave composta (`codigo do Hotel + Numero do Quarto`).

**Lógica de Dados:** Traduz os objetos de Entidade C++ (quarto) em comandos SQL (ou lógica de `std::map`) e vice-versa (recupera o objeto a partir das colunas do banco de dados).

**Dependências:** Depende apenas do **Módulo de Domínio** (para manipular objetos Quarto, Numero, Codigo, DataInicio, etc.). Não depende da Camada de Serviço.

Módulo Controladora Autenticação

**Responsabilidade Primária:** Gerenciar a interface de usuário específica para o **login** e **cadastro** do Gerente. É a primeira tela interativa que o usuário vê.

**Funções Chave:** Coleta o `email` e a `senha` do usuário via `cin/cout` e exibe mensagens de erro ou sucesso (`cout`).

**Lógica de Comunicação:** Não contém lógica de negócio. Ela delega a validação das credenciais para o `ServicoAutenticacao` e atua como uma ponte entre o usuário e o Serviço.

**Dependências:** Depende do `ServicoAutenticacao` (o contrato) para realizar as operações de login e cadastro. Não conecta o banco de dados.

Módulo Serviço de Autenticação

**Responsabilidade Primária:** Gerenciar o fluxo de entrada do sistema. É responsável por validar as credenciais do gerente e permitir ou bloquear o acesso.

**Funções Chave:** Implementa a lógica para **autenticar** um gerente existente e para **cadastar** uma nova conta de gerente (registro inicial).

**Lógica de Negócio:** Aplica a regra de autenticação (consultando a senha no banco de dados e comparando-a com a senha fornecida).

**Dependências:** Depende da `PersistenciaGerente` para buscar os dados de login no repositório. Não depende da Camada de Apresentação.

Módulo Controladora Reserva

**Responsabilidade Primária:** Gerenciar o fluxo de interações do usuário (via console) para as operações **CRUD** (Criar, Ler, Editar, Excluir) e **Listagem** da entidade **Reserva**.

**Funções Chave:** Exibir o menu de gestão de reservas, coletar dados de entrada, e chamar os métodos do `ServicoReserva`.

**Lógica de Comunicação:** Atua como a **interface direta** entre o usuário e a Camada de Serviço. É responsável por capturar a entrada do usuário e criar os objetos de domínio (ex: `Reserva`, `Quarto`, `Reservacao`) antes de enviá-los ao serviço.

**Dependências:** Depende do `ServicoReserva` (o contrato de serviço) para realizar as ações de negócio.

Módulo Serviço Reserva

**Responsabilidade Primária:** Gerenciar o **CRUD** (Criar, Ler, Editar, Excluir) da entidade **Reserva** e aplicar as regras oficiais da **lógica de negócio** (ex: `validarReserva`).

**Funções Chave:** Implementa os métodos do contrato `ServicoReserva` (ex: `cadastroReserva`, `consultarReserva`).

**Lógica de Negócio:** Aplica a regra **Anti-Conflito de Datas**. Verifica se o `quarto` já está reservado no período solicitado antes de permitir o cadastro de uma nova reserva.

**Dependências:** Depende da `PersistenciaGerente` (para operações primárias) e da `PersistenciaQuarto` (para a checagem de disponibilidade do quarto e para a criação da reserva).

Módulo Persistência Reserva

**Responsabilidade Primária:** Realizar as operações de **CRUD** (Criar, Ler, Editar, Excluir) da entidade **Reserva** no repositório de dados (SQLite).

**Funções Chave:** Implementa o contrato `PersistenciaGerente`, que inclui métodos específicos para manipular registros (ex: `consultarReserva`, `consultarReserva`).

**Lógica de Dados:** Traduz os objetos de Entidade C++ (reserva, data, codigo) em comandos SQL e vice-versa.

**Dependências:** Depende apenas do **Módulo de Domínio** (para manipular objetos Reserva, Codigo, Data, etc.). Não depende da Camada de Serviço.

Módulo Controladora Gerente

**Responsabilidade Primária:** Atuar como o **Menu Principal** do sistema após o login ser bem-sucedido. Gerencia a navegação do subnível.

**Funções Chave:** Exibe as opções principais (Gerenciar Hotel, Quartos, Hospedes, Reservas) e **delega** o fluxo de **Lógica de Navegação** para a opção do usuário (via `cin`) e chama a controladora de CRUD específica (ex: chama `ControladorHotel::executar()`).

**Dependências:** Depende das interfaces de **todas** as controladoras de CRUD (`ControladorHotel`, `ControladorQuarto`, etc.) e das interfaces de Serviço (`ServicoHotel`, `ServicoQuarto`, etc.) se a aplicação for ligada dessa forma.

Módulo Serviço Gerente

**Responsabilidade Primária:** Gerenciar o **CRUD** (Criar, Ler, Editar, Excluir) da entidade **Gerente**.

**Funções Chave:** Implementa os métodos do contrato `ServicoGerente` (ex: `consultarGerente`, `editarGerente`, `excluirGerente`).

**Lógica de Negócio:** Aplica a regra de que a **chave primária** (`email`) do Gerente não pode ser alterada durante a edição. Delega todas as operações de IO de dados para a Persistência.

**Dependências:** Depende da interface `PersistenciaGerente` para salvar e buscar dados no repositório.

Módulo Persistência Gerente

**Responsabilidade Primária:** Realizar as operações de **CRUD** (Criar, Ler, Editar, Excluir) da entidade **Gerente** no repositório de dados.

**Funções Chave:** Implementa o contrato `PersistenciaGerente`, que inclui métodos específicos para manipular registros (ex: `consultarGerente`, `consultarGerente`, `listarGerentes`).

**Lógica de Dados:** Traduz os objetos de Entidade C++ (gerente) em comandos SQL (ou lógica de `std::map`) e vice-versa.

**Dependências:** Depende apenas do **Módulo de Domínio** (para manipular objetos Gerente, Email, Senha). Não depende da Camada de Serviço.

Módulo Controladora Hotel

**Responsabilidade Primária:** Gerenciar o fluxo de interações do usuário (via console) para as operações **CRUD** (Criar, Ler, Editar, Excluir) da entidade **Hotel**.

**Funções Chave:** Exibir o menu de gestão de hotéis, coletar dados de entrada, realizar a validação de formato de entrada e chamar os métodos do `ServicoHotel`.

**Lógica de Comunicação:** Atua como a **interface direta** entre o usuário e a Camada de Serviço. Recebe dados da UI, cria os objetos de domínio (ex: `Codigo`, `nome`) e os envia ao serviço.

**Dependências:** Depende do `ServicoHotel` (o contrato de serviço) para realizar as ações de negócio.

Módulo Serviço Hotel

**Responsabilidade Primária:** Gerenciar o **CRUD** (Criar, Ler, Editar, Excluir) da entidade **Hotel** e garantir a **integridade referencial** com as Quartos.

**Funções Chave:** Implementa os métodos do contrato `ServicoHotel` (ex: `cadastroHotel`, `listarHotéis`).

**Lógica de Negócio:** Aplica a regra de **Consistência**. Antes de excluir um Hotel, consulta o **Módulo de Persistência** (DAO) de Quartos para verificar se há dependências ativas (quartos cadastrados). Se houver, a exclusão é negada.

**Dependências:** Depende da `PersistenciaHotel` (para operações primárias) e da `PersistenciaQuarto` (para a checagem de exclusão).

Módulo Persistência Hotel

**Responsabilidade Primária:** Realizar as operações de **CRUD** (Criar, Ler, Editar, Excluir) da entidade **Hotel** no repositório de dados (SQLite).

**Funções Chave:** Implementa o contrato `PersistenciaGerente`, que inclui métodos específicos para manipular registros (ex: `consultarHotel`, `consultarHotel`, `listarHotéis`).

**Lógica de Dados:** Traduz os objetos de Entidade C++ (hotel) em comandos SQL e vice-versa (recupera o objeto a partir das colunas do banco de dados).

**Dependências:** Depende apenas do **Módulo de Domínio** (para manipular objetos Hotel, Codigo, etc.). Não depende da Camada de Serviço.

Módulo Controladora Hospede

**Responsabilidade Primária:** Gerenciar o fluxo de interações do usuário (via console) para as operações **CRUD** (Criar, Ler, Editar, Excluir) e **Listagem** da entidade **Hospede**.

**Funções Chave:** Exibir o menu de gestão de hóspedes, coletar dados de entrada, e chamar os métodos do `ServicoHospede`.

**Lógica de Comunicação:** Atua como a **interface direta** entre o usuário e a Camada de Serviço. É responsável por capturar a entrada do usuário e criar os objetos de domínio (ex: `email`, `Cartao`) antes de enviá-los ao serviço.

**Dependências:** Depende do `ServicoHospede` (o contrato de serviço) para realizar as ações de negócio.

Módulo Serviço Hospede

**Responsabilidade Primária:** Gerenciar o **CRUD** (Criar, Ler, Editar, Excluir) da entidade **Hospede** e garantir a **integridade referencial** com as Reservas.

**Funções Chave:** Implementa os métodos do contrato `ServicoHospede` (ex: `cadastroHospede`, `listarHospedes`).

**Lógica de Negócio:** Aplica a regra de **Consistência**. Antes de excluir um Hospede, consulta o **Serviço de Reservas** para verificar se há **reservas ativas** (já excluído é negado se houver reservas futuras).

**Dependências:** Depende da `PersistenciaHospede` (para operações primárias) e da interface `ServicoReservas` (para a checagem de consistência).

Módulo Persistência Hospede

**Responsabilidade Primária:** Realizar as operações de **CRUD** (Criar, Ler, Editar, Excluir) e **Listagem** da entidade **Hospede** no repositório de dados (SQLite).

**Funções Chave:** Implementa o contrato `PersistenciaHospede`, manipulando comandos SQL para salvar, buscar e atualizar registros.

**Lógica de Dados:** Traduz os objetos de Entidade C++ (hospede) em comandos SQL (ou lógica de `std::map`) e vice-versa (recupera o objeto a partir das colunas do banco de dados).

**Dependências:** Depende apenas do **Módulo de Domínio** (para manipular objetos Hospede, Email, Cartao, etc.). Não depende da Camada de Serviço.