

## **Universidade Federal do Cariri**

**Curso:** Bacharelado em Ciência da Computação.

**Disciplina:** Programação Orientada à Objetos.

**Professora:** Paola Accioly.

**Equipe:** João Isaac Alves Farias e Vinícius Bezerra Leite.

### **SISTEMA DE GERENCIAMENTO DE FUNCIONÁRIOS**

#### **1. DESCRIÇÃO DO SISTEMA**

Este sistema é um projeto CRUD que tem o intuito de gerir os funcionários de uma empresa, de forma que seja possível manipular as informações dos mesmos, tais como código de cadastro, nome, cargo, remuneração, telefone e endereço. As manipulações possíveis serão cadastro, edição, exclusão e listagem de informações dos funcionários. Tal sistema tem o intuito de facilitar o gerenciamento dos recursos humanos (R.H.) da empresa, centralizando todas as informações necessárias de forma que tanto o responsável pelo R.H. possa acessar, quanto o gerente da empresa.

Desta forma, o sistema terá um módulo, que é um app web acessível pelo R.H. da empresa em questão e vai permitir que o funcionário encarregado possa criar um funcionário novo na empresa, editar informações pré-existentes ou acessar as informações necessárias, e acessível também pelo gerente da empresa, de forma que possa acessar as informações dos funcionários, alterar alguns dados e removê-lo, caso necessário. Nossa escolha foi baseada na necessidade de um sistema unificado que possa gerir estas informações nas empresas atualmente, para assim manter um controle mais rígido sobre os dados dos funcionários, além de remuneração por cargo e um gasto total da empresa.

## 2. BACKLOG

Funcionalidade	Responsável
Classes de Dados	Vinícius
Repositórios CRUD	João
Testes dos Repositórios	João

### 3. ARQUITETURA DO SISTEMA

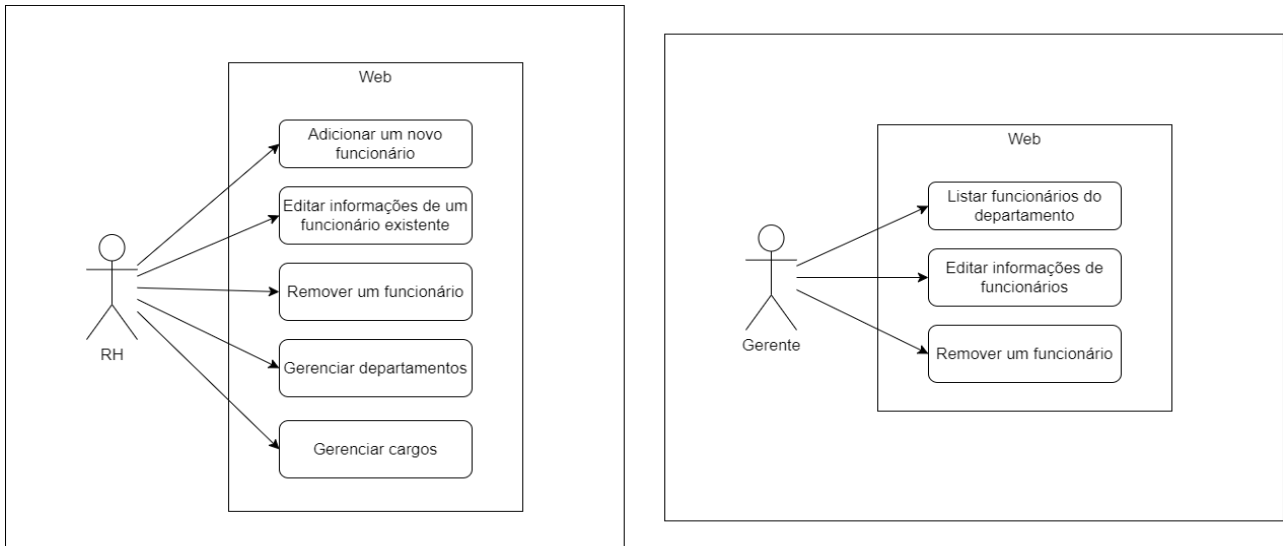


Figura 1: Diagrama de casos de uso.

[Link para diagramas](#)

O primeiro caso de uso descreve a interação do RH da empresa com o sistema de gerenciamento de funcionários. O RH pode cadastrar novos funcionários preenchendo seus dados, bem como gerenciar os cargos e departamentos da empresa. O RH também pode visualizar e editar informações de cargos e departamentos existentes no sistema.

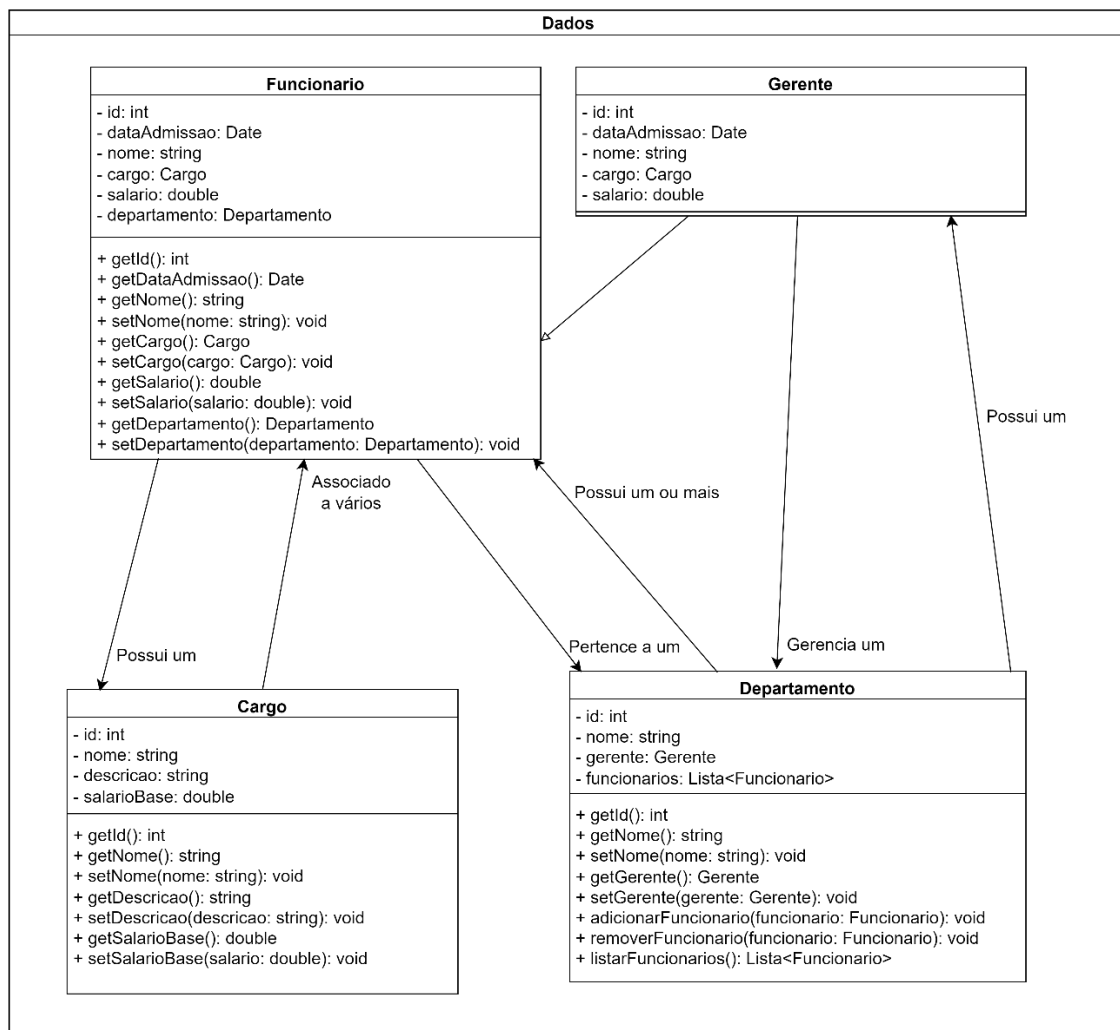


Figura 2: Diagrama de classes.

Já o segundo caso de uso descreve a interação dos gerentes da empresa com o sistema de gerenciamento de funcionários. Os gerentes acessam o sistema usando suas credenciais de acesso e podem visualizar os dados dos funcionários sob sua responsabilidade, bem como gerenciar as informações dos funcionários, como cargos, departamentos e salários. Os gerentes também podem realizar buscas e filtrar os funcionários por diferentes critérios.

O diagrama de classes desse projeto consiste em duas partes principais: as classes de domínio, que modelam as entidades do mundo real como Funcionário, Gerente, Cargo e Departamento; e as classes de repositório, que definem as operações básicas de CRUD para cada entidade do sistema.

As classes de dados são responsáveis por modelar as entidades do mundo real e seus relacionamentos. Por exemplo, a classe Funcionário contém informações pessoais do funcionário, como nome e data de admissão, bem como relacionamentos com outras entidades do sistema, como seu cargo e gerente.

As interfaces de repositório foram criadas para desacoplar a implementação do repositório em si do resto do sistema. Isso permite que diferentes implementações de repositório possam ser usadas, como armazenamento em memória ou em banco de dados, sem afetar o restante do sistema.

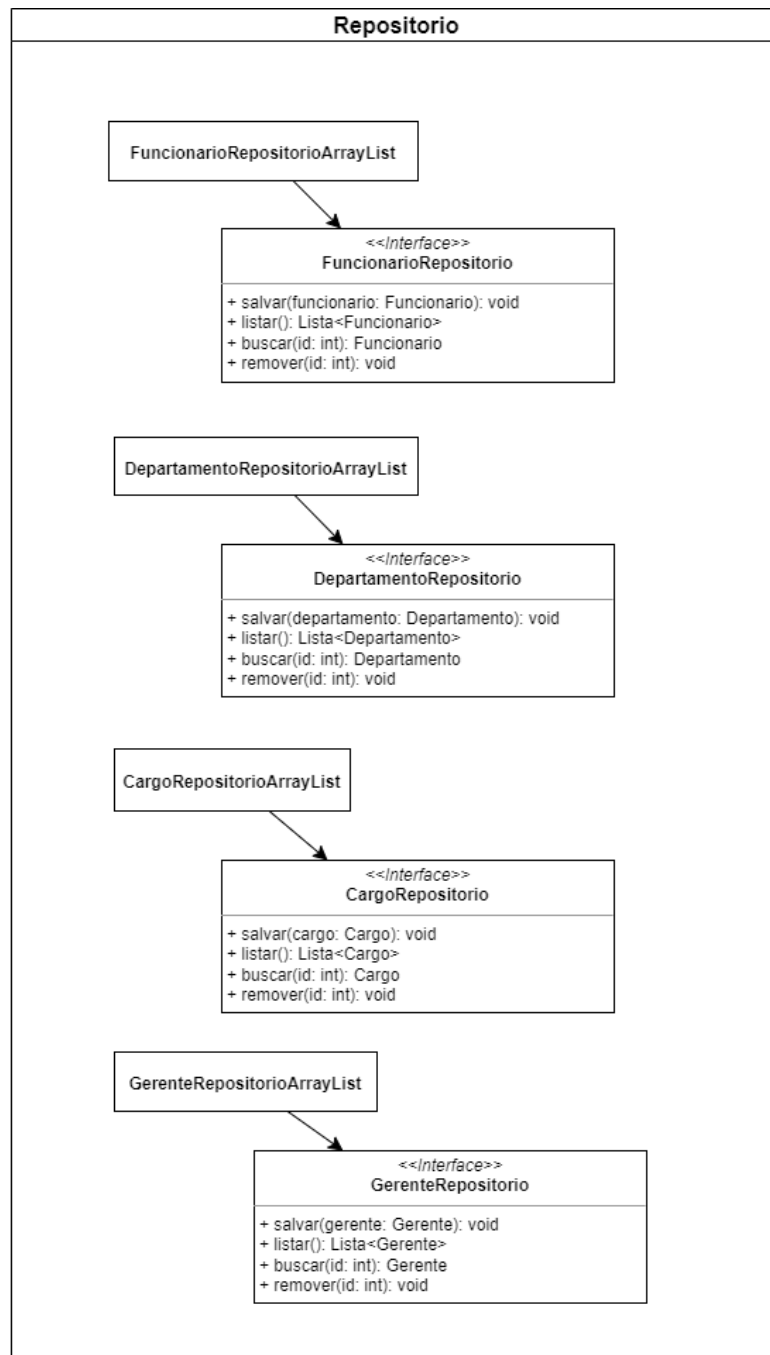


Figura 3: Diagrama de repositório.

Em resumo, o uso de interfaces permite desacoplar a implementação do repositório do resto do sistema, permitindo diferentes implementações de armazenamento de dados. A implementação em ArrayList fornece uma maneira simples de armazenar dados em memória para prototipação e testes, embora não seja útil em produção já que os dados não são persistidos.