

### Prompt:

Estou especificando um projeto em Java que tem como objetivo simular um sistema interno utilizado por uma **empresa de desenvolvimento de software** para gerenciar diferentes **tipos de funcionários**, como **gerentes, programadores e analistas de dados**.

A proposta busca modelar os dados reais desses profissionais e permitir a manipulação de informações como **nome**, **salário**, **departamento**, entre outros atributos específicos de cada categoria.

.

O principal usuário é:

- Administrador do sistema da empresa

As principais funcionalidades incluem:

- O usuário deve poder cadastrar novos funcionários
- O usuário deve poder visualizar uma lista de funcionários cadastrados.
- O usuário deve poder editar dados já cadastrados de um funcionário
- O usuário deve poder excluir um funcionário do sistema.
- O sistema deve exibir o salário total da folha de pagamento de todos os funcionários.
- O sistema deve armazenar todos os dados de funcionários em um banco de dados relacional.
- O sistema deve validar dados no momento da inserção/edição
- O sistema deve gerar relatórios dinâmicos (ex.: quantidade de funcionários, média salarial, folha por departamento) a partir do BD.

Gere 9 histórias de usuário cobrindo diferentes funcionalidades e perfis de usuário.

Use o seguinte formato:

Como [tipo de usuário], eu quero [ação] para [benefício].

Organize as histórias por perfil de usuário.

Crie também os critérios de aceitação para cada história.

### Resposta do GPT:

Aqui vão **9 histórias de usuário**, organizadas por **perfil de usuário**, no formato solicitado — cada uma com **critérios de aceitação** claros.

## Administrador do Sistema

**1) Como Administrador do Sistema, eu quero cadastrar novos funcionários para manter o quadro de pessoal atualizado.**

**Critérios de aceitação**

- Dado que estou na tela “Novo Funcionário”, quando preencho **nome, cargo, salário e departamento** válidos e clico em **Salvar**, então o registro é **persistido no BD**

**relacional** e recebe **ID único**.

- Salário deve ser **numérico e  $> 0$** ; campos obrigatórios **não podem ficar vazios**.
- O **departamento deve existir** (validação de **chave estrangeira**).
- Em caso de erro de validação, o sistema **não salva** e exibe **mensagens específicas por campo**.

**2) Como Administrador do Sistema, eu quero editar dados de um funcionário para corrigir informações ou atualizar seu vínculo.**

**Critérios de aceitação**

- Ao abrir a tela de **Detalhes do Funcionário** e alterar um ou mais campos válidos, ao **Salvar** as mudanças são **persistidas no BD**.
- Se qualquer novo valor for inválido (ex.: salário  $\leq 0$ , departamento inexistente), o sistema **bloqueia o salvamento** e mostra **mensagens de erro por campo**.
- Após salvar, a **lista de funcionários** reflete imediatamente as alterações.

**3) Como Administrador do Sistema, eu quero excluir um funcionário para remover cadastros que não são mais necessários.**

**Critérios de aceitação**

- Ao clicar em **Excluir**, o sistema solicita **confirmação**.
- Confirmando, o registro é **removido do BD** e **não aparece** mais na listagem.
- Se houver **restrições de integridade** (ex.: vínculos obrigatórios), o sistema **impede a exclusão** e informa o motivo.

**4) Como Administrador do Sistema, eu quero visualizar uma lista de funcionários com filtros e ordenação para localizar rapidamente registros.**

**Critérios de aceitação**

- A listagem é **paginada** e permite **ordenar** por nome, cargo, salário e departamento.
- É possível **filtrar** por nome (contém), cargo, departamento e **faixa salarial**.
- Cada linha exibe **Nome, Cargo, Departamento, Salário** e ações (**Editar/Excluir**).

**5) Como Administrador do Sistema, eu quero que os dados sejam armazenados em um banco de dados relacional para garantir integridade e recuperação.**

**Critérios de aceitação**

- O sistema utiliza um **SGBD relacional** com esquema **normalizado** (tabelas de **Funcionários** e **Departamentos** com **PK/FK**).
  - **Constraints**: NOT NULL em campos obrigatórios; **CHECK** para salário > 0; **UNIQUE** para identificadores definidos (ex.: matrícula interna).
  - **Migrations** de schema disponíveis para criação/alteração de tabelas.
  - Os dados **persistem após reiniciar** a aplicação.
- 

## Analista de RH

6) Como Analista de RH, eu quero validações claras no cadastro/edição para garantir a qualidade dos dados.

### Critérios de aceitação

- Regras mínimas: **nome não vazio**, **salário > 0**, **departamento existente**, **cargo válido** (dentro do catálogo: gerente, programador, analista de dados, etc.).
- **Data de admissão** (se informada) não pode ser futura.
- Mensagens de erro devem ser **específicas por campo** e impedir o salvamento até correção.
- Todas as validações são aplicadas **tanto na inserção quanto na edição**.

7) Como Analista de RH, eu quero relatórios dinâmicos de quantidade de funcionários e média salarial para monitorar o quadro.

### Critérios de aceitação

- O relatório permite **filtrar** por **departamento**, **cargo** e **período de admissão**.
- Mostra **cards** com: total de funcionários, média salarial global e por filtro aplicado.
- Os números são calculados **diretamente do BD** considerando os filtros selecionados.
- Permite **drill-down** para ver a **lista** correspondente ao número agregado.

8) Como Analista de RH, eu quero consultar e exportar uma visão consolidada por departamento para apoiar decisões de alocação.

### Critérios de aceitação

- Painel mostra, por **departamento**: **nº de funcionários**, **média salarial** e **faixa salarial** (mín/máx).
  - Filtros por **cargo** e **status** (ativo/inativo, se existir no modelo).
  - Exportação da visão atual para **CSV** (ou similar) mantendo os filtros aplicados.
  - Todos os dados são carregados **do BD** com tempos de resposta adequados (ex.: < 2s para até 10k registros agregados).
- 

## Analista Financeiro (Folha)

9) Como Analista Financeiro, eu quero ver o total da folha de pagamento (geral e por departamento) para acompanhar custos.

### Critérios de aceitação

- O sistema exibe o **total da folha** como a **soma dos salários cadastrados** de todos os funcionários **ativos**.
- É possível **quebrar por departamento** e **ordenar** do maior para o menor custo.
- A moeda é apresentada em **BRL** com formatação adequada.
- Os cálculos vêm **diretamente do BD** e se atualizam quando cadastros são incluídos/alterados/excluídos.