

Sistema de Gestão de Evolução de Carreira – Estudo de Caso URGENT

Aluno: Italo Madeira Fernandes Sousa **RA:** 22553690

Aluno: Luís Flávio Gois da Silva **RA:** 22553964

Disciplina: Engenharia de Software

Curso: Análise e Desenvolvimento de Sistemas

Professor: Leonardo Pol Suarez

Repositório GitHub: <https://github.com/italosoousa/universidade-URGEN>

Brasília – DF

Dezembro / 2025



Sumário

ARTEFATO 01 – ANÁLISE DO PROBLEMA	4
1. Contextualização do Estudo de Caso	4
2. Situação Atual e Problemas Enfrentados	4
3. Necessidades dos Stakeholders	5
4. Visão da Solução: Sistema Web em React com Login por Perfil	6
5. Objetivos da Solução	8
6. Escopo do Problema e Escopo do Sistema	8
ARTEFATO 02 – ESPECIFICAÇÃO DAS REGRAS DE NEGÓCIO (RN)	9
1. Introdução	9
2. Regras Gerais	9
3. Regras Específicas para Professores	10
4. Regras Específicas para Funcionários	11
5. Regras de Relatório	11
6. Regras de Acesso e Login	12
7. Regras de Histórico e Auditoria	13
ARTEFATO 03 – ESPECIFICAÇÃO DE REQUISITOS DE SOFTWARE (ERS)	13
1. Objetivo da ERS	13
2. Requisitos Funcionais (RF)	14
3. Requisitos Não Funcionais (RNF)	16
ARTEFATO 04 – ESPECIFICAÇÃO DOS CASOS DE USO	18
1. Atores	18
2. Lista de Casos de Uso (Resumo)	18
3. Casos de Uso Detalhados (principais)	19
UC01 – Realizar Login	19
UC02 – Encerrar Sessão	20
UC03 – Registrar Titulação (Professor)	20
UC04 – Registrar Produção Científica (Professor)	20
UC05 – Visualizar Histórico (Professor / Funcionário)	21
UC06 – Registrar Avaliação (Chefia)	21
UC07 – Registrar Função (RH)	22
UC08 – Cadastrar Servidor (RH)	22
UC09 – Validar Evoluções (RH)	22
UC10 – Executar Cálculo Anual de Salário (RH)	23
UC11 – Gerar Relatório de Professores (RH / Reitor)	23
UC12 – Gerar Relatório de Funcionários (RH / Reitor)	23

ARTEFATO 01 – ANÁLISE DO PROBLEMA

1. Contextualização do Estudo de Caso

A Universidade Regional de Gênios (URGEN) é uma instituição pública de ensino superior que possui um quadro de servidores composto por professores e funcionários técnico-administrativos. Esses servidores possuem um plano de carreira que permite aumento salarial através de gratificações, adicionais e extras, de acordo com critérios específicos para cada tipo de servidor.

ESW_TCD 2s_2025 - G-14 Estudo d...

- Para professores, a evolução está fortemente ligada a:
 - Titulação acadêmica (Especialização, Mestrado, Doutorado, Pós-Doutorado);
 - Produção científica (livros, artigos, palestras, etc.).
- Para funcionários, o foco está em:
 - Função exercida (administrativa, financeira, acadêmica, etc.);
 - Avaliação de desempenho realizada pela chefia.

Além disso, cada servidor está vinculado a um setor, que por sua vez pertence a uma unidade administrativa, o que influencia na organização dos relatórios e na visão gerencial da universidade.

Anualmente, a URGEN precisa recalcular o salário dos servidores com base nas evoluções de carreira ocorridas ao longo do ano. O estudo de caso também informa que o Reitor recebe semestralmente dois relatórios:

ESW_TCD 2s_2025 - G-14 Estudo d...

1. Professores com maiores salários classificados por titulação;

2. Funcionários com maiores salários classificados por setor.

2. Situação Atual e Problemas Enfrentados

Hoje, o processo de evolução salarial tende a ser:

- **Manual ou baseado em planilhas isoladas;**
- **Dependente de conhecimento tácito de pessoas do RH;**
- **Pouco padronizado entre diferentes setores.**

Isso gera uma série de problemas:

1. Altíssimo risco de erro humano

- **Cálculos incorretos de adicionais.**
- **Esquecimento de alguma evolução registrada tardiamente.**

2. Dificuldade em consolidar informações

- **Produção científica e titulação de professores em um lugar,**
- **Avaliações e funções de funcionários em outro,**
- **Relatórios para o Reitor exigindo muito retrabalho.**

3. Baixa rastreabilidade

- **Não há um histórico claro de “ano a ano” das evoluções:**
 - **Quando o professor obteve a titulação?**
 - **Quando o funcionário mudou de função?**

4. Falta de transparência e auditoria

- **Dificuldade em comprovar que todos foram tratados segundo as mesmas regras.**

5. Demora na geração de relatórios estratégicos

- **Os relatórios semestrais podem levar dias para serem montados manualmente.**
-

3. Necessidades dos Stakeholders

Professores

- **Registrar novas titulações;**
- **Registrar produções científicas;**
- **Consultar histórico de evolução e salário atual.**

Funcionários técnico-administrativos

- **Consultar funções exercidas ao longo do tempo;**
- **Visualizar avaliações de desempenho;**
- **Ver histórico de evolução e salário.**

Setor de Recursos Humanos (RH)

- **Cadastrar servidores, setores e unidades;**
- **Validar evoluções informadas pelos servidores;**
- **Executar o cálculo anual de salário;**
- **Gerar relatórios solicitados pelo Reitor;**
- **Manter histórico completo para auditorias.**

Chefias de Setor

- **Registrar avaliações de funcionários;**
- **Acompanhar servidores sob sua responsabilidade.**

Reitor

- **Receber relatórios confiáveis e atualizados:**
 - **Professores com maiores salários por titulação;**
 - **Funcionários com maiores salários por setor.**
-

4. Visão da Solução: Sistema Web em React com Login por Perfil

Para enfrentar esses problemas, será desenvolvido um sistema web, com front-end em React, focado em:

- **Interface moderna e responsiva;**
- **Organização por rotas/páginas, facilitando separar a experiência por tipo de usuário.**

O sistema contará com:

- 1. Página de Login (React)**
 - **Formulário de usuário/senha;**
 - **Autenticação;**
 - **Identificação do tipo de usuário;**
 - **Redirecionamento automático para a página correspondente ao perfil.**
- 2. Páginas específicas para cada tipo de usuário**
 - **Professor:** página com:
 - **Registro de titulação;**
 - **Registro de produção científica;**
 - **Consulta de histórico de evolução.**
 - **Funcionário:** página com:
 - **Consulta de avaliações;**

- Consulta de funções exercidas;
- Histórico de evolução.
- RH: painel administrativo:
 - Cadastro de servidores;
 - Validação de evoluções;
 - Execução do cálculo anual;
 - Geração de relatórios;
 - Gerenciamento de setores e unidades.
- Chefias: tela para:
 - Registro de avaliações;
 - Visualização dos funcionários sob sua responsabilidade.
- Reitor: painel de relatórios:
 - Visualização de relatórios de professores e funcionários;
 - Opção de filtrar, ordenar e exportar.

A escolha por React favorece:

- Componentização de telas (Login, Dashboard, Listas, Formulários);
 - Reaproveitamento de componentes (tabelas, cards, formulários);
 - Estrutura SPA (Single Page Application), com navegação fluida entre perfis.
-

5. Objetivos da Solução

- Reduzir erros no processo de cálculo de salário;
- Padronizar o registro de evoluções;

- **Facilitar a emissão de relatórios semestrais;**
 - **Garantir rastreabilidade, histórico e transparência;**
 - **Oferecer uma interface intuitiva e moderna em React;**
 - **Entregar páginas separadas para cada tipo de usuário após login.**
-

6. Escopo do Problema e Escopo do Sistema

Escopo do problema:

- **Evolução salarial mal controlada;**
- **Falta de integração e padronização;**
- **Dificuldade em gerar relatórios confiáveis.**

Escopo do sistema:

- **Não vai fazer folha de pagamento completa;**
- **Não vai tratar férias, ponto, admissões/demissões;**
- **Vai focar em: plano de carreira, evoluções, cálculo anual e relatórios.**

Esse artefato define claramente “em que mundo o problema vive” e “que tipo de sistema em React será necessário para resolvê-lo”.

ARTEFATO 02 – ESPECIFICAÇÃO DAS REGRAS DE NEGÓCIO (RN)

1. Introdução

As Regras de Negócio (RN) são as regras que o sistema deve obedecer para refletir corretamente o plano de carreira da URGEN. Elas definem como o salário deve ser calculado, como evoluções são tratadas e como relatórios devem ser gerados.

ESW_TCD 2s_2025 - G-14 Estudo d...

Essas regras serão implementadas no back-end, mas o front-end em React precisa respeitá-las, exibindo e coletando dados corretamente, especialmente nas telas específicas de cada perfil.

2. Regras Gerais

RN01 – Vinculação organizacional

Todo servidor deve estar vinculado a um setor e a uma unidade administrativa.

RN02 – Tipo de servidor

Cada servidor deve ser classificado como Professor ou Funcionário. Essa classificação determina as regras usadas no cálculo salarial.

RN03 – Evolução anual

O cálculo do novo salário é feito uma vez por ano, considerando todas as evoluções validadas no período.

RN04 – Evolução informada pelo servidor

As evoluções (titulação, produção, etc.) são informadas voluntariamente pelo servidor ao RH, através do sistema.

RN05 – Validação obrigatória

Nenhuma evolução impacta o salário antes de ser validada pelo RH.

3. Regras Específicas para Professores

RN06 – Titulação como base de adicional

O professor recebe adicionais conforme seu maior grau de titulação:

- **Especialização**
- **Mestrado**
- **Doutorado**
- **Pós-Doutorado**

Cada nível representa um percentual ou valor adicional a ser definido em regulamento interno.

RN07 – Produção científica pontuada

Cada produção científica (livro, artigo, palestra, etc.) gera pontos que serão convertidos em valor ou percentual adicional sobre o salário.

RN08 – Limite de contagem por período

A produção científica considerada no cálculo anual deve ser a que ocorreu dentro do período de referência daquele ano.

RN09 – Registro detalhado de produção

Cada produção deve incluir, no mínimo:

- **Tipo (livro, artigo, palestra);**
- **Título;**
- **Data;**
- **Evidência (arquivo/url);**
- **Autor(es).**

4. Regras Específicas para Funcionários

RN10 – Função exercida influencia salário

A função do funcionário (administrativa, financeira, acadêmica, etc.) deve ter um peso ou faixa salarial específica que impacta o cálculo final.

RN11 – Avaliação de desempenho

A cada ciclo de avaliação, a chefia atribui uma nota/indicador que pode resultar em bônus ou adicional.

RN12 – Mudança de função

Quando um funcionário muda de função:

- A função anterior deve ter data de término;
 - A nova função passa a vigorar a partir de uma data de início;
 - O cálculo anual deve considerar a função vigente no período da evolução.
-

5. Regras de Relatório

RN13 – Relatório de Professores

Semestralmente, deve ser gerado um relatório contendo:

- Professores com maiores salários;
- Classificação por titulação;
- Possibilidade de ordenação por:
 - Titulação;
 - Salário;
 - Nome.

RN14 – Relatório de Funcionários

Semestralmente, deve ser gerado um relatório contendo:

- Funcionários com maiores salários;
 - Classificação por setor;
 - Ordenação por:
 - Setor;
 - Salário;
 - Nome.
-

6. Regras de Acesso e Login

RN15 – Perfis de acesso

O sistema deve possuir, no mínimo, os perfis:

- Professor
- Funcionário
- RH
- Chefia
- Reitor

RN16 – Acesso por tela/página (React)

Após login, cada perfil deve ser redirecionado automaticamente para uma página específica:

- Professor → página com funcionalidades de evolução acadêmica;
- Funcionário → página com visualização de avaliações e funções;
- RH → painel administrativo completo;
- Chefia → página para avaliação de funcionários;
- Reitor → página de relatórios.

RN17 – Restrições de visualização

- Professor só vê seus próprios dados;
- Funcionário só vê seus próprios dados;
- Chefia só vê funcionários do seu setor;
- RH vê todos;
- Reitor vê apenas dados consolidados em relatórios.

7. Regras de Histórico e Auditoria

RN18 – Histórico obrigatório

Toda alteração relevante deve gerar registro no histórico:

- **Evolução aprovadas/rejeitadas;**
- **Mudança de função;**
- **Execução do cálculo anual;**
- **Geração de relatórios.**

RN19 – Registro de decisão do RH

Cada evolução validada deve indicar: aprovador, data e decisão.

ARTEFATO 03 – ESPECIFICAÇÃO DE REQUISITOS DE SOFTWARE (ERS)

Aqui entram os Requisitos Funcionais (RF) e Não Funcionais (RNF), já bem alinhados com React e a tela de login com páginas específicas.

1. Objetivo da ERS

Definir o que o sistema deve fazer (RF) e como ele deve se comportar (RNF), de forma que o desenvolvimento em React e demais camadas siga uma base clara e bem estruturada.

[ESW_TCD 2s_2025 - G-14 Estudo d...](#)

2. Requisitos Funcionais (RF)

Vou numerar de forma organizada.

RF01 – Autenticação de Usuários (Login)

O sistema deve disponibilizar uma tela de login onde o usuário informe credenciais (e-mail/usuário e senha) para acessar o sistema.

RF02 – Identificação de Perfil

Após autenticação, o sistema deve identificar o perfil do usuário (Professor, Funcionário, RH, Chefia ou Reitor).

RF03 – Redirecionamento de Página por Perfil (React)

Após login bem-sucedido, o front-end em React deve redirecionar o usuário para a rota/página correspondente ao seu perfil.

RF04 – Página do Professor

Esta página deve permitir que o professor:

- Registre titulações;
- Registre produções científicas;
- Acompanhe histórico de evolução;
- Visualize salário atual.

RF05 – Página do Funcionário

Esta página deve permitir que o funcionário:

- **Visualize suas avaliações;**
- **Consulte sua função atual e anteriores;**
- **Visualize histórico de evolução;**
- **Veja seu salário atual.**

RF06 – Painel do RH

O painel do RH deve permitir:

- **Cadastrar novos servidores;**
- **Atualizar dados de servidores;**
- **Registrar e alterar funções;**
- **Validar evoluções (titulação, produção, avaliação);**
- **Executar o cálculo anual de salários;**
- **Gerar relatórios.**

RF07 – Página da Chefia

A página da Chefia deve permitir:

- **Selecionar funcionários sob sua responsabilidade;**
- **Registrar avaliações de desempenho;**
- **Consultar histórico de avaliações.**

RF08 – Página do Reitor (Relatórios)

A página do Reitor deve permitir:

- **Visualizar relatório de professores (maiores salários por titulação);**
- **Visualizar relatório de funcionários (maiores salários por setor);**
- **Filtrar e ordenar dados;**
- **Exportar para PDF/CSV.**

RF09 – Cadastro de Servidores

O RH poderá cadastrar servidores informando:

- **Dados pessoais;**
- **Tipo de servidor;**
- **Setor;**
- **Unidade administrativa;**
- **Salário base;**
- **Perfil de acesso (Professor, Funcionário etc.).**

RF10 – Registro de Titulação

Professores podem registrar novas titulações, que entrarão como “pendentes” para aprovação do RH.

RF11 – Registro de Produção Científica

Professores podem registrar produções científicas com detalhes e anexos.

RF12 – Registro de Função de Funcionário

O RH pode cadastrar e alterar funções dos funcionários, com datas de início/fim.

RF13 – Registro de Avaliação

Chefias podem registrar avaliações de funcionários, com notas e comentários.

RF14 – Validação de Evoluções pelo RH

O RH pode aprovar ou rejeitar evoluções registradas por professores e funcionários.

RF15 – Cálculo Anual de Salário

O sistema deve calcular o novo salário de todos os servidores, uma vez por ano, com base nas regras de negócio.

RF16 – Histórico de Evolução

O sistema deve manter histórico completo de cada servidor, incluindo:

- **Evolução;**
- **Salários anuais;**
- **Funções;**
- **Avaliações.**

RF17 – Geração de Relatório de Professores

O sistema deve gerar relatório com professores de maiores salários, por titulação.

RF18 – Geração de Relatório de Funcionários

O sistema deve gerar relatório com funcionários de maiores salários, por setor.

RF19 – Exportação de Relatórios

Permitir exportar relatórios em PDF/CSV.

RF20 – Recuperação de Senha

Disponibilizar processo de recuperação de senha.

3. Requisitos Não Funcionais (RNF)

RNF01 – Plataforma Web (React)

O sistema deve ser acessado via navegador, com interface construída em React.

RNF02 – Responsividade

As telas devem ser responsivas, permitindo uso em diferentes tamanhos de tela (desktop, notebook, tablet).

RNF03 – Desempenho

- Cálculo anual de salários para até 1.000 servidores deve ocorrer em menos de 5 segundos.
- Navegação entre páginas no front-end React deve ser fluida.

RNF04 – Segurança

- As senhas devem ser armazenadas usando hashing;
- Deve haver controle de sessão;
- Perfis não podem acessar dados de outros usuários não autorizados.

RNF05 – Usabilidade

As interfaces devem ser claras, com textos explicativos, botões bem identificados e feedbacks de sucesso/erro.

RNF06 – Auditabilidade

Toda ação relevante (validação de evolução, cálculo anual, cadastro, alteração de dados) deve gerar um registro de log.

RNF07 – Disponibilidade

O sistema deve estar disponível pelo menos 99% do tempo em período letivo.

RNF08 – Manutenibilidade

A arquitetura deve permitir evolução futura, por exemplo, inclusão de novos tipos de relatórios ou regras salariais.

ARTEFATO 04 – ESPECIFICAÇÃO DOS CASOS DE USO

Aqui focamos em “como o usuário interage com o sistema”, agora já considerando a tela de login e as páginas específicas em React.

1. Atores

- Professor
- Funcionário
- RH
- Chefia
- Reitor
- Sistema de Autenticação

2. Lista de Casos de Uso (Resumo)

- UC01 – Realizar Login
- UC02 – Encerrar Sessão
- UC03 – Registrar Titulação
- UC04 – Registrar Produção Científica
- UC05 – Visualizar Histórico

- **UC06 – Registrar Avaliação**
 - **UC07 – Registrar Função**
 - **UC08 – Cadastrar Servidor**
 - **UC09 – Validar Evoluções**
 - **UC10 – Executar Cálculo Anual**
 - **UC11 – Gerar Relatório de Professores**
 - **UC12 – Gerar Relatório de Funcionários**
-

3. Casos de Uso Detalhados (principais)

UC01 – Realizar Login

Atores: Professor, Funcionário, RH, Chefa, Reitor

Objetivo: Autenticar usuário e redirecionar para a página correspondente (React).

Pré-condições:

- Usuário cadastrado no sistema.

Pós-condições:

- Sessão iniciada;
- Usuário redirecionado à página do seu perfil.

Fluxo Principal:

1. Usuário acessa a página de login.
2. Informa usuário/e-mail e senha.
3. Sistema valida credenciais.
4. Sistema identifica o perfil do usuário.

5. O front-end em React redireciona:

- **Professor** → `/professor`
- **Funcionário** → `/funcionario`
- **RH** → `/rh`
- **Chefia** → `/chefia`
- **Reitor** → `/reitor`

Fluxos Alternativos:

- **3A. Credenciais inválidas** → sistema exibe mensagem de erro.
 - **3B. Usuário bloqueado** → sistema exibe mensagem e orienta recuperação de senha.
-

UC02 – Encerrar Sessão

Atores: Todos

Objetivo: Encerrar sessão do usuário.

Fluxo Principal:

1. Usuário clica em “Sair”.
 2. Sistema encerra sessão.
 3. Front-end retorna à página de login.
-

UC03 – Registrar Titulação (Professor)

Ator Principal: Professor

Objetivo: Registrar nova titulação para futura evolução salarial.

Fluxo Principal:

1. Professor, já logado na sua página, acessa “Titulações”.

- 2. Clica em “Adicionar nova titulação”.**
 - 3. Informa nível (especialização, mestrado, etc.), data e anexa comprovante.**
 - 4. Sistema salva registro com status “Pendente de validação”.**
 - 5. RH é notificado.**
-

UC04 – Registrar Produção Científica (Professor)

Autor: Professor

Objetivo: Registrar produção científica.

Fluxo Principal:

- 1. Professor acessa “Produção Científica”.**
 - 2. Clica em “Nova produção”.**
 - 3. Informa tipo, título, data, evidência.**
 - 4. Sistema registra produção com status pendente.**
 - 5. RH é notificado.**
-

UC05 – Visualizar Histórico (Professor / Funcionário)

Atores: Professor, Funcionário

Objetivo: Consultar histórico de evoluções e salários.

Fluxo Principal:

- 1. Usuário acessa “Meu Histórico”.**
- 2. Sistema exibe linha do tempo com:**
 - Titulações ou funções;**
 - Avaliações;**
 - Cálculos anuais de salário.**

UC06 – Registrar Avaliação (Chefia)

Ator: Chefia

Objetivo: Registrar avaliação de desempenho de funcionário.

Fluxo Principal:

- 1. Chefia acessa página de Chefia.**
 - 2. Seleciona funcionário sob sua supervisão.**
 - 3. Preenche formulário de avaliação (nota, comentários).**
 - 4. Sistema salva avaliação.**
 - 5. RH é notificado.**
-

UC07 – Registrar Função (RH)

Ator: RH

Objetivo: Registrar ou alterar função de funcionário.

Fluxo Principal:

- 1. RH entra no painel.**
 - 2. Seleciona funcionário.**
 - 3. Informa nova função, setor, data de início.**
 - 4. Se necessário, define data de término da função anterior.**
 - 5. Sistema salva alteração no histórico.**
-

UC08 – Cadastrar Servidor (RH)

Ator: RH

Objetivo: Cadastrar um novo servidor no sistema.

Fluxo Principal:

- 1. RH acessa “Cadastro de Servidor”.**
 - 2. Informa dados pessoais e funcionais.**
 - 3. Define tipo de servidor e perfil de acesso.**
 - 4. Sistema cria usuário e credenciais.**
-

UC09 – Validar Evoluções (RH)

Ator: RH

Objetivo: Validar evoluções registradas por professores e funcionários.

Fluxo Principal:

- 1. RH acessa “Evoluçãoes Pendentes”.**
 - 2. Seleciona uma evolução (titulação, produção, avaliação).**
 - 3. Analisa dados e comprovantes.**
 - 4. Aprova ou rejeita.**
 - 5. Sistema atualiza status e registra histórico.**
-

UC10 – Executar Cálculo Anual de Salário (RH)

Ator: RH

Objetivo: Calcular novo salário de todos servidores, aplicando regras de negócio.

Fluxo Principal:

- 1. RH acessa “Cálculo Anual”.**
- 2. Seleciona ano de referência.**
- 3. Sistema coleta dados de todas as evoluções válidas daquele ano.**
- 4. Aplica regras de negócio para:**
 - Professores (titulação + produção);

- **Funcionários (função + avaliação).**
- 5. Calcula novo salário de cada servidor.**
 - 6. Salva resultado no histórico.**
 - 7. Exibe resumo do cálculo.**
-

UC11 – Gerar Relatório de Professores (RH / Reitor)

Atores: RH, Reitor

Objetivo: Visualizar professores com maiores salários, por titulação.

UC12 – Gerar Relatório de Funcionários (RH / Reitor)

Atores: RH, Reitor

Objetivo: Visualizar funcionários com maiores salários, por setor.

Artefato 5 - Implementação da prototipagem de média e alta fidelidade das telas

1. Visão geral da prototipagem

Neste artefato foi desenvolvida a prototipagem de **média e alta fidelidade** do sistema de controle de carreira dos servidores da Universidade Regional de Gênios (URGEN).

A prototipagem foi implementada em código, simulando o comportamento real do sistema, porém **sem integração com banco de dados ou APIs reais**.

Todo o funcionamento é baseado em **dados mocados**, que representam exemplos das informações de servidores, avaliações, titulações e relatórios utilizados nos fluxos principais do estudo de caso.

O frontend foi desenvolvido utilizando:

- **React** (componentes funcionais em TypeScript – arquivos `.tsx`);
- Projeto criado com **Vite**, ferramenta de build moderna originária do ecossistema do **Vue**, o que proporciona carregamento rápido, hot reload e uma experiência de desenvolvimento mais fluida.

2. Organização das telas no projeto

As telas principais do protótipo foram implementadas na pasta:

`frontend/src/pages/`

Arquivos criados:

- `LoginPage.tsx`
- `ProfessorDashboard.tsx`
- `FuncionarioDashboard.tsx`
- `ChefiaDashboard.tsx`
- `ReitorDashboard.tsx`
- `RhDashboard.tsx`

Cada arquivo representa uma tela de navegação do sistema, compondo o fluxo básico de uso descrito a seguir.

3. Uso de dados mocados (`mockApi.tsx`)

Para simular o backend, foi criado o arquivo:

`frontend/src/services/mockApi.tsx`

Neste arquivo estão **todas as estruturas de dados mocadas** usadas pelo sistema, como:

- Lista de servidores (professores e funcionários) com nome, setor, unidade, salário e tipo de vínculo;
- Informações de titulação e produção científica dos professores (especialização, mestrado, doutorado, pós-doutorado, livros, artigos, palestras etc.);
- Dados de avaliações e funções exercidas pelos funcionários (administrativo, financeiro, acadêmico, entre outros);
- Relatórios consolidados utilizados pelos perfis de Reitor e RH, como:
 - Professores com maiores salários por titulação;
 - Funcionários com maiores salários por setor.

As telas consomem esses dados através de funções simuladas (ex.: `getProfessores()`, `getFuncionarios()`, `getRelatoriosReitor()`, etc.) definidas em `mockApi.tsx`. Essas funções imitam o comportamento de chamadas HTTP reais (como se fossem requisições REST), porém retornando imediatamente os dados estáticos em memória.

Dessa forma, o protótipo consegue **exibir informações coerentes com o estudo de caso**, testar fluxos de navegação e validar regras de apresentação, sem a necessidade de implementar o backend definitivo neste momento.

4. Descrição das telas implementadas

4.1. Tela de Login – `LoginPage.tsx`

A tela de login é o ponto de entrada do sistema.

Principais elementos e comportamentos:

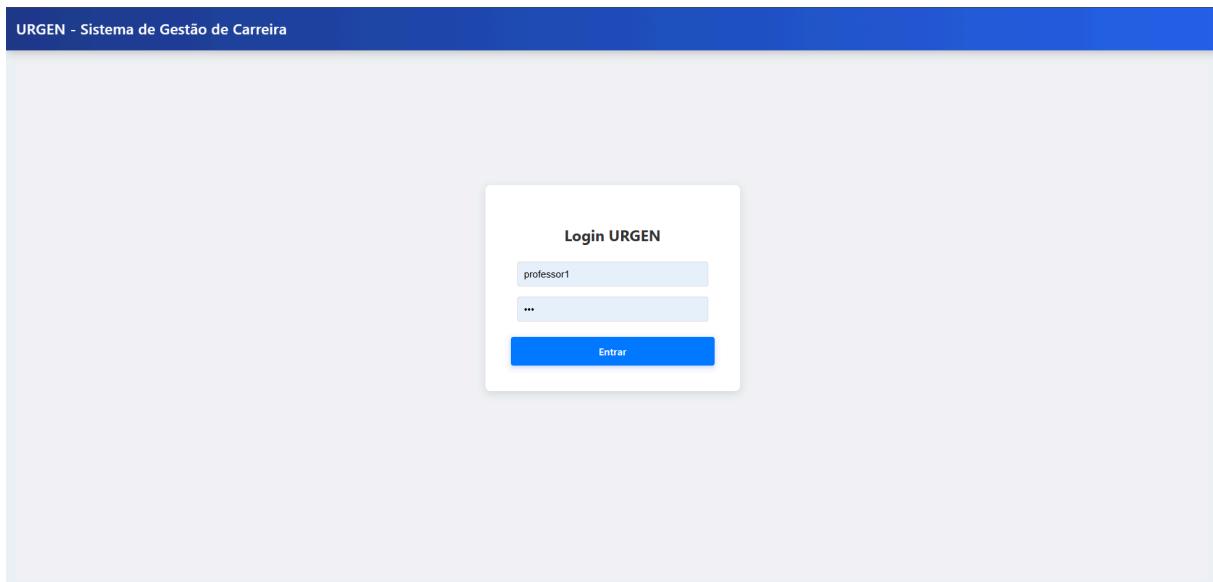
- Campos de **usuário** e **senha** (apenas validados de forma simplificada, usando regras mocadas);

Engenharia de Software – TCD URGEN

- Seleção ou detecção do **perfil de acesso** (Professor, Funcionário, Chefia, RH, Reitor);
- Ao efetuar login com sucesso, o usuário é redirecionado para o **Dashboard** correspondente ao seu perfil.

Nesta fase de prototipagem, a validação do login é totalmente baseada em dados do `mockApi.tsx`, sem criptografia ou autenticação real, servindo apenas para simular o fluxo de acesso.

Modelo da página:



4.2. Dashboard do Professor – `ProfessorDashboard.tsx`

Tela voltada ao perfil **Professor**, permitindo visualizar informações diretamente relacionadas ao plano de carreira acadêmica.

Funcionalidades simuladas:

- Exibição dos **dados cadastrais** do professor (nome, unidade, setor, data de admissão e salário atual simulado);
- Área de **titulação** mostrando o grau máximo obtido (Especialização, Mestrado, Doutorado, Pós-Doutorado) e o adicional correspondente;
- Listagem da **produção científica** (livros, artigos, palestras, etc.), usando os dados mokados;

Engenharia de Software – TCD URGEN

- Visão de como essas produções poderiam impactar o **cálculo do novo salário** (apenas exemplificado em texto e/ou em campos calculados de forma estática).

O objetivo dessa tela é representar visualmente como o professor acompanha sua evolução na carreira.

Modelo da página:

4.3. Dashboard do Funcionário – `FuncionarioDashboard.tsx`

Voltada ao perfil **Funcionário Técnico-Administrativo**.

Funcionalidades simuladas:

- Exibição das **informações pessoais e funcionais** (nome, função exercida, setor, unidade, data de admissão e salário atual simulado);
- Área com o **histórico de avaliações de desempenho** (notas e comentários), carregadas a partir da `mockApi`;
- Destaque para a **função exercida** e sua influência no plano de carreira (administrativa, financeira, acadêmica etc.);

Engenharia de Software – TCD URGEN

- Campo ou seção ilustrando como essas avaliações poderiam gerar **incrementos salariais** ao final do período.

Essa tela busca ilustrar como o funcionário visualizaria seu progresso e feedback ao longo do tempo.

Modelo da página:

The screenshot shows a user interface for a career management system. At the top, there's a blue header bar with the text "URGEN - Sistema de Gestão de Carreira" on the left and "Olá, funcionario1 (FUNCIONARIO) Sair" on the right. Below the header, the main content area has a light gray background. It displays the following information:

- Painel do Funcionário - Maria Oliveira**
- Matrícula:** s2
- Sector:** Financeiro
- Função Atual:** Analista Financeiro Júnior
- Salário Base:** R\$ 3000,00
- Salário Atual:** R\$ 3000,00

The interface is divided into two main sections:

- Funções Exercidas** (Exercised Functions):
 - Assistente Administrativo (2012-01-01 - 2018-12-31)
 - Analista Financeiro Júnior (2019-01-01 - Atualmente)
- Avaliações de Desempenho** (Performance Assessments):
 - Nota: 8 - 2023-01-20 (Status: approved)
"Bom desempenho geral."
 - Nota: 9 - 2024-01-25 (Status: pending)
"Superou as expectativas no último projeto."

At the bottom, there's a section titled **Histórico de Evolução** (Evolution History) with a list of events:

- Função: Assistente Administrativo Início: 2012-01-01, Fim: 2018-12-31 (2012-01-01) - Status: approved
- Função: Analista Financeiro Júnior Início: 2019-01-01 (2019-01-01) - Status: approved
- Avaliação de Desempenho: Pontuação: 8 - Bom desempenho geral. (2023-01-20) - Status: approved
- Avaliação de Desempenho: Pontuação: 9 - Superou as expectativas no último projeto. (2024-01-25) - Status: pending

4.4. Dashboard da Chefia – `ChefiaDashboard.tsx`

Focada no perfil de **chefia imediata**, que acompanha e avalia servidores sob sua responsabilidade.

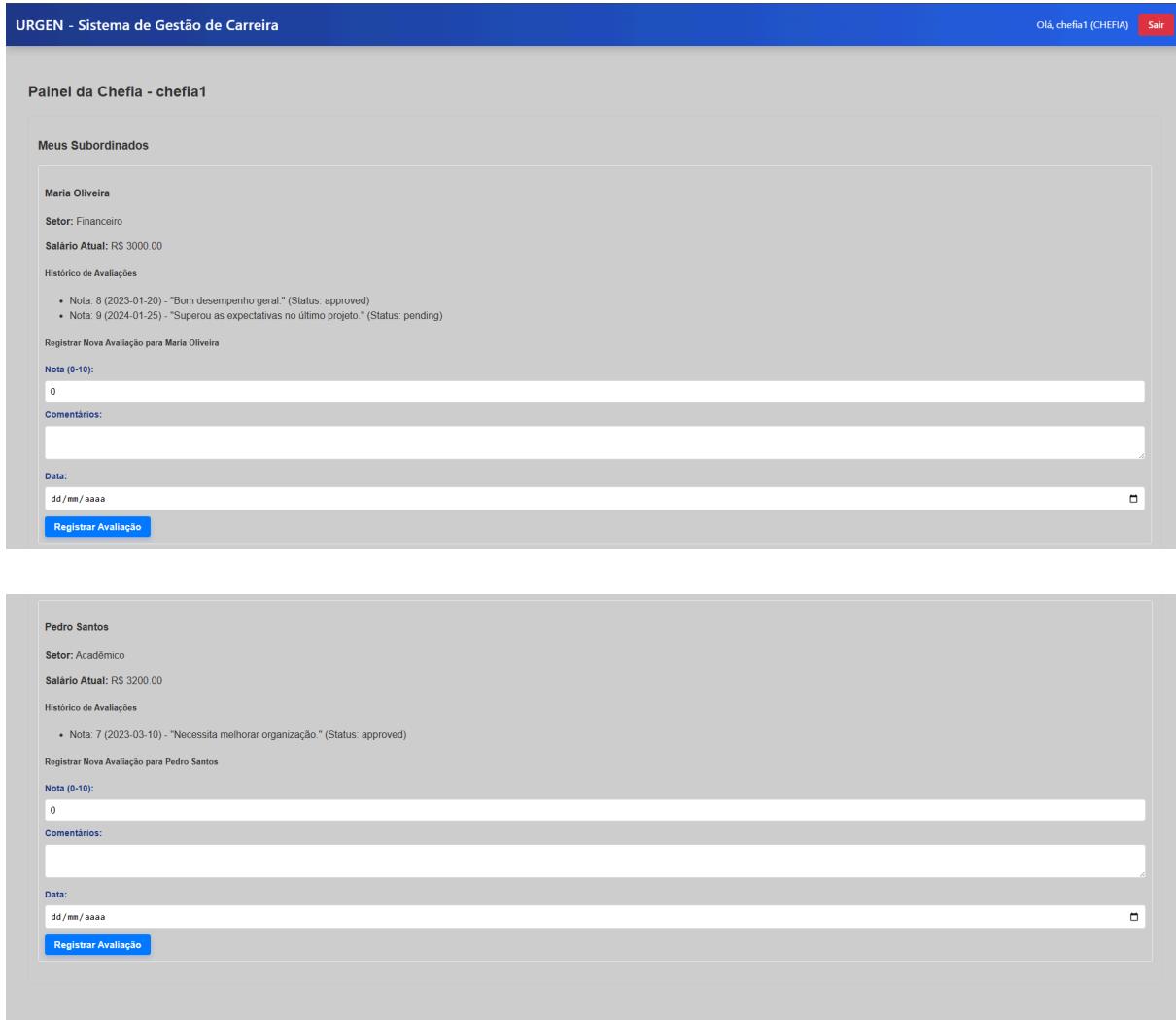
Funcionalidades simuladas:

- Lista dos **servidores subordinados** (professores e funcionários do setor);
- Visualização resumida das últimas **avaliações realizadas** e status de cada servidor;
- Acesso a detalhes de desempenho e possíveis **pendências de avaliação** (por exemplo, cards indicando “Avaliação pendente”);
- Indicadores visuais (cards, tabelas ou gráficos simplificados) com o panorama geral do setor.

Tudo isso é alimentado por dados mockados, permitindo avaliar a experiência de uso antes de construir o backend.

Engenharia de Software – TCD URGEN

Modelo da página:



Painel da Chefia - chefia1

Meus Subordinados

Maria Oliveira

Setor: Financeiro

Salário Atual: R\$ 3000.00

Histórico de Avaliações

- Nota: 8 (2023-01-20) - "Bom desempenho geral." (Status: approved)
- Nota: 9 (2024-01-25) - "Superou as expectativas no último projeto." (Status: pending)

Registrar Nova Avaliação para Maria Oliveira

Nota (0-10):

Comentários:

Data:

Registrar Avaliação

Pedro Santos

Setor: Acadêmico

Salário Atual: R\$ 3200.00

Histórico de Avaliações

- Nota: 7 (2023-03-10) - "Necessita melhorar organização." (Status: approved)

Registrar Nova Avaliação para Pedro Santos

Nota (0-10):

Comentários:

Data:

Registrar Avaliação

4.5. Dashboard do Reitor – `ReitorDashboard.tsx`

Tela destinada ao perfil **Reitor**, responsável por análises estratégicas.

Funcionalidades simuladas:

- Exibição dos relatórios semestrais definidos no estudo de caso:
 - **Professores com maiores salários classificados por titulação;**
 - **Funcionários com maiores salários classificados por setor;**
- Tabelas e/ou cards com ordenação e destaque para os maiores salários;

Engenharia de Software – TCD URGEN

- Possível visão consolidada de indicadores (por exemplo, quantidade de professores por titulação, setores com maior média salarial, etc.).

Todos esses dados são obtidos da `mockApi.tsx`, demonstrando como serão os relatórios no sistema final.

The screenshot shows the 'URGEN - Sistema de Gestão de Carreira' dashboard. At the top right, there are 'Olá, reitor1 (REITOR)' and 'Sair' buttons. Below the header, the title 'Painel do Reitor' is displayed. The dashboard is divided into two main sections: 'Relatório de Professores' and 'Relatório de Funcionários'. Each section has a dropdown filter for 'Filtrar por Titulação:' or 'Filtrar por Setor:' and a 'Todos' option. The 'Relatório de Professores' table shows two rows:

Nome	Titulação Principal	Salário Atual
Dr. João Silva	Doutorado	R\$ 5000,00
Profa. Ana Costa	Especialização	R\$ 4800,00

The 'Relatório de Funcionários' table shows three rows:

Nome	Setor	Salário Atual
Carlos Souza (Chefia)	RH	R\$ 4500,00
Pedro Santos	Acadêmico	R\$ 3200,00
Maria Oliveira	Financeiro	R\$ 3000,00

At the bottom left, there is a 'Opções de Relatório' section with a 'Simular Exportação' button.

4.6. Dashboard de Recursos Humanos – `RhDashboard.tsx`

Voltada ao perfil de **Recursos Humanos**, responsável pela gestão dos dados e cálculos de carreira.

Funcionalidades simuladas:

- Listagem completa de **servidores da URGEN** com informações básicas (nome, tipo de servidor, setor, unidade);
- Acesso a detalhes sobre **titulações, produções científicas, avaliações e funções**;
- Interface que sugere operações de **cálculo de novos salários**, representando os processos anuais de atualização com base nas evoluções cadastradas;
- Visualização de registros que teriam sido informados voluntariamente pelos servidores ao RH (novas titulações, produções, mudanças de função etc.).

Engenharia de Software – TCD URGEN

Nessa fase, as ações de “salvar”, “editar” ou “calcular” são apenas simuladas na interface, sem persistência real, mas já permitem validar o fluxo visual e a experiência de uso.

The screenshot displays the URGEN - Sistema de Gestão de Carreira application. At the top, there's a blue header bar with the title "URGEN - Sistema de Gestão de Carreira" and a user menu "Olá, rh1 (RH) | Sair".

The main content area is divided into several sections:

- Painel do RH**: A summary section with a green button "Adicionar Servidor".
- Cadastro de Servidor**: A form for adding an employee. Fields include "Nome" (Name), "Tipo" (Type: Professor selected), "Unidade" (Unit), and "Salário Base" (Base Salary: 0). A green button "Adicionar Servidor" is present.
- Cálculo Anual de Salários**: A section with a blue button "Calcular Salários Anuais".
- Evoluçãoes Pendentes de Validação**: A list of pending validation items:
 - Professor: Profa Ana Costa - Pós-Doutorado (2023-11-01) with buttons "Aprovar" (Approve) and "Rejeitar" (Reject).
 - Produções Científicas Pendentes: Professor: Profa Ana Costa - Palestra: O Futuro da IA na Educação (2022-03-10) with buttons "Aprovar" and "Rejeitar".
 - Avaliações Pendentes: Funcionário: Maria Oliveira - Score: 9 (2024-01-25) with buttons "Aprovar" and "Rejeitar".
- Todos os Servidores**: A table listing all employees:

Nome	Tipo	Setor/Unidade	Salário Atual
Dr. João Silva	PROFESSOR	Ciência da Computação	R\$ 5000,00
Maria Oliveira	FUNCIONARIO	Financeiro	R\$ 3000,00
Carlos Souza (Cheia)	FUNCIONARIO	RH	R\$ 4500,00
Profa. Ana Costa	PROFESSOR	Engenharia Civil	R\$ 4800,00
Pedro Santos	FUNCIONARIO	Acadêmico	R\$ 3200,00

5. Nível de fidelidade da prototipagem

- **Média fidelidade:**
 - Estrutura de navegação entre telas;
 - Layout geral dos dashboards;
 - Agrupamento de informações conforme os requisitos do estudo de caso;
- **Alta fidelidade:**

- Interface já codificada em React, com componentes reutilizáveis;
- Uso de estilos, responsividade básica e organização visual próxima ao sistema final;
- Dados de exemplo suficientes para simular cenários reais de uso para cada perfil (Professor, Funcionário, Chefia, Reitor, RH).

Embora não exista backend implementado, a prototipagem em código permite **testar a usabilidade, a organização das informações e o fluxo do usuário** de forma muito próxima ao produto final.

Referências

- 1. SOMMERVILLE, Ian. *Engenharia de Software*. 10. ed. São Paulo: Pearson, 2019.**
(Referência principal para conceitos de requisitos, casos de uso, regras de negócio e ciclo de vida.)
- 2. PRESSMAN, Roger; MAXIM, Bruce. *Engenharia de Software: Uma Abordagem Profissional*. 9. ed. Porto Alegre: AMGH, 2016.**
(Utilizado como base para definição de artefatos, ERS, boas práticas e documentação padrão.)
- 3. ALMEIDA, Marco Túlio. *Requisitos de Software: Fundamentos, Técnicas e Aplicações*. Rio de Janeiro: Brasport, 2020.**
(Base para estruturação de requisitos funcionais e não funcionais.)
- 4. JACOBSON, Ivar; BOOCHE, Grady; RUMBAUGH, James. *UML – Guia do Usuário*. 2. ed. Rio de Janeiro: Campus, 2000.**
(Referência para criação dos casos de uso e diagramas.)
- 5. WAZLAWICK, Raul. *Análise e Projeto de Sistemas de Informação Orientados a Objetos*. Rio de Janeiro: Elsevier, 2014.**
(Conceitos aplicados na estruturação dos fluxos, atores e modelagem conceitual.)
- 6. ESTUDO DE CASO – UNIVERSIDADE GÊNIOS (URGEN).** Documento fornecido pelo professor no Moodle, integrante do TCD da disciplina de Engenharia de Software.
(Base oficial do problema, regras de cálculo, necessidades do Reitor, e contexto organizacional.)

7. ReactJS Documentation. Disponível em: <https://react.dev>
(Referência para estrutura de rotas, navegação e páginas separadas por perfil de usuário.)
8. MDN Web Docs – Web Applications Security & Authentication
(Referências para boas práticas de login, autenticação, controle de acesso e usabilidade.)

DECLARAÇÃO DE USO DE INTELIGÊNCIA ARTIFICIAL (IA)

Este trabalho utilizou **Inteligência Artificial (IA) como ferramenta de apoio**, especificamente o modelo **Chat GPT**, para:

- Auxílio na estruturação e organização dos artefatos do TCD;
- Revisão textual e ampliação de clareza;
- Sugestões de padronização e formatação documental;
- Geração de descrições técnicas consistentes com bibliografia reconhecida;
- Apoio na redação de requisitos, regras de negócio e casos de uso;
- Contribuição na análise do estudo de caso proposto pela disciplina.

A IA **não substituiu** a análise crítica, decisões de modelagem, interpretação do estudo de caso nem a responsabilidade intelectual dos autores.

Os conteúdos gerados foram revisados, validados e adaptados conforme:

- O estudo de caso oficial URGEN;
- O conhecimento adquirido na disciplina;
- Referências bibliográficas indicadas pelo professor e pela literatura clássica de Engenharia de Software.

O uso da IA foi **instrumental**, atuando como ferramenta auxiliar de edição e elaboração, respeitando princípios de honestidade, clareza e coerência acadêmica.

Agradecimentos

Agradeço profundamente ao **Professor Leonardo Pol Suarez** pela orientação, dedicação e pelas aulas que contribuíram diretamente para o desenvolvimento deste projeto.

O aprendizado adquirido ao longo da disciplina foi essencial para a construção dos artefatos, para a compreensão dos conceitos de Engenharia de Software e para a evolução das habilidades práticas aplicadas no sistema desenvolvido.

Como forma de deixar o trabalho acessível e aberto para futuras melhorias, o projeto estará **disponível no GitHub**, onde poderá ser consultado, analisado e testado livremente caso o professor deseje visualizar a implementação completa.

Este projeto representa não apenas a entrega final da disciplina, mas também um marco importante na minha jornada acadêmica e profissional.

Muito obrigado!