



e-DefPR

Software para Controle de Processos

Defensoria Pública do Paraná



UNICENTRO

Campus CEDETEG

DECOMP -Departamento de Ciência da Computação

R. Simeão Varela de Sá, 03 - Vila Carli

Guarapuava - PR, 85040-080

Equipe:

Alexandre Gueths

Alexandre Szabo (orientador)

Gabriel Lucas Pimenta

Enrique Augusto Roza

Jean Pierri Novacoski Iensen

João Ricardo Ribas de Oliveira

Marcos Antonio Quináia (orientador)

Mauro Miazaki (orientador)

Paulo Henrique Pieczarka da Silva

Rafaela Schvarz (Defensoria)

Thomas Nichay

Tiago Remes Nunes

21.08.2017

Resumo

A Defensoria Pública do Paraná (DPPR) presta assistência jurídica gratuita às pessoas hipossuficientes. Atualmente a DPPR não possui um sistema eficiente para realização e controle das atividades realizadas pelos seus funcionários, relativos aos processos assistenciais prestados por este órgão. Um sistema baseado em planilhas onde são preenchidos formulários que alimentam as etapas de cada processo de assistência. O processamento e reescrita de informações já trabalhadas é outra constante no dia a dia de trabalho. Atividades burocráticas como realizar relatório de atividades executadas também tomam bastante tempo de cada funcionário. Atualmente está sendo desenvolvido um sistema para gerenciamento dos processos que são assistidos pela DPPR. Este sistema deve possibilitar o gerenciamento de todos os processos, possibilitando agilidade na execução das atividades. O sistema está sendo desenvolvido em um projeto extensionista que conta com a participação de professores e alunos do Departamento de Ciência da Computação da UNICENTRO e também de funcionários da Defensoria Pública de Guarapuava.

1. INTRODUÇÃO

A Gestão eficiente de Processos organizacionais é imprescindível para que se obtenha o máximo de eficiência e qualidade no trabalho final e nas diversas atividades relativas ao negócio de qualquer instituição.

A Defensoria Pública do Paraná (DPPR) é um órgão público do Estado do Paraná que tem por finalidade oferecer assistência jurídica gratuita às pessoas consideradas hipossuficientes.

Atualmente, não existe um sistema computacional para gerenciamento dos processos assistenciais que tramitam no âmbito da DPPR. É usado um “sistema” baseado em planilhas eletrônicas para preencher formulários, como por exemplo de triagem inicial, atendimento, processos, entre outros. Diversos modelos de documentos são utilizados e é necessário imprimir e preencher manualmente os dados de uma pessoa já atendida.

A reescrita de informações é um outro problema constantemente enfrentado pelos funcionários, já que para cada parte de um atendimento é necessário reescrever informações que já constavam em uma fase prévia do atendimento.

Outra necessidade mensal dos funcionários é a emissão de relatório de todos os atendimentos realizados, sendo uma tarefa repetitiva e cansativa.

A falta de um sistema que auxilie os funcionários é fator que dificulta muito a celeridade e controle de fluxo de informações e tarefas, diminuindo a eficiência do local de trabalho.

Para agilizar as atividades da DPPR se faz necessário desenvolver um sistema para ser utilizado no gerenciamento dos processos, controle das atividades, possibilidade de criar e facilitar a impressão de documentos com os dados já cadastrados.

O sistema deve permitir a geração de relatórios automaticamente conforme o funcionário trabalhe e registre as informações no sistema.

Este documento mostra o trabalho desenvolvido por professores e alunos do Departamento de Ciência da Computação da UNICENTRO para desenvolver um software que será usado pela DPPR no gerenciamento dos processos dos assistidos por este órgão público.

As ferramentas e tecnologias que estão sendo usadas no desenvolvimento deste projeto são: PHP, Laravel, REST, MySQL, React e Docker.

2. REQUISITOS

2.1. Elicitação e análise dos requisitos

O levantamento das necessidades foi realizada através de reuniões entre os funcionários da DPPR, escritório de Guarapuava-PR e professores e alunos do Departamento de Ciência da Computação, juntamente com os alunos de computação.

Em cada reunião foram elicitados os requisitos, os quais foram analisados e são descritos a seguir.

2.2. Descrição dos requisitos

Segundo Sommerville [1], os requisitos de um sistema podem ser expressos em duas visões diferentes:

- *Requisito de Usuário*: descrevem as funções e restrições do sistema de forma abstrata Inteligível pelo usuário / cliente; representam o ponto de vista das necessidades da empresa cliente sem indicar qualquer solução; são escritos em linguagem natural; podem ser usados diagramas simples (ex. tabelas) na descrição.
- *Requisitos de Sistema*: são descrições mais detalhadas que os requisitos do usuário; devem ser padronizadas, completas e consistente; são usados pela

equipe de desenvolvimento; geralmente fazem parte do contrato entre cliente e desenvolvedor.

2.2.1. Visão de Usuário

Para melhor entendimento das necessidades dos usuários, levantou-se um perfil dos possíveis usuários do sistema. Os usuários são divididos em:

2.2.1.1. Defensores Públicos

Os Defensores Públicos são pessoas formadas em Direito que ingressam na Defensoria Pública através de aprovação em concurso público de provas e títulos. Na defesa dos interesses de seus assistidos, os Defensores Públicos têm atuação no primeiro e no segundo grau de jurisdição, com titularidade e atribuições específicas em razão da matéria a ser examinada.

Os Defensores possuem o maior cargo na hierarquia de uma Defensoria Pública e devem possuir acesso irrestrito ao *e-def*.

Acesso: Total.

2.2.1.2. Assessores Jurídicos

Assessores Jurídicos têm a função de ajudar os defensores públicos. Cada assessor trabalha com um defensor. Podem fazer atendimento jurídico, tendo a responsabilidade de confeccionar petições e encaminhar para um defensor para que ele possa fazer correções, assinar e ajuizar. Possuem a responsabilidade também de cumprir prazos no sistema PROJUDI.

Acesso: Os assessores devem possuir acesso aos processos que realizou atendimento. Isso inclui o estado do processo e os dados da pessoa assistida. Não pode visualizar dados sigilosos como por exemplo consulta com psicóloga.

2.2.1.3. Psicólogo

Os funcionários “Psicólogos” devem conseguir, se for encaminhado, realizar uma consulta com o assistido, podem fazer seu laudo, anotações e guardar no sistema.

- Para redigir o parecer técnico, os Psicólogos deverão escrever o documento em algum editor de texto externo ou digitalizar uma cópia de suas anotações e realizar o upload do arquivo para o sistema.

Acesso: Possui acesso aos dados dos assistidos atendidos por ele.

2.2.1.4. Assistente Social

Atuam na triagem dos atendimentos, na conciliação e na mediação de conflitos nas questões familiares e no acompanhamento de assistidos. Com eles fica a responsabilidade

de confeccionar relatórios socioeconômicos de cada um dos assistidos. São esses documentos que garantem todo o embasamento necessário para o prosseguimento das ações judiciais e extrajudiciais.

Os funcionários “Assistentes sociais” assim como os “Psicólogos”, quando o cliente é encaminhado para eles, deve conseguir utilizar uma área específica para sua categoria, onde é possível fazer *upload* de suas anotações e seu parecer sobre o cliente.

- Para realizar o parecer técnico, os Assistentes Sociais deverão redigir o documento em algum editor de texto externo e realizar o upload do arquivo para o sistema.

Acesso: Devem possuir o acesso somente aos dados dos assistidos atendidos por eles (dados pessoais e socioeconômicos).

2.2.1.5. Técnico Administrativo

Na Defensoria Pública de Guarapuava existe basicamente dois tipos de técnicos, os que possuem a formação em Direito e contribuem com os processos judiciais e os que possuem formação em outra área ou apenas o ensino médio completo. Assim possuindo técnicos com diferentes níveis de acesso ao sistema. Os técnicos que possuem curso de Direito, podem acessar dados dos processos em que ele efetuou atendimento e os demais técnicos tem somente acesso aos dados dos assistidos atendidos.

Os técnicos em geral, ao utilizar o sistema, deve conseguir realizar o cadastro do atendimento de assistidos que estão chegando pela primeira vez, ou que já tem um processo em andamento, mas que foi necessário marcar mais um atendimento por algum fator.

Acesso: Técnicos sem formação em direito devem possuir acesso aos dados dos assistidos atendidos em geral, porém não podem possuir acesso aos processos, enquanto que o Técnico com formação em direito além de possuir esses acessos também possui aos processos dos atendidos por eles.

2.2.1.6. Estagiários de Direito

São responsáveis por iniciar a confecção de petições e enviar para uma defensora corrigir. Utilizam as informações do assistido para gerar um documento denominado “Ação”, por isso esses funcionários devem ter acesso a consulta das informações dos assistidos.

- Para criar o documento “Ação” o estagiário deverá redigir o documento em algum editor de texto externo e realizar o *upload* do arquivo para o sistema.
- Os documentos, para upload no sistema, devem estar nos seguintes formatos: “.odt”, “.doc”, “.xlsx”, “.xls” ou em “.docx”.

Acesso: Possuem o mesmo nível de acesso que os Assessores Jurídicos.

2.2.1.7. Estagiários de Ensino Médio

Responsável pelo acolhimento dos assistidos.

Acesso: Deve possuir acesso somente aos dados pessoais do assistido e algumas informações internas, como por exemplo última pessoa a efetuar o atendimento.

2.2.1.8 Funcionalidades do sistema

Os usuários devem conseguir, respeitando o devido acesso:

- Salvar as informações pertinentes dos funcionários e atualizar em caso de mudança;
- Gerar um relatório das atividades realizadas por um funcionário;
- Salvar, alterar ou remover informações dos assistidos pelos funcionários;
- Gerar relatório dos atendimentos feitos ao assistido;
- Gerar relatório de geral de atividades;
- Realizar uma triagem inicial;
- Cadastrar documentos necessários para se realizar a triagem;
- Fazer a abertura de um processo;
- Consultar um processo existente;
- Realizar alterações no processo através de protocolos;

2.2.2. Visão de Sistema

A visão do sistema documenta o ambiente geral de processos desenvolvidos para o sistema, fornecendo a todos os envolvidos uma descrição compreensível deste e suas macro-funcionalidades.

2.2.2.1 Abrir Processo

Procedimento de abrir um processo.

Usuários: Técnico Administrativo Jurídico, Técnico Administrativo, Assistente Social, Assessor Jurídico, Defensor Público.

Entradas e pré-condições:

- Entradas: código processo, descrição ou observação.
- Pré-Condições: Estar logado no sistema com algum usuário que tenha permissão de abrir um processo.

Saídas e pós-condições:

- Saídas: Em caso de sucesso exibir mensagem de concluído. Em caso de falha exibir as possíveis razões.
- Pós-Condições: exibir uma tela com o processo criado.

2.2.2.2 Consultar Processo

Procedimento para consultar um processo em andamento.

Usuários: Técnico Administrativo Jurídico, Técnico Administrativo, Assistente Social, Estagiário de Direito, Assessor Jurídico, Defensor Público, Psicólogo.

Entradas e pré-condições:

- Entradas: Nome do processo ou número.
- Pré-Condições: Estar logado no sistema com uma conta que tenha permissão de consultar processos e é necessário que o assistido atendido tenha um processo cadastrado.

Saídas e pós-condições:

- Saídas: Em caso de sucesso exibir tela com informações referentes ao processo pesquisado, em caso de falha exibir motivo.
- Pós-Condições: Tela exibindo os dados do processo ou dizendo que o processo não existe.

2.2.2.3 Realizar Atendimento Jurídico

Procedimento de atendimento jurídico.

Usuários: Técnico Administrativo Jurídico, Estagiário de Direito, Assessor Jurídico, Defensor Público.

Entradas e pré-condições:

- Entradas: Triagem inicial, socioeconômica do atendido, relato.
- Pré-Condições: Estar logado no sistema com uma conta que permita realizar atendimento jurídico (Técnico Administrativo Jurídico, Assessor Jurídico, Defensor, Estagiário de Direito), ter a triagem inicial preenchida e o registro socioeconômico preenchido e aprovado.

Saídas e pós-condições:

- Saídas: contagem de atendimento para o funcionário.
- Pós-Condições: nenhuma.

2.2.2.4 Confeccionar Petição

Realizar upload de uma petição no sistema.

Usuários: Técnico Administrativo Jurídico, Estagiário de Direito, Assessor Jurídico, Defensor Público.

Entradas e pré-condições:

- Entrada: Título da petição, número de controle, nome funcionário, data.
- Pré-Condições: Estar logado no sistema com um usuário que tenha permissão de confeccionar petições, o assistido atendido deve ter passado pela triagem inicial e socioeconômica, ter feito o atendimento jurídico.

Saídas e pós-condições:

- Saídas: Exibir mensagem de envio com sucesso ou falha.
- Pós-Condições: Sugerir ao usuário enviar a petição para correção do defensor.

2.2.2.5 Enviar para Análise do Defensor

Enviar uma petição para análise do defensor designado. Necessário ter confeccionado e upado para o sistema uma petição.

Usuários: Técnico Administrativo Jurídico, Estagiário de Direito, Assessor Jurídico.

Entradas e pré-condições:

- Entrada: Upar um arquivo, nome do defensor que irá corrigir, nome do autor, código, data.
- Pré-Condições: Estar logado no sistema e já ter feito upload de pelo menos uma petição e que essa não tenha sido enviada para correção ainda.

Saídas e pós-condições:

- Saídas: Exibir mensagem de envio com sucesso ou falha.
- Pós-Condições: Aparecer na tela do defensor uma mensagem avisando que chegou uma petição para correção e atualizar sua lista de petições para correção pendentes.

2.2.2.6 Consultar Atendimento Psicológico

Consultar um atendimento psicológico no sistema. Nem todos os atendimentos estarão livres para acesso. Em alguns casos terá que solicitar o acesso.

Usuários: Técnico Administrativo Jurídico, Estagiário de Direito, Defensor Público, Psicólogo.

Entradas e pré-condições:

- Entrada: Nome do atendido.
- Pré-Condições: Estar logado no sistema e possuir acesso ao atendimento.

Saídas e pós-condições:

- Saídas: Em caso de sucesso exibir tela com dados do atendimento psicológico e em caso de falha exibir mensagem com o motivo da falha (sendo possível não possuir acesso).
- Pós-Condições: Tela com os dados da consulta ou mensagem de falha.

2.2.2.7 Abrir Protocolo

Abrir um protocolo, deve ser permitido deletar um protocolo também. Caso o usuário tente deletar um protocolo que esteja anexado em um ou mais processos deve ser exibido uma mensagem avisando que esse protocolo está anexado em determinado processo e perguntar se realmente deseja excluir pois caso queira o protocolo será excluído do processo também.

Usuários: Técnico Administrativo Jurídico, Técnico Administrativo, Assistente Social, Estagiário de Direito, Assessor Jurídico, Estagiário Ensino Médio, Defensor Público, Psicólogo.

Entradas e pré-condições:

- Entrada: Título protocolo, código, conteúdo, autor, data criação.
- Pré-Condições: Estar logado no sistema.

Saídas e pós-condições:

- Saídas: nenhuma.
- Pós-Condições: Protocolo armazenado no sistema e mensagem de sucesso.

2.2.2.8 Triagem Inicial

Triagem inicial onde são coletados os dados pessoais do assistido. Um dos primeiros contatos de um assistido com a defensoria. Deve ser possível editar os dados de um assistido após ter passado pela triagem inicial.

Usuários: Técnico Administrativo Jurídico, Técnico Administrativo, Assistente Social, Defensor Público.

Entradas e pré-condições:

- Entradas: Nome do funcionário que atendeu, número atendimento, Data e nome da pessoa para qual será encaminhado o atendimento.

- Dados pessoais: Nome, sobrenome, gênero, data de nascimento, idade, RG, órgão emissor, CPF, naturalidade, estado civil, profissão, local de trabalho.
- Endereço: Rua, número, complemento bairro, cidade, estado, CEP, endereço de trabalho, cidade, estado, CEP, bairro, telefone residencial, telefone celular, telefone trabalho, telefone recado, e-mail.
- Dados da Parte Contrária: Nome, sobrenome, RG, CPF, profissão, endereço familiar, remuneração, bairro, cidade, estado, CEP, telefone, celular, e-mail.
- Atendimentos: Data do atendimento, responsável, descrição.
- Andamentos: área necessária para comentar a situação atual do atendimento. Como um histórico de tudo que já foi feito e a situação atual do atendimento. Campo de texto.
- Contatos: Sendo possível adicionar vários. Data, responsável, meio de contato, descrição.
- Documentos Necessários para atendimento inicial:
 - Carteira de Identidade (RG);
 - CPF;
 - Comprovante de residência (talão de água, luz ou telefone com CEP atualizado);
 - Comprovante de renda atualizado de todos os membros da família (Contracheque, holerite, comprovante de aposentadoria, pensão, benefício);
 - Autônomos: trazer declaração com o valor de rendimento recebido mensalmente;
 - Desempregados: trazer declaração de não rendimento e carteira de trabalho (CTPS);
 - Carnê de IPTU, caso possua casa própria (com valor venal descrito);
 - Extrato de conta poupança, caso possua investimento bancário;
 - Endereço, dados pessoais e contato da parte contrária;
 - Dados e documentos referente ao processo: certidão de nascimento, certidão de casamento, cópia de processo, ou acordo, etc;
 - Nome de 03 testemunhas que conheçam o fato.

A princípio para primeira consulta são necessários no mínimo o RG, CPF e comprovante de residência. Os documentos devem ser upados no sistema (tentar colocar o upload no modo de arrastar os arquivos para dentro do sistema).

- Pré-Condições: Estar logado com o login de um funcionário no sistema.

Saídas e pós-condições:

- Saídas: Exibir mensagem de cadastro com sucesso ou mensagem de falha apontando os possíveis motivos.

- Pós-Condições: Após a mensagem de sucesso exibir uma página com todos os dados cadastrados para revisão ou retornar a página de triagem inicial vazia novamente.

2.2.2.9 Triagem Socioeconômica

Registrar no sistema os dados socioeconômicos de um assistido. O assistido só poderá ser atendido caso a renda de todos os membros da família que moram na mesma casa seja menor que três salários mínimos (aprox. R\$ 2.800,00); valor total de bens não pode ultrapassar 1.500 Unidades Fiscais do Paraná (aprox. R\$120.000,00); não possuir recursos depositados em conta poupança ou outros investimentos em valor superior a doze salários mínimos (aprox. 11.500,00). Deve ser possível editar quando necessário os dados do registro de hipossuficiência.

Usuários: Técnico Administrativo Jurídico, Técnico Administrativo, Assistente Social, Defensor Público.

Entradas e pré-condições:

- Entradas: Nome de quem registrou, data, nome do assistido que já possui cadastro pois já passou pela triagem inicial.
- Composição Familiar: Nome, sobrenome, data de nascimento, grau de parentesco, atividade laboral, renda. Dados necessário para calcular a renda mensal da família. Ao final somar a renda de todos e armazenar a renda familiar mensal.
- Situação Habitacional: Tipo de residência, situação, valor do aluguel.
- Bens: Bem, valor do bem, situação, valor, valor de aplicações em banco. Disponibilizar três somas, uma do valor total de bens, do valor total quitado, do valor total de aplicações em banco.
- Pré-Condições: Estar logado no sistema.

Saídas e pós-condições:

- Saídas: Exibir mensagem de sucesso ou falha e gerar um PDF de declaração de hipossuficiência socioeconômica para que seja possível imprimir, assinar e upar no sistema.
- Pós-Condições: Exibir os dados socioeconômicos de forma resumida e pedir para upar o PDF de saída assinado pela pessoa atendida.

2.2.2.10 Cadastrar Documentos

Utilizado quando falta algum documento durante um atendimento e então o assistido traz posteriormente, necessitando assim cadastrar esses novos documentos.

Usuários: Técnico Administrativo Jurídico, Técnico Administrativo, Assistente Social, Estagiário de Direito, Assessor Jurídico, Estagiário Ensino Médio, Defensor Público, Psicólogo.

Entradas e pré-condições:

- Entradas: Descrição, Arquivo e nome do documento a ser cadastrado, data que foi adicionado.
- Pré-Condições: Necessário estar logado com uma conta que permita cadastrar documentos, um scanner.

Saídas e pós-condições:

- Saídas: Exibir mensagem de sucesso ou falha.
- Pós-Condições: Voltar a página de cadastro de documentos.

2.2.2.11 Apensar Protocolo ao Processo

Anexar um protocolo a um processo cadastrado no sistema.

Usuários: Técnico Administrativo Jurídico, Estagiário de Direito, Assessor Jurídico, Defensor Público.

Entradas e pré-condições:

- Entradas: Nome/número do processo e nome/número do protocolo a ser apensado.
- Pré-Condições: Estar logado no sistema e possuir o protocolo em aberto para apensar (anexar) ao processo.

Saídas e pós-condições:

- Saídas: Exibir mensagem de sucesso ou falha.
- Pós-Condições: Exibir o processo já com os novos anexos.

2.2.2.12 Encaminhar Atendimento

Encaminhar um atendimento a outro funcionário.

Usuários: Técnico Administrativo Jurídico, Técnico Administrativo, Assistente Social, Estagiário de Direito, Assessor Jurídico, Estagiário Ensino Médio, Defensor Público, Psicólogo.

Entradas e pré-condições:

- Entrada: Nome da pessoa a qual será encaminhado, data, quem encaminhou.
- Pré-Condições: Estar logado no sistema e estar realizando um atendimento.

Saídas e pós-condições:

- Saídas: Mensagem de sucesso ou falha.
- Pós-Condições: Exibir uma notificação na tela da pessoa em que recebeu o encaminhamento.

2.2.2.13 Cadastrar Funcionário

Cadastrar novo funcionário no sistema.

Usuários: Técnico Administrativo Jurídico, Técnico Administrativo, Defensor Público.

- Entradas e pré-condições:
- Entrada: Nome do funcionário que cadastrou, Data, nome novo funcionário, sobrenome, gênero, data de nascimento, RG, órgão emissor, CPF, e-mail, telefone, celular, rua, número, complemento, bairro, CEP, cidade, estado.
- Pré-Condições: Estar logado no sistema.

Saídas e pós-condições:

- Saídas: Mensagem de sucesso ou falha.
- Pós-Condições: voltar a tela de cadastro de funcionário em branco.

2.2.2.14 Atualizar Funcionário

Atualizar funcionário no sistema.

Usuários: Técnico Administrativo Jurídico, Técnico Administrativo, Defensor Público.

Entradas e pré-condições:

- Entrada: Os mesmos do cadastro de funcionário. Mostrar a mesma página de cadastro já com todos os dados preenchidos para que a pessoa altere os que desejar e aperte em salvar.
- Pré-Condições: Estar logado no sistema.

Saídas e pós-condições:

- Saídas: Mensagem de sucesso ou falha.
- Pós-Condições: voltar a tela de cadastro de funcionário em branco.

2.2.2.15 Gerar Relatório

Gerar relatórios de determinado período de tempo (Mensal, Anual, Customizado) com todas as operações realizadas no sistema.

Usuários: Técnico Administrativo Jurídico, Técnico Administrativo, Assistente Social, Estagiário de Direito, Assessor Jurídico, Estagiário Ensino Médio, Defensor Público, Psicólogo

Entradas e pré-condições:

- Entrada: Data inicial e data final ou selecionar um relatório mensal ou anual.
- Pré-Condições: Estar logado no sistema.

Saídas e pós-condições:

- Saídas: Mensagem de sucesso ou falha.
- Pós-Condições: Exibir página com os atendimentos realizados e atividades.

2.2.2.16 Registrar Atividade

Cadastrar atividades no sistema. Como por exemplo participação em reuniões, realização de orçamentos, pedido de malotes, etc. Para poder apresentar essas atividades nos relatórios.

Usuários: Técnico Administrativo Jurídico, Técnico Administrativo, Assistente Social, Estagiário de Direito, Estagiário Ensino Médio, Defensor Público, Psicólogo.

Entradas e pré-condições:

- Entrada: Nome da pessoa, data, atividade, descrição.
- Pré-Condições: Estar logado no sistema.

Saídas e pós-condições:

- Saídas: Mensagem de sucesso ou falha.
- Pós-Condições: Exibir página com os atendimentos.

2.2.2.17 Gerar Correspondência

Gerar correspondências (ofícios, memorandos e cartas). Cada um desses deve ter uma numeração exclusiva e ser possível upar as correspondências no sistema para questões de controle. Deve ser possível também consultar todas as correspondências já registradas.

A página de correspondências cadastradas deve possuir um filtro que separe ofícios de memorandos, cartas e vice-versa.

Usuários: Técnico Administrativo Jurídico, Técnico Administrativo, Assistente Social, Estagiário de Direito, Estagiário Ensino Médio, Defensor Público, Psicólogo.

Entradas e pré-condições:

- Entrada: Nome da pessoa, data, descrição, número do processo a qual corresponde (opcional, nem todos os ofícios estão ligados a um processo), arquivo, número de controle.
- Pré-Condições: Estar logado no sistema.

Saídas e pós-condições:

- Saídas: Mensagem de sucesso ou falha.
- Pós-Condições: Exibir página com os as correspondências já cadastradas.

2.2.2.18 Atendimento Social

Realizar um atendimento social. A Assistente Social faz suas anotações em um documento e faz o upload para dentro do sistema.

Usuário: Assistente Social

Entradas e pré-condições:

- Entrada: Arquivo, número de controle, nome funcionário, data, relato.
- Pré-Condições: Estar logado no sistema, o assistido já deve possuir cadastro de seus dados pessoais (triagem inicial).

Saídas e pós-condições:

- Saídas: Mensagem de sucesso ou falha.
- Pós-Condições: Exibir página com as movimentações já cadastradas no processo.

2.2.2.19 Consultar Atendimentos

Consultar os atendimentos em aberto ou arquivados. Através do nome do assistido ou do CPF deve ser possível encontrar os dados relacionados aos atendimentos desse. O que importa para o estagiário de ensino médio é saber por quem esse assistido recém-chegado deve ser atendido. Caso o assistido nunca tenha sido atendido ele deve ser encaminhado para os técnicos para fazer a sua triagem inicial, caso contrário verificar no sistema quem foi a última pessoa que atendeu e encaminhar o atendimento.

Usuário: Estagiário Ensino Médio

Entradas e pré-condições:

- Entradas: Nome ou CPF do assistido.
- Pré-Condições: Estar logado no sistema.

Saídas e pós-condições:

- Saídas: Dados relacionados aos atendimentos.

- Pós-Condições: Exibir uma página para encaminhar atendimento.

2.2.2.20 Despacho de Instauração

Gerar um despacho de instauração. Um despacho de instauração possui uma pasta e essa possui uma capa que deve ser gerada automaticamente pelo sistema através dos dados inseridos no sistema. O modelo da capa está no arquivo "Anexo 1 – Capa Despacho de Instauração".

Usuário: Defensor Público

Entradas e pré-condições:

- Entradas: código, nome do defensor que gerou, data, objeto, interessado, matéria.
- Pré-Condições: estar logado no sistema.

Saídas e pós-condições:

- Saídas: Em caso de sucesso exibir mensagem de concluído. Em caso de falha exibir as possíveis razões.
- Pós-Condições: Exibir PDF da capa da pasta para impressão.

2.2.2.21 Transferir Correção

Caso um defensor queira passar a correção de uma determinada petição para outro defensor, ou até mesmo para o estagiário voltar a corrigir esse caso de uso será usado.

Usuário: Defensor Público

Entradas e pré-condições:

- Entradas: nome do funcionário ao qual será transferido a correção, descrição, data, nome de quem transferiu.
- Pré-Condições: estar logado no sistema.

Saídas e pós-condições:

- Saídas: Em caso de sucesso exibir mensagem de concluído. Em caso de falha exibir as possíveis razões.
- Pós-Condições: voltar a página de correções pendentes do sistema.

2.2.2.22 Ajuizar Petição

Registrar no sistema quando um processo foi ajuizado (inserido no PROJUDI). Para isso será necessário indicar qual processo interno da defensoria foi ajuizado (pelo código),

posteriormente o PROJUDI irá gerar um código o qual deverá ser cadastrado junto ao processo interno da defensoria.

Usuário: Defensor Público

Entradas e pré-condições:

- Entradas: data, nome de quem ajuizou, código do processo a ser ajuizado, código PROJUDI.
- Pré-Condições: estar logado no sistema.

Saídas e pós-condições:

- Saídas: Em caso de sucesso exibir mensagem de concluído. Em caso de falha exibir as possíveis razões.
- Pós-Condições: Voltar a página de processos em aberto.

2.2.2.23 Realizar Atendimento Jurídico

[RF002] Caso de Uso – Atendimento Psicológico

Caso de uso necessário para o Psicólogo registrar seus atendimentos. Um atendimento consiste de cadastrar no sistema um arquivo de Word.

Usuário: Psicólogo

Entradas e pré-condições:

- Entrada: nome do funcionário que realizou o atendimento, data, arquivo.
- Pré-Condições: Estar logado no sistema.

Saídas e pós-condições:

- Saídas: Em caso de sucesso exibir tela com dados do atendimento psicológico e em caso de falha exibir mensagem com o motivo da falha.
- Pós-Condições: Tela com os dados da consulta ou mensagem de falha.

3. CRONOGRAMA

Como parte do protocolo de formalização, foi desenvolvido um cronograma do projeto extensionista que prevê tarefas de requisitos, projeto, codificação, implantação, ajustes de código, testes, validação e relatório final. A Figura 3.1 apresenta o Gráfico de Gantt do cronograma onde são mostradas as atividades, data inicial, duração e data final. São mostrados também a ocorrência de cada tarefa no espaço temporal do cronograma bem como as interdependências entre as atividades.

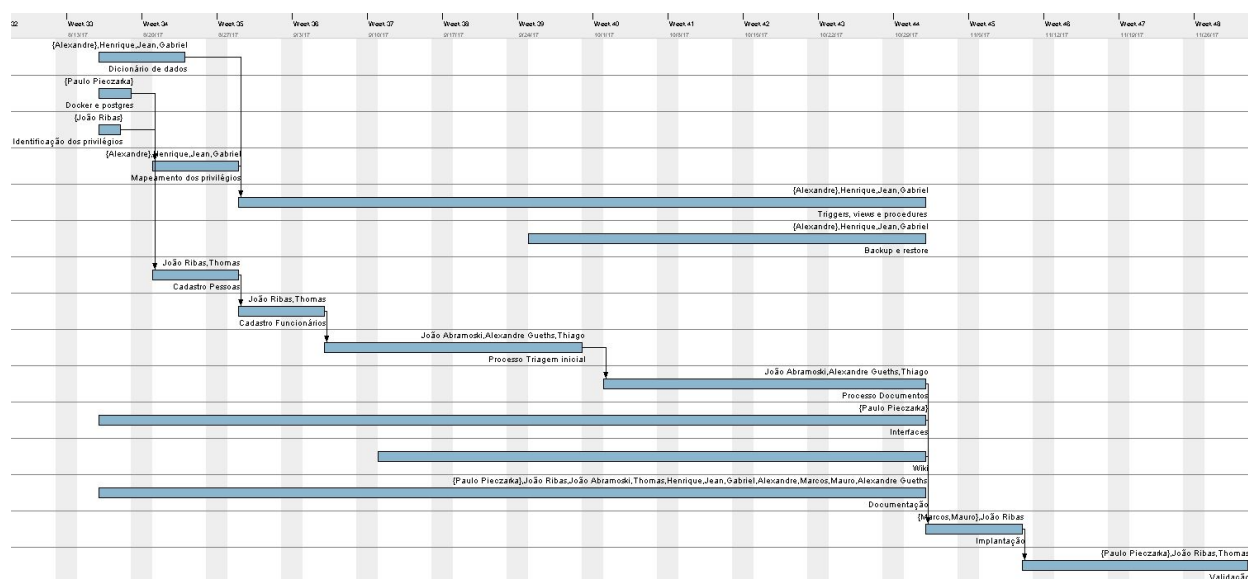


Figura 3.1 - Cronograma do projeto e-DefPR

4. MODELAGEM

Com o objetivo de entender as partes e o funcionamento do software, são desenvolvidos modelos abstratos que simbolizam perspectivas do software. Neste trabalho foi utilizada a Linguagem de Modelagem Unificada - UML (do inglês Unified Modeling Language).

4.1. Casos de Uso

A modelagem dos casos de uso é uma técnica da UML que possibilita entender o sistema a partir de uma visão dinâmica em cenários de utilização (interação com o ambiente). Segundo Sommerville [1] em sua forma mais simples, um caso de uso identifica os atores envolvidos em uma interação com o software e dá nome ao tipo de interação. Logo depois são inseridas informações adicionais descrevendo a interação com o software. Essas informações podem ser dispostas de forma tabular, descrevendo cada o caso de uso [2].

4.1.1. Diagrama(s) de casos de uso

Para melhor compreensão dos requisitos, foram gerados diagramas de casos de uso separados por módulo. A Figura 4.1 é o diagrama de casos de uso referentes ao módulo de Cadastro, onde são cadastrados funcionários, assistidos e as atividades realizadas por um funcionário. A Figura 4.2 é o diagrama de casos de uso referentes ao módulo de Processo, o qual permite realizar a triagem inicial e requisição de documentos de um assistido.

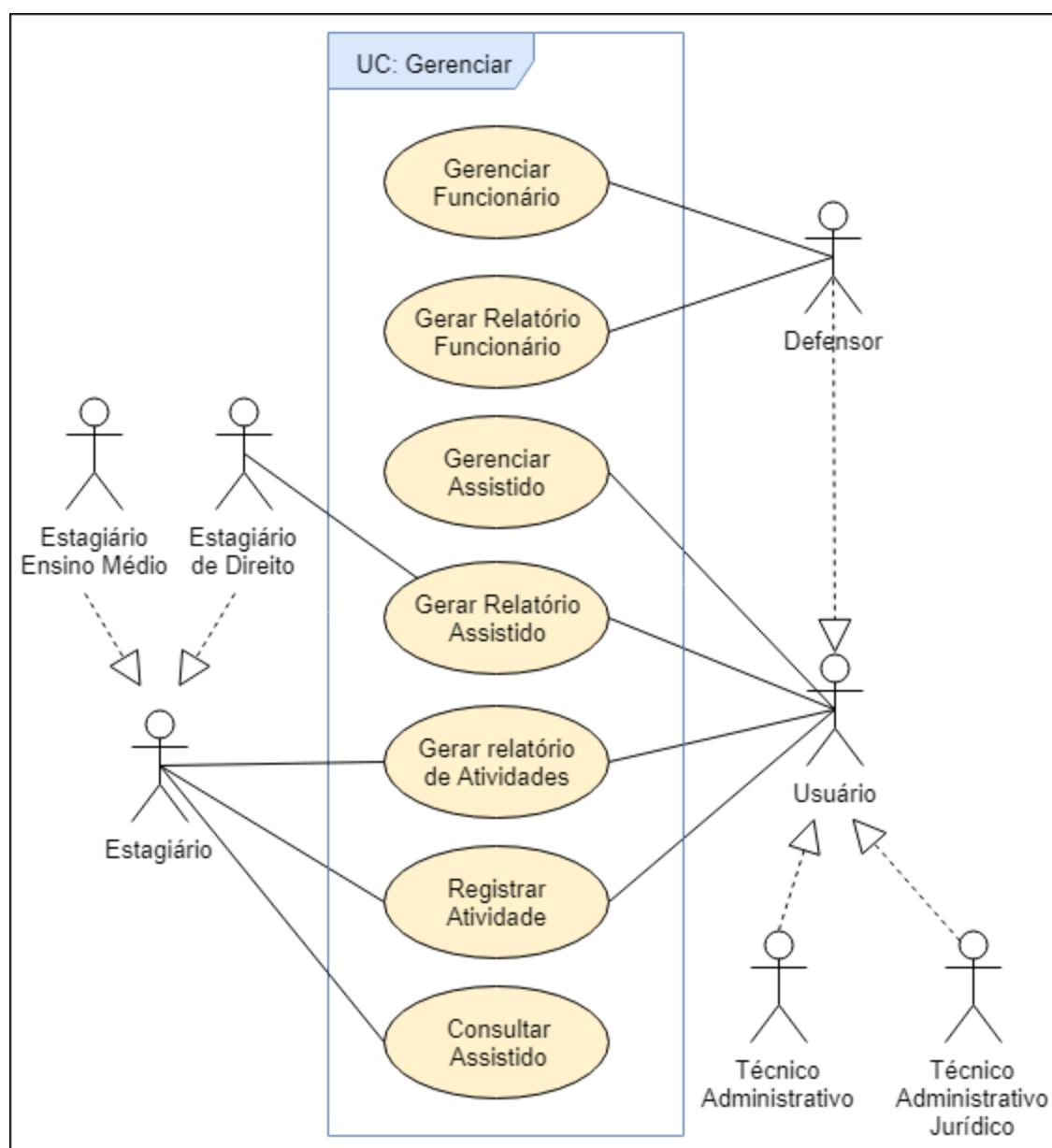


Figura 4.1 - Casos de Uso: Gerenciar pessoas

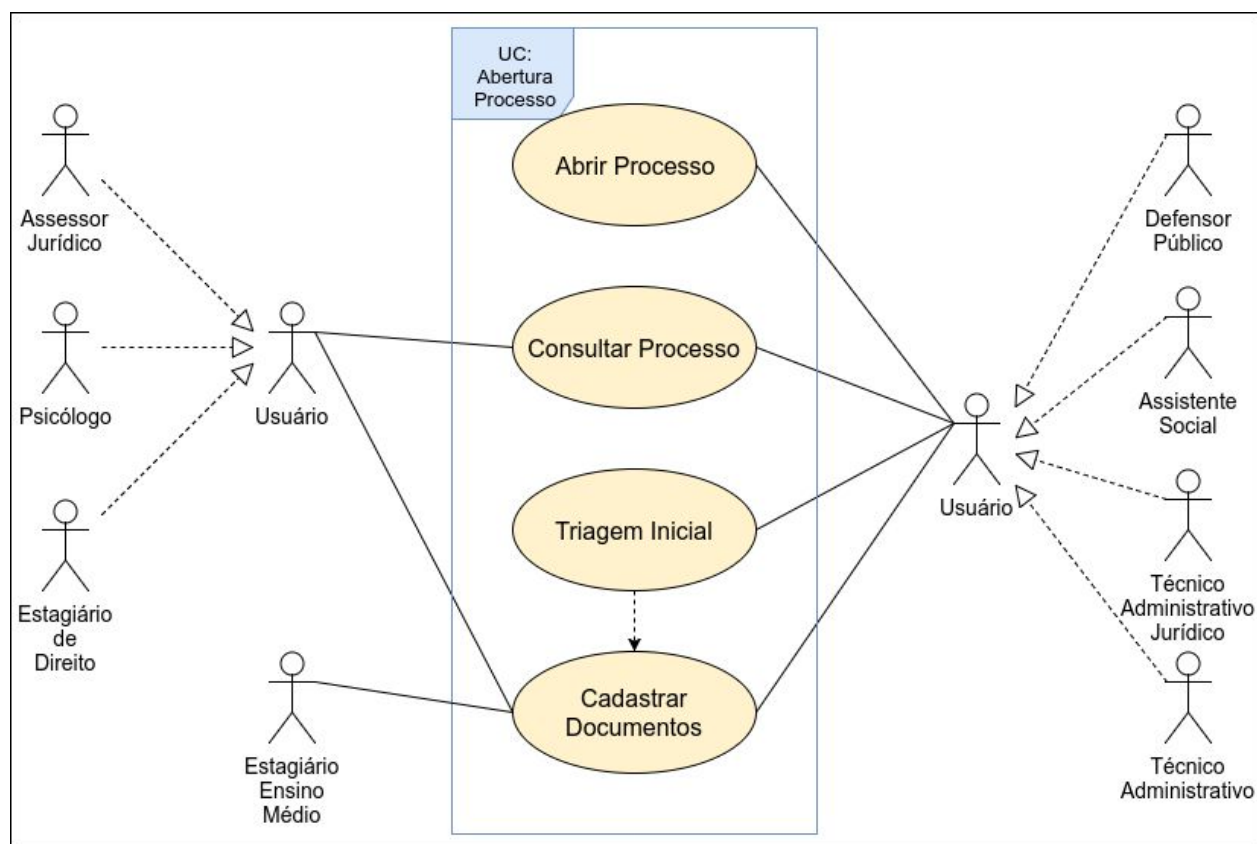


Figura 4.2 Casos de uso - Abertura de processos

4.1.2. Descrição dos casos de uso

Os diagrama de casos de uso são ótimos para mostrar o fluxo de interação entre os casos de uso e atores do sistema. Entretanto, estes diagramas são muito simplificados em termos de detalhes; não mostrando o que acontece dentro de cada caso de uso. Para resolver este problema, foram criados, para cada caso de uso, uma tabela que o descreva. Existem diversos modelos de tabelas descritivas. Para fins deste projeto vamos usar o modelo prescrito por Guedes [2]. As tabelas 4.1 a 4.7 descrevem os casos de uso do módulo de Cadastro e as tabelas 4.8 a 4.11 descrevem os casos de uso do módulo de Processo.

Tabela 4.1 - Gerenciar Funcionário

Nome do Caso de Uso	Gerenciar Funcionário
Caso de Uso Geral	
Ator Principal	Defensor
Atores Secundários	
Resumo	Este caso de uso descreve as etapas percorridas por um Defensor Público para incluir um funcionário.
Pré-condições	O Defensor Público deve estar logado no sistema.
Pós-condições	Mostrar o cadastro do funcionário inserido.
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Selecionar a opção de incluir funcionário.	
	2. Carregar/exibir interface de preenchimento dos dados
3. Inserir dados do funcionário.	
	4. Verificar os dados preenchidos
	5. Em caso de sucesso exibir mensagem de concluído. Em caso de falha exibir possíveis razões.
Restrições / Validações	

Tabela 4.2 - Gerar relatório funcionário

Nome do Caso de Uso	Gerar relatório funcionário
Caso de Uso Geral	
Ator Principal	Defensor Público
Atores Secundários	
Resumo	Gerar relatórios de determinado período de tempo (Mensal, Anual, Customizado) com as atividades do funcionário.
Pré-condições	Estar logado no sistema.
Pós-condições	Exibir página com os atendimentos realizados e atividades.
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Selecionar a opção de gerar relatório funcionário	
	2. Carregar/exibir interface de preenchimento dos dados
3. Inserir data inicial e data final ou selecionar um relatório mensal ou anual.	
	4. Verificar os dados preenchidos
	5. Em caso de sucesso exibir mensagem de concluído. Em caso de falha exibir possíveis razões.
	6. Exibir página com os atendimentos realizados e atividades.
Restrições / Validações	

Tabela 4.3 - Gerenciar assistido

Nome do Caso de Uso	Gerenciar assistido		
Caso de Uso Geral			
Ator Principal	Usuário		
Atores Secundários	Defensor, Técnico Administrativo Jurídico, Técnico Administrativo e Estagiário de Direito.		
Resumo	Este caso de uso descreve as etapas percorridas por um usuário para incluir, alterar e remover um assistido.		
Pré-condições	Estar logado no sistema.		
Pós-condições	Exibir uma tela com os funcionários gerenciados.		
Fluxo Principal			
Ações do Ator		Ações do Sistema	
1. Selecionar a opção de inserir assistido.			
		2. Carregar/exibir interface de preenchimento dos dados	
3. Inserir dados do assistido			
		4. Verificar os dados preenchidos	
		5. Em caso de sucesso exibir mensagem de concluído. Em caso de falha exibir possíveis razões.	
Restrições / Validações			

Tabela 4.4 - Gerar relatório assistido

Nome do Caso de Uso	Gerar relatório assistido	
Caso de Uso Geral		
Ator Principal	Usuário	
Atores Secundários	Defensor, Técnico Administrativo Jurídico, Técnico Administrativo e Estagiário de Direito.	
Resumo	Gerar relatório de determinado período de tempo (Mensal, Anual, Customizado) com as atividades do assistido.	
Pré-condições	Estar logado no sistema.	
Pós-condições	Exibir página com as informações e as atividades.	
Fluxo Principal		
Ações do Ator		Ações do Sistema
1. Selecionar a opção de gerar relatório assistido.		
		2. Carregar/exibir interface de preenchimento dos dados
3. Inserir data inicial e data final ou selecionar um relatório mensal ou anual.		
		4. Verificar os dados preenchidos
		5. Em caso de sucesso exibir mensagem de concluído. Em caso de falha exibir possíveis razões.
		6. Exibir página com as informações e as atividades.
Restrições / Validações		

Tabela 4.5 - Gerar Relatório de Atividades

Nome do Caso de Uso	Gerar Relatório de Atividades		
Caso de Uso Geral			
Ator Principal	Usuário		
Atores Secundários	Defensor, Técnico Administrativo Jurídico, Técnico Administrativo, Estagiário Ensino Médio e Estagiário de Direito		
Resumo	Este caso de uso descreve as etapas percorridas por um usuário para gerar relatório de suas atividades		
Pré-condições	O usuário deve estar logado no sistema		
Pós-condições	Exibir uma tela com o relatório gerado		
Fluxo Principal			
Ações do Ator		Ações do Sistema	
1. Selecionar a opção para gerar relatório de atividades			
		2. Exibir caixa de seleção de filtros	
3. Selecionar os filtros de geração			
		3. Se for possível gerar: exibir uma o relatório gerado; senão mostrar os possíveis motivos.	
		4. Registrar uma atividade de log sobre a requisição do mesmo.	
Restrições / Validações			

Tabela 4.6 - Registrar Atividade

Nome do Caso de Uso	Registrar Atividade		
Caso de Uso Geral			
Ator Principal	Usuário		
Atores Secundários	Defensor, Técnico Administrativo Jurídico, Técnico Administrativo, Estagiário Ensino Médio e Estagiário de Direito		
Resumo	Este caso de uso descreve as etapas percorridas por um usuário para registrar uma atividade		
Pré-condições	O usuário deve estar logado no sistema		
Pós-condições	Exibir uma mensagem de sucesso ou falha no registro		
Fluxo Principal			
Ações do Ator		Ações do Sistema	
1. Selecionar a opção de registro de atividade			
		2. Exibir o formulário de para registro	
3. Inserir dados do registro			
4. Descrever o registro			
		5. Colocar a data e hora nos dados	
		6. Em caso de sucesso exibir mensagem de concluído. Em caso de falha exibir possíveis razões.	
Restrições / Validações			

Tabela 4.7 - Consultar assistido

Nome do Caso de Uso	Consultar assistido		
Caso de Uso Geral			
Ator Principal	Estagiário Ensino Médio		
Atores Secundários			
Resumo	Este caso de uso descreve as etapas percorridas por um Estagiário para consultar um assistido		
Pré-condições	Estar logado no sistema		
Pós-condições	Exibir uma tela com as informações do assistido		
Fluxo Principal			
Ações do Ator		Ações do Sistema	
1. Selecionar a opção de consulta de assistido			
		2. Exibir a lista de assistidos	
3. Inserir nome, CPF ou telefone do assistido			
		4. Retornar a lista de possíveis assistidos	
5. Selecionar o assistido			
		6. Em caso de sucesso exibir os dados do assistido. Em caso de falha exibir possíveis razões.	
Restrições / Validações			

Tabela 4.8 - Abrir Processo

Nome do Caso de Uso	Abrir Processo	
Caso de Uso Geral		
Ator Principal	Defensor Público	
Atores Secundários	Técnico Administrativo Jurídico, Técnico Administrativo e Assistente Social	
Resumo	Este caso de uso descreve as etapas percorridas por um Defensor Público para iniciar o procedimento de abrir um processo	
Pré-condições	O Defensor Público deve estar logado no sistema com algum usuário que tenha permissão de abrir um processo	
Pós-condições	Exibir uma tela com o processo criado	
Fluxo Principal		
Ações do Ator		Ações do Sistema
1. Selecionar a opção de abertura de processo		
		2. Carregar/exibir interface de preenchimento dos dados
3. Inserir dados do processo		
		4. Verificar os dados preenchidos
		5. Em caso de sucesso exibir mensagem de concluído. Em caso de falha exibir possíveis razões.
Restrições / Validações		

Tabela 4.9 - Consultar Processo

Nome do Caso de Uso	Consultar Processo		
Caso de Uso Geral			
Ator Principal	Defensor Público		
Atores Secundários	Técnico Administrativo Jurídico, Técnico Administrativo, Assistente Social, Estagiário de Direito, Assessor Jurídico e Psicólogo		
Resumo	Este caso de uso descreve as etapas percorridas por um Defensor Público para consultar um processo em andamento		
Pré-condições	O Defensor Público deve estar logado no sistema com algum usuário que tenha permissão de abrir um processo e é necessário que o assistido atendido tenha um processo cadastrado		
Pós-condições	Exibir uma tela com os dados do processo ou dizendo que o processo não existe		
Fluxo Principal			
Ações do Ator		Ações do Sistema	
1. Selecionar a opção de consulta de processo			
		2. Carregar/exibir interface de preenchimento dos dados	
3. Inserir nome do processo ou número			
		4. Verificar os dados preenchidos	
		5. Em caso de sucesso exibir tela com informações referentes ao processo pesquisado. Em caso de falha exibir motivo.	
Restrições / Validações			

Tabela 4.10 - Triagem Inicial


Nome do Caso de Uso	Triagem Inicial
Caso de Uso Geral	
Ator Principal	Defensor Público
Atores Secundários	Técnico Administrativo Jurídico, Técnico Administrativo e Assistente Social
Resumo	Este caso de uso descreve a triagem inicial do assistido, onde é coletado os dados pessoais do assistido. Um dos primeiros contatos de um assistido com a defensoria. Deve ser possível editar os dados de um assistido após ter passado pela triagem inicial.
Pré-condições	Estar logado com login de um funcionário no sistema.
Pós-condições	Após a mensagem de sucesso exibir uma página com todos os dados cadastrados para revisão ou retornar a página de triagem inicial vazia novamente
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Selecionar a opção de triagem inicial	
	2. Carregar/exibir interface de preenchimento dos dados
3. Inserir dados necessários (dados pessoais, endereço, dados da parte contrária, atendimentos, andamentos, contatos e anexos)	
	4. Verificar os dados preenchidos
	5. Em caso de sucesso exibir mensagem de sucesso, em caso de falha exibir motivo
Restrições / Validações	

Tabela 4.11 - Cadastrar Documentos

Nome do Caso de Uso	Cadastrar Documentos
Caso de Uso Geral	
Ator Principal	Defensor Público
Atores Secundários	Técnico Administrativo Jurídico, Técnico Administrativo, Assistente Social, Estagiário de Direito, Assessor Jurídico, Psicólogo e Estagiário Ensino Médio
Resumo	Utilizado quando falta algum documento durante um atendimento e então o assistido traz posteriormente, necessitando assim cadastrar esses novos documentos.
Pré-condições	Estar logado com uma conta que permita cadastrar documentos.
Pós-condições	Após a mensagem de sucesso voltar a página de cadastrar documentos
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Selecionar a opção de cadastrar documentos	
	2. Carregar/exibir interface de preenchimento dos dados
3. Inserir dados necessários (descrição, nome e caminho para o arquivo)	
	4. Verificar os dados preenchidos
	5. Em caso de sucesso exibir mensagem de sucesso, em caso de falha exibir motivo
Restrições / Validações	

4.1.3. Interfaces do software

Logo após terminar os casos de uso, é um bom momento para se criar as interfaces do sistema. Partindo do princípio que os casos de uso representam as funcionalidades requeridas do sistema, logo é necessário, para cada caso de uso, projetar/desenhar uma interface (janela) de interação entre o sistema e o(s) usuário(s). A Figura 4.3 apresenta a tela de login do software eDef-PR, já a Figura 4.4 apresenta a tela principal.



The image shows a login interface for EDEF-PR. It features a white rectangular box centered on a light gray background. Inside the box, at the top left, is a logo consisting of three stylized human figures, followed by the text "EDEF-PR". Below this, there are two input fields: the first is labeled "CPF" and the second is labeled "Senha". To the right of the "Senha" field is a green button with the text "Acessar". Below the "CPF" field, there is a link that says "Esqueci minha senha". At the bottom center of the box, there is a link that says "Ajuda".

Figura 4.3 - Interface de Login do eDef-PR

4.2. Classes

A modelagem de classes proporciona uma visão estática das classes, que são os elementos centrais da orientação a objetos, na formação da estrutura do software.

Neste trabalho, a modelagem das classes foi feita de duas formas; através do diagrama das classes, complementado com a descrição tabular de cada classe, de seus atributos e de seus métodos.

4.2.1. Diagrama de classes

O diagrama de classes, de acordo com Guedes [2], possibilita a visualização das classes, com seus métodos e atributos, bem como os relacionamentos entre as classes. A Figura 4.6 apresenta as classes que compõem o software eDef-PR.

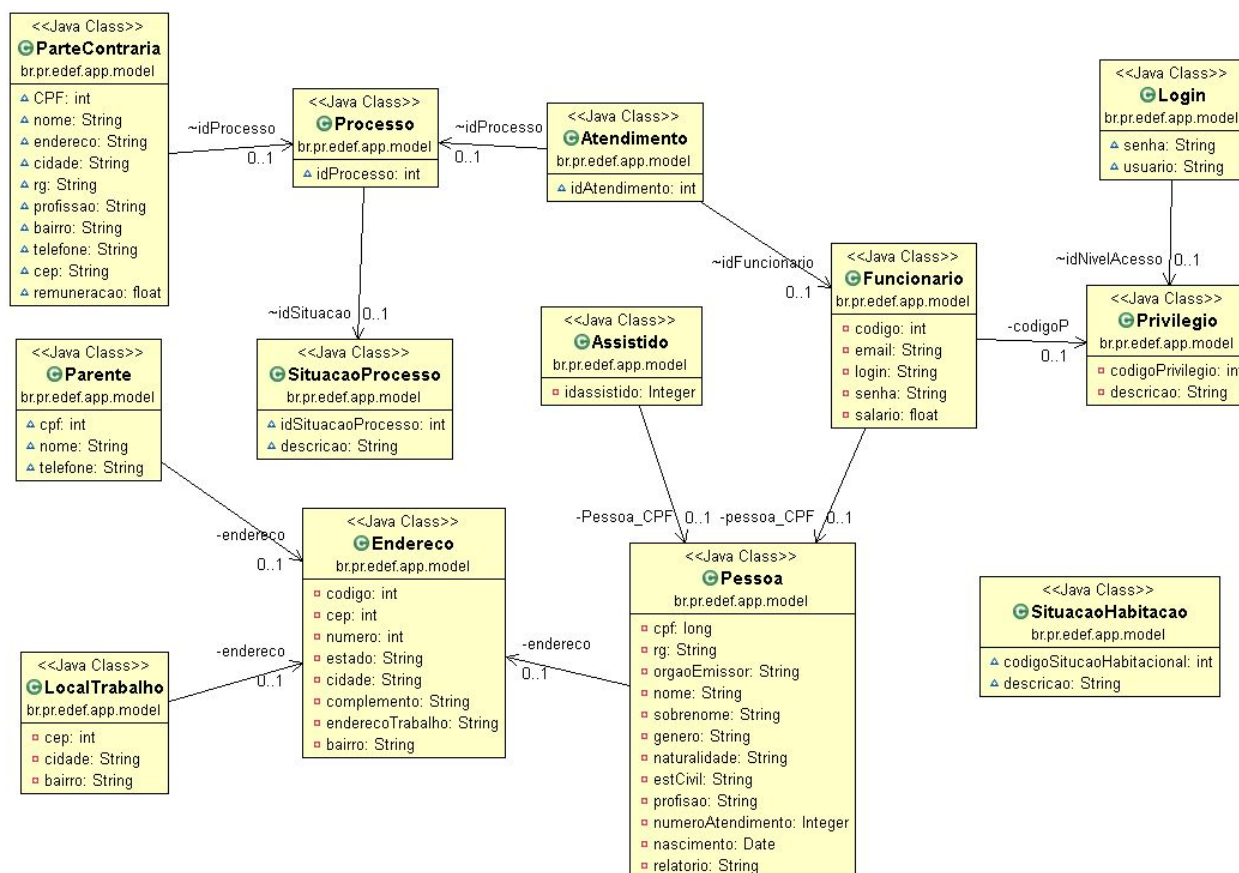


Figura 4.6 - Diagrama de classes

4.2.2. Descrição das classes

Em projetos com várias pessoas trabalhando é comum se esquecer para que serve cada classe, atributo, ou método. Para resolver este problema, foi criada, para cada classe, uma tabela que a descreva. Cada uma das tabelas 4.12 a 4.25 mostra as características de uma classe do sistema.

Tabela 4.12 - Tabela descritiva da classe Endereco

ATRIBUTO	TAMANHO	TIPO	DESCRIÇÃO
bairro		String	o bairro do endereço
cep		int	o Código de Endereçamento Postal do endereço
cidade		String	a cidade do endereço
complemento		String	algum complemento do endereço para melhor identificação
estado		String	estado do endereço
codigo		int	id de controle interno no sistema
numero		int	número do endereço
rua		String	nome do logradouro do endereço
MÉTODO	DESCRIÇÃO		
Endereco()	um construtor com todos os atributos e um sem nenhum atributo		
getters()	métodos para adquirir atributos de uma instância da classe Endereco		
setters()	métodos para definir/atualizar atributos de uma instância da classe Endereco		

Tabela 4.13 - Tabela descritiva da classe Pessoa

ATRIBUTO	TAMANHO	TIPO	DESCRIÇÃO
cpf		long	número do documento da pessoa, usado como id de controle internamente no sistema
endereco		Endereco	instância de uma objeto Endereco, usado para registrar o endereço de moradia da pessoa
estCivil		String	estado civil cadastrado a pessoa
genero		String	gênero cadastrado a pessoa
nascimento		Date	a data de nascimento da pessoa
naturalidade		String	a naturalidade da pessoa
nome		String	o primeiro nome da pessoa
numeroAtendimento		Integer	o número do atendimento da pessoa, existe caso ela seja um assistido
orgaoEmissor		String	o órgão emissor do documento
profisao		String	a profissão da pessoa
relatorio		String	relatório do funcionário, existe caso ela seja um funcionário
rg		String	número do documento de registro geral da pessoa
sobrenome		String	o sobrenome da pessoa
MÉTODO	DESCRIÇÃO		
Pessoa()	um construtor sem nenhum atributo		
getters()	métodos para adquirir atributos de uma instância da classe Pessoa		
setters()	métodos para definir/atualizar atributos de uma instância da classe Pessoa		
toString()	método retorna uma String contendo nome, sobrenome e CPF		

Tabela 4.14 - Tabela descritiva da classe ParteContraria

ATRIBUTO	TAMANHO	TIPO	DESCRIÇÃO
idPartecContraria		int	id de controle interno no sistema
complemento		String	algum complemento, descrição ou relação da parte contrária
remuneracao		float	remuneração da parte contrária
MÉTODO	DESCRIÇÃO		
ParteContraria()	um construtor com todos os atributos e um sem nenhum atributo		
getters()	métodos para adquirir atributos de uma instância da classe ParteContraria		
setters()	métodos para definir/atualizar atributos de uma instância da classe ParteContraria		

Tabela 4.15 - Tabela descritiva da classe Assistido

ATRIBUTO	TAMANHO	TIPO	DESCRIÇÃO
idAssistido		int	id de controle interno no sistema
Pessoa_CPF		Pessoa	referência a uma instância da classe Pessoa com os dados do assistido
MÉTODO	DESCRIÇÃO		
Assistido()	um construtor com todos os atributos e um sem nenhum atributo		
getters()	métodos para adquirir atributos de uma instância da classe Assisto		
setters()	métodos para definir/atualizar atributos de uma instância da classe Assisto		

Tabela 4.16 - Tabela descritiva da classe Privilegio

ATRIBUTO	TAMANHO	TIPO	DESCRIÇÃO
idPrivilegio		int	id de controle interno no sistema
descricao		String	descrição do privilégio associado a um ou mais funcionários
MÉTODO	DESCRIÇÃO		
Privilegio()	um construtor com todos os atributos e um sem nenhum atributo		
getters()	métodos para adquirir atributos de uma instância da classe Privilegio		
setters()	métodos para definir/atualizar atributos de uma instância da classe Privilegio		

Tabela 4.17 - Tabela descritiva da classe Funcionario

ATRIBUTO	TAMANHO	TIPO	DESCRIÇÃO
codigo		int	id de controle interno no sistema
email		String	o Código de Endereçamento Postal do endereço
login		String	login de acesso ao sistema
Pessoa_CPF		Pessoa	referência a uma instância da classe Pessoa com os dados do funcionário
codigoP		Privilegio	referência a uma instância da classe Privilegio com os dados do privilégio do funcionário
salario		Float	salário do funcionário
senha		String	senha de acesso ao sistema
MÉTODO	DESCRIÇÃO		
Funcionario()	um construtor sem nenhum atributo		
getters()	métodos para adquirir atributos de uma instância da classe Funcionario		
setters()	métodos para definir/atualizar atributos de uma instância da classe Funcionario		

Tabela 4.18 - Tabela descritiva da classe Processo

ATRIBUTO	TAMANHO	TIPO	DESCRIÇÃO
codigoProcesso		int	id de controle interno no sistema
descricao		String	um espaço de texto para uma breve descrição sobre o processo
dataAtualizacao		Date	data da última atualização realizada neste processo
MÉTODO	DESCRIÇÃO		
Processo()	um construtor com todos os atributos e um sem nenhum atributo		
getters()	métodos para adquirir atributos de uma instância da classe Processo		
setters()	métodos para definir/atualizar atributos de uma instância da classe Processo		

Tabela 4.19 - Tabela descritiva da classe Pendencia

ATRIBUTO	TAMANHO	TIPO	DESCRIÇÃO
codigoPendencia		int	id de controle interno no sistema
descricaoPendencia		String	um espaço de texto para uma descrição sobre a pendencia
Entrega		boolean	espaco para definir se apendencia foi resolvida ou não
processo		Processo	referencia ao processo relacionaldo a essa pendencia
data solicitacao		Date	data em que foi realizada a solicitação desta pendencia ao assistido
MÉTODO	DESCRIÇÃO		
Pendencia()	um construtor com todos os atributos e um sem nenhum atributo		
getters()	métodos para adquirir atributos de uma instância da classe Pendencia		
setters()	métodos para definir/atualizar atributos de uma instância da classe Pendencia		

Tabela 4.20 - Tabela descritiva da classe Atendimento

ATRIBUTO	TAMANHO	TIPO	DESCRIÇÃO
codigo		int	id de controle interno no sistema
descricao		String	um espaço de texto para uma descrição/anotações sobre o que ocorreu no atendimento
MÉTODO	DESCRIÇÃO		
Atendimento()	um construtor com todos os atributos e um sem nenhum atributo		
getters()	métodos para adquirir atributos de uma instância da classe Atendimento		
setters()	métodos para definir/atualizar atributos de uma instância da classe Atendimento		

Tabela 4.21 - Tabela descritiva da classe Assistencia

ATRIBUTO	TAMANHO	TIPO	DESCRIÇÃO
codigoAssistencia		int	id de controle interno no sistema
relato		String	um espaço de texto para uma descrição/ anotações sobre o que ocorreu na assistência
data		Date	data da onde ocorreu essa assistência
nomeFuncionario		Funcionario	referencia ao Funcionario que forneceu essa assistência
formatoArquivo		String	indicação do formato do anexo
Atendimento_codigo		Atendimento	referência ao atendimento dessa assistência
MÉTODO	DESCRIÇÃO		
Assistencia()	um construtor com todos os atributos e um sem nenhum atributo		
getters()	métodos para adquirir atributos de uma instância da classe Assistencia		
setters()	métodos para definir/atualizar atributos de uma instância da classe Assistencia		

Tabela 4.22 - Tabela descritiva da classe TriagemInicial

ATRIBUTO	TAMANHO	TIPO	DESCRIÇÃO
codigo		int	id de controle interno no sistema
qualificação		String	nível de escolaridade
contato		Contato	contato do assistido
contatos_CPF		Contato	referencia aos contatos
Atendimento_codigo		Atendimento	referência ao atendimento dessa assistência
MÉTODO	DESCRIÇÃO		
TriagemInicial()	um construtor com todos os atributos e um sem nenhum atributo		
getters()	métodos para adquirir atributos de uma instância da classe TriagemInicial		
setters()	métodos para definir/atualizar atributos de uma instância da classe TriagemInicial		

Tabela 4.23 - Tabela descritiva da classe Contatos

ATRIBUTO	TAMANHO	TIPO	DESCRIÇÃO
CPF		int	CPF do contato, usado como controle interno no sistema
descricao		String	descrição do contato (relação com o assistido, processo, etc)
endereco_CEP		Endereco	endereço do contato
telefoneResidencial		String	telefone residencial do contato
telefoneTrabalho		String	telefone do trabalho do contato
telefoneCelular		String	telefone celular do contato
MÉTODO	DESCRIÇÃO		
Contatos()	um construtor com todos os atributos e um sem nenhum atributo		
getters()	métodos para adquirir atributos de uma instância da classe Contatos		
setters()	métodos para definir/atualizar atributos de uma instância da classe Contatos		

Tabela 4.24 - Tabela descritiva da classe TriagemSocioEconomic

ATRIBUTO	TAMANHO	TIPO	DESCRIÇÃO
triagemInicial		TriagemInicial	referência a triagem inicia desse processo
composicaoFamiliar		String	composição familiar do assistido
bens		String	breve descrição dos bens do assistido
nome		String	nome
sobrenome		String	sobrenome
GralDeParentesco		String	
renda		float	renda total do assistido
descricao_Renda		String	descrição da renda total do assistido
codigoProcesso		Processo	referência ao processo dessa triagem
situacaoHabitacao		SituacaoHabitacao	referência a SituacaoHabitacao da triagem
Qualificacao		String	qualificação do assistido
MÉTODO	DESCRIÇÃO		
TriagemSocioEconomic	um construtor com todos os atributos e um sem nenhum atributo		
getters()	métodos para adquirir atributos de uma instância da classe TriagemSocioEconomic		
setters()	métodos para definir/atualizar atributos de uma instância da classe TriagemSocioEconomic		

Tabela 4.25 - Tabela descritiva da classe SituacaoHabitacao

ATRIBUTO	TAMANHO	TIPO	DESCRIÇÃO
codigoSituacaoHabitacao		int	id de controle interno no sistema
descricao		String	um espaço de texto para uma descrever a situação da habitação do assistido
MÉTODO	DESCRIÇÃO		
SituacaoHabitacao()	um construtor com todos os atributos e um sem nenhum atributo		
getters()	métodos para adquirir atributos de uma instância da classe SituacaoHabitacao		
setters()	métodos para definir/atualizar atributos de uma instância da classe SituacaoHabitacao		

4.3. Base de Dados

O projeto DefensoriaPR utiliza um padrão de DAO genérico, assim aproveitando do reuso dos métodos repetitivos para toda classe a ser salva. essa classe genérica contém todas as operações básicas de manipulação de dados, criar, ler, atualizar e deletar, ou CRUD. Portanto todas as tabelas do banco possuem sua classe onde é definida, onde ficam suas anotações necessárias para o funcionamento do Hibernate e uma classe DAO que implementa o DAO genérico, assim, sempre que um método de manipulação de dados contido no DAO genérico for invocado, qualquer que seja o tipo do objeto passado como parâmetro, será utilizado para determinar sua tabela correspondente, e como interpretar os dados contidos no objeto, sem a necessidade de desenvolver essas consultas manualmente para toda classe.

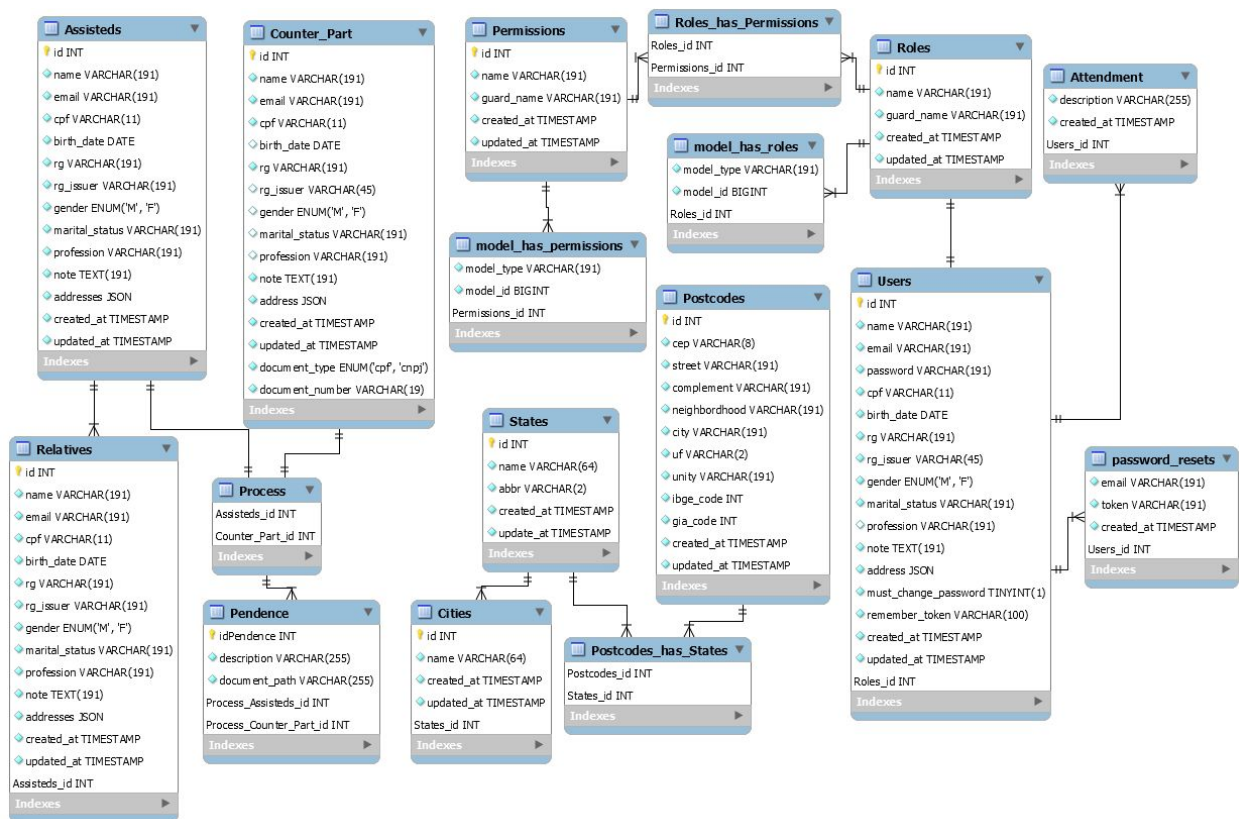


Figura 4.7 - Banco de dados do sistema E-Def Pr

4.4. Diagramas de Sequência

O objetivo dos diagramas de sequência é descrever as interações entre os objetos das classes do sistema. Um diagrama de sequência contém detalhes das comunicações necessárias entre objetos das classes para a execução de um caso de uso do software. São mostradas ao longo de linhas de tempo, a sequência de comunicações entre objetos [2].

As figuras de 4.8 a 4.14 apresentam os diagramas de sequência relativos aos casos de uso do software.

A Figura 4.8 apresenta o diagrama de sequência sobre o caso de uso Excluir Funcionário. Esse diagrama foi criado exemplificando a comunicação entre as partes envolvidas no sistema. Sendo elas: o usuário, o cliente de interface, o servidor da aplicação e a base de dados do sistema.

As figuras de 4.9 a 4.14 apresentam diagramas de funcionamento interno do sistema. Ressalta-se que o cliente de interface está representado com a cor verde, enquanto o servidor e a base de dados está representado na cor azul. Essa distinção ocorre para mostrar a natureza distribuída da aplicação.

A Figura 4.9 apresenta o diagrama de sequência para as funcionalidades de gerenciamento de entidades. O gerenciamento consiste em criação, leitura, alteração, e exclusão da entidade em questão. Como essas funcionalidades não possuem variações significativas entre cada entidade, estão todas representadas neste diagrama.

A Figura 4.10 apresenta o diagrama de sequência para o caso de uso Cadastrar Atividade, podendo-se observar a criação de uma atividade realizada. Já a Figura 4.11 apresenta o diagrama de sequência para o caso de uso Consultar Assistido, que observa-se a leitura das informações de um Assistido. Estes diagramas exemplificam como o diagrama de sequência de gerenciamento engloba com precisão as funcionalidades que descreve.

As figuras 4.12 a 4.14 apresentam diagramas para geração de relatório de atividades, do assistido e de funcionário, respectivamente.

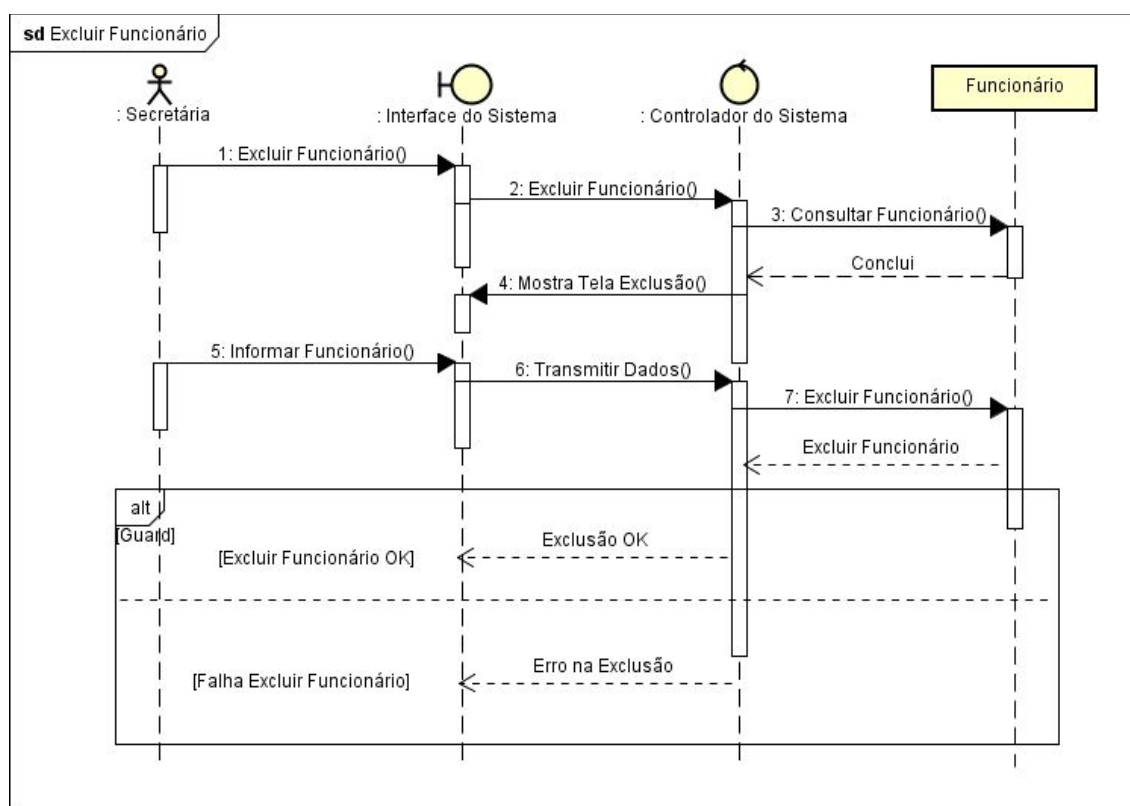


Figura 4.8 - Diagrama de sequência para o caso de uso Excluir Funcionário

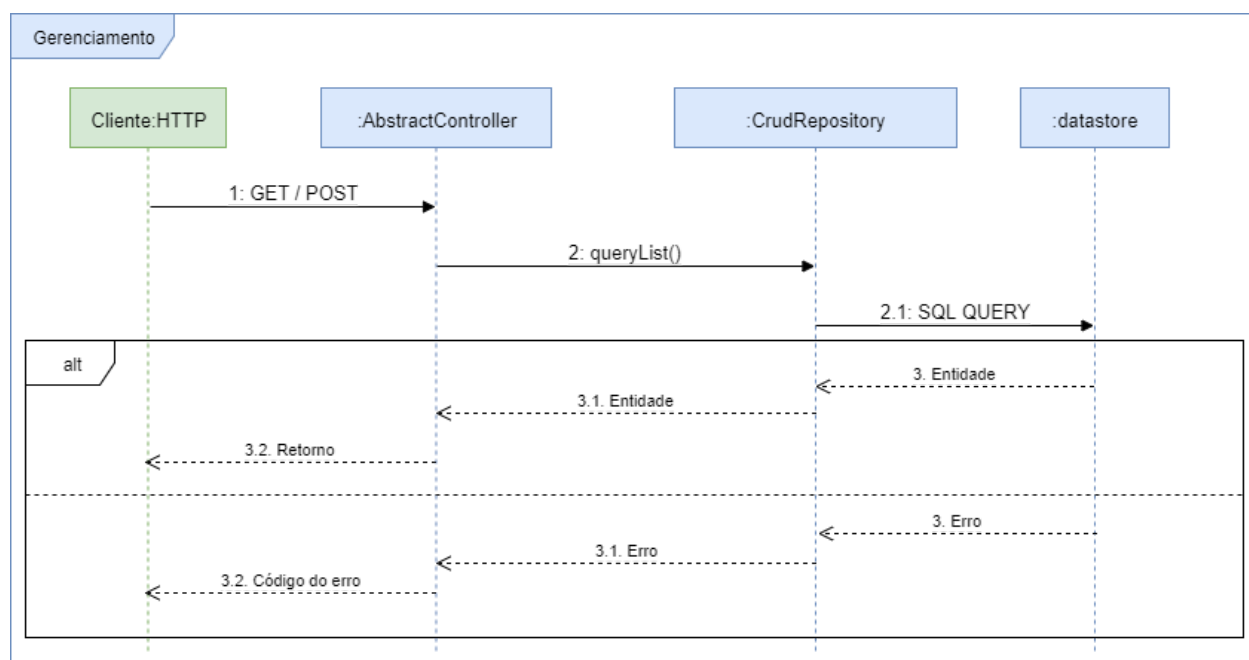


Figura 4.9 - Diagrama de sequência para as operações de gerenciamento

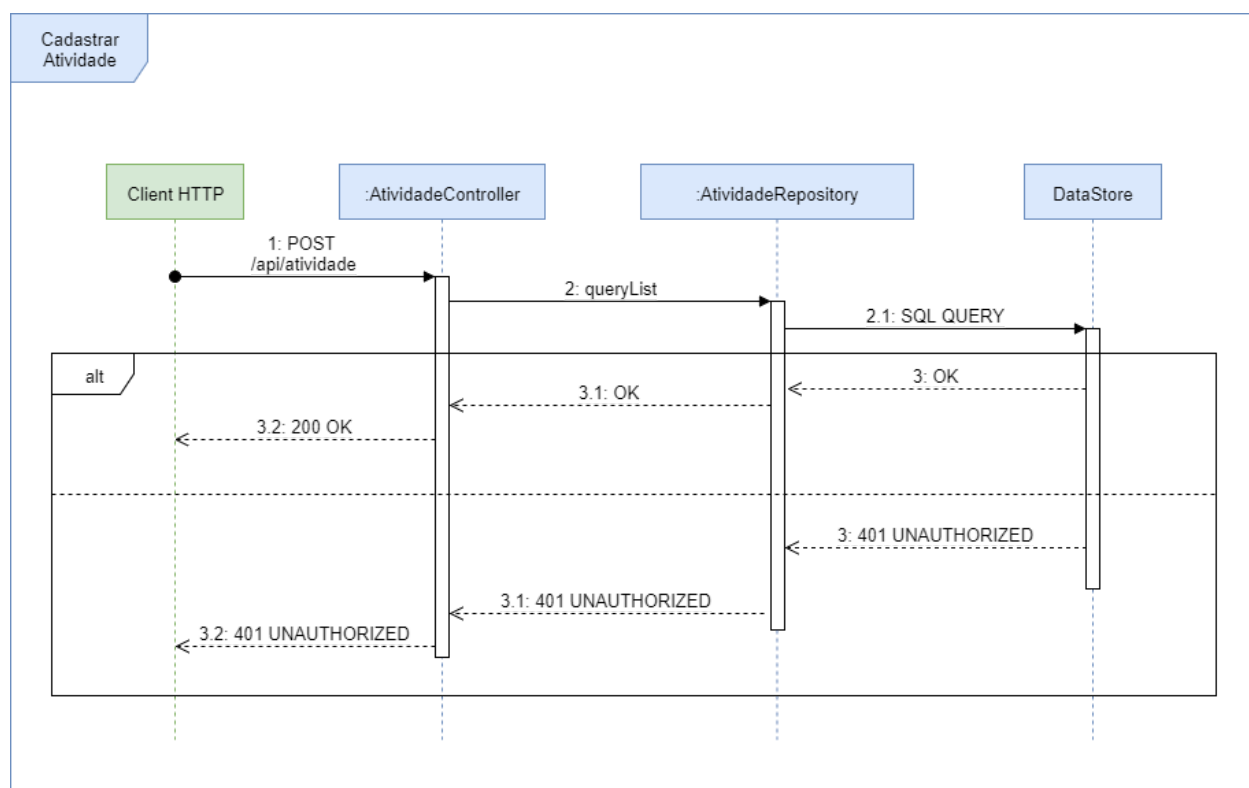


Figura 4.10 - Diagrama de sequência para o caso de uso Cadastrar Atividade

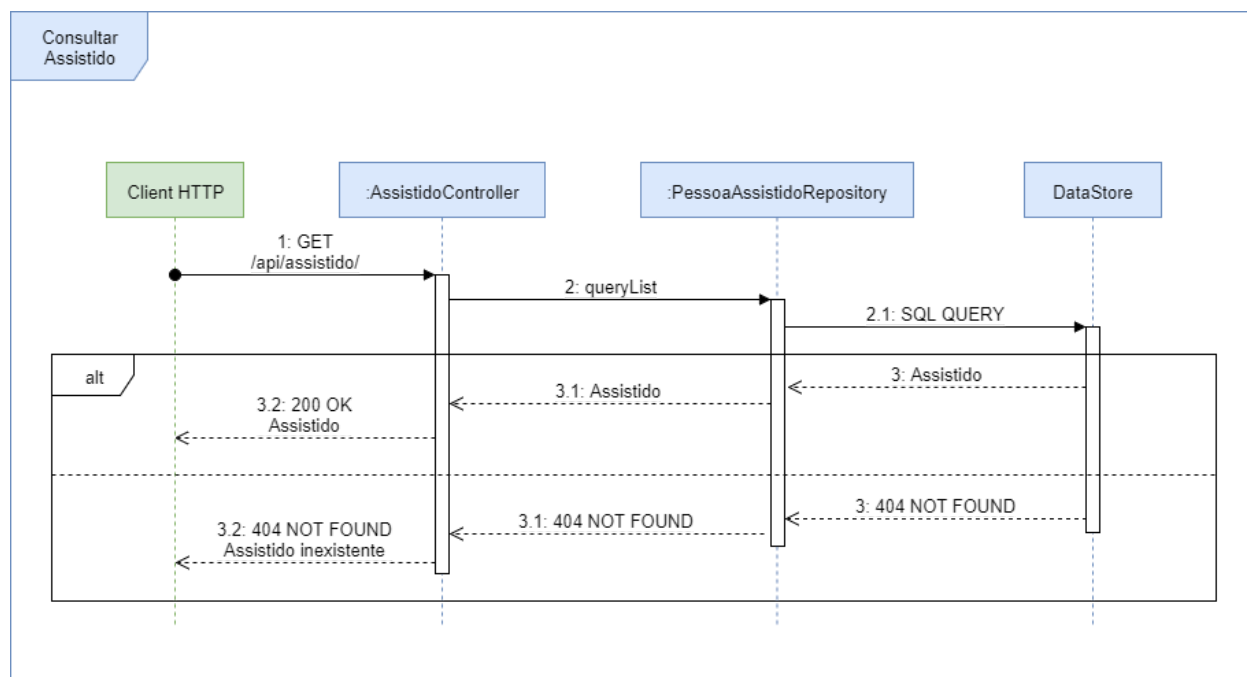


Figura 4.11 - Diagrama de sequência para o caso de uso Consultar Assistido

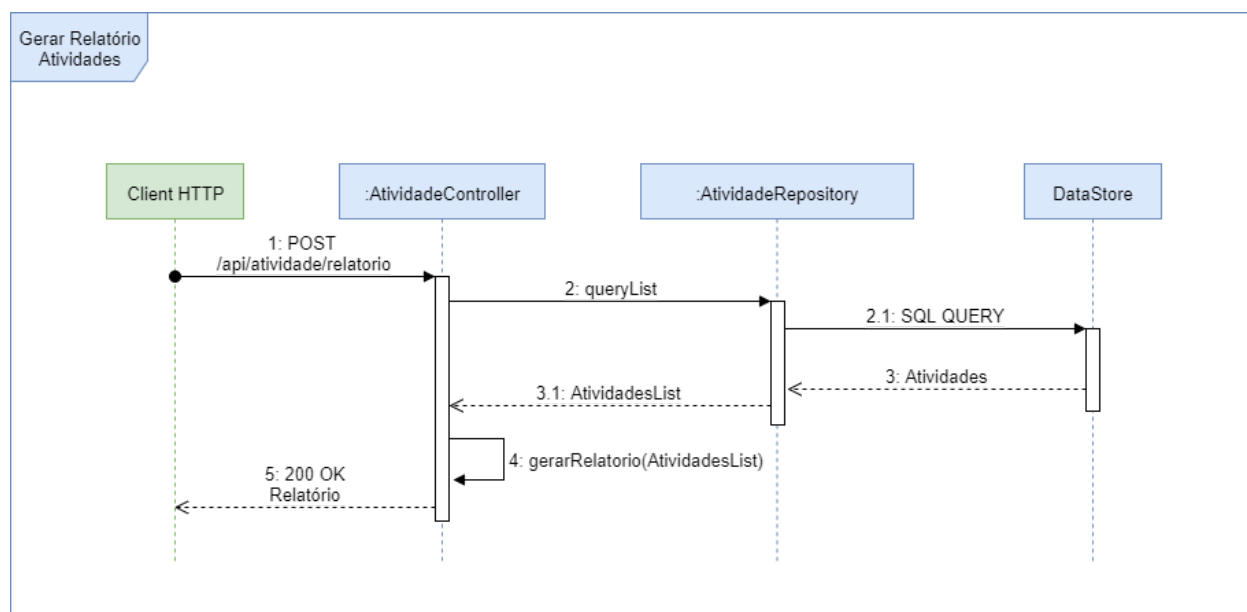


Figura 4.12 - Diagrama de sequência para o caso de uso Gerar Relatório Atividades

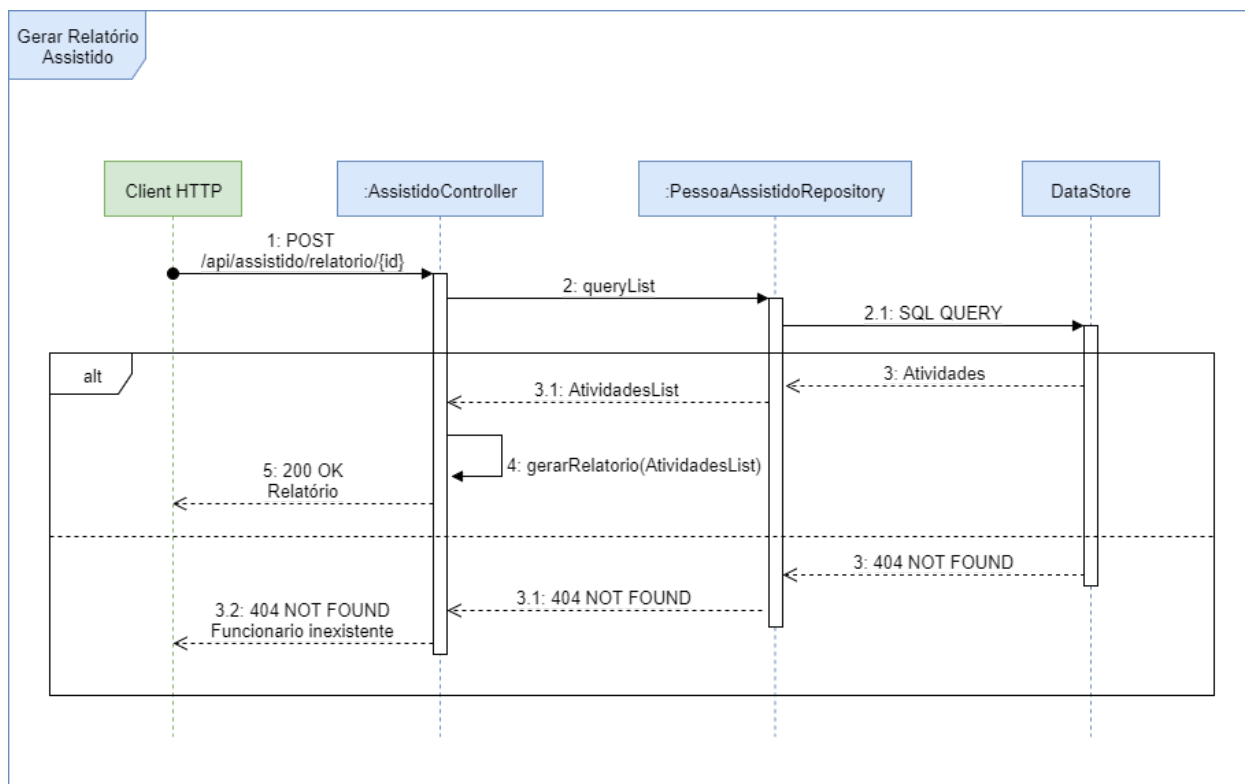


Figura 4.13 - Diagrama de sequência para o caso de uso Gerar Relatório Assistido.

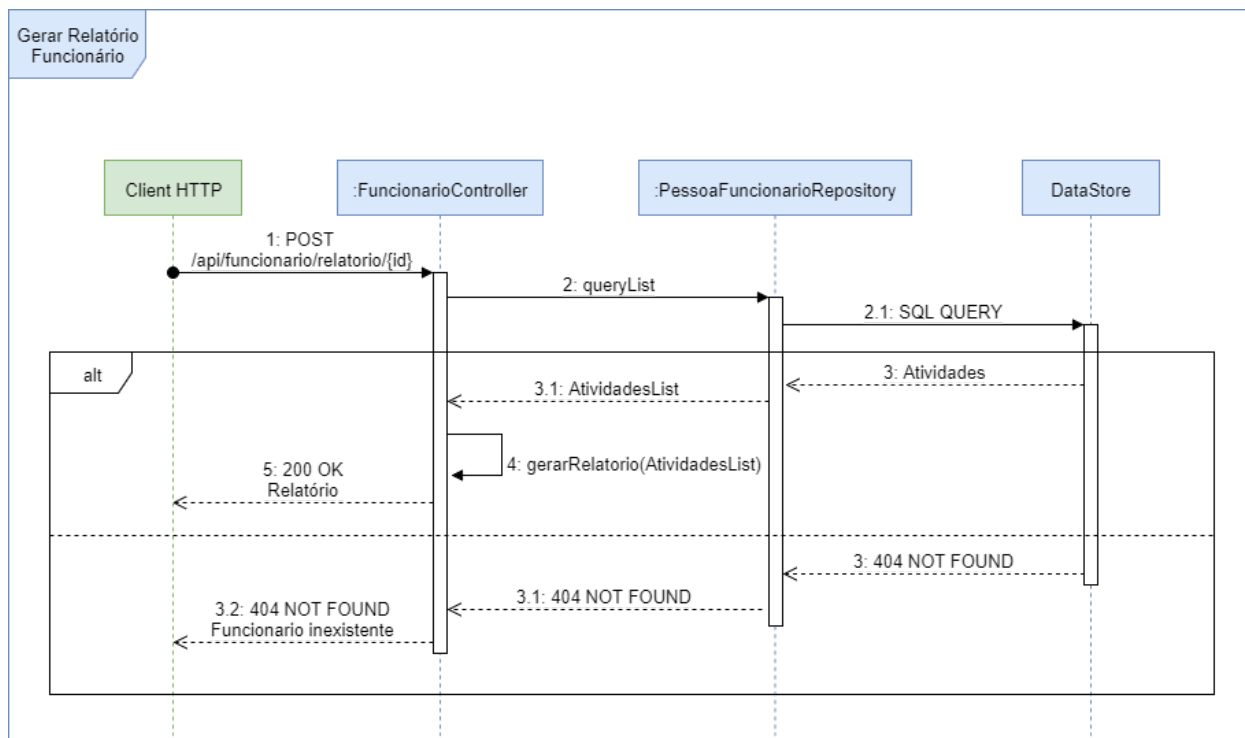


Figura 4.14 - Diagrama de sequência para o caso de uso Gerar Relatório Funcionário

5. ARQUITETURA

Para Bosch [2000] citado por Sommerville [1] a definição da arquitetura do software é importante por afetar o desempenho e a robustez, bem como influenciar na capacidade de distribuição e de manutenibilidade do sistema. Sabe-se também que é impossível que qualquer sistema seja otimizado em todos os atributos de sua arquitetura. Nas subseções a seguir são descritos alguns aspectos e características da arquitetura do software objeto de trabalho deste projeto.

5.1. Confiabilidade

Sommerville [1] define a confiabilidade de um sistema como sendo a probabilidade de, durante determinado período, o sistema prestar serviços corretamente, conforme o esperado pelo usuário, em determinado ambiente, para uma finalidade específica. “A confiabilidade inclui ‘correção’ (garantia de que os serviços de sistema são conforme diz suas especificações), ‘precisão’ (garantia de que a informação é entregue com o nível de detalhamento adequado) e ‘em tempo certo’ (garantindo que a informação é entregue quando necessário)”.

5.2. Proteção

A proteção reflete a capacidade do software resistir às invasões acidentais ou deliberadas. A proteção de sistemas não depende apenas das decisões de projeto locais, mas também é afetada pela proteção de aplicações externas, pelos *web services* e pela infraestrutura da rede [1].

Tendo isso como principais características de proteção, o acesso ao sistema é de grande importância, por isso o sistema deve possuir uma forma de autenticação de acesso, mediante ao CPF e senha de um usuário. O acesso do usuário ao sistema é feito utilizando-se do protocolo OAuth2, que permite autenticação de forma segura à web-services [4]. Ademais, a senha do usuário é armazenada no banco de dados criptografada com bcrypt, que foi criada com o intuito de manter-se segura de ataques de força bruta [5].

Uma autenticação de URL deve ser feita para restringir acesso ao sistema, somente os usuários autorizados devem conseguir concluir esse acesso.

Outra função necessária seria se o usuário esquecer a senha, deve ser possível ele recuperar a mesma através de uma funcionalidade dentro do sistema, essa funcionalidade deve ser segura para que somente o usuário que fez a solicitação de recuperar a senha consiga utilizá-la.

5.3. Disponibilidade

A disponibilidade diz respeito a probabilidade de um sistema estar ativo e funcionando para fornecer serviços aos usuários quando estes solicitarem em determinado momento [Sommerville, 1].

O sistema deve estar disponível para uso no horário de atendimento da Defensoria, em alguns casos específicos o sistema deve entrar em manutenção e então não estará acessível durante esse período.

5.4. Robustez

Segundo SWEBOK [6] Robustez de um software refere-se à capacidade do mesmo tolerar entradas de dados erradas. Desta forma, um software é dito ser robusto se continuar a funcionar mesmo quando forem fornecidas entradas de dados erradas.

Fazer um tratamento de entrada de dados, quando dados errados forem enviados.

5.5. Resiliência

Segundo Sommerville [1], a resiliência, também chamada de sobrevivência é característica importante e desejável que reflete a capacidade do software continuar em operação e prestar serviços essenciais enquanto estiver sob ataque, ou após parte do sistema ter sido danificada.

5.6. Portabilidade

A portabilidade reflete a capacidade de um software poder ser portado de um ambiente para outro; capacidade de ser compilado ou executado em diferentes arquiteturas.

Para garantir a portabilidade do sistema e também aumentar a manutenibilidade do sistema o desenvolvimento usará como guia o padrão Model View Controller (MVC). O desacoplamento dos modelos de entidades e visão, permite o software sofra evoluções incrementais em módulos.

5.7. Usabilidade

Essa propriedade reflete quão fácil é usar o sistema [1]. A usabilidade depende dos componentes técnicos de sistema, seus operadores e seu ambiente operacional.

Utilizou-se padrões de usabilidade para que o usuário se sinta confortável na utilização do sistema. Ademais, o sistema foi desenvolvido com design responsivo, ou seja, o usuário terá uma boa experiência com o sistema independente da plataforma que utilizará para acessar o sistema.

A biblioteca React foi utilizada para aumentar a usabilidade e também acessibilidade da interface do sistema. Esta biblioteca além de verificar por erros e renderizar os componentes que formam a interface, procura por falhas de responsividade e acessibilidade, aumentando a usabilidade do sistema final. Os resultados das métricas aplicadas no desenvolvimento podem ser visualizado na Figura 5.1, esses são dados fornecidos após um teste de usabilidade (Best Practices) e acessibilidade com a ferramenta Lighthouse¹ da Google.

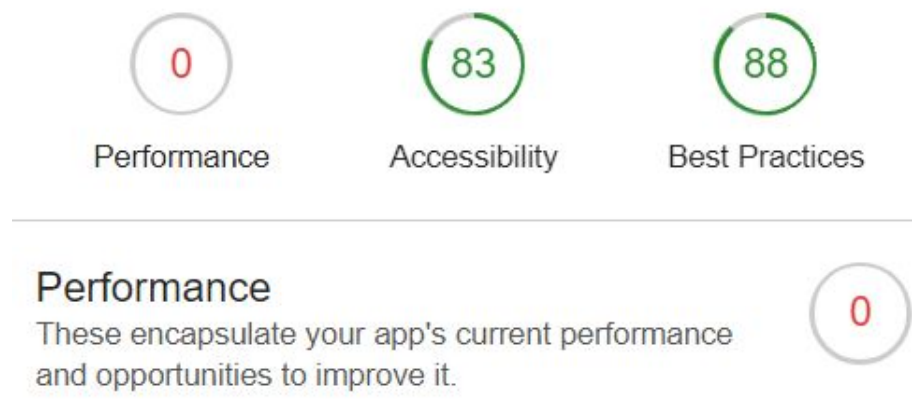


Figura 5.1 - Teste realizado pelo Lighthouse

6. VALIDAÇÃO DE REQUISITOS E PROJETO

A validação dos requisitos e do projeto foi feita através de reuniões com a representante do DPPR, onde era-se discutidas as necessidades da DPPR que seriam sanadas com o sistema. Tais reuniões eram realizadas periodicamente, tanto para validar os requisitos quanto para acompanhar o desenvolvimento do projeto.

Os requisitos levantados foram descritos previamente em outro documento e, posteriormente, transcritos para o presente documento. Com a aprovação do projeto dada pela representante do DPPR, deu-se continuidade com o desenvolvimento do projeto.

7. PROJETO E EXECUÇÃO DE CASOS DE TESTE

Para Sommerville [1], O teste visa mostrar que um programa cumpre o que é proposto a fazer, como também para descobrir os defeitos do programa antes do uso.

Os casos de teste são usados em cenários de execução dinâmica com o objetivo de exercitar adequadamente as estruturas internas do software. Cada caso de teste é composto

¹ <https://developers.google.com/web/tools/lighthouse/>

por valores de entrada, o resultado ou comportamento esperado e, caso existam, as restrições para a sua execução.

Uma abordagem para compor os casos de teste usa os casos de uso do software. Briand e Labiche [7] propuseram uma metodologia de teste chamada *Totem*, na qual as especificações dos testes são feitas a partir dos Casos de Uso.

Cada subseção a seguir apresenta um caso de teste desenvolvido a partir de um caso de uso do software construído, o qual será testado usando os casos de teste aqui apresentados.

7.1. Caso de teste: Gerenciamento de Funcionário .

Objetivo do Teste	Este caso de Teste valida o Gerenciamento do Funcionário.	
Cenários	Cenário 1	Cenário 2
Entrada	Dados para incluir um novo funcionário foram inseridos corretamente.	Adicionar um novo funcionário com um CPF que já possui registro no sistema.
Saída	Exibe mensagem de concluído.	Exibe mensagem de erro e suas razões.

7.2. Caso de teste: Gerar relatório funcionário

Objetivo do Teste	Gerar relatório de determinado período de tempo com as atividades de um funcionário.	
Cenários	Cenário 1	Cenário 2
Entrada	Selecionar o funcionário e a data que deseja gerar o relatório.	Seleciona o funcionário, porém a data desejada não existe.
Saída	Exibe mensagem de concluído.	Exibe mensagem de erro e suas razões.



7.3. Caso de teste: Gerenciar assistindo

Objetivo do Teste	Gerenciar as etapas percorridas por um usuário para incluir, alterar e remover um assistido.	
Cenários	Cenário 1	Cenário 2
Entrada	Os dados do assistindo são inseridos corretamente.	Os dados do assistidos foram inseridos incorretamente.
Saída	Exibe mensagem de concluído.	Exibe mensagem de erro e suas razões.

7.4. Caso de teste: Gerar relatório assistido

Objetivo do Teste	Gerar relatório de uma determinada data para as atividades do assistido.	
Cenários	Cenário 1	Cenário 2
Entrada	Os dados do assistido foram inseridos corretamente e a data tbm.	A data do relatório não existe.
Saída	Exibe mensagem de concluído.	Exibe mensagem de erro e suas razões.

7.5. Caso de teste: Gerar relatório de atividades

Objetivo do Teste	Gerar relatório relatório de atividades.	
Cenários	Cenário 1	Cenário 2
Entrada	Funcionário seleciona a opção de gerar relatório e completa corretamente os campos.	Funcionário seleciona a opção de gerar relatório e completa incorretamente os campos ou não completa.
Saída	Exibe mensagem de concluído e gerar uma atividade de log.	Exibe mensagem de erro e suas razões.

7.6. Caso de teste: Registrar atividade

Objetivo do Teste	Registrar atividade de um funcionário.	
Cenários	Cenário 1	Cenário 2
Entrada	O funcionário insere os dados corretamente e registra uma atividade.	O funcionário não insere os dados necessário e registrar uma atividade
Saída	Exibe mensagem de concluído.	Exibe mensagem de erro e suas razões.

7.7. Caso de teste: Consultar assistido

Objetivo do Teste	Consulta de um assistindo	
Cenários	Cenário 1	Cenário 2
Entrada	Inserir o CPF correto do assistido	Inserir um CPF inválido.
Saída	Exibe mensagem de concluído e exibe dados do assistido.	Exibe mensagem de erro e suas razões.

7.8. Caso de teste: Abrir um processo

Objetivo do Teste	Abrir um processo.	
Cenários	Cenário 1	Cenário 2
Entrada	Os dados do processo foram todos inseridos corretamente.	Algum dos dados do processo foi inserido incorretamente e a validação interrompeu a abertura do processo.
Saída	Exibe mensagem de concluído.	Exibe mensagem de erro e suas razões.

7.9. Caso de teste: Consulta de processo

Objetivo do Teste	Consulta de um processo existente.	
Cenários	Cenário 1	Cenário 2
Entrada	Procura um processo existente.	Procura um processo mas ele não existe.
Saída	Exibe tela de informações do processo.	Exibe mensagem de erro e suas razões.

7.10. Caso de teste: Triagem inicial

Objetivo do Teste	Coletar e cadastrar os dados dos assistidos.	
Cenários	Cenário 1	Cenário 2
Entrada	Assistido fornece todos os dados necessários.	Assistido não fornece todos os dados necessários.
Saída	Exibe uma mensagem de concluído e passa para próxima etapa.	Exibe mensagem de erro e suas razões.

7.11. Caso de teste: Cadastro de documentos

Objetivo do Teste	Cadastrar documentos	
Cenários	Cenário 1	Cenário 2
Entrada	Documentos corretos que faltaram na hora do cadastro.	Procura um processo mas ele não existe
Saída	Exibe mensagem de concluído.	Exibe mensagem de erro e suas razões.

8. RELATÓRIO DA IMPLEMENTAÇÃO/DEPURAÇÃO

O desenvolvimento do sistema foi realizado através de projetos de extensão de curso e dentro de disciplinas do curso de Ciência da Computação da UNICENTRO. Cada projeto menor ficou encarregado de desenvolver uma parte do sistema aqui descrito.

Para acompanhamento do desenvolvimento do sistema, foram utilizadas tanto as aulas das disciplinas quanto reuniões específicas para o sistema.

Durante o desenvolvimento do sistema, observou-se a dificuldade dos alunos em compreender com exatidão a necessidade do sistema, sendo esse problema resolvido com reuniões em conjunto com os professores e também a representante da DPPR.

Outra dificuldade encontrada foi a inexperiência do corpo discente com as ferramentas que foram utilizadas no desenvolvimento do sistema. Essa dificuldade foi solucionada com cursos sobre as ferramentas utilizadas, ministrados pelos alunos mais experientes e supervisionados pelos professores.

Além disso, a necessidade de segurança e de mecanismos que limitam o acesso às funcionalidades do sistema, demandou mais tempo de desenvolvimento que o esperado. Como contramedida para o ocorrido, optou-se por aumentar o número de alunos envolvidos diretamente no projeto para que os prazos estipulados fossem cumpridos.

Para melhor aproveitamento das competências dos alunos, os módulos do sistema foram divididos em desenvolvimento do servidor Java e desenvolvimento do cliente React. O sistema ainda não foi concluído, ficando a cargo do grupo de projeto de extensão concluir o desenvolvimento dos requisitos restantes, sendo eles:

- Registro de Atividades;
- Gerenciamento de atendimentos;
- Geração de relatórios.

9. RELATÓRIO DA IMPLANTAÇÃO

O processo de implantação será descrito posteriormente, visto que o sistema ainda está em fase de desenvolvimento. A implantação do sistema ficará designado para o Projeto de Extensão de Curso.

10. RELATÓRIO DA VALIDAÇÃO

Como o sistema está em desenvolvimento, a validação do sistema foi adiada para após a conclusão do sistema.

11. CONCLUSÕES

A DPPR enfrentava dificuldades com o gerenciamento dos processos assistenciais do órgão. Para solucionar esse problema elaborou-se e desenvolveu-se o projeto descrito no presente relatório. O projeto foi desenvolvido por alunos e professores da UNICENTRO, através de projetos de extensão e disciplinas do curso de Ciência da Computação.

Foram levantados os requisitos do sistema e deles foram gerados diagramas de casos de uso, de sequência e de classe, para documentar o projeto e facilitar o entendimento de outros desenvolvedores a dar continuidade ao projeto. O desenvolvimento foi realizado por diversos grupos, separados pelas disciplinas do curso de Ciência da Computação.

Foram desenvolvidos, até agora:

- o modelo do banco de dados, dentro do projeto de extensão de curso;
- controle de acesso (login e permissões), dentro da disciplina de Projeto e Desenvolvimento de Software;
- o cadastro de funcionários, dentro da disciplina de Engenharia de Software 2; e
- a triagem inicial, com gerenciamento de assistido, de processo e de documentos, dentro da disciplina de Projeto e Desenvolvimento de Software;

Restam ainda o desenvolvimento da geração de relatórios, registro de atividades, cadastro dos atendimentos e implantação do sistema, que serão desenvolvidos nos anos subsequentes a este.

12. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Sommerville, I. Engenharia de Software, 9ª Ed. Pearson, 2011.
- [2] Guedes, G. T. A. UML 2 - Uma Abordagem Prática, 2ª Ed. Novatec, 2011.
- [3] Bauer, C. e King, G. Hibernate in action. Manning Publications. 2005.
- [4] Politze, M., Schaffert, S., e Decker, B. A secure infrastructure for mobile blended learning applications. J. Bergström, European Journal of Higher Education IT 2016-1. Umeå
- [5] Malvoni, K., Solar Designer, e Knezovic, J. Are your passwords safe: Energy-efficient bcrypt cracking with low-cost parallel hardware. USENIX Workshop on Offensive Technologies. 2014.
- [6] SWEBOK. Guide to the Software Engineering Body of Knowledge. Version 3.0. 2014.
- [7] Briand, L. e Labiche, Y. A uml-based approach to system testing. In: UML'01: Proceedings of the 4th International Conference on The Unified Modeling Language, Modeling Languages, Concepts, and Tools. Springer-Verlag, London, UK, pp. 194–208. 2001.