DOCUMENTO DE REQUISITOS DO SISTEMA CR/NOVACAP

1. Introdução

O presente documento descreve os requisitos funcionais e não funcionais do sistema Central de Relacionamento da NOVACAP (CR/NOVACAP). O sistema foi desenvolvido com o objetivo de organizar, controlar e acompanhar processos administrativos e protocolos de atendimento no âmbito da NOVACAP, atendendo tanto às Regiões Administrativas quanto às demandas internas da empresa.

2. Escopo do Sistema

O sistema CR/NOVACAP é composto por dois módulos principais:

Módulo de Controle de Processos: Responsável pelo cadastro, tramitação, visualização, alteração e análise de processos administrativos enviados à NOVACAP pelas Regiões Administrativas.

Módulo de Protocolo de Atendimento: Responsável por registrar, consultar, responder e gerenciar os atendimentos prestados pela Central de Relacionamento a cidadãos, servidores ou administrações regionais.

3. Requisitos Funcionais

3.1. Cadastro e Gerenciamento de Processos

RF01: O sistema deve permitir o cadastro de novos processos com número, status inicial, diretoria de destino e observações.

RF02: O sistema deve associar cada processo à sua entrada (RA de origem, data de entrada, tipo, demanda, usuário responsável).

RF03: O sistema deve registrar todas as movimentações associadas à entrada de um processo (status, data, responsável, observação).

RF04: O sistema deve permitir alterar processos e registrar novas tramitações.

RF05: O sistema deve permitir a visualização completa do processo, suas entradas e histórico de movimentações.

3.2. Cadastro e Gerenciamento de Protocolos de Atendimento

RF06: O sistema deve permitir o registro de protocolos de atendimento com dados do solicitante, contato, assunto, RA, demanda e encaminhamento inicial.

RF07: O sistema deve permitir o registro de interações (respostas) associadas a cada atendimento.

RF08: O sistema deve permitir a consulta por número de protocolo, data, RA ou tipo de solicitante.

3.3. Gestão de Usuários

RF09: O sistema deve permitir o cadastro, aprovação e bloqueio de usuários por administradores.

RF10: O sistema deve permitir o login com autenticação segura.

RF11: O sistema deve registrar logs de ações realizadas pelos usuários.

3.4. BI e Relatórios

RF12: O sistema deve gerar painéis de BI com gráficos interativos (status, RA, diretoria, período).

RF13: O sistema deve permitir exportar relatórios em CSV e Excel a partir de filtros (status, RA, responsável, período).

RF14: O sistema deve permitir visualizar indicadores como tempo médio de atendimento, processos por RA, e tramitações por status.

4. Diagrama de Casos de Uso

O Diagrama de Casos de Uso apresentado abaixo ilustra os principais atores do sistema CR/NOVACAP e suas respectivas interações com os módulos funcionais. Este diagrama é uma representação visual das funcionalidades descritas nos requisitos anteriores, organizadas por módulo e perfil de acesso.

Atores identificados:

Administrador: possui acesso completo ao sistema, incluindo aprovação e bloqueio de usuários, atribuição de permissões, e acesso a relatórios e dashboards.

Usuário da CR: responsável por cadastrar processos, registrar tramitações, atender solicitações via protocolo e gerar relatórios.

Servidor da RA: inicia processos administrativos e consulta status.

Cidadão/Servidor Solicitante: interage com a Central por meio de protocolo de atendimento, podendo registrar demandas e acompanhar interações.

Casos de uso principais:

Cadastro, tramitação e análise de processos;

Registro e resposta de protocolos de atendimento;

Geração de relatórios gerenciais e painéis de BI;

Gestão de permissões e administração de usuários.

O diagrama será incluído em anexo como imagem ilustrativa e também pode ser visualizado de forma independente para apresentação ou análise técnica.

5. Diagrama de Classes

O Diagrama de Classes do sistema CR/NOVACAP descreve a estrutura lógica das entidades implementadas no sistema, seus atributos, e os relacionamentos entre elas. Ele representa a base de dados e as regras de negócio utilizadas no controle de processos e atendimento aos usuários.

Classes principais:

Usuário: representa os usuários autenticados no sistema, incluindo administradores. Possui atributos de controle como nome, login, senha criptografada, e flags de aprovação e privilégio.

Processo e Entrada de Processo: um Processo é o elemento principal que contém informações gerais como número, status e diretoria de destino. A EntradaProcesso representa o registro da entrada deste processo, com datas, RA de origem, demanda, tipo e responsável pelo cadastro.

Movimentação: representa cada etapa de tramitação de uma entrada de processo, com status, data, usuário executor e observação.

ProtocoloAtendimento e InteracaoAtendimento: o ProtocoloAtendimento representa um atendimento prestado pela CR. A InteracaoAtendimento representa cada resposta ou encaminhamento interno vinculado a esse protocolo.

Relacionamentos destacados:

Um Processo está ligado a uma EntradaProcesso.

Uma EntradaProcesso pode possuir diversas Movimentacoes.

Um Usuario está associado ao cadastro de entradas, movimentações e interações.

Cada ProtocoloAtendimento pode conter múltiplas InteracaoAtendimento.

Esse diagrama está representado graficamente em anexo para fins de apresentação visual e análise técnica.

6. Diagrama de Componentes

O Diagrama de Componentes do sistema CR/NOVACAP apresenta a estrutura de alto nível da aplicação web desenvolvida em Flask. Ele representa os principais blocos funcionais, as tecnologias empregadas e como esses componentes se integram entre si.

Principais componentes representados:

Usuário e Interface Web: A interação inicial ocorre através de páginas HTML/CSS renderizadas pelo Flask, com formulários para entrada de dados, filtros e navegação.

Backend em Flask: Centraliza o controle da aplicação. Realiza o roteamento (Blueprints), autenticação (Flask-Login), lógica de negócios e comunicação com o banco de dados por meio do ORM SQLAlchemy.

SQLAlchemy ORM: Intermediário entre o código Python e o banco de dados relacional MySQL. Permite persistência e recuperação segura de dados estruturados.

Banco de Dados MySQL: Contém todas as tabelas principais (usuários, processos, movimentações, atendimentos) e a VIEW vw\_processos\_completos, usada nos dashboards analíticos.

Painel de BI (Dash + Plotly): Subaplicação incorporada ao sistema para visualização de dados com filtros dinâmicos, gráficos interativos e indicadores gerenciais.

Scripts Administrativos: Utilizados para criação da estrutura inicial (criar\_tabelas.py), geração da view (criar\_view.py), atribuição de permissões, verificação e desbloqueio de usuários.

Esses componentes estão organizados de forma modular, permitindo a manutenção isolada de cada parte da aplicação (frontend, backend, dados e análises).

7. Diagrama Entidade-Relacionamento (DER)

O Diagrama Entidade-Relacionamento do sistema CR/NOVACAP representa a estrutura relacional do banco de dados implementado em MySQL e mapeado por meio da biblioteca SQLAlchemy no backend Flask. O diagrama identifica as entidades centrais do sistema, seus atributos, chaves primárias e estrangeiras, e os principais relacionamentos entre tabelas.

Entidades principais:

Usuario: armazena as credenciais e configurações de acesso dos usuários do sistema, incluindo atributos como nome, e-mail, login, senha criptografada, status de aprovação e privilégios de administração.

Processo: representa os processos administrativos recebidos pela NOVACAP. Contém dados como número do processo, status atual, observações e a diretoria de destino.

EntradaProcesso: tabela intermediária que associa um processo à sua origem (RA), tipo, demanda, data de criação e usuário responsável. Serve como ponto de partida para as movimentações.

Movimentacao: registra o histórico de tramitações realizadas em uma entrada de processo. Cada movimentação está associada a um usuário, a um status e a uma data específica.

ProtocoloAtendimento: representa atendimentos realizados pela Central de Relacionamento, com dados do solicitante, tipo de demanda, RA de origem e encaminhamento inicial.

InteracaoAtendimento: representa as respostas ou atualizações relacionadas a um atendimento, com registro de data, conteúdo e responsável técnico.

Relacionamentos:

Um Processo possui uma ou mais EntradasProcesso.

Uma EntradaProcesso pode gerar várias Movimentacoes.

Cada Usuario pode estar relacionado ao cadastro de entradas, movimentações e interações.

Um ProtocoloAtendimento pode conter várias InteracaoAtendimento.

Esse modelo garante a integridade relacional, rastreabilidade de ações e permite a geração de relatórios confiáveis e análises estatísticas a partir da view consolidada vw\_processos\_completos.

8. Modelo Entidade-Relacionamento (MER)

O Modelo Entidade-Relacionamento (MER) do sistema CR/NOVACAP representa, em nível conceitual, a organização dos dados e entidades envolvidas nos dois módulos principais do sistema: Controle de Processos e Protocolo de Atendimento.

Esse modelo foi desenvolvido utilizando o MySQL Workbench e contempla entidades fundamentais como Usuário, Processo, Entrada de Processo, Movimentação, Protocolo de Atendimento e Interação de Atendimento, além de tabelas auxiliares como Demandas, Tipos de Demanda, Status e Regiões Administrativas.

No MER, cada entidade possui seus atributos essenciais, e os relacionamentos foram definidos conforme as regras de negócio identificadas na análise funcional do sistema. As cardinalidades e dependências foram modeladas de forma a garantir:

Normalização dos dados (evitando redundâncias)

Integridade referencial entre as tabelas

Flexibilidade para geração de relatórios e visualizações interativas

Este modelo serviu como base lógica para a posterior implementação do DER (Diagrama Entidade-Relacionamento) físico e para a definição do banco de dados operacional do sistema.

9. Requisitos Não Funcionais

RNF01: O sistema deve ser acessado via navegador, utilizando padrões responsivos (HTML5, CSS3, JS).

RNF02: O sistema deve utilizar banco de dados MySQL, acessado via SQLAlchemy (ORM).

RNF03: A autenticação deve ser realizada com Flask-Login e senha criptografada.

RNF04: Os dados devem ser persistidos com segurança e integridade, com chaves estrangeiras e controle transacional.

RNF05: O sistema deve exportar dados em formatos interoperáveis (CSV, XLSX).

RNF06: Os gráficos devem ser gerados com Plotly e visualizados com Dash e/ou Plotly.js.

RNF07: As operações de criação de estrutura devem ser feitas via scripts Python (criar\_tabelas.py, criar\_view.py).

10. Regras de Negócio

R01: Um processo pode possuir apenas uma entrada, mas uma entrada pode possuir várias movimentações.

R02: Um atendimento pode ter várias interações, mas apenas um criador.

R03: Os processos de "Indivíduo Arbóreo" devem ser considerados improcedentes e devolvidos automaticamente à RA.

R04: Apenas usuários administradores podem aprovar, bloquear ou atribuir permissões a outros usuários.

R05: Um status que finaliza um processo deve ser marcado com a flag finaliza\_processo=True.

11. Restrições

O sistema não deve permitir o cadastro de processos com número repetido.

Os campos obrigatórios devem ser validados tanto no frontend quanto no backend.

Os relatórios devem considerar apenas os dados realmente registrados (sem campos nulos para data ou status).

12. Considerações Finais

Este documento apresenta de forma consolidada a estrutura lógica, técnica e funcional do sistema CR/NOVACAP. Ele reúne os requisitos, diagramas UML, modelagem de dados e visão geral dos componentes internos e externos do sistema. As informações aqui contidas servirão como base para desenvolvimento, manutenção, documentação oficial e apresentação institucional do projeto.