

|  |
| --- |
| **Sistema IA – Portal Web**  Documento de Requisitos  (Tela de Login) |
|  |
| Versão 1.0 |

|  |
| --- |
| Clovis Ap. Moratta  10/9/2025 |

|  |  |  |  |
| --- | --- | --- | --- |
| **HISTÓRICO DE REVISÕES** | | | |
| Data | Versão | Descrição | Autor |
| 10/09/2025 | 1.0 | Criação do documento de requisitos Tela de Login | Clovis Ap. Moratta |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**SUMÁRIO**

[1) Informações do Projeto 3](#_Toc208397026)

[2) Observações para IA Cursor 3](#_Toc208397027)

[3) Arquitetura DDD + API REST 3](#_Toc208397028)

[4) Estrutura de Banco de Dados 5](#_Toc208397029)

[5) Requisitos Funcionais (RF) 5](#_Toc208397030)

[6) Requisitos Não Funcionais (RNF) 6](#_Toc208397031)

[7) Estrutura da Tela 6](#_Toc208397032)

[8) Integração com APIs Externas 7](#_Toc208397033)

[9) API REST Interna 9](#_Toc208397034)

[10) Segurança 11](#_Toc208397035)

[11) Referências 11](#_Toc208397036)

# 1) Informações do Projeto

* **Projeto:** IA - Portal Web - Protótipo
* **Módulo:** Tela de Login
* **Descrição:** Tela de autenticação de usuários.
* **Objetivo:** Validar login consumindo API Odoo, registrar acessos e logs via API interna (REST com PostgreSQL), e permitir recuperação/redefinição de senha.

# 2) Observações para IA Cursor

O desenvolvimento deve seguir as regras descritas nos arquivos MD:

* **A-react\_guide.md** → Estrutura, componentes e padrões para o front-end em React.
* **A-guia\_fastapi.md** → Estrutura, organização, regras de DDD, endpoints, models, validações e padrões para API REST em FastAPI.
* **NomenclatureGuide.md** → Padrões de nomenclatura (arquivos, variáveis, funções, classes, schemas, constantes).

Todo front-end deve referenciar diretamente o **A-react\_guide.md**.  
Todo back-end deve referenciar diretamente o **A-guia\_fastapi.md**.  
Padrões de nomenclatura devem seguir o **NomenclatureGuide.md**.

# 3) Arquitetura DDD + API REST

**3.1 Bounded Context**

**Autenticação & Registro de Acesso**

**3.2** **Aggregates**:

* Usuario (root) → Repositório UsuarioRepo
* Acesso (apenas leitura/auditoria) → AcessoRepo
* LogAcesso (auditoria) → LogAcessoRepo

3.3 **Domain Services**: AutenticacaoService (encapsula chamada ao Odoo e regras), RecuperacaoSenhaService.

3.4 **Domain Events**: LoginSucceeded, LoginFailed, PasswordResetRequested, PasswordChanged.

3.5 **Application Layer (Use Cases)**: AutenticarUsuarioHandler, RegistrarAcessoHandler, SolicitarResetSenhaHandler, ConfirmarResetSenhaHandler.

3.6 **Infra**: Adaptadores para Odoo (JSON-RPC), Postgres (ORM), Email (se aplicável).

**3.7 Entidades**

* **Acesso** (usuario\_id, data\_hora, ip, device, navegador, session\_id)
* **LogAcesso** (usuario\_id, timestamp, status, motivo, ip, user\_agent)
* **RecuperacaoSenha** (usuario\_id, token\_hash, data\_expiracao, usado)

**3.8 Casos de Uso**

| Código | Nome do Caso de Uso | Ator Principal | Fluxo Principal | Fluxos Alternativos / Mensagens |
| --- | --- | --- | --- | --- |
| UC01 | Autenticar Usuário | Usuário | 1. Usuário insere usuário e senha → 2. API REST consome API Odoo → 3. Se sucesso, retorna dados do usuário | - Credenciais inválidas → "Usuário ou senha inválidos"  - Falha na API Odoo → "Erro de comunicação com servidor Odoo" |
| UC02 | Registrar log de acesso | Sistema | 1. Após login, registrar log de entrada no banco via API REST | - Falha de persistência → "Erro ao registrar log de acesso" |
| UC03 | Registrar acesso do usuário | Sistema | 1. Após login, registrar data e hora do acesso no banco via API REST | - Falha de persistência → "Erro ao registrar acesso do usuário" |
| UC04 | Redirecionar usuário | Sistema | 1. Login validado → 2. Redireciona para tela principal | Nenhum, fluxo direto |
| UC05 | Recuperar senha | Usuário | 1. Usuário clica “Esqueci minha senha” → 2. Modal solicita e-mail → 3. Envia para API Odoo → 4. Recebe link de redefinição | - E-mail inválido → "E-mail inválido"  - E-mail não cadastrado → "Por favor, insira seu e-mail cadastrado"  - Falha no envio → "Erro ao enviar link de redefinição de senha" |
| UC06 | Alterar senha | Usuário | 1. Usuário abre link recebido → 2. Página solicita nova senha e confirmação → 3. Envia JSON para API Odoo | - Senhas não conferem → "Senhas não conferem"  - Token inválido/expirado → "Token inválido ou expirado, solicite novo link"  - Token já utilizado → "Este link já foi utilizado, solicite um novo link"  - Sucesso → "Senha alterada com sucesso", redirecionar para login |

**3.9 Regras de Negócio**

* Senhas devem ser seguras e criptografadas.
* Token de redefinição expira em **1h** e só pode ser usado uma vez.
* Logs de acesso devem ser persistidos sempre (sucesso ou falha).
* E-mails devem ser únicos.

# 4) Estrutura de Banco de Dados

* **odoo\_acessos**: id (PK), usuario\_id, data\_hora, ip, device, navegador, session\_id.
* **odoo\_log\_acesso**: id (PK), usuario\_id, timestamp, status, motivo, ip, user\_agent.
* **odoo\_recuperacao\_senha**: id (PK), usuario\_id, token\_hash (UNIQUE), data\_expiracao, usado.

**Restrições:**

* NOT NULL em todos os campos obrigatórios.
* Campos de data com CURRENT\_TIMESTAMP.
* Token único e rastreável.

**4.1 Conexão com o Banco de Dados**

* **SGBD**: PostgreSQL
* **ORM**: SQLAlchemy (modo assíncrono via async SQLAlchemy + alembic para migrações)
* **Configuração**:
  + Conexão deve ser parametrizada por variáveis de ambiente:
    - DB\_HOST → 51.8.188.218
    - DB\_PORT → 5432
    - DB\_NAME → THEGED\_HOMOLOG
    - DB\_USER → thegedapp\_homolog
    - DB\_PASSWORD → RGM@@2025@@
    - DB\_SCHEMA → public
* **Regras adicionais**:
  + Pool mínimo: 1 conexão
  + Pool máximo: 10 conexões
  + Timeout de conexão: 30s
  + Conexões devem ser fechadas corretamente após uso (boa prática para evitar vazamento).

# 5) Requisitos Funcionais (RF)

* RF01 Validar usuário e senha via API do Odoo.
* RF02 Registrar log de acesso.
* RF03 Registrar acesso do usuário.
* RF04 Redirecionar usuário autenticado.
* RF05 Retornar mensagens de erro claras.
* RF06 Enviar link de redefinição de senha via Odoo.
* RF07 Redefinir senha do usuário via link.

# 6) Requisitos Não Funcionais (RNF)

* RNF01 Tempo de resposta do login < 3s.
* RNF02 Comunicação segura (HTTPS, criptografia de senha).
* RNF03 Logs persistidos para auditoria.
* RNF04 Mensagens de erro padronizadas.
* RNF05 Código front-end segue A-react\_guide.md.
* RNF06 Código back-end segue A-guia\_fastapi.md.
* RNF07 Token expira em 1h, válido uma vez.
* RNF08 Nomenclaturas conforme NomenclatureGuide.md.
* RNF09 Mensagens unificadas para front-end.

# 7) Estrutura da Tela

**Tela de Login**

* Campos: usuário, senha.
* Botões: **Entrar** (autentica via Odoo + registra acesso), “**Esqueci minha senha”** (abre modal).
* Mensagens: “Usuário ou senha inválidos”, “Erro de comunicação com Odoo”, “Login realizado com sucesso”.

**Modal Esqueci Minha Senha**

* Campo: e-mail (obrigatório).
* Fluxo: valida e-mail → consome Odoo → envia link de redefinição.
* Mensagens: “E-mail inválido”, “Por favor, insira seu e-mail cadastrado”, “Link enviado com sucesso”.

**Página de Redefinição de Senha**

* Campos: nova senha, confirmar senha, alterar senha.
* Fluxo: valida token → envia senha para Odoo → retorna mensagem.
* Mensagens: “Senhas não conferem”, “Token inválido ou expirado”, “Senha alterada com sucesso”.

**Tela Principal**

* Mensagem de boas-vindas: “Bem-vindo, {nome}”.
* Menu principal.
  + Segurança
    - Submenus
      * Usuários → cadastro e gerenciamento de usuários
      * Cadastro de Grupo → criação e gerenciamento de grupos de usuários
      * Grupo x Usuários → associação de usuários a grupos
      * Permissões → cadastro e gerenciamento de permissões
      * Logs de Acesso → consulta de logs de login e acessos
  + Chamados
    - Submenus
      * Cadastro de Chamados → inclusão, alteração e exclusão de chamados via API externa

# 8) Integração com APIs Externas

**8.1 API Odoo Login**

* **Endpoint:** ${ODOO\_API\_URL} = https://edu-rgm-ia-base-teste.odoo.com/jsonrpc
* **Variáveis:**
  + ODOO\_API\_URL = https://edu-rgm-ia-base-teste.odoo.com/jsonrpc
  + ODOO\_DB\_NAME = edu-rgm-ia-base-teste
  + ODOO\_UID = 6
  + ODOO\_API\_KEY = 3771bef71f270b7430a5b60d4129915b6d711aaa
* **Request (JSON):**

{

"jsonrpc": "2.0",

"method": "call",

"params": {

"service": "object",

"method": "execute\_kw",

"args": [

"${ODOO\_DB\_NAME}",

${ODOO\_UID},

"${ODOO\_API\_KEY}",

"hr.employee",

"search\_read",

[["work\_email", "=", "<email\_do\_usuario>"]],

{"fields": ["id","work\_email","name","message\_partner\_ids","x\_studio\_senha"]}

]

},

"id": 2

}

* **Resposta:** se retorna registro → sucesso; se vazio → “Usuário ou senha inválidos”; se erro → “Erro de comunicação com Odoo”.

**8.2 Envio de E-mail no Odoo**

* **Request (JSON):**

{

"jsonrpc": "2.0",

"method": "call",

"params": {

"service": "object",

"method": "execute\_kw",

"args": [

"${ODOO\_DB\_NAME}",

${ODOO\_UID},

"${ODOO\_API\_KEY}",

"mail.mail",

"create",

[{

"subject": "Teste de E-mail via API",

"body\_html": "{{emailBody}}", //html

"email\_to": "francisco.barros@rgm.com.br",

"email\_from": "customer-care@edu-rgm-ia-base-teste.odoo.com"

}]

]

},

"id": 2

}

**8.3 Alteração de Senha no Odoo**

* **Request (JSON):**

{

"jsonrpc": "2.0",

"method": "call",

"params": {

"service": "object",

"method": "execute\_kw",

"args": [

"${ODOO\_DB\_NAME}",

${ODOO\_UID},

"${ODOO\_API\_KEY}",

"hr.employee",

"write",

[[<id\_funcionario>], { "x\_studio\_senha": "<nova\_senha>" }]

]

},

"id": 2

}

# 9) API REST Interna

**9.1 POST /api/log\_acesso/login\_attempt**

* **Request:**

{ "usuario\_id": 1, "status": "FALHA", "motivo": "Senha incorreta", "ip": "192.168.1.10", "user\_agent": "Mozilla/5.0" }

* **Persistência:** Deve armazenar o registro na tabela odoo\_log\_acesso, conforme estrutura definida no item 4.

**9.2 POST /api/acessos**

* **Request:**

{ "usuario\_id": 1, "ip": "192.168.1.10", "device": "Desktop", "navegador": "Chrome", "session\_id": "abcd1234" }

* **Persistência:** Deve armazenar o registro na tabela odoo\_acessos, conforme estrutura definida no item 4.

**9.3 POST /api/recuperacao\_senha**

* **Request (JSON)**:

{ "usuario\_id": 1, "token\_hash": "abc123xyz", "data\_expiracao": "2025-09-09T20:00:00", "usado": false}

* **Persistência:** Deve armazenar o registro na tabela odoo\_recuperacao\_senha, conforme estrutura definida no item 4.

**9.4 GET /api/log\_acesso**

* **Parâmetros de consulta (query params)**:
  + usuario\_id (opcional, int) → filtra por usuário.
  + status (opcional, string: SUCESSO ou FALHA).
  + data\_inicio (opcional, string no formato YYYY-MM-DD).
  + data\_fim (opcional, string no formato YYYY-MM-DD).
  + ip (opcional, string).
* **Exemplo de Request**:
  + GET /api/log\_acesso?usuario\_id=1&status=FALHA&data\_inicio=2025-09-01&data\_fim=2025-09-09
* **Exemplo de Response:**

{

"status": 200,

"logs": [

{

"id": 124,

"usuario\_id": 1,

"login\_odoo": "user1",

"status": "FALHA",

"motivo": "Senha incorreta",

"ip": "192.168.1.10",

"user\_agent": "Mozilla/5.0",

"timestamp": "2025-09-05T08:35:00"

}

]

}

# 10) Segurança

* **HTTPS obrigatório**.
* **Senhas e tokens criptografados**.
* **Logs imutáveis para auditoria**.
* **Tokens expirados em 1h**.

# 11) Referências

* **A-react\_guide.md** — guia front-end React.
* **A-guia\_fastapi.md** — guia back-end FastAPI + DDD.
* **NomenclatureGuide.md** — padrões de nomenclatura.