



CarrisPlus

Sistema de Gestão de Documentos Online

Documentação Técnica Completa

Equipa de Desenvolvimento:

Davi (2024301) • Iago (2024195) • Ana (2024184)

Projeto Académico - Engenharia Informática

Ano Letivo 2024/2025

CarrisPlus - Documentação Técnica do Projeto

Technical Project Documentation

Projeto Académico - Engenharia Informática ---

Informação do Projeto / Project Information

Nome do Projeto: CarrisPlus - Sistema de Gestão de Documentos Online **Instituição:** [Nome da Instituição] **Curso:** Engenharia Informática **Ano Letivo:** 2024/2025

Equipa de Desenvolvimento / Development Team

Nome Número de Estudante Papel	----- ----- -----	Davi
2024301 Desenvolvedor Full-Stack Iago 2024195 Desenvolvedor Backend	----- -----	
Ana 2024184 Desenvolvedora Frontend ---		

1. RESUMO EXECUTIVO / EXECUTIVE SUMMARY

1.1 Visão Geral do Projeto

O CarrisPlus é uma plataforma web inovadora desenvolvida para modernizar e digitalizar o processo de submissão de documentos e gestão de passes da Carris, a empresa municipal de transportes públicos de Lisboa. O sistema visa eliminar a necessidade de deslocações físicas aos balcões de atendimento, permitindo que os cidadãos realizem todo o processo de forma online, com validação automática através de Inteligência Artificial. CarrisPlus is an innovative web platform developed

to modernize and digitize the document submission and pass management process for Carris, Lisbon's municipal public transport company. The system aims to eliminate the need for physical travel to service counters, allowing citizens to complete the entire process online with automatic validation through Artificial Intelligence.

1.2 Contexto e Motivação

Problema Identificado

Atualmente, os cidadãos que necessitam de criar ou renovar passes da Carris enfrentam diversos desafios:

- Deslocação Obrigatória:** Necessidade de se deslocar fisicamente aos postos de atendimento
- Tempo de Espera:** Longas filas para tarefas administrativas simples
- Horário Limitado:** Dependência dos horários de funcionamento dos balcões
- Constrangimentos:** Dificuldades para pessoas com mobilidade reduzida ou falta de tempo
- Processamento Manual:** Validação manual de documentos é lenta e propensa a erros

Impacto Social

O problema afeta especialmente:

- Idosos com mobilidade reduzida
- Trabalhadores com horários rígidos
- Estudantes com horários escolares
- Pessoas em áreas remotas de Lisboa
- População em geral durante períodos de pandemia ou emergência

1.3 Solução Proposta

O CarrisPlus oferece uma solução digital completa que permite:

- Submissão Online de Documentos:** Upload digital de documentos necessários (CC, Passaporte, Atestados, etc.)
- Criação de Passes:** Pedido de novos passes sem sair de casa
- Renovação Automática:** Sistema de renovação de passes existentes
- Validação Inteligente:** Verificação automática de dados usando IA e OCR
- Acompanhamento em Tempo Real:** Status do pedido disponível 24/7

Diferencial Tecnológico

- **OCR Avançado:** Extração automática de dados dos documentos - **Validação de NIF:** Verificação cruzada com base de dados - **Matching Inteligente:** Comparação de nomes e dados pessoais - **Deteção de Fraude:** Identificação de documentos suspeitos - **Interface Acessível:** Design responsivo e intuitivo

1.4 Objetivos do Projeto

Objetivos Primários

1. Reduzir o tempo médio de processamento de pedidos em 70%
2. Eliminar 90% das deslocações físicas desnecessárias
3. Disponibilizar o serviço 24 horas por dia, 7 dias por semana
4. Processar documentos com precisão de 95%+ através de IA

Objetivos Secundários

1. Melhorar a experiência do utilizador
2. Reduzir custos operacionais da Carris
3. Criar um sistema escalável e seguro
4. Estabelecer base para futuras integrações com outros serviços municipais ---

2. REQUISITOS FUNCIONAIS / FUNCTIONAL REQUIREMENTS

RF-001: Gestão de Utilizadores

RF-001.1 - Registo de Utilizador

Descrição: O sistema deve permitir que novos utilizadores se registem na plataforma.

Prioridade: Alta **Entradas:** - Nome completo - Email - NIF (Número de Identificação Fiscal) - Palavra-passe - Confirmação de palavra-passe - Número de telefone

(opcional) Processamento: 1. Validar formato do email 2. Verificar se email já existe 3. Validar formato do NIF (9 dígitos) 4. Verificar força da palavra-passe (mínimo 8 caracteres, maiúsculas, minúsculas, números) 5. Criar hash seguro da palavra-passe

(bcrypt) 6. Enviar email de confirmação **Saídas:** - Conta criada com sucesso - Email de confirmação enviado - Mensagem de erro se dados inválidos **Critérios de Aceitação:** - ✓ Utilizador consegue criar conta com dados válidos - ✓ Sistema rejeita emails duplicados - ✓ NIF é validado segundo algoritmo português - ✓ Palavra-passe é armazenada com hash seguro - ✓ Email de confirmação é enviado em menos de 30 segundos **Status Atual:** 🚧 Não Implementado ---

RF-001.2 - Login de Utilizador

Descrição: O sistema deve permitir que utilizadores registados façam login.

Prioridade: Alta **Entradas:** - Email - Palavra-passe **Processamento:** 1. Verificar se email existe 2. Comparar hash da palavra-passe 3. Verificar se conta está ativa 4. Criar sessão segura 5. Registar log de acesso **Saídas:** - Sessão iniciada - Redirecionamento para dashboard - Token de autenticação (JWT) **Critérios de Aceitação:** - ✓ Login bem-sucedido com credenciais válidas - ✓ Mensagem de erro com credenciais inválidas - ✓ Proteção contra força bruta (máximo 5 tentativas) - ✓ Sessão expira após 24 horas de inatividade **Status Atual:** 🚧 Não Implementado ---

RF-001.3 - Recuperação de Palavra-passe

Descrição: O sistema deve permitir recuperação de palavra-passe via email.

Prioridade: Média **Status Atual:** 🚧 Não Implementado ---

RF-001.4 - Gestão de Perfil

Descrição: Utilizador pode visualizar e editar seus dados pessoais. **Prioridade:** Média

Status Atual: 🚧 Não Implementado ---

RF-002: Submissão de Documentos

RF-002.1 - Upload de Documento

Descrição: O sistema deve permitir upload de documentos digitalizados. **Prioridade:** Alta **Entradas:** - Ficheiro (PDF, JPG, PNG) - Tipo de documento (dropdown) - Notas adicionais (opcional) **Processamento:** 1. Validar formato do ficheiro (magic bytes) 2. Verificar tamanho máximo (5MB) 3. Gerar nome único para ficheiro 4. Armazenar em

diretório seguro 5. Criar registo na base de dados 6. Atribuir status "Pendente"

Saídas: - Documento submetido com sucesso - Número de protocolo gerado - Confirmação visual **Regras de Negócio:** - Tamanho máximo: 5MB por ficheiro - Formatos aceites: PDF, JPG, JPEG, PNG - Resolução mínima: 300 DPI (recomendado) - Máximo de 3 ficheiros por submissão **Critérios de Aceitação:** - ✓ Upload bem-sucedido com formatos válidos - ✓ Rejeição de ficheiros muito grandes - ✓ Rejeição de formatos não permitidos - ✓ Geração de número de protocolo único - ✓ Feedback visual do progresso de upload **Status Atual:** Parcialmente Implementado - Upload funcional - Validação básica de formato - Falta validação de tamanho - Falta validação de magic bytes - Falta geração de protocolo ---

RF-002.2 - Validação Automática de Documento

Descrição: O sistema deve validar automaticamente documentos submetidos usando IA. **Prioridade:** Alta **Entradas:** - Ficheiro de documento - Tipo de documento declarado - Dados do utilizador **Processamento:** 1. **Extração OCR:** - Aplicar OCR ao documento - Extrair texto estruturado - Identificar campos-chave (Nome, NIF, Data) 2. **Validação de Dados:** - Comparar nome extraído vs. nome no registo - Validar NIF extraído vs. NIF do utilizador - Verificar data de validade do documento - Validar formato e estrutura do documento 3. **Verificação de Autenticidade:** - Análise de qualidade da imagem - Deteção de adulterações - Verificação de elementos de segurança (se aplicável) 4. **Scoring de Confiança:** - Calcular score de confiança (0-100%) - Determinar se aprovação automática ou revisão manual **Saídas:** - Status: Aprovado / Rejeitado / Revisão Manual - Score de confiança - Lista de discrepâncias encontradas - Dados extraídos estruturados **Regras de Negócio:** - Score > 90%: Aprovação automática - Score 70-90%: Revisão manual - Score < 70%: Rejeição automática com motivo **Critérios de Aceitação:** - ✓ OCR extrai dados com 95%+ de precisão - ✓ Validação de NIF funciona corretamente - ✓ Matching de nomes aceita variações (ex: "João Silva" = "Joao Silva") - ✓ Sistema identifica documentos vencidos - ✓ Tempo de processamento < 10 segundos **Status Atual:** Não Implementado --

RF-002.3 - Histórico de Documentos

Descrição: Utilizador pode visualizar todos os documentos submetidos. **Prioridade:** Média **Entradas:** - Utilizador autenticado **Saídas:** - Lista de documentos - Data de submissão - Status atual - Ações disponíveis (visualizar, download) **Status Atual:** 🚧 Não Implementado ---

RF-003: Gestão de Passes

RF-003.1 - Criação de Novo Passe

Descrição: O sistema deve permitir criação de novos passes da Carris. **Prioridade:** Alta **Entradas:** - Tipo de passe (Normal, Estudante, Social, Séniior) - Zona tarifária - Documentos comprovativos - Foto tipo passe (opcional) **Processamento:** 1. Validar documentos necessários para tipo de passe 2. Calcular preço baseado em zona e tipo 3. Validar condições especiais (idade para séniior, certificado para estudante) 4. Gerar passe provisório 5. Enviar para validação **Saídas:** - Passe criado (status: Pendente) - Número do passe - Data estimada de ativação - Comprovante em PDF **Regras de Negócio:** **Passe Estudante:** - Requer: Certificado de matrícula válido - Idade: 4-23 anos - Renovação anual obrigatória **Passe Séniior:** - Idade: 65+ anos - Desconto: 40% - Documento: Cartão de Cidadão **Passe Social:** - Requer: Declaração de rendimentos - Rendimento máximo: 1.5x IAS - Renovação semestral **Critérios de Aceitação:** - ✓ Criação bem-sucedida com documentos válidos - ✓ Cálculo correto de preço - ✓ Validação automática de condições - ✓ Geração de PDF com comprovante **Status Atual:** 🚧 Não Implementado ---

RF-003.2 - Renovação de Passe

Descrição: Renovação de passes existentes. **Prioridade:** Alta **Status Atual:** 🚧 Não Implementado ---

RF-003.3 - Consulta de Passe

Descrição: Visualização de detalhes do passe ativo. **Prioridade:** Média **Status Atual:** 🚧 Não Implementado ---

RF-004: Painel de Administração

RF-004.1 - Dashboard Administrativo

Descrição: Painel para administradores visualizarem estatísticas. **Prioridade:** Média
Informações Exibidas: - Documentos pendentes de revisão - Total de documentos processados (dia/semana/mês) - Taxa de aprovação automática - Passes ativos - Utilizadores registados - Gráficos de tendências **Status Atual:**  Não Implementado

RF-004.2 - Revisão Manual de Documentos

Descrição: Interface para revisão de documentos sinalizados. **Prioridade:** Alta
Funcionalidades: - Visualização de documento original - Dados extraídos pelo OCR - Score de confiança - Discrepâncias identificadas - Ações: Aprovar / Rejeitar / Solicitar novo documento - Campo para comentários **Status Atual:**  Não Implementado ---

RF-004.3 - Gestão de Utilizadores

Descrição: Administradores podem gerenciar contas de utilizadores.
Funcionalidades: - Listagem de utilizadores - Busca e filtros - Desativar/Ativar contas - Resetar palavra-passe - Ver histórico de atividades **Status Atual:**  Implementado ---

RF-005: Notificações

RF-005.1 - Notificações por Email

Descrição: Sistema envia emails automáticos para eventos importantes. **Prioridade:** Alta
Eventos de Notificação: 1. Confirmação de registo 2. Documento submetido com sucesso 3. Documento aprovado 4. Documento rejeitado (com motivo) 5. Passe ativado 6. Passe próximo do vencimento (30 dias antes) 7. Recuperação de palavra-passe
Template de Email: - Logo da Carris - Mensagem personalizada - Call-to-action (botão) - Informações de contato - Footer com links úteis **Status Atual:**  Não Implementado ---

RF-005.2 - Notificações In-App

Descrição: Notificações dentro da plataforma. **Prioridade:** Baixa **Status Atual:**  Não Implementado ---

RF-006: Relatórios e Exportação

RF-006.1 - Geração de Relatórios

Descrição: Sistema gera relatórios para administradores. **Prioridade:** Baixa **Tipos de Relatórios:** - Documentos processados por período - Taxa de aprovação vs. rejeição - Tempo médio de processamento - Passes criados por tipo - Utilizadores ativos

Formatos de Exportação: - PDF - Excel (XLSX) - CSV **Status Atual:**  Não Implementado ---

RF-007: API REST

RF-007.1 - API para Integrações

Descrição: API RESTful para integrações externas. **Prioridade:** Baixa **Endpoints Principais:**

- `POST /api/v1/documents` - Submeter documento - `GET /api/v1/documents/{id}` - Consultar status - `GET /api/v1/passes/{user_id}` - Listar passes - `POST /api/v1/passes` - Criar passe **Autenticação:** JWT Bearer Token **Status Atual:**  Não Implementado ---

3. REQUISITOS NÃO-FUNCIONAIS / NON-FUNCTIONAL REQUIREMENTS

RNF-001: Performance

RNF-001.1 - Tempo de Resposta

Descrição: O sistema deve responder rapidamente às ações do utilizador. **Prioridade:** Alta **Requisitos:** - Carregamento de página: < 2 segundos - Upload de documento: <

5 segundos (ficheiro 5MB) - Processamento OCR: < 10 segundos - Consultas à base de dados: < 500ms - API endpoints: < 1 segundo **Métricas:** - 95% das requisições devem cumprir os tempos acima - Medição com ferramentas: Google Lighthouse, Apache JMeter **Status:**  Não testado ---

RNF-001.2 - Escalabilidade

Descrição: O sistema deve suportar crescimento de utilizadores. **Requisitos:** - Suportar 1.000 utilizadores simultâneos - Processar 10.000 documentos por dia - Crescimento horizontal via containerização - Load balancing preparado **Estratégias:** - Docker Compose para múltiplas instâncias - Redis para cache - CDN para assets estáticos - Database connection pooling **Status:**  Parcialmente preparado (Docker ready) ---

RNF-001.3 - Disponibilidade

Descrição: O sistema deve estar disponível 24/7. **Requisitos:** - Uptime: 99.5% (downtime máximo: 3.65 horas/mês) - Backup automático diário - Plano de recuperação de desastres - Monitorização contínua **Status:**  Não implementado -- -

RNF-002: Segurança

RNF-002.1 - Autenticação e Autorização

Descrição: Acesso seguro e controlado ao sistema. **Requisitos:** - Autenticação via email/password - Hash de passwords com bcrypt (salt rounds: 12) - Tokens JWT com expiração (24h) - Renovação automática de tokens - Roles: User, Admin, SuperAdmin **Proteções:** - Rate limiting: máximo 5 tentativas de login - Bloqueio temporário após falhas (15 minutos) - Logout automático após inatividade (30 minutos) **Status:**  Não implementado ---

RNF-002.2 - Proteção de Dados

Descrição: Dados sensíveis devem ser protegidos. **Requisitos:** - HTTPS obrigatório (TLS 1.3) - Encriptação de dados sensíveis em repouso (AES-256) - Sanitização de

inputs - Proteção contra SQL Injection (ORM) - Proteção contra XSS - CSRF tokens em formulários - Content Security Policy headers **Compliance:** - RGPD (GDPR) - Regulamento Geral de Proteção de Dados - Política de privacidade clara - Consentimento explícito - Direito ao esquecimento **Status:** Parcial (HTTPS ready, mas sem encriptação implementada) ---

RNF-002.3 - Segurança de Ficheiros

Descrição: Upload de ficheiros deve ser seguro. **Requisitos:** - Validação de tipo via magic bytes (não apenas extensão) - Scan antivírus (ClamAV) - Armazenamento fora do webroot - Nomes de ficheiro sanitizados - Limites de tamanho enforçados - Isolamento de ficheiros por utilizador **Status:** Parcial (validação básica, sem antivírus) ---

RNF-002.4 - Auditoria e Logs

Descrição: Todas as ações importantes devem ser registadas. **Eventos Logged:** - Login/Logout - Submissão de documentos - Aprovação/Rejeição - Alterações administrativas - Acessos não autorizados - Erros do sistema **Informações Registadas:** - Timestamp - User ID - IP Address - Ação realizada - Resultado (sucesso/falha) **Retenção:** 1 ano **Status:** Não implementado ---

RNF-003: Usabilidade

RNF-003.1 - Interface Intuitiva

Descrição: Interface fácil de usar para todos os públicos. **Requisitos:** - Design limpo e minimalista - Navegação clara (máximo 3 cliques para qualquer função) - Feedback visual para todas as ações - Mensagens de erro claras e acionáveis - Ajuda contextual disponível **Acessibilidade:** - Conformidade com WCAG 2.1 Level AA - Suporte a leitores de tela - Contraste adequado (mínimo 4.5:1) - Navegação por teclado - Tamanho de fonte ajustável **Status:** Parcialmente implementado (design básico funcional) ---

RNF-003.2 - Responsividade

Descrição: Funcionar em todos os dispositivos. **Requisitos:** - Desktop (1920x1080 e superiores) - Tablet (768x1024) - Mobile (375x667 e superiores) - Touch-friendly (botões mínimo 44x44px) **Browsers Suportados:** - Chrome 90+ - Firefox 88+ - Safari 14+ - Edge 90+ **Status:** Implementado (CSS Grid responsivo) ---

RNF-003.3 - Multilíngue

Descrição: Suporte a múltiplos idiomas. **Idiomas:** - Português (PT-PT) - Principal - Inglês (EN-GB) - Secundário **Status:** Não implementado (conteúdo em PT/EN misturado) ---

RNF-004: Manutenibilidade

RNF-004.1 - Código Limpo

Descrição: Código deve ser fácil de manter. **Requisitos:** - Padrões PEP 8 (Python) - Comentários em código complexo - Docstrings em todas as funções - Nomes de variáveis descritivos - Funções pequenas (< 50 linhas) - DRY (Don't Repeat Yourself) **Status:** Parcial (código funcional mas falta documentação) ---

RNF-004.2 - Testes

Descrição: Código deve ser testado. **Requisitos:** - Cobertura de testes: mínimo 80% - Testes unitários (pytest) - Testes de integração - Testes end-to-end (Selenium) - CI/CD pipeline **Status:** Não implementado ---

RNF-004.3 - Documentação

Descrição: Projeto deve estar bem documentado. **Documentos:** - README.md completo - API documentation (Swagger) - Manual do utilizador - Manual do administrador - Diagramas de arquitetura **Status:** Parcial (README completo, resto em falta) ---

RNF-005: Compatibilidade

RNF-005.1 - Containerização

Descrição: Aplicação deve rodar em containers. **Requisitos:** - Docker image otimizada (< 500MB) - Docker Compose para orquestração - Variáveis de ambiente configuráveis - Volumes para persistência **Status:** Implementado ---

RNF-005.2 - Base de Dados

Descrição: Suporte a diferentes ambientes. **Requisitos:** - PostgreSQL para produção - SQLite para desenvolvimento - Migrations com Alembic - Seed data para testes **Status:** Preparado mas não implementado ---

RNF-006: Legal e Compliance

RNF-006.1 - RGPD/GDPR

Descrição: Conformidade com proteção de dados. **Requisitos:** - Consentimento explícito para recolha de dados - Política de privacidade visível - Direito ao esquecimento - Portabilidade de dados - Notificação de violações (72h) **Status:** Não implementado ---

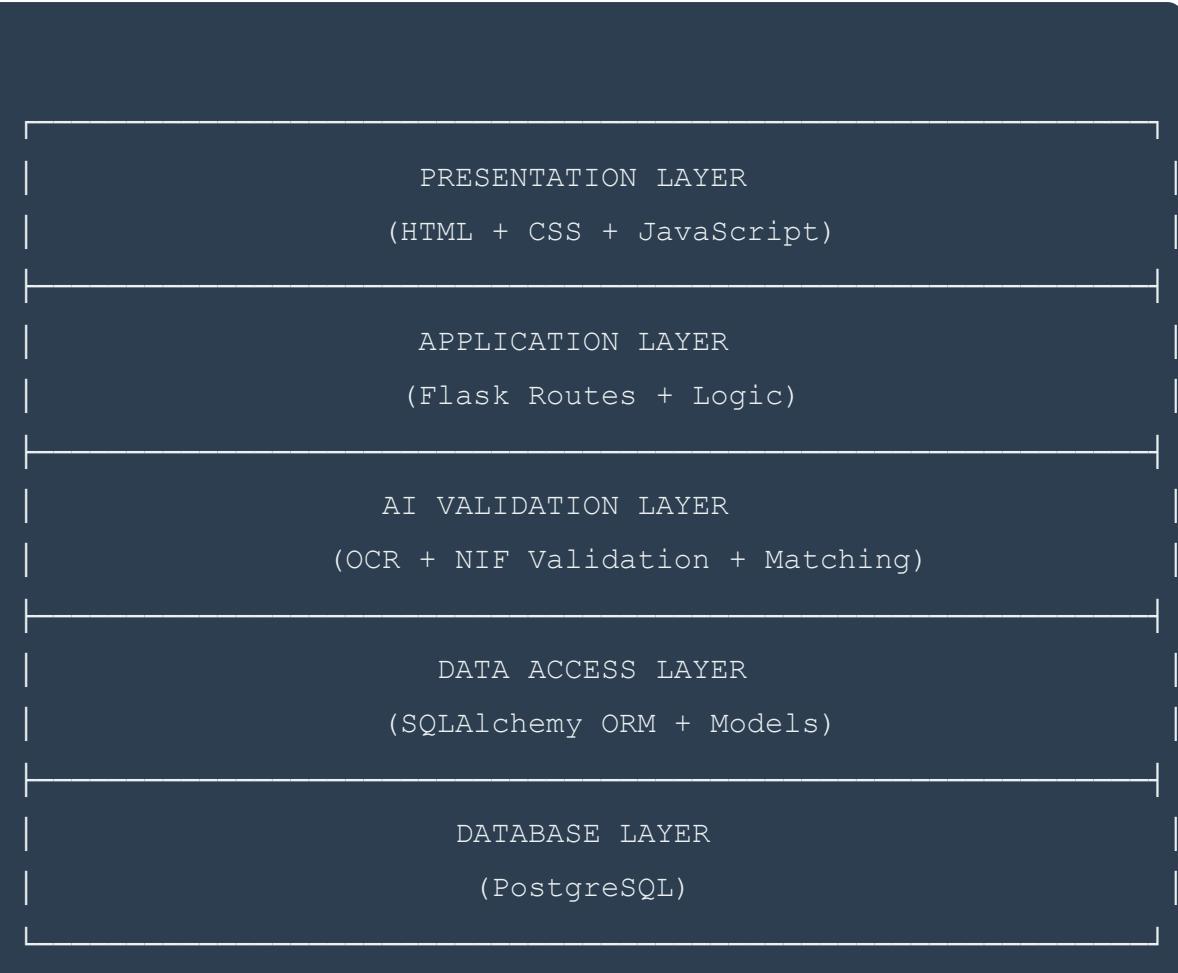
RNF-006.2 - Termos de Serviço

Descrição: Termos claros de utilização. **Conteúdo:** - Condições de uso - Responsabilidades - Limitações de responsabilidade - Política de cookies **Status:** Não implementado ---

4. ARQUITETURA DO SISTEMA / SYSTEM ARCHITECTURE

4.1 Visão Geral

O CarrisPlus utiliza uma arquitetura em camadas (Layered Architecture) com separação clara de responsabilidades:



4.2 Componentes Principais

4.2.1 Frontend (Presentation Layer)

- HTML5, CSS3, JavaScript ES6+
- Custom CSS Grid
- Jinja2
- - Renderização de interface
 - Validação de formulários (client-side)

- Feedback visual ao utilizador
- Upload de ficheiros

4.2.2 Backend (Application Layer)

- Flask 2.3.3
- Python 3.11
- MVC (Model-View-Controller)
- - Roteamento de requisições
 - Lógica de negócio
 - Validação de dados (server-side)
 - Gestão de sessões
 - Autenticação e autorização

4.2.3 AI Validation Layer

- pytesseract + OpenCV
- TensorFlow ou PyTorch
- - Extração de texto de documentos
 - Validação de NIF
 - Matching de nomes
 - Deteção de fraudes
 - Scoring de confiança

4.2.4 Data Access Layer

- SQLAlchemy
- Alembic
- - Abstração de acesso a dados
 - Queries otimizadas

- Transações
- Relacionamentos entre entidades

4.2.5 Database Layer

- PostgreSQL 15+
- - Armazenamento persistente
 - Integridade referencial
 - Índices e otimizações
 - Backups

4.3 Modelo de Dados

Entidades Principais

- id: UUID (PK)
- email: String (unique)
- password_hash: String
- nif: String (9 digits, unique)
- full_name: String
- phone: String (optional)
- created_at: DateTime
- updated_at: DateTime
- is_active: Boolean

```
- is_admin: Boolean

- id: UUID (PK)
- user_id: UUID (FK -> User)
- document_type: Enum (CC, PASSPORT, CERTIFICATE, etc.)
- file_path: String
- original_filename: String
- file_size: Integer
- upload_date: DateTime
- status: Enum (PENDING, APPROVED, REJECTED, REVIEW)
- confidence_score: Float
- validated_at: DateTime
- validated_by: UUID (FK -> User) [nullable]
- rejection_reason: Text [nullable]

- id: UUID (PK)
- user_id: UUID (FK -> User)
- pass_type: Enum (NORMAL, STUDENT, SENIOR, SOCIAL)
- zone: String
- start_date: Date
- end_date: Date
- status: Enum (ACTIVE, EXPIRED, PENDING, CANCELLED)
- price: Decimal
```



```
User (1) ———< (N) Document
|
|—————< (N) Pass
|
|—————< (N) AuditLog

Document (1) ———< (N) ValidationLog
```

4.4 Fluxo de Dados

Submissão de Documento

```
[Utilizador]
    ↓ Upload ficheiro
[Frontend] - Validação client-side
    ↓ POST /submit-document
[Flask Route] - Validação server-side
    ↓ Salvar ficheiro
[File Storage]
    ↓ Criar registo
[Database] ← Status: PENDING
    ↓ Processar
[AI Validation Layer]
    |— OCR → Extrair texto
    |— NIF → Validar número
    |— Name → Comparar com BD
```

```
└ Scoring → Calcular confiança
  ↓ Resultado
  [Database] ← Atualizar status
    ↓ Email
  [User] ← Notificação
```

4.5 Segurança da Arquitetura

1. HTTPS/TLS 1.3
2.
 - CSRF tokens
 - Input sanitization
 - Rate limiting
3.
 - JWT tokens
 - bcrypt hashing
4.
 - SQL injection prevention (ORM)
 - Prepared statements
5.
 - Encrypted files at rest
 - Access control lists

5. TECNOLOGIAS E FERRAMENTAS /

5.1 Stack Tecnológico Completo

Backend

Tecnologia	Versão	Propósito
Python	3.11	Linguagem principal
Flask	2.3.3	Web framework
SQLAlchemy	2.0+	ORM
Alembic	1.12+	Database migrations
Flask-Login	0.6+	Session management
Flask-CORS	4.0+	Cross-origin requests
bcrypt	4.1+	Password hashing
PyJWT	2.8+	JWT tokens
python-dotenv	1.0.0	Environment variables

AI/ML Processing

Tecnologia	Versão	Propósito
pytesseract	0.3+	OCR engine
OpenCV	4.8+	Image processing
Pillow	10.0.1	Image manipulation
TensorFlow	2.14+	ML framework (opcional)
NumPy	1.24+	Numerical computing

Database

Tecnologia	Versão	Propósito
PostgreSQL	15+	Database principal
psycopg2	2.9+	PostgreSQL adapter
Redis	7.2+	Cache + message broker

Frontend

Tecnologia	Versão	Propósito
HTML5	-	Markup
CSS3	-	Styling

JavaScript ES6+ Client-side logic
Jinja2 3.1.2 Template engine

DevOps

Tecnologia Versão Propósito
----- ----- -----
Docker 24.0+ Containerization
Docker Compose 2.20+ Container orchestration
Git 2.40+ Version control
GitHub - Code repository
Nginx 1.25+ Reverse proxy (produção)

Testing

Tecnologia Versão Propósito
----- ----- -----
pytest 7.4+ Unit testing
pytest-cov 4.1+ Coverage
Selenium 4.15+ E2E testing
Postman - API testing

Monitoring & Logging

Tecnologia Versão Propósito
----- ----- -----
Sentry 1.38+ Error tracking
ELK Stack 8.11+ Logging (opcional)

5.2 Justificativas das Escolhas

Por que Flask?

- ✓ Lightweight e flexivel
- ✓ Excelente documentação

- ✓ Grande comunidade
- ✓ Fácil integração com Python ML libraries
- ✓ Adequado para projeto académico

Por que PostgreSQL?

- ✓ Open source e gratuito
- ✓ ACID compliant
- ✓ Suporte a JSON
- ✓ Escalável
- ✓ Excelente integração com Python/SQLAlchemy

Por que Docker?

- ✓ Ambiente consistente (dev = prod)
- ✓ Fácil setup para equipa
- ✓ Isolamento de dependências
- ✓ Facilita deployment

6. IMPLEMENTAÇÃO ATUAL / CURRENT

6.1 Funcionalidades Implementadas



1. Estrutura base do projeto
2. Containerização Docker completa
3. Interface web responsiva
4. Upload de documentos (básico)
5. Sistema de rotas Flask

6.2 Estrutura de Ficheiros

```
CarrisPlus-GP/
|
|   app.py                      # Aplicação Flask principal
|
|   requirements.txt             # Dependências Python
|
|   Dockerfile                   # Container configuration
|   docker-compose.yml          # Orchestration
|   .dockerignore                # Docker exclusions
|
|   templates/                  # Templates Jinja2
|     base.html                 # ⚠ CORROMPIDO (necessário)
|     index.html                # ✅ Homepage funcional
|     submit_document.html      # ✅ Formulário de submissão
|     renew_document.html       # ❌ EM FALTA
|     create_pass.html          # ❌ EM FALTA
|
|   static/                      # Assets estáticos
|     css/
|       style.css               # ✅ Estilos principais
|     js/
|       main.js                 # ✅ JavaScript básico
|
|   uploads/                     # Armazenamento de documentos
|
|   README.md                    # ✅ Documentação completa
```

6.3 Estado dos Componentes

Componente	Status	Completude	Notas

Docker Setup	✓	Completo	100%	Funcional
Flask App	⚠	Parcial	30%	Rotas básicas funcionais
Templates	⚠	Parcial	60%	base.html corrompido
CSS	✓	Funcional	100%	Design responsivo
JavaScript	✓	Funcional	100%	Funcionalidade de interface
Database	✗	Não iniciado	0%	Comentado no código
Authentication	✗	Não iniciado	0%	-
AI/OCR	✗	Não iniciado	0%	-
Admin Panel	✗	Não iniciado	0%	-
Email System	✗	Não iniciado	0%	-
Tests	✗	Não iniciado	0%	-

6.4 Problemas Conhecidos



1. `base.html` corrompido (conteúdo substituído)
2. Falta implementação de database
3. Sem sistema de autenticação



1. Templates em falta (`renew_document.html`, `calendar.html`)
2. Validação de ficheiros insuficiente
3. Secret key hardcoded
4. Sem tratamento de erros



1. Adicionar testes
2. Implementar logging
3. Documentação inline do código

7. ROADMAP DE DESENVOLVIMENTO

Fase 1: Correções e Fundação 1-

- [x] ~~Setup inicial do projeto~~
- [x] ~~Containerização Docker~~
- []
- []
- [] Implementar .env para variáveis de ambiente
- [] Adicionar validação robusta de ficheiros
- [] Implementar tratamento de erros básico

- Aplicação sem erros críticos
- Todos os templates funcionais
- Validação de upload segura

Fase 2: Database e Autenticação

- [] Setup PostgreSQL
- [] Criar models SQLAlchemy
 - [] User model
 - [] Document model
 - [] Pass model
 - [] ValidationLog model
 - [] AuditLog model
- [] Implementar migrations com Alembic
- [] Sistema de registo de utilizadores
- [] Sistema de login/logout
- [] Recuperação de password
- [] Gestão de sessões com Flask-Login
- [] Hashing de passwords com bcrypt

- Database funcional
- Utilizadores podem registar-se e fazer login
- Dados persistentes

Fase 3: Validação IA e OCR 3-4

- [] Integrar pytesseract
 - [] Implementar processamento de imagem com OpenCV
 - [] Desenvolver módulo de extração de dados
 - [] Extração de nome
 - [] Extração de NIF
 - [] Extração de datas
 - [] Implementar validação de NIF
 - [] Desenvolver algoritmo de matching de nomes
 - [] Sistema de scoring de confiança
 - [] Pipeline de validação automática
 - [] Testes com documentos reais
-
- OCR funcional com 90%+ precisão
 - Validação automática de documentos
 - Sistema de scoring implementado

Fase 4: Gestão de Passes 2 semanas

- [] Implementar criação de passes
 - [] Lógica de negócio para tipos de passe
 - [] Passe Normal
 - [] Passe Estudante
 - [] Passe Sénior
 - [] Passe Social
 - [] Validação de requisitos por tipo
 - [] Cálculo de preços
 - [] Geração de PDF com comprovante
 - [] Sistema de renovação
 - [] Histórico de passes
-
- Utilizadores podem criar passes online
 - Validação automática de requisitos
 - PDFs gerados automaticamente

Fase 5: Painel Administrativo 2

- [] Dashboard com estatísticas
- [] Interface de revisão manual
- [] Gestão de utilizadores
- [] Sistema de aprovação/rejeição
- [] Visualização de logs
- [] Geração de relatórios
- [] Exportação de dados (Excel, CSV)

- Admin pode gerir todo o sistema
- Revisão manual de documentos
- Relatórios exportáveis

Fase 6: Notificações e Comunicação

- [] Integrar Flask-Mail
- [] Templates de email
- [] Sistema de notificações automáticas
- [] Configuração SMTP
- [] Queue de emails com Celery + Redis
- [] Histórico de notificações

- Emails automáticos funcionais
- Utilizadores sempre informados

Fase 7: Segurança e Compliance 3

- [] Implementar CSRF protection
- [] Rate limiting
- [] Content Security Policy
- [] Sanitização completa de inputs
- [] Scan antivírus (ClamAV)
- [] Audit logging completo
- [] Política de privacidade (RGPD)
- [] Termos de serviço
- [] Cookie consent

- Sistema seguro e em conformidade
- RGPD compliant

Fase 8: Testes e Qualidade 🕒 2 semanas

- [] Testes unitários (pytest)
- [] Testes de integração
- [] Testes E2E (Selenium)
- [] Cobertura mínima 80%
- [] CI/CD pipeline (GitHub Actions)
- [] Code linting (flake8, black)
- [] Performance testing

- Cobertura de testes > 80%
- CI/CD automático
- Código limpo e padronizado

Fase 9: API REST 🚀 1-2 semanas

- [] Endpoints RESTful
- [] Autenticação JWT

- [] Documentação Swagger
- [] Rate limiting por API key
- [] Versionamento (v1)

- API pública documentada
- Integrações possíveis

Fase 10: Deployment e Produção ⏳

- [] Setup servidor de produção
- [] Nginx como reverse proxy
- [] SSL/TLS certificates (Let's Encrypt)
- [] Variáveis de ambiente de produção
- [] Backup automático configurado
- [] Monitoring (Sentry)
- [] Logs centralizados
- [] Performance tuning
- [] Load testing

- Aplicação em produção
- Monitorização ativa
- Backups automáticos

8. CASOS DE USO / USE CASES

UC-001: Registrar Novo Utilizador

Cidadão

- Utilizador não tem conta
- Possui email válido e NIF

1. Utilizador acede à página de registo
2. Preenche formulário (nome, email, NIF, password)
3. Sistema valida dados
4. Sistema cria conta
5. Sistema envia email de confirmação
6. Utilizador confirma email
7. Conta ativada

- Sistema exibe erro "Email já registado"
- Oferece opção de recuperar password

- Sistema valida algoritmo do NIF
- Exibe erro se inválido

- Conta criada e ativa
- Utilizador pode fazer login

UC-002: Submeter Documento para Validação

Utilizador Autenticado

- Utilizador com login efetuado
- Possui documento digitalizado

1. Utilizador acede a "Submeter Documento"
2. Seleciona tipo de documento
3. Faz upload do ficheiro
4. Adiciona notas (opcional)
5. Submete formulário
6. Sistema valida ficheiro
7. Sistema processa com OCR

8. Sistema extrai dados
9. Sistema valida NIF
10. Sistema compara nome
11. Sistema calcula score
12. Se score > 90%: aprovação automática
13. Sistema envia email de confirmação

- Sistema rejeita upload
- Exibe mensagem de erro

- Documento enviado para revisão manual
- Administrador notificado

- Documento rejeitado automaticamente
- Utilizador notificado com motivo

- Documento armazenado
- Status atualizado
- Utilizador notificado

UC-003: Criar Novo Passe

Utilizador Autenticado

- Login efetuado
- Documentos aprovados

1. Utilizador acede a "Criar Passe"
2. Seleciona tipo de passe
3. Seleciona zona tarifária
4. Anexa documentos comprovativos
5. Sistema valida requisitos
6. Sistema calcula preço
7. Utilizador confirma
8. Sistema cria passe (status: PENDING)
9. Sistema gera PDF provisório
10. Sistema envia para aprovação
11. Aprovação automática ou manual
12. Sistema ativa passe

13. Utilizador recebe email com comprovante

- Sistema lista documentos em falta
- Utilizador deve submeter documentos

- Admin revisa pedido
- Aprova ou rejeita

- Passe criado
- PDF disponível para download
- Utilizador notificado

UC-004: Renovar Passe Existente

Utilizador Autenticado

- Possui passe existente
- Passe expira em < 60 dias

1. Sistema envia lembrete (30 dias antes)
2. Utilizador acede a "Renovar Passe"
3. Visualiza passe atual
4. Confirma dados
5. Atualiza documentos (se necessário)
6. Sistema valida
7. Sistema calcula novo preço
8. Utilizador confirma
9. Sistema renova passe
10. Novo período de validade aplicado
11. PDF atualizado gerado

- Passe renovado
- Nova data de validade
- Utilizador notificado

UC-005: Revisar Documento (Admin)

Administrador

- Login como admin
- Existem documentos em REVIEW

1. Admin acede ao dashboard
2. Visualiza lista de documentos pendentes
3. Seleciona documento
4. Visualiza:
 - Imagem original
 - Dados extraídos (OCR)
 - Score de confiança
 - Discrepâncias
5. Analisa documento
6. Decide: APROVAR ou REJEITAR
7. Adiciona comentários (se rejeitado)
8. Confirma decisão
9. Sistema atualiza status
10. Utilizador notificado

- Admin solicita resubmissão
- Utilizador notificado com motivo

- Documento processado
- Status atualizado
- Utilizador notificado
- Log de auditoria criado

9. TESTES E VALIDAÇÃO / TESTES

Caso Input Output Esperado Status

----- ----- ----- -----				
TC-001.1 Dados válidos Conta criada				
TC-001.2 Email duplicado Erro "Email já ex"				
TC-001.3 NIF inválido Erro "NIF inválido"				
TC-001.4 Password fraca Erro "Password ins				

TC-002: Upload de Documento

Caso	Input	Output Esperado	Status
----- ----- ----- -----			
TC-002.1 PDF válido (2MB) Upload sucesso			
TC-002.2 Ficheiro 10MB Erro "Tamanho exec..."			
TC-002.3 Ficheiro .exe Erro "Formato invál..."			
TC-002.4 JPG válido Upload sucesso			

TC-003: Validação OCR

Caso	Input	Output Esperado	Status
----- ----- ----- -----			
TC-003.1 CC português Nome + NIF extraídos			
TC-003.2 Imagem borrada Score baixo → Revisão			
TC-003.3 NIF diferente Detetar discrepâncias			

9.3 Critérios de Aceitação

- ✓ Cobertura de testes > 80%
 - ✓ Zero bugs críticos
 - ✓ Todos os testes E2E passam
 - ✓ Performance: 95% requests < 2s
 - ✓ Security scan sem vulnerabilidades HIGH
 - ✓ RGPD compliant
 - ✓ Documentação completa
-

10. DEPLOYMENT E INFRAESTRUTURA

10.1 Ambientes

Desenvolvimento (Development)

- Local (Docker)
- SQLite ou PostgreSQL local
- Ativado
- Console
- <http://localhost:5000>

Homologação (Staging)

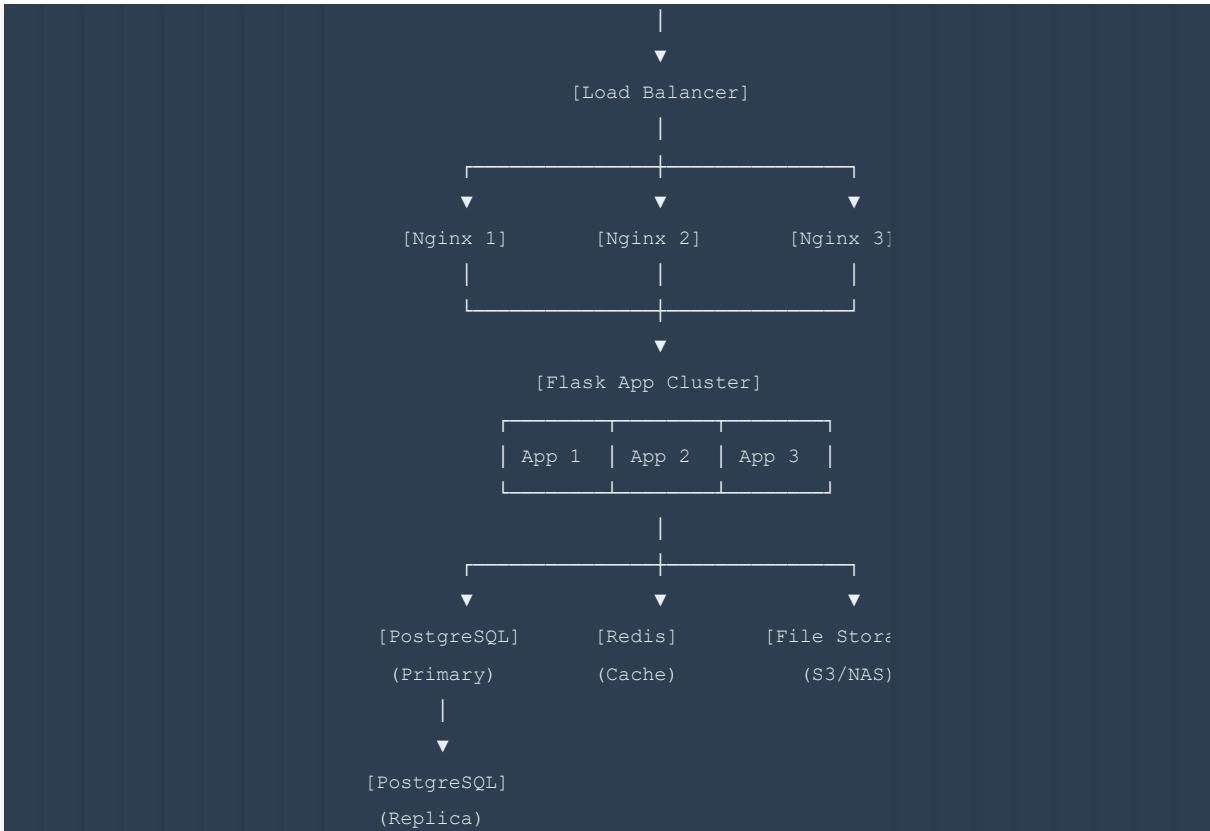
- Cloud VM
- PostgreSQL
- Desativado
- Ficheiro + Sentry
- <https://staging.carrisplus.pt>

Produção (Production)

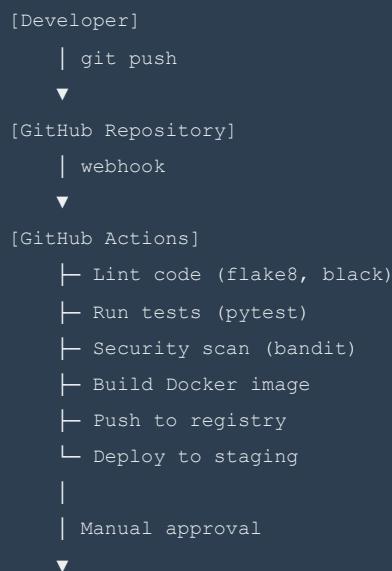
- Cloud VM (Alta disponibilidade)
- PostgreSQL com replicação
- Desativado
- Centralizado (ELK)
- <https://carrisplus.pt>
- Let's Encrypt
- Nginx
- Sentry + UptimeRobot

10.2 Arquitetura de Deployment

[INTERNET]



10.3 CI/CD Pipeline



[Deploy to Production]

10.4 Backup Strategy

- Backup completo: Diariamente às 02:00
 - Backup incremental: A cada 6 horas
 - Retenção: 30 dias
 - Teste de restore: Semanalmente
-
- Sincronização em tempo real para stor
 - Snapshot diário
 - Retenção: 90 dias
-

11. MANUTENÇÃO E SUPORTE

11.1 Procedimentos de Manutenção

- Patches críticos: Aplicados em 24h
- Patches não-críticos: Aplicados semanalmente
- Dependency updates: Mensalmente

- Uptime monitoring (UptimeRobot)
- Error tracking (Sentry)
- Performance monitoring (New Relic ou CloudWatch Metrics)
- Log analysis (ELK Stack)

- Downtime > 5 minutos → Email + SMS
- Erro rate > 5% → Email
- CPU > 80% por 10 min → Email
- Disk space < 20% → Email

11.2 Suporte aos Utilizadores

- Email: suporte@carrisplus.pt
- FAQ online
- Chat (horário comercial)

- Resposta inicial: < 24h
- Resolução crítica: < 48h
- Resolução normal: < 5 dias

12. RISCOS E MITIGAÇÃO

Riscos Técnicos

Risco	Probabilidade	Impacto	Mitigação
OCR com baixa precisão	Média	Alto	
Database corruption	Baixa	Critico	
Ataque DDoS	Média	Alto	CDN, rate limiting
Violiação de dados	Baixa	Critico	

Riscos de Projeto

Risco	Probabilidade	Impacto	Mitigação
Atraso no desenvolvimento	Alta	Médio	
Mudança de requisitos	Média	Médio	
Indisponibilidade de membros	Média	Médio	

13. CONCLUSÕES E PRÓXIMAS AÇÕES

13.1 Estado Atual do Projeto

O CarrisPlus encontra-se em:

- Infraestrutura Docker funcional
- Interface web responsiva básica
- Sistema de upload de ficheiros

Contudo, as funcionalidades core ainda:

- Sistema de autenticação
- Base de dados
- Validação IA/OCR
- Gestão de passes completa

13.2 Próximas Ações Imediatas

1. - Recuperação de dados
2. - renew_doc
- 3.
4. - Database funciona
5. - User, Documentos

1. Sistema de autenticação completo
2. OCR e validação IA
3. Lógica de criação de passes

13.3 Viabilidade e Perspetivas

O projeto é viável e tem potencial para ser implementado.

- Stack tecnológico bem escolhido
- Problema real com impacto social
- Arquitetura escalável

- Roadmap bem definido

- Complexidade da validação IA
- Compliance com RGPD
- Garantia de segurança
- Tempo limitado do projeto académico

13.4 Valor Académico

Este projeto demonstra:

- ✓ Aplicação de
- ✓ Uso de (Flask,
- ✓ Resolução de
- ✓ Integração de
- ✓ Trabalho em com Git/GitHub

14. REFERÊNCIAS / REFF

Documentação Técnica

1. - <https://flask.palletsprojects.com/en/2.2.x/>
2. - <https://www.w3.org/TR/WCAG21/>
3. - <https://docs.opencv.org/4.x/>
4. - <https://github.com/madabh/PyTorch-Image-Classification>
5. - <https://docs.opencv.org/4.x/>

Normas e Compliance

1. - Regulamento Geral de Proteção de Dados
2. - Web Content Accessibility Guidelines (WCAG)
3. - Security Best Practices

Inspiração

1. - <https://www.carris.com.br>
2. - Sistema de passes

15. ANEXOS / APPENDICE

Anexo A: Glossário

- Número de Identificação Fiscal (
- Optical Character Recognition
- Object-Relational Mapping
- Regulamento Geral de Proteção de Dados Pessoais
- Cartão de Cidadão
- JSON Web Token
- Cross-Site Request Forgery
- Cross-Site Scripting
- Vulnerabilidade de segurança

Anexo B: Variáveis de Ambiente

```
FLASK_APP=app.py
FLASK_ENV=production
SECRET_KEY=
```

```
DATABASE_URL=postgresql://user:pas  
DB_POOL_SIZE=10
```

```
TESSERACT_PATH=/usr/bin/tesseract  
OCR_LANGUAGE=por
```

```
MAIL_SERVER=smtp.gmail.com  
MAIL_PORT=587  
MAIL_USERNAME=suporte@carrisplus.p  
MAIL_PASSWORD=
```

```
BCRYPT_LOG_ROUNDS=12  
JWT_SECRET_KEY=  
JWT_EXPIRATION_HOURS=24
```

```
MAX_UPLOAD_SIZE=5242880 # 5MB  
UPLOAD_FOLDER=/app/uploads
```

```
SENTRY_DSN=
```

Anexo C: Comandos Út

```
docker-compose up --build      #  
docker-compose down           #  
docker-compose logs -f web   #
```

```
docker exec -it carrisplus-db psql
flask db init          #
flask db migrate -m "message"  #
flask db upgrade        #
```

```
pytest                 #
pytest --cov=app       #
pytest tests/unit      #
```

```
black .                #
flake8 .              #
bandit -r app/         #
```

```
gunicorn -w 4 -b 0.0.0.0:5000 app:
```

INFORMAÇÃO DE DOCUMENTO

1.0

2025-11-05

Equipa CarrisPlus (

[Nome do Professor]

[Data]

- *Este documento está preparado para:
- *Página de rosto com logotipo*
- *Índice automático*
- *Cabeçalhos e rodapés*
- *Numeração de páginas*
- *Diagramas coloridos para

