Uma imagem com texto, Tipo de letra, Gráficos, logótipo

Descrição gerada automaticamente

**Fundamentos, Administração e**

**Gestão de Base da Dados**

CTeSP em Cloud e Cibersegurança

Projeto nº 2:

**Sistema de Gestão Hospitalar**

Uma imagem com texto, computador, computador portátil, software

Descrição gerada automaticamente

Realizado por:

Amira Babkir nº2024126219

Vicente Gonçalves nº2024122708

Docente: Prof. Fábio Sampaio

ESTSetúbal/IPS, ano letivo 2024/2025

**Índice**

[1. Introdução 2](#_Toc65157191)

[2. Rede de Teste 2](#_Toc65157192)

[2.1. Topologia da Rede 2](#_Toc65157193)

[2.2. Endereçamento IP 2](#_Toc65157194)

[3. Configuração e Teste da Rede de Teste 3](#_Toc65157195)

[4. Conclusões 3](#_Toc65157196)

[Bibliografia 3](#_Toc65157197)

# Introdução

A informatização da gestão hospitalar é fundamental para garantir eficiência, organização e segurança no tratamento de dados clínicos e administrativos. Este projeto tem como objetivo o desenvolvimento de um sistema de gestão hospitalar funcional e seguro, utilizando Python e a base de dados SQLite3, com interface em consola.

O tema foi proposto pelo professor da UC e tem como finalidade aplicar os conhecimentos teóricos em um contexto prático. O sistema abrange funcionalidades como a gestão de pacientes, médicos, consultas, tratamentos, prescrições e funcionários, além de mecanismos de segurança como o controle de acesso por tipo de usuário e o registo de ações por meio de um log.

O projeto permite consolidar os conteúdos abordados em aula e desenvolver competências práticas como o trabalho em grupo, o raciocínio lógico, a organização do código e a documentação técnica. Essas habilidades são fundamentais na área de tecnologia da informação, especialmente em contextos que lidam com dados sensíveis, como os da área da saúde.

# Requesitos e Funcionalidades

## Requisitos funcionais

O sistema de gestão hospitalar oferece as seguintes funcionalidades:

* Descrever funcionalidades que temos.
* Todas elas.
* Se tiver incompletas/parcialmente implementadas indicar o mesmo.
* Verificar se temos tudo o que o prof pede.

## Requisitos não funcionais

Aqui pretende-se escrever o que funciona no programa (não em termos de funcionalidades), segue-se um exemplo em seguida: EDITAR TEXTO

* Persistência de Dados: O sistema utiliza a base de dados relacional SQLite3 para armazenar e gerir as informações dos pacientes, médicos, consultas, prescrições, tratamentos e funcionários. Todos os dados permanecem disponíveis entre diferentes sessões de uso.
* Interface: A aplicação foi desenvolvida para funcionar em consola (linha de comandos), permitindo a interação por menus de texto claros e diretos.
* Segurança: O sistema implementa autenticação por nome de utilizador e palavra-passe. Existe uma conta de administrador com permissões totais, enquanto outros usuários têm permissões restritas. Os dados sensíveis dos pacientes, como o histórico clínico, são cifrados na base de dados. O sistema também mantém um registo (log) de ações realizadas, acessível apenas ao administrador.
* Portabilidade: Por ter sido desenvolvido em Python e utilizar o SQLite3 (bibliotecas padrão), o sistema pode ser executado em qualquer ambiente que tenha Python instalado, sem necessidade de instalação adicional.
* Desempenho: O sistema realiza as operações básicas de inserção, consulta e atualização com desempenho satisfatório em testes realizados com um volume pequeno de dados.

# Estrutura da base de dados

O sistema utiliza o SQLite3 como mecanismo de base de dados relacional, com seis tabelas principais interligadas, de acordo com os requisitos do projeto. Abaixo são descritas as tabelas, os campos e os relacionamentos definidos.

## Tabelas

FAZER TODAS AS TABELAS AQUI ABAIXO COM CAMPO, TIPO.

## Relações

DESCREVER TODAS AS RELAÇÕES ENTRE TABELAS.

## Diagrama Entidade-Relacionamento (ER)

INSERIR FOTO DO DIAGRAMA ER

# Segurança e acesso

A segurança da informação é um elemento necessário no contexto hospitalar onde dados sensíveis como históricos clínicos, dados pessoais e prescrições são manipulados. Para garantir a confidencialidade, integridade e controle de acesso, forma implementadas as seguintes medidas no sistema:

## Autenticação de Utilizador

**EXEMPLO ABAIXO:**

O sistema exige autenticação por nome de utilizador e palavra-passe para qualquer operação.

Existem diferentes tipos de usuário, cada um com permissões específicas:

Administrador: acesso total ao sistema, incluindo visualização do log de acessos e alteração de dados de qualquer usuário.

Funcionários (médicos/enfermeiros): acesso limitado às funcionalidades relacionadas às suas funções.

Pacientes: acesso restrito aos seus próprios dados, podendo visualizar e editar informações pessoais.

## Controle de Permissões

**EXEMPLO ABAIXO:**

Cada funcionalidade do menu é validada com base no tipo de usuário autenticado.

Apenas o administrador pode:

Ver o log de acessos

Modificar dados de outros usuários

Visualizar todas as tabelas

Pacientes podem apenas:

Ver suas consultas, prescrições e tratamentos

Alterar o próprio contato

## Proteção de Dados Sensíveis

**EXEMPLO ABAIXO:**

O histórico médico dos pacientes é armazenado de forma cifrada na base de dados, de modo a garantir a confidencialidade em caso de acesso indevido ao banco.

As senhas dos usuários também são armazenadas de forma cifrada.

## Registos de Log

**EXEMPLO ABAIXO:**

Todas as ações relevantes no sistema são registadas em um arquivo de log, contendo:

ID do usuário

Ação executada

Data e hora

Status da operação (sucesso, erro, etc.)

Esse log só pode ser consultado por usuários com perfil de administrador.

# Interface do utilizador

## Menu Principal

COLOCAR:

Lista de principais opções do utilizador após o login

Colocar que o menu é dinâmico variando conforme o tipo de utilizador

## Navegação

Explicar como o utilizador interage com o menu

Mencionar a validação de entradas e mensagens de erro

## Tipos de Utilizador

Resumir as permições de cada perfil – admin, funcionário (medico/enfermeiro) e paciente

## Mensagens e Feedback

Informar que o sistema dá respostas claras ao utilizador como confirmações e erro.

# Implementação técnica

## Estrutura do Código

Como o programa foi desenvolvido – num único python ou em módulos;

As funcionalidades e como foi organizada, se foi em funções separadas para melhor legibilidade e reutilização.

Se o código esta comentado e os blocos de tratamento de erros para evitar falhas

## Bibliotecas Utilizadas

Bibliotecas tais como

sqlite3: para conexão, criação de tabelas, inserção e consulta de dados.

getpass: para entrada segura de palavras-passe, sem exibição no terminal.

datetime: para trabalhar com datas (consultas, prescrições, logs).

hashlib ou cryptography (caso tenham usado): para cifragem de palavras-passe ou dados sensíveis.

## Criação e Utilização da Base de Dados

Se ainda não existir a base de dados é criada automaticamente.

Que comandos foram utilizados e porque

A integridade foi respeitada com fk entre as tabelas

## Controle de Acesso

Função de login que identifica o tipo de utilizador com base no login

Depois do login o sistema esta limitado a cada privilegio de cada user

## Encriptação de Dados sensíveis

Histórico medical dos pacientes foi encriptado antes de ser armazenado

As palavras passes dos utilizadores também são armazenadas para evitar o armazenamento em texto plano

## Registo de Logs

Ações importantes realizadas no sistema são registadas numa tabela de log, com identificação do usuário, tipo de ação, data/hora e status (sucesso ou erro).

Esse log é acessível apenas ao administrador.

# Exemplos de testes e resultados?

Para garantir a funcionalidade do sistema, foram realizados alguns testes. Como seguido de exemplo:

Teste a) adição de paciente

Teste b – agendamento de consulta

Teste c – visualização de prescrições por período e medico

Teste d – teste de login com acesso restrito

# Conclusão

NÃO ESQUECER DE ESCREVER CONCLUSÃO

# Bibliografia

[1] Guia do Laboratório 1 - Configurações Básicas de Routers, *site Moodle do IPS*, consultado em abril de 2023. Link: “https://moodle.ips.pt/2223/mod/resource” “/view.php?id=79165”.