

Relatório do Trabalho Prático II da disciplina de Integração de Sistemas

Informáticos

Gestão de Contactos

Arthur Fellipe Cerqueira Gomes – 24200

Engenharia de Sistemas Informáticos – Pós-Laboral

Dezembro de 2024

Afirmo por minha honra que não recebi qualquer apoio não autorizado na realização deste trabalho prático. Afirmo igualmente que não copiei qualquer material de livro, artigo, documento web ou de qualquer outra fonte exceto onde a origem estiver expressamente citada.

Arthur Fellipe Cerqueira Gomes – 24200

Lista de Figuras

Figura 1 – Diagrama de Entidade-Relacionamento (ERD)	6
--	---

ÍNDICE

1. INTRODUÇÃO AO PROJETO	5
2. MODELO DE DADOS: DIAGRAMA DE ENTIDADE-RELACIONAMENTO (ERD)	6
3. DESCRIÇÃO DAS OPERAÇÕES PRINCIPAIS A IMPLEMENTAR	7
4. ARQUITETURA DO SISTEMA	8
4.1. OPERAÇÕES NO SERVIÇO SOAP:	8
4.2. OPERAÇÕES NA API RESTFUL:	8
4.3. IMPLEMENTAÇÃO E ACESSO À BASE DE DADOS	8
4.4. SERVIÇOS EXTERNOS A INTEGRAR	9
5. CONCLUSÃO.....	10
6. BIBLIOGRAFIA	11

1. Introdução ao Projeto

O presente relatório descreve um projeto de Gestão de Contactos, que visa desenvolver uma aplicação para gerir informações de contactos, incluindo nome, email e telefone. A aplicação será estruturada utilizando uma arquitetura multicamadas, com um serviço SOAP XML Web Service a funcionar como camada de acesso a dados (Data Layer) e uma API RESTful Web API a servir como camada de apresentação e autenticação. A autenticação será realizada com base em JWT (JSON Web Token), garantindo comunicação segura entre as camadas.

Todos os ficheiros de implementação do projeto se encontram anexados ao presente relatório, bem como foram disponibilizados online através do repositório no *GitHub* https://github.com/arthur-fellipe/ISI_TP2_24200.

2. Modelo de Dados: Diagrama de Entidade-Relacionamento (ERD)

O modelo de dados baseia-se em duas entidades principais: Contactos e Utilizadores. A entidade Contactos armazena informações essenciais como nome, email e telefone, sendo identificada por um campo único ID. A entidade Utilizadores é responsável pela autenticação e contém campos como Username e PasswordHash, também identificados por um ID exclusivo. As duas entidades se relacionam através do ID do utilizador que criou o contacto, que funciona como chave estrangeira na tabela Contactos.

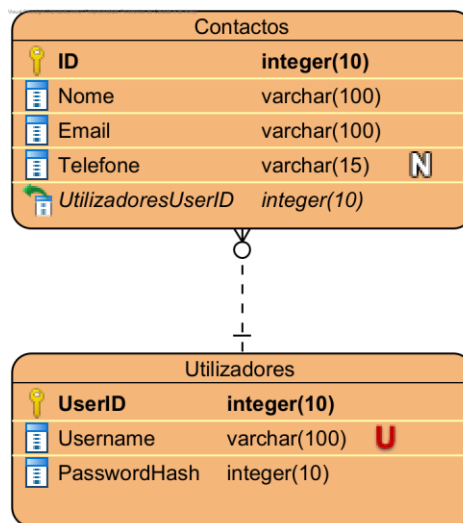


Figura 1 – Diagrama de Entidade-Relacionamento (ERD)

3. Descrição das Operações Principais a Implementar

As operações principais da aplicação serão expostas através de endpoints RESTful que permitirão a gestão de contactos e a autenticação de utilizadores. A API oferecerá métodos para criar, atualizar, listar e excluir contactos, além de permitir a criação de um utilizador e a autenticação segura, conforme lista de endpoints a seguir:

01- Autenticação:

- POST /api/auth/register: Cria um novo utilizador.
- POST /api/auth/login: Autentica o utilizador e gera um token JWT.

02 - Gestão de Contactos:

- GET /api/contact: Retorna todos os contactos.
- GET /api/contact/{id}: Retorna um contacto específico pelo ID.
- POST /api/contact: Adiciona um novo contacto.
- PUT /api/contact/{id}: Atualiza um contacto existente.
- DELETE /api/contact/{id}: Remove um contacto pelo ID.

4. Arquitetura do Sistema

A arquitetura da aplicação será dividida em duas camadas principais: a camada de dados implementada via **SOAP XML Web Service** e a camada de apresentação realizada pela **API RESTful Web API**.

4.1. Operações no Serviço SOAP:

As operações implementadas no serviço SOAP incluirão métodos para gestão de contactos e autenticação de utilizadores. Os principais métodos serão:

- **GetAllContacts():** Retorna todos os contactos.
- **GetContactById(int id):** Retorna um contacto específico pelo ID.
- **AddContact(Contact contact):** Adiciona um novo contacto.
- **UpdateContact(Contact contact):** Atualiza informações de um contacto existente.
- **DeleteContact(int id):** Remove um contacto pelo ID.
- **AddUser(string username, string passwordHash):** Cria as credenciais de um novo utilizador.
- **ValidateUser(string username, string passwordHash):** Valida as credenciais de um utilizador.

4.2. Operações na API RESTful:

A API RESTful será responsável por gerir a autenticação dos utilizadores e encaminhar chamadas ao serviço SOAP para realizar operações CRUD. A autenticação será realizada através de tokens JWT, que protegerão os endpoints contra acessos não autorizados.

4.3. Implementação e Acesso à Base de Dados

A base de dados será implementada utilizando o SQL Server, sendo o acesso realizado diretamente através de consultas SQL.

Estrutura da Base de Dados:

- **Tabela Contactos:** Contém informações dos contactos.

- **Tabela Utilizadores:** Armazena credenciais de autenticação.

As operações no banco de dados serão realizadas através de Stored Procedures e consultas SQL embutidas.

4.4. Serviços Externos a Integrar

Para enriquecer a aplicação, será integrado o serviço externo NumVerify para verificar a validade do número de telefone sempre que um novo contacto for criado ou atualizado. Isso aumentará a confiabilidade do sistema. A conexão com a API externa será feita através da camada REST.

5. Conclusão

Este projeto resultará numa aplicação segura e robusta, combinando um serviço SOAP para acesso a dados, uma API RESTful para apresentação e autenticação, e integração com serviços externos para fornecer uma experiência completa e fiável ao utilizador.

Além disto, o projeto contribuirá significativamente para o desenvolvimento de competências em integração de sistemas, consolidando conceitos aprendidos na Unidade Curricular e potenciando o uso de tecnologias modernas no contexto de fornecimento de serviços web. A partir do que foi construído, há espaço para expandir os serviços, de forma a explorar ainda mais o poder da integração de dados de forma segura.

6. Bibliografia

JWT.IO - JSON Web Tokens Introduction. Disponível em: <https://jwt.io/introduction>. Acesso em: 15 dez. 2024.

NumVerify - Free Phone Number Validation & Lookup API. Disponível em: <https://numverify.com/documentation>. Acesso em: 15 dez. 2024.