Relatório do trabalho da disciplina de Integração de Sistemas de Informação

Desenvolvimento SOAP e RESTful

João Rafael Azevedo Cunha – a21598

Licenciatura em Engenharia 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.

João Rafael Azevedo Cunha – a21598

Índice

Índice Figuras

1. Enquadramento

Este trabalho, desenvolvido no âmbito da disciplina de Integração de Sistemas de Informação (ISI), tem como objetivo a aplicação e experimentação de ferramentas para serviços Web essenciais para a integração em sistemas de informação.

O trabalho propõe o desenvolvimento de uma arquitetura de software que integra Web Services SOAP, APIs REST e acesso direto a bancos de dados SQL com serviços externos aplicados, com foco em gestão de projetos. O objetivo principal é demonstrar a aplicação de diferentes abordagens e tecnologias na construção de um sistema, respeitando as boas práticas de design e implementação.

2. Problema

Atualmente, pequenas e médias empresas (PMEs) enfrentam muitos desafios, especialmente quando se trata de modernização e digitalização de processos. A escassez de especialistas nas áreas de desenvolvimento de software é um dos problemas mais comuns. As razões para isso são restrições financeiras e a falta de especialistas envolvidos nesta área. No entanto, isso muitas vezes resulta na necessidade de contratar terceiros para o desenvolvimento de novas soluções tecnológicas. Isso não apenas sai caro aos PMEs, mas também pode tornar os sistemas muito complexos.

Em resposta a essa dificuldade, foi desenvolvido um sistema que ajuda a gestão de projetos, no qual entrega uma ferramenta necessitaria para as PMEs aumentarem a eficientemente dos seus processos.

A arquitetura do sistema foi projetada em SOAP e REST, tendo como armazenamento SQL Server. Com esta abordagem, o sistema cobre todas as operações relacionadas ao backend.

3. Estratégia Utilizada

Para alcançar os objetivos do projeto, adotei uma estratégia em várias etapas que me guiou ao longo do processo, tornando-o mais organizado e eficiente. Aqui está um resumo das principais ações que realizei:

1. **Armazenamento de dados**: O armazenamento de dados é realizado no SQL Server.
2. **SOAP**: A gestão de dados é realizada pelo SOAP, no qual realiza toda a comunicação com o SQL Server, este serviço não é disposto ao cliente final.
3. **REST**: O REST será fornecido ao cliente pois este funciona como intermediário entre o cliente e o serviço SOAP, aumentando a segurança do sistema. É possível as funções de POST, GET, PUT, UPDATE, DELETE.
4. **Interface do Utilizador**: É disponibilizado uma interface programada para o usso destes serviços.

4. Modelo de Dados

No seguinte tópico é a presentado o modelo de dados construído sobre o presente projeto construído.

a. Diagrama de Entidade-Relacionamento (ERD)

O diagrama ERD é constituído por duas tabelas:

- **Projects**: Designada para armazenar e gerir os projetos existentes.
- **Users**: Designada para armazenar e gerir os utilizadores.

Estás são as duas tabelas que realizam todo o armazenamento de dados do software. A seguinte figura representa o diagrama de ERD.

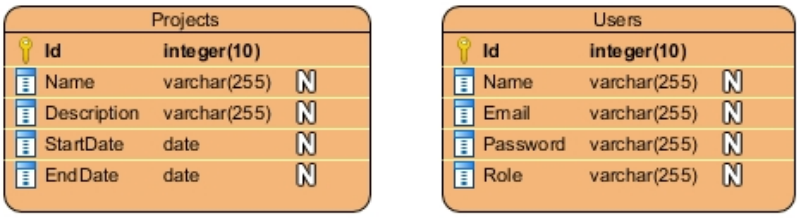

Figura 1 - Diagrama Entidade-Relação (ERD)

5. Endpoints

No seguinte tópico serão apresentados os seguintes endpoints disponíveis para realizar ações que alterem o fluxo de dados do sistema.

a. IProjectService

- **GetAllProjects** - Destinado a obter todos os projetos.
- **GetProjectById** - Destinado a procurar um projeto pelo seu id.
- **AddProject** - Destinado a criar um novo registo.
- **UpdateProject** - Destinado a atualizar os registos.
- **DeleteProject** - Destinado a obter todos os projetos.

```
public interface IProjectService
{
    [OperationContract]
    1 referência
    List<Project> GetAllProjects();

    [OperationContract]
    1 referência
    Project GetProjectById(int id);

    [OperationContract]
    1 referência
    void AddProject(Project project);

    [OperationContract]
    1 referência
    void UpdateProject(Project project);

    [OperationContract]
    1 referência
    void DeleteProject(int id);
}
```

Figura 2 - Interface IProjectService

b. IAuthenticateService

- **Login** – Destinado a realizar o login.

```
public interface IAuthenticateService
{
    [OperationContract]
    1 referência
    User Login(string username, string password);
}
```

Figura 3 - Interface IAuthenticateService

c. ProjectsController

- **/api/ObterProjetos** - Destinado a obter todos os projetos,

- **/api/ObterProjetos/{id}** - Destinado a procurar um projeto pelo seu id.
- **/api/CriarNovoProjeto** - Destinado a criar um novo registo.
- **/api/AtualizarProjeto/{id}** - Destinado a atualizar os registos.
- **/api/EliminarProjeto/{id}** - Destinado a obter todos os projetos.

```
public class ProjectsController : ControllerBase
{
    private readonly ProjectServiceClient _soapClient;

    0 referências
    public ProjectsController(ProjectServiceClient soapClient)
    {
        _soapClient = soapClient;
    }

    GetAllProjects

    GetProjectById

    CreateProject

    UpdateProject

    DeleteProject
}
```

Figura 4 - ProjectsController

d. AutenticacaoController

- **/api/SingIn** – Destinada a realização de um login, ao qual retorna um token de entrada.

```
public class AutenticacaoController : ControllerBase
{
    private readonly AuthenticateServiceClient _soapClient;

    0 referências
    public AutenticacaoController(AuthenticateServiceClient soapClient)
    {
        _soapClient = soapClient;
    }

    GetUserFromDatabase

}
```

Figura 5 - AutenticacaoController

6. Arquitetura

No seguinte tópico será apresentada a arquitetura usada para a construção do serviço web, sendo uma das etapas mais importantes para o bom funcionamento do sistema.

a. Web service SOAP XML

O protocolo SOAP usado para as seguintes operações:

- **GetAllProjects**
- **GetProjectById**
- **AddProject**
- **UpdateProject**
- **DeleteProject**
- **Login**

O SOAP realiza a comunicação diretamente com a base de dados, e não é fornecido qualquer acesso ao cliente. Deste modo é construído um *layer* de segurança.

b. API REST

O protocolo REST usado para as seguintes operações:

- ***/api/ObterProjetos***
- ***/api/ObterProjetos/{id}***
- ***/api/CriarNovoProjeto***
- ***/api/AtualizarProjeto/{id}***
- ***/api/EliminarProjeto/{id}***
- **/api/SingIn**

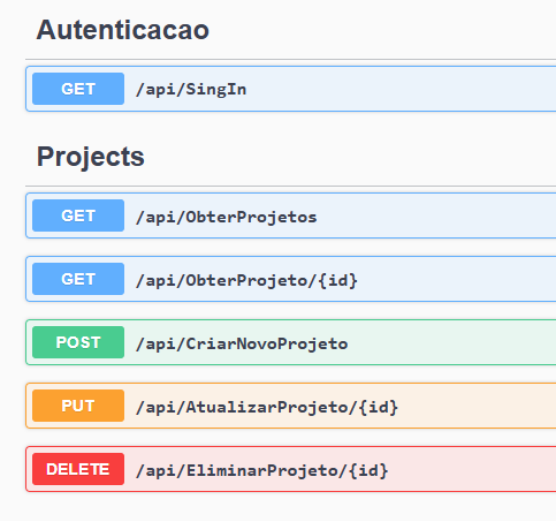

Figura 6 - Arquitetura REST

O REST é fornecido ao cliente sendo o intermediário entre cliente e o SOAP (Data Layer), desta forma o não é fornecido nenhum dado critico ao cliente.

c. Data Base

O acesso ao data base será realizado utilizando a biblioteca *System.Data.SqlClient*, assim é para comunicação direta com bases de dados SQL. Para garantir uma gestão segura e organizada da configuração de conexão, a *string* está guardada no ficheiro *web.config*, diminuindo o risco de exposição. As consultas são mantidas diretamente dentro das funções responsáveis pelas operações específicas no sistema.

Esta abordagem evita a utilizar ORMs, dando um maior nível de granularidade na interação com data base e maximizando o desempenho através de consultas SQL otimizadas.

```
  <!-- Definição da string de conexão -->
  <add name="SqlServer"
  connectionString="Server=.;Database=GanttDb;User Id=admin;Password=admin123;"
  providerName="System.Data.SqlClient" />
</connectionStrings>
```

Figura 7 - String de conexão

Conclusão

Em conclusão, este projeto de Web Service desenvolvido com o ASP.NET cumpriu com sucesso, o objetivo disponibilizar um serviço web. Aplicando arquiteturas como SOAP, REST e SQL Server foi possível garantir o correto e bom funcionamento do serviço. Em registo pessoal sinto um

enorme aprendizado a nível destas arquiteturas e de como elas se relacionam para criar um serviço mais seguro, eficiente e autónomo.

Bibliografia

Documentação SOAP - https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/webservices/soap-web-services

Documentação RESTful - https://learn.microsoft.com/pt-pt/azure/architecture/best-practices/api-design

SQL Server - https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver16