Índice

[Estrangeirismos e Acrónimos 4](#_Toc113812942)

[Introdução 6](#_Toc113812943)

[Conceitos 8](#_Toc113812944)

[Sistema de Gestão de Conteúdos 8](#_Toc113812945)

[Serviço *Web* 8](#_Toc113812946)

[SOAP 8](#_Toc113812947)

[RESTful 9](#_Toc113812948)

[Projeto 10](#_Toc113812949)

[Frontend 12](#_Toc113812950)

[Processo de Desenvolvimento 16](#_Toc113812951)

[1. Registo e Base de Dados de *Users* 16](#_Toc113812952)

[2. Criação de Sessão e Redireccionamento 17](#_Toc113812953)

[3. Estilização da Página de Registo e *Login* 17](#_Toc113812954)

[4. Criação da Barra de Reprodução 18](#_Toc113812955)

[5. Criação da Barra Lateral de Navegação 18](#_Toc113812956)

[6. Criação das Várias Tabelas na Base de Dados 18](#_Toc113812957)

[7. Configuração da Página de Entrada do Utilizador 19](#_Toc113812958)

[8. Configuração da Página do Álbum 20](#_Toc113812959)

[9. Implementação do Reprodutor de Áudio 20](#_Toc113812960)

[10. Configuração da Página do Artista 20](#_Toc113812961)

[11. Configuração da Página de Procura 20](#_Toc113812962)

[12. Configuração da Página de Listas de Reprodução 21](#_Toc113812963)

[13. Menu de Controlo do Áudio 21](#_Toc113812964)

[14. Página de Perfil do Utilizador 22](#_Toc113812965)

[15. Menu de Controlo do Administrador 22](#_Toc113812966)

[Conceito Final 24](#_Toc113812967)

[Implementações Complementares 30](#_Toc113812968)

[Web Service – Spotify 30](#_Toc113812969)

[JQuery 30](#_Toc113812970)

[Observações 32](#_Toc113812971)

[Validação do Registo por parte do Utilizador 32](#_Toc113812972)

[Adição de Conteúdos Dinamicamente 32](#_Toc113812973)

[Guardar Informação do Tema na Sessão 32](#_Toc113812974)

[Implementações Futuras 34](#_Toc113812975)

[Alteração da Imagem de Utilizador 34](#_Toc113812976)

[Adição de Conteúdos por Parte do Utilizador 34](#_Toc113812977)

[Listas de Reprodução Públicas 34](#_Toc113812978)

[Conclusão 36](#_Toc113812979)

[Bibliografia 38](#_Toc113812980)

Índice de Figuras

[Figura 1: Logótipo Sporkify. 10](#_Toc113812981)

[Figura 2: Wireframe do Ecrã de Login e Registo. 12](#_Toc113812982)

[Figura 3: Wireframe do Ecrã Principal. 12](#_Toc113812983)

[Figura 4: Ecrã de Detalhes do Utilizador. 13](#_Toc113812984)

[Figura 5: Ecrã de Edição dos Detalhes de Utilizador. 13](#_Toc113812985)

[Figura 6: Ecrã das Listas de Reprodução. 14](#_Toc113812986)

[Figura 7: Ecrã de Informação do Álbum. 14](#_Toc113812987)

[Figura 8: Ecrã de Controlo do Administrador. 15](#_Toc113812988)

[Figura 9: Estrutura da Tabela "Users". 16](#_Toc113812989)

[Figura 10: Estrutura da Tabela "Challenges". 16](#_Toc113812990)

[Figura 11: Estrutura da Tabela "Email-Accounts". 17](#_Toc113812991)

[Figura 12: Estrutura da Tabela "Albums". 18](#_Toc113812992)

[Figura 13: Estrutura da Tabela "Artists". 19](#_Toc113812993)

[Figura 14: Estrutura da Tabela "Genres". 19](#_Toc113812994)

[Figura 15: Estrutura da Tabela "Songs". 19](#_Toc113812995)

[Figura 16: Estrutura da Tabela "Playlists". 21](#_Toc113812996)

[Figura 17: Estrutura da Tabela "PlaylistSongs". 21](#_Toc113812997)

[Figura 18: Ecrã de Login e Registo, com Processo de Registo em Curso. 24](#_Toc113812998)

[Figura 19: Controlo de Utilizadores do Administrador, e Ativação de Utilizador. 25](#_Toc113812999)

[Figura 20: Página Inicial do Utilizador. 25](#_Toc113813000)

[Figura 21: Página de Pesquisa. 26](#_Toc113813001)

[Figura 22: Página das Listas de Reprodução e Detalhes da Mesma. 26](#_Toc113813002)

[Figura 23: Painel de Perfil do Utilizador, e Edição dos Detalhes de Utilizador. 27](#_Toc113813003)

[Figura 24: Vista do Álbum. 27](#_Toc113813004)

[Figura 25: Vista do Artista. 28](#_Toc113813005)

[Figura 26: Controlos para Remover Canções e Consultar o Spotify. 28](#_Toc113813006)

# Introdução

No âmbito do desenvolvimento da componente prática da disciplina de Infraestruturas Computacionais Distribuídas, foi proposto o desenvolvimento, ao longo do semestre, do jogo multijogador da Batalha Naval, onde os jogadores tentam afundar a armada do adversário.

No início do jogo, cada jogador posiciona os seus navios, representados por um grupo de quadrados em linha reta que não podem ser contíguos a outro navio.

Cada jogador dispõe dos seguintes navios:

* 1 porta-aviões (cinco quadrados);
* 2 navios-tanque (quatro quadrados);
* 3 contratorpedeiros (três quadrados);
* 4 submarinos (dois quadrados).

Cada jogador possui uma grelha que apresenta a disposição dos seus navios.

Em cada turno, um jogador indica um quadrado da grelha do adversário. Se houver um navio nessa posição é identificado tiro certeiro, caso contrário é identificado tiro na água. O jogo termina quando um jogador adivinhar todas as posições de todos os navios do adversário.

Os jogadores farão um auto-registo indicando o nickname, password e foto. Na componente de multijogador, cada jogador poderá convidar outros jogadores previamente “logados” no sistema que não estejam a participar noutras partidas, e iniciar de imediato uma nova partida após o convite ter sido aceite.

O jogo proposto deverá ser capaz de estabelecer a comunicação entre todos os clientes no sistema, bem como manter a informação estritamente necessária sobre os jogadores e os jogos em curso. Assim, será necessário implementar um sistema com base numa arquitetura cliente-servidor e definir cuidadosamente as mensagens trocadas entre as diversas componentes do sistema.

No decorrer deste documento iremos apresentar as diversas abordagens e tomadas de decisões necessárias de forma a garantir o cumprimento dos requisitos.

# Desenvolvimento

O sistema proposto, conforme indicado anteriormente, deve ser capaz de estabelecer a comunicação entre diversos jogadores que se autenticam no mesmo. Assim, é necessário manter a informação sobre todos os jogadores autenticados, bem como sobre os jogos em curso. Para isso, iniciámos o processo de desenvolvimento pela construção da arquitetura cliente-servidor do nosso sistema. Cada cliente solicita recursos ao servidor e recebe uma resposta de acordo com o pedido efetuado.

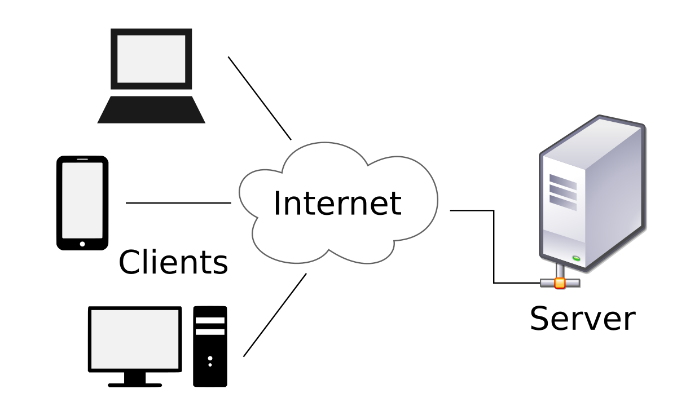


Figura 1 - Arquitetura Cliente - Servidor

Esta arquitetura opera com base num modelo de pedido-resposta, pelo que as mensagens trocadas para realizar os pedidos e repostas devem ser cuidadas e bem definidas. Assim, definimos as mensagens trocadas entre o cliente e o servidor. Todas as mensagens foram definidas em XML e obedecem às regras de validação criadas em XSD.

O protocolo de comunicação criado obedece à seguinte estrutura:

<protocolo>

<pedido tipo = ”*operação*”>

*(dados do pedido)*

</pedido>

</protocolo>

Figura 2 - Estrutura Pedido XML

<protocolo>

<pedido tipo = ”*operação*”>

*(dados adicionais do pedido)*

<resposta>

*(dados adicionais da resposta)*

</resposta>

</pedido>

</protocolo>

Figura 3 - Estrutura Resposta XML

## Mensagens para Login

<protocolo>

<pedido tipo="login">

<nome\_utilizador>Ana</nome\_utilizador>

<password>Testes1234</password>

<foto src="C:\Users\anaso\Desktop\galaxy.jpg"/>

</pedido>

</protocolo>

Figura 4 - Exemplo pedido login

Os dados dos utilizadores anteriormente “logados” são armazenados no ficheiro *utilizadores.xml*. Assim, após o auto-registo do jogador é possível verificar se a autenticação é válida ou não. Desta forma, podem ser produzidas várias respostas para o login. Adicionalmente, o servidor apenas permite uma sessão por jogador, pelo que se o jogador se tentar autenticar com uma sessão ativa não conseguirá fazê-lo.

<protocolo>

<pedido tipo="login">

<nome\_utilizador>Ana</nome\_utilizador>

<password>Testes1234</password>

<foto src="C:\Users\anaso\Desktop\galaxy.jpg"/>

<resposta>

<codigo>Autorizado</codigo>

<mensagem/>

</resposta>

</pedido>

</protocolo>

Figura 5 - Exemplo resposta login com sucesso

<protocolo>

<pedido tipo="login">

<nome\_utilizador>Ana</nome\_utilizador>

<password>Testes1234</password>

<foto src="C:\Users\anaso\Desktop\galaxy.jpg"/>

<resposta>

<codigo>Não Autorizado</codigo>

<mensagem>Utilizador já logado</mensagem>

</resposta>

</pedido>

</protocolo>

Figura - Exemplo resposta login de jogador login já efetuado

<protocolo>

<pedido tipo="login">

<nome\_utilizador>Tiago</nome\_utilizador>

<password>UmaPassword</password>

<foto src="C:\Users\anaso\Desktop\galaxy.jpg"/>

<resposta>

<codigo>Não Autorizado</codigo>

<mensagem>Password incorreta</mensagem>

</resposta>

</pedido>

</protocolo>

Figura 7 - Exemplo resposta login com password incorreta

## Mensagens para listar utilizadores

Os jogadores podem listar os utilizadores disponíveis para iniciar um novo jogo.

<protocolo>

<pedido tipo="listaUtilizadores"/>

</protocolo>

<protocolo>

<pedido tipo="listaUtilizadores">

<resposta>

<utilizadores>

<utilizador>

<nome\_utilizador>Ana</nome\_utilizador>

</utilizador>

</utilizadores>

</resposta>

</pedido>

</protocolo>

## Mensagens para convidar utilizadores

A mensagem para o convite apenas é enviada após o jogador indicar o nickname do jogador que pretende convidar para iniciar um novo jogo. Assim, inicialmente é enviada a mensagem para listar os utilizadores, apresentada acima, e, posteriormente, é enviada uma mensagem de convite para o jogador indicado.

<protocolo>

<pedido tipo="convite">

<anfitriao>

<nome\_utilizador>Tiago</nome\_utilizador>

</anfitriao>

<convidado>

<nome\_utilizador>Ana</nome\_utilizador>

</convidado>

</pedido>

</protocolo>

O servidor reencaminha o pedido com o convite para o jogador convidado. Após ter enviado esta mensagem, o servidor responde, com a seguinte mensagem, ao jogador que efetuou o convite sinalizando que o mesmo foi enviado.

<protocolo>

<pedido tipo="convite">

<anfitriao>

<nome\_utilizador>Tiago</nome\_utilizador>

</anfitriao>

<convidado>

<nome\_utilizador>Ana</nome\_utilizador>

</convidado>

<resposta>

<estado>Enviado</estado>

</resposta>

</pedido>

</protocolo>

O jogador convidado recebe a seguinte mensagem, podendo aceitar ou recusar o convite.

<protocolo>

<pedido tipo="convite">

<anfitriao>

<nome\_utilizador>Tiago</nome\_utilizador>

</anfitriao>

<convidado>

<nome\_utilizador>Ana</nome\_utilizador>

</convidado>

</pedido>

</protocolo>

Caso o jogador convidado recuse o convite, envia a seguinte mensagem para o servidor, propagando-a até ao jogador que iniciou o convite.

<protocolo>

<pedido tipo="convite">

<anfitriao>

<nome\_utilizador>Tiago</nome\_utilizador>

</anfitriao>

<convidado>

<nome\_utilizador>Ana</nome\_utilizador>

</convidado>

<resposta>

<estado>Recusado</estado>

</resposta>

</pedido>

</protocolo>

Caso o jogador convidado aceite o convite, o servidor, ao receber a mensagem abaixo, envia uma nova mensagem para ambos os jogadores iniciarem o jogo.

<protocolo>

<pedido tipo="convite">

<anfitriao>

<nome\_utilizador>Tiago</nome\_utilizador>

</anfitriao>

<convidado>

<nome\_utilizador>Ana</nome\_utilizador>

</convidado>

<resposta>

<estado>Aceite</estado>

</resposta>

</pedido>

</protocolo>

## Mensagens para iniciar o jogo

O pedido para iniciar o jogo é proveniente do servidor indicando quais os navios necessários, a quantidade de determinado tipo de navio e o tamanho que cada um ocupa no tabuleiro.

<protocolo>

<pedido jogador="Ana" tipo="setup\_game">

<navios>

<portaAvioes quantidade="1" tamanho="5"/>

<tanque quantidade="2" tamanho="4"/>

<contratorpedeiros quantidade="3" tamanho="3"/>

<submarino quantidade="4" tamanho="2"/>

</navios>

</pedido>

</protocolo>

O jogador indica a posição inicial e a posição final de cada navio respondendo da seguinte forma ao servidor.

<protocolo>

<pedido jogador="Ana" tipo="setup\_game">

<navios>

<portaAvioes quantidade="1" tamanho="5"/>

<tanque quantidade="2" tamanho="4"/>

<contratorpedeiros quantidade="3" tamanho="3"/>

<submarino quantidade="4" tamanho="2"/>

</navios>

<resposta>

<navios>

<PortaAvioes tamanho="5">

<posIni>

<linha>0</linha>

<coluna>0</coluna>

</posIni>

<posFin>

<linha>0</linha>

<coluna>4</coluna>

</posFin>

</PortaAvioes>

<Tanque tamanho="4">

<posIni>

<linha>2</linha>

<coluna>0</coluna>

</posIni>

<posFin>

<linha>2</linha>

<coluna>3</coluna>

</posFin>

</Tanque>

(…)

<Contratorpedeiro tamanho="3">

<posIni>

<linha>6</linha>

<coluna>0</coluna>

</posIni>

<posFin>

<linha>6</linha>

<coluna>2</coluna>

</posFin>

</Contratorpedeiro>

(…)

<Submarino tamanho="2">

<posIni>

<linha>4</linha>

<coluna>6</coluna>

</posIni>

<posFin>

<linha>4</linha>

<coluna>7</coluna>

</posFin>

</Submarino>

(…)

</navios>

</resposta>

</pedido>

</protocolo>

## Mensagens para ataque

Após estarem configurados os tabuleiros de ambos os jogadores, estes iniciam o jogo propriamente dito, atacando estrategicamente os navios do adversário. Para isso, cada jogador deve indicar a posição a atacar enviando a seguinte mensagem ao servidor.

<protocolo>

<pedido tipo="joga"/>

</protocolo>

<protocolo>

<pedido tipo="atualiza">

<resposta resultado="Acertou">

<tiro>

<linha>3</linha>

<coluna>4</coluna>

</tiro>

<atacante>Tiago</atacante>

</resposta>

</pedido>

</protocolo>

## Mensagem para término do jogo

Quando o servidor deteta que um dos tabuleiros dos jogadores não tem mais navios por afundar, envia a mensagem abaixo para ambos os jogadores, indicado que o jogo terminou e anunciando o vencedor.

Caso um dos jogadores abandone o jogo, esta mensagem também é enviada para o outro jogador, anunciando-o vencedor.

<protocolo>

<pedido tipo="termina">

<estado>Ganho</estado>

<vencedor>Ana</vencedor>

</pedido>

</protocolo>

NOTAS ADICIONAIS PARA O RELATÓRIO

1 – Diagrama geral da arquitetura proposta representando os componentes e os fluxos de dados

2 – Identificação dos protocolos ao nível da camada de transporte (TCP e/ou UDP) e respetivos portos

3 – Descrição de de cada componente (iterativo ou concorrente) enumerando as funcionalidades

4 – Descrição das estruturas usadas para manter os dados dos jogadores de forma persistente (XSD dos documentos XML e/ou classes dos objetos seriados)

5 – Descrição dos protocolos ao nível da aplicação (sintaxe, semântica e temporização) incluindo a estrutura das mensagens (XSD dos documentos XML e/ ou classes dos objetos seriados)

6 – Apresentação de ecrãs com exemplos de utilização mais relevantes

7 – Na conclusão devem ser enumeradas as vantagens e desvantagens da arquitetura/solução proposta, destacando as características relevantes, designadamente: expansibilidade, tolerância às falhas, segurança, transparência e concorrência.

# Implementações Futuras

Gostaríamos agora de salientar implementações para uma futura evolução do projeto.

## Alteração da Imagem de Utilizador

A personalização é uma parte importante de qualquer sistema de partilha de conteúdos por parte dos utilizadores, pois ajuda a diferenciar cada um deles. Como tal, pretende-se a implementação da mecânica de alteração da imagem de utilizador, com recurso a interface gráfica.

## Adição de Conteúdos por Parte do Utilizador

De momento o sistema implementado não permite aos utilizadores criarem diretamente os seus próprios conteúdos – os seus próprios áudios para adicionar à base de dados. Por motivos de criar um CMS similar a outros já existentes, não considerámos esta implementação prioritária. No entanto, no conceito da disciplina, será algo importante a ter em conta, pelo que se pretende que seja uma implementação futura.

## Listas de Reprodução Públicas

As listas de reprodução de cada utilizador são privadas, isto porque no conceito inicial do projeto fazia sentido serem conteúdo direcionado à personalização direta de cada utilizador. No entanto, achámos que, numa futura melhoria do sistema, fazia sentido dar a capacidade de cada utilizador poder partilhar as suas listas com outros utilizadores, onde poderemos incluir a sua procura no mecanismo de pesquisa já implementado.

# Conclusões

Durante o desenvolvimento deste projeto foi possível explorar vários conceitos relevantes à disciplina de Sistemas de Multimédia para a Internet. Alguns dos temas abordados que se refletem na elaboração deste projeto:

* Configuração Inicial da Base de Dados;
* Registo e Criação de Sessão de Utilizador;
* Implementação de CAPTCHA no Registo;
* Autenticação e Permissões de Utilizadores;
* Utilização de Web Services.

O grupo tem noção da complexidade que um CMS envolve no que toca à sua implementação, manutenção e proteção. Não deve ser implementado de forma elementar, porque trabalha com a informação dos seus utilizadores para dinamizar o controlo dos mesmos sobre os seus próprios conteúdos.

O grupo acredita ter tido a capacidade de demonstrar conhecimento técnico na execução deste projeto. Apesar de algumas das implementações necessitarem aprofundamento ou melhorias, trata-se de um CMS funcional e intuitivo.

# Bibliografia

* Stackoverflow:
  + <https://stackoverflow.com/>
* Spotify:
  + <https://open.spotify.com/>
* Documentação PHP:
  + <https://www.php.net/>
* Documentação Apache HTTP:
  + <https://httpd.apache.org/>
* Wikipedia:
  + <https://pt.wikipedia.org/>
* REST API Tutorial:
  + <https://restfulapi.net/>
* Uizard:
  + <https://uizard.io/>
* Draw IO:
  + <https://app.diagrams.net/>