



INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
DEPARTAMENTO DE ENGENHARIA ELETRÓNICA E DE
TELECOMUNICAÇÕES E COMPUTADORES

LICENCIATURA EM ENGENHARIA INFORMÁTICA E MULTIMÉDIA
UNIDADE CURRICULAR DE PROJETO

Anime-Arena

(eventual) imagem ilustrativa do trabalho – *dimensão*: até 13cm x 4cm

Autor

Pedro Dias (45170)

Orientador

Professor [Doutor] Porfírio Filipe

setembro, 2022

Resumo

A reedição de um jogo antigo chamado *Naruto-Arena*, trouxe nostalgia a um número considerável de pessoas. Tendo sido parte da infância de muitos jogadores durante vários anos, estes têm demonstrado interesse por versões alternativas do seu jogo preferido.

Neste âmbito, de forma a satisfazer este mercado inexplorado, considera-se relevante a criação de um jogo que englobe diferentes séries animadas japonesas (animes), introduzindo novas personagens, combinações, missões e classificações.

O objetivo deste projeto, denominado *Anime-Arena*, é conceber e implementar uma plataforma para a web, que seja capaz de gerir a interação entre pares de jogadores que competem para melhorar a sua classificação e desbloquear personagens.

Para solução da aplicação, utilizou-se JSP de modo a criar páginas dinâmicas baseadas em HTML que possuem também Javascript, de forma a melhorar a experiência do utilizador e enviar pedidos ao servidor (Servlets) com Ajax, sendo possível ler e escrever na base de dados que foi construída em MySQL e XML.

Abstract

Write here an overview of your work . . .

Motivation, most relevant ideas, main contributions, evaluations and brief conclusions.

Short sentences. Succinct paragraphs. Top-down approach.

Agradecimentos

O objetivo deste texto é agradecer às pessoas que contribuíram para a realização deste projeto.

À minha família, pilar indispensável que me deram todo o seu apoio, tanto emocional como financeiro, ao longo do meu percurso académico.

Ao Engenheiro Porfírio Filipe, por toda a sua disponibilidade no desenvolvimento deste projeto. Esteve sempre presente tanto para tirar questões, como para dar a sua opinião e sugestões, permitindo-me realizar este projeto de forma fluida.

À minha equipa que visiona lançar e manter o jogo online, pelo debate de ideias, incansável ajuda na organização e planeamento, por toda a componente gráfica e pela criação de personagens, habilidades e missões. Obrigado “Betraye”, “Clover”, “Aby” e “Alerane”.

Aos docentes da cadeira de projeto, Paulo Trigo e Hélder Bastos, pelas críticas construtivas, disponibilidade a responder às questões durante a unidade curricular e pelos materiais fornecidos.

Eventual texto de dedicatória . . .

. . . mais texto,

. . . e o fim do texto.

Índice

Resumo	i
Abstract	iii
Agradecimentos	v
Índice	ix
Lista de Tabelas	xi
Lista de Figuras	xiii
1 Introdução	1
1.1 Enquadramento	1
1.2 Motivação	2
1.3 Objetivo	3
1.4 Requisitos	3
1.4.1 Funções do Sistema	3
1.4.2 Atributos do Sistema	4
1.4.3 Atributos e Funções do Sistema	5
1.5 Contribuição	6
1.6 Organização do Documento	6
2 Tecnologias	9
2.1 Linguagens	9
2.2 Sistemas de Gestão de Dados	10
2.2.1 Base de Dados Relacional	10
2.2.2 Base de Dados Não Relacional	11

2.3	Servidores Aplicacionais	11
2.4	Ferramentas	12
3	Projeto	15
3.1	Casos de Utilização	15
3.2	Arquitetura	16
3.2.1	Definição de Personagens	16
3.2.2	Modelo EA	16
3.2.3	Configuração de Habilidades	19
3.2.4	Configuração de Missões	19
3.3	Interface Humana	19
4	Demonstração	21
5	Conclusão	23
5.1	Sumário	23
5.2	Desenvolvimentos Futuros	23
A	Um Detalhe Adicional	25
B	Outro Detalhe Adicional	27
	Bibliografia	29

Lista de Tabelas

1.1	Funções do Sistema	4
1.2	Requisitos Funcionais Site	4
1.3	Requisitos Funcionais Jogo	5
1.4	Atributos do Sistema	5
1.5	Requisitos Não-Funcionais	5
1.6	Atributos e Funções do Sistema	6

Lista de Figuras

1.1	Fim do Naruto-Arena antigo	2
3.1	Entidade User	17
3.2	Entidade Mission	17
3.3	Entidade Ability	18

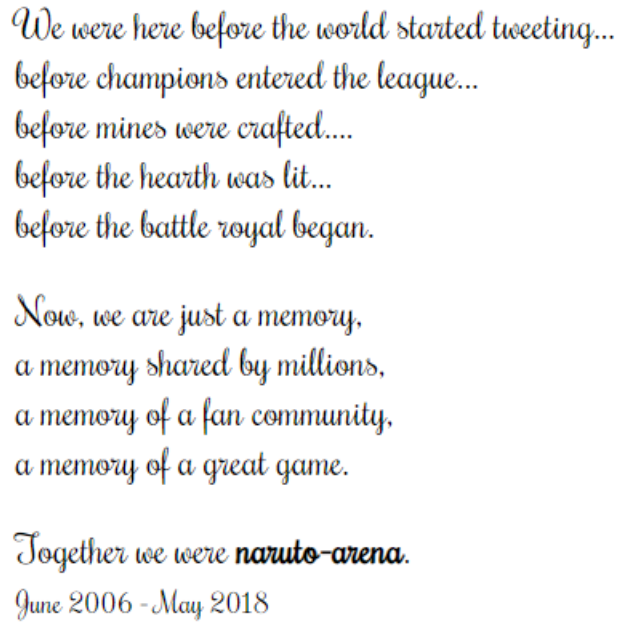
Capítulo 1

Introdução

A reedição de um jogo antigo chamado *Naruto-Arena*, trouxe nostalgia a um número considerável de pessoas. Tendo sido parte da infância de muitos jogadores durante vários anos, estes têm demonstrado interesse por versões alternativas do seu jogo preferido.

1.1 Enquadramento

Tudo começou em 2006 com a criação do *Naruto-Arena* [Wespro, 2019]. Devido ao facto de nesta altura o anime *Naruto* [Kishimoto, 2002] ser bastante popular entre as crianças e os jovens, muitos procuravam por jogos relacionados com este seu interesse. Inevitavelmente encontraram este jogo e jogaram-no até 2018. Infelizmente, no final do ano 2015 e início de 2016, os criadores abandonaram o projeto. A inexistência de atualizações fez diminuir consideravelmente o número de jogadores e o jogo terminou permanentemente em 2018. No final de maio desse ano, quando uma pessoa tentava entrar no site, apenas encontrava a seguinte imagem:

A screenshot of a tweet from the Naruto-Arena Twitter account. The text of the tweet is: "We were here before the world started tweeting... before champions entered the league... before mines were crafted.... before the hearth was lit... before the battle royal began. Now, we are just a memory, a memory shared by millions, a memory of a fan community, a memory of a great game. Together we were **naruto-arena**. June 2006 - May 2018". The text is in a cursive font, and the words "naruto-arena" are in bold. The date "June 2006 - May 2018" is at the bottom.

*We were here before the world started tweeting...
before champions entered the league...
before mines were crafted....
before the hearth was lit...
before the battle royal began.*

*Now, we are just a memory,
a memory shared by millions,
a memory of a fan community,
a memory of a great game.*

*Together we were **naruto-arena**.*

June 2006 - May 2018

Figura 1.1: Fim do Naruto-Arena antigo

Não ser possível voltar a jogar deixou tristeza, mas também esperança que os administradores voltassem e/ou alguém recriasse o seu jogo de infância que tanto gostavam. Este desejo concretizou-se em 2019 quando uma pessoa exterior (que não fazia parte da equipa do jogo original) “reviveu” o jogo que tantos queriam. Este ainda estava numa fase inicial, onde lhe deu o nome de “versão alpha”. Em 2020 lançou oficialmente o produto final, chamando-lhe “versão beta”.

1.2 Motivação

A reedição do *Naruto-Arena* [Wespro, 2019] trouxe alegria a muitas pessoas, mas também desejo por mais. Desde o abandono do jogo original até à atualidade, os jogadores têm exprimido o seu desejo por versões alternativas. Uns por quererem algo diferente onde as personagens, habilidades e classificações não são iguais, outros por preferirem diferentes animes ao Naruto [Kishimoto, 2002]. As razões variam entre os jogadores, mas é o facto que um número considerável de pessoas tem interesse por algo novo baseado nesta versão original. Este projeto visa satisfazer este mercado, englobando neste

estilo de jogo vários animes diferentes. Foi pensado em fazer para um anime em particular, no entanto a popularidade varia consoante o tempo, e nem todas os jogadores preferem o mesmo anime. Assim, este problema fica resolvido com a existência de vários animes no mesmo jogo, e garante conteúdo para o futuro, pois todos os anos aparecem animes novos e populares.

1.3 Objetivo

O objetivo de *Anime-Arena* é conceber e implementar um jogo para a web, capaz de gerir a interação entre pares de jogadores que competem para melhorar a sua classificação e desbloquear personagens.

Este projeto visa manter as características do jogo original que são familiares e apelativas aos jogadores, mas também fazer uma interpretação pessoal e melhorar funcionalidades existentes.

1.4 Requisitos

Os requisitos descrevem as necessidades e propósitos do projeto. Esta fase serve para colecionarmos dados necessários para alcançar o objetivo e vai determinar o sucesso ou o fracasso do projeto. De forma ao trabalho estar bem estruturado, estes requisitos relevantes para o projeto devem ser bem analisados e detalhados.

1.4.1 Funções do Sistema

As funções do sistema são os requisitos funcionais e servem para representar aquilo que o sistema é suposto fazer. Estas devem ser categorizadas, de modo a definir prioridades e identificar as que, sendo evidentes, consomem recursos temporais. Os requisitos podem assim ser divididos nas seguintes categorias:

Com a utilização destas categorias, são apresentadas, nas próximas tabelas, todas as funções deste sistema.

Categoria	Significado
Evidente	Tem que ser realizada O utilizador tem que ter conhecimento da sua realização
Invisível	Tem que ser realizada Não é visível para os utilizadores
Adorno	Opcional Não afeta significativamente o custo ou outras funções

Tabela 1.1: Funções do Sistema

Ref	Função	Categoria
R1.1	Criação do perfil do jogador	Evidente
R1.2	Autenticação do jogador (nome e senha)	Evidente
R1.3	Identificação do perfil do jogador	Evidente
R1.4	Consulta de perfis dos jogadores	Evidente
R1.5	Criar missões	Evidente
R1.6	Consultar missões	Evidente
R1.7	Criar personagens	Invisível
R1.8	Consultar habilidades das personagens	Evidente
R1.9	Notificações na página principal	Evidente

Tabela 1.2: Requisitos Funcionais Site

1.4.2 Atributos do Sistema

Os atributos do sistema são os requisitos não-funcionais e representam as características ou dimensões do sistema. Estes, podem ser definidos em duas categorias:

Neste projeto, os requisitos não-funcionais são focados maioritariamente na experiência do utilizador, com o objetivo de melhorar a sua compreensão e facilitar a sua utilização do jogo. Assim, os atributos do sistema são os seguintes:

Ref	Função	Categoria
R2.1	Formar uma equipa	Evidente
R2.2	Jogar contra outros jogadores	Evidente
R2.3	Animações em tempo real que facilitem a compreensão do jogo	Adorno
R2.4	Reprodução de sons	Adorno

Tabela 1.3: Requisitos Funcionais Jogo

Categoria	Significado
Obrigatório	Tem que ser contemplado
Desejável	Deve estar preparado para alcançar

Tabela 1.4: Atributos do Sistema

Atributos	Detalhe / Restrição de Fronteira	Categoria
Facilidade de Utilização	Responsividade	Obrigatório
	Fácil aprendizagem das mecânicas do jogo	Desejável
	Representação de símbolos para pessoas daltónicas	Desejável
	Existência de vários idiomas	Desejável
Interação Pessoa-Máquina	Utilização de rato e de teclado	Obrigatório
Plataforma	Windows	Obrigatório
Tolerância a falhas	Minimizar o máximo possível	Desejável
Tempo de resposta	O mais rápido possível	Desejável

Tabela 1.5: Requisitos Não-Funcionais

1.4.3 Atributos e Funções do Sistema

É importante evidenciar as relações entre funções e atributos do sistema, ou seja, descrever os atributos relacionados com funções específicas. Estas relações podem ser visualizadas de seguida:

Atributos	Funções
Facilidade de Utilização	R1.4, R1.6, R1.8, R1.9, R2.3, R2.4
Interação Pessoa-Máquina	R1.1, R1.2, R1.3, R1.4, R1.5, R1.6, R1.7, R1.8, R1.9, R2.1, R2.2, R2.3, R2.4
Plataforma	R1.1, R1.2, R1.3, R1.4, R1.5, R1.6, R1.7, R1.8, R1.9, R2.1, R2.2, R2.3, R2.4
Tolerância a falhas	R1.1, R1.2, R1.3, R1.4, R1.5, R1.6, R1.7, R1.8, R1.9, R2.1, R2.2, R2.3, R2.4
Tempo de resposta	R1.1, R1.2, R1.3, R1.4, R1.5, R1.6, R1.7, R1.8, R1.9, R2.1, R2.2, R2.3, R2.4

Tabela 1.6: Atributos e Funções do Sistema

1.5 Contribuição

Existe uma comunidade interessada em desenvolver a parte gráfica e manter o jogo online. Nomeadamente, foi formada uma equipa de pessoas que contribui no planeamento e organização do projeto, assim como na criação das imagens do jogo: personagens, habilidades, missões, classificações, energias e botões.

A contribuição principal deste projeto assenta na satisfação do mercado que deseja versões alternativas do jogo original, querendo jogar com novas personagens, realizar diferentes missões e obter classificações de outros animes.

1.6 Organização do Documento

No documento presente, existem cinco capítulos que descrevem o projeto desenvolvido.

Este capítulo de introdução começa por enquadrar o projeto, explicando a origem e o tipo deste jogo de uma forma geral. Fala-se da motivação do projeto, nomeadamente a atualidade e o interesse do público. É dito concretamente o objetivo que se pretende alcançar e nos requisitos, são enumeradas as características a implementar. Por fim, na parte da contribuição é explicado o que foi feito e as considerações sobre o produto final.

No segundo capítulo é descrito as tecnologias consideradas relevantes para a elaboração deste projeto, nomeadamente linguagens, sistemas para gestão de dados, servidores aplicativos e ferramentas.

O terceiro capítulo trata a solução apresentada do projeto, mais concretamente os casos de utilização, a arquitetura para descrever os serviços (definição de personagens, modelo EA e configuração de habilidades e missões), os servidores aplicativos e as ferramentas utilizadas.

São apresentados cenários de demonstração e testes no quarto capítulo, fazendo referência aos casos de utilização.

O documento termina no quinto capítulo com a conclusão e desenvolvimentos futuros.

Capítulo 2

Tecnologias

É possível desenvolver uma aplicação para a Web de diversas formas. Este capítulo destaca diferentes tecnologias consideradas para o projeto. Salienta-se que foi focado as que são gratuitas, de modo a reduzir o custo do projeto.

2.1 Linguagens

Relativamente a linguagens para aplicações Web, salientam-se: Java, PHP, JavaScript, Python, C Sharp, HTML e CSS.

Java é uma linguagem de programação orientada a objetos, independente da plataforma e possuidora de uma biblioteca extensa que facilita a cooperação com protocolos TCP/IP. Com Java [Gosling,], é possível criar páginas dinâmicas com interfaces interativas e que sejam compatíveis com grande variedade de dispositivos, independentemente de serem computadores ou telemóveis. Existem sites bastante conhecidos que utilizam Java na sua programação, nomeadamente LinkedIn [Hoffman, 2002] e Amazon [Bezos, 1994].

PHP [Lerdorf,] é uma linguagem interpretada livre, utilizada originalmente para o desenvolvimento de aplicações no lado do servidor, capaz de criar conteúdo dinâmico na *World Wide Web*. É usado em conjunto com outras linguagens no desenvolvimento Web para interagir diversas páginas com base de dados, permitindo rápidas atualizações dos conteúdos nos sites exibidos aos utilizadores. Alguns sites famosos que utilizam PHP são o Facebook [Zuckerberg, 2004] e a Wikipédia.

JavaScript [Eich,] é uma linguagem de programação interpretada estru-

turada, e uma das principais tecnologias da World Wide Web, sendo utilizado no desenvolvimento de aplicações para a web, principalmente na criação de sites mais interativos. Google [Page e Brin, 1998] e Youtube [Hurley et al., 2005] são exemplos de sites que utilizam esta linguagem.

Python [van Rossum, 1991] é uma linguagem de programação interpretada de script, imperativa, orientada a objetos e funcional. Em desenvolvimento Web, é focado especialmente no “back end”, de modo a controlar as conexões entre servidores e páginas, apresentando o conteúdo da melhor forma possível aos utilizadores. Alguns exemplos de sites que utilizam Python são Spotify [Ek e Lorentzon, 2006] e Netflix [Hastings e Randolph, 1997].

C Sharp [Microsoft,] é uma linguagem de programação multi-paradigma desenvolvida pela Microsoft como parte da plataforma *.NET*. A sua sintaxe orientada a objetos foi baseada no C++ mas inclui muitas influências de outras linguagens, nomeadamente *Object Pascal* [Wirth,] e, principalmente, *Java*. C Sharp tem duas principais vantagens: acesso a várias *frameworks* e bibliotecas *.NET* [Microsoft, 2002] que ajudam na eficiência e otimização de aplicações Web; compatibilidade com códigos de linguagens diferentes. Entre os sites que utilizam C Sharp, cito Ancestry [Sittner, 1983].

É difícil falar de desenvolvimento Web sem falar de HTML [W3C e WHATWG,]. É uma linguagem de marcação utilizada na estrutura de páginas para a Internet e que pode ser interpretada pelos navegadores. Na maioria das vezes tem associado CSS [Consortium, 1996] - linguagem utilizada para embelezar um documento HTML.

2.2 Sistemas de Gestão de Dados

No que toca a base de dados, existem vários concorrentes no mercado, separados em duas categorias: relacional e não relacional.

2.2.1 Base de Dados Relacional

Um sistema relacional tem uma forma de estrutura organizada em tabelas, armazenando e fornecendo acesso a pontos de dados que estão relacionados entre si. Cada linha da tabela é um registo com uma identificação única chamada chave primária. As colunas da tabela contêm os atributos dos dados, facilitando o estabelecimento das relações entre os dados. O mais

popular entre eles é o *MySQL*, mas também temos outras opções, como por exemplo *Microsoft SQL Server* e *PostgreSQL*.

O sistema de base de dados MySQL [Corporation, 1995] utiliza a linguagem *Structured Query Language (SQL)* e é um dos sistemas de gestão de dados mais populares da *Oracle Corporation*, com mais de 10 milhões de instalações pelo mundo. Algumas das suas características são: portabilidade (suporta qualquer plataforma atual), excelente desempenho, pouco exigente em termos de hardware, fácil manuseamento e controle transacional.

Microsoft SQL Server [Microsoft, 1989] tem como principal funcionalidade armazenar e recuperar dados solicitados por outras aplicações de software. Existem várias versões, destinadas a públicos e trabalho diferentes, variando de pequenas aplicações que leem e escrevem dados no mesmo computador, até milhões de utilizadores que acessam grandes quantidades de dados a partir da Internet ao mesmo tempo.

PostgreSQL [Group, 1996] foi desenvolvido como um projeto de código aberto. Hoje é um dos sistemas de bancos de dados mais avançado. Algumas das suas características são: facilidade de acesso, controlo transacional e suporte de consultas complexas.

2.2.2 Base de Dados Não Relacional

Um sistema de base de dados não relacional utiliza um modelo de armazenamento otimizado para os requisitos específicos do tipo de dados armazenados, por exemplo simples pares de chave-valor como documentos JSON [Crockford, 2002] e XML [W3C, 1998].

O mais popular designa-se MongoDB [Inc., 2009], software orientado a documentos, livre, de código aberto, multiplataforma e escrito na linguagem C++. Permite que as aplicações modelem informações mais naturalmente, pois os dados podem ser organizados em hierarquias complexas e continuarem a ser indexáveis e fáceis de aceder.

2.3 Servidores Aplicacionais

Num aplicação para a Web, é necessário um servidor que entenda os pedidos feitos no protocolo HTTP, de modo a ser possível comunicar com clientes

espalhados pela Internet. Para este projeto, destacam-se os seguintes servidores aplicativos: Tomcat, LiteSpeed e JBoss.

Tomcat [Foundation, 1999], desenvolvido por Apache Software Foundation, é um servidor Web Java que implementa as tecnologias Java Servlet e JavaServer Pages (JSP) [Microsystems, 1999]. Tem a capacidade de funcionar como um servidor Web, ou integrado a outro, por exemplo com Apache [McCool, 1995]. Como servidor Web HTTP, é puramente em Java e inclui ferramentas para configuração e gestão. Por padrão, o porto é no número 8080.

LiteSpeed Web Server (LSWS) [Wang, 2003] é um software de servidor Web proprietário. Desenvolvido por LiteSpeed Technologies, estima-se que tenha sido utilizado por 10% dos sites em julho de 2021, sendo o quarto servidor Web mais popular. Utiliza o mesmo formato de configuração do Apache HTTP Server e é compatível com a maioria das suas funcionalidades.

JBoss [Fleury,] é um servidor aplicativo de código fonte aberto baseado na plataforma JEE e implementado na linguagem de programação Java e é possível ser utilizado em qualquer sistema operacional que suporte esta linguagem. Desenvolvido por uma empresa de serviços chamada “JBoss Inc”, fundada por Marc Fleury, criador da primeira versão do JBoss. Em abril de 2006 foi adquirida pela *Red Hat* [Young e Ewing, 1993].

2.4 Ferramentas

De modo a ser possível desenvolver o projeto, é necessário escolher as ferramentas adequadas. Existem diversas opções, salientando-se Eclipse, NetBeans, IntelliJ e MySQL Workbench.

O Eclipse [IBM, 2001] é um excelente editor, muito utilizada no mercado, para desenvolvimento em Java, suportando também diversas outras linguagens se instalados os respetivos *plugins*, como por exemplo C, PHP, Python e Kotlin. Dentro das suas características podemos citar: o utilizador consegue incrementar as funcionalidades; criação de aplicações gráficas multiplataforma; possibilidade de utilizar diversos pacotes de desenvolvimento para criar aplicações em Java tanto para a Web como para o computador.

NetBeans [Stanek, 2000] é um editor gratuito e de código fonte aberto para desenvolvimento Java, no entanto é também extensível para outras lin-

guagens como PHP, Python e JavaScript. É possível utilizar nos principais sistemas operacionais (Windows, Linux e macOS) e possui suporte para a criação de interfaces para aplicações Web, computador e telemóvel.

IntelliJ [JetBrains, 2001] é um dos principais editores do mercado que teve um crescimento impressionante nos últimos anos. Tem como principais vantagens ser possível desenvolver em diversas tecnologias e possuir um bom assistente de código que auto-completa determinadas linhas, facilitando a criação de aplicações.

MySQL Workbench [Corporation, 2015] é uma das ferramentas que torna possível a implementação do sistema de base de dados relacional e permite ao utilizador ver o estado do servidor. Tendo sido desenvolvido pela Oracle (empresa por detrás do próprio MySQL), dá a certeza e tranquilidade ao utilizador que sempre será compatível com as funcionalidades mais recentes.

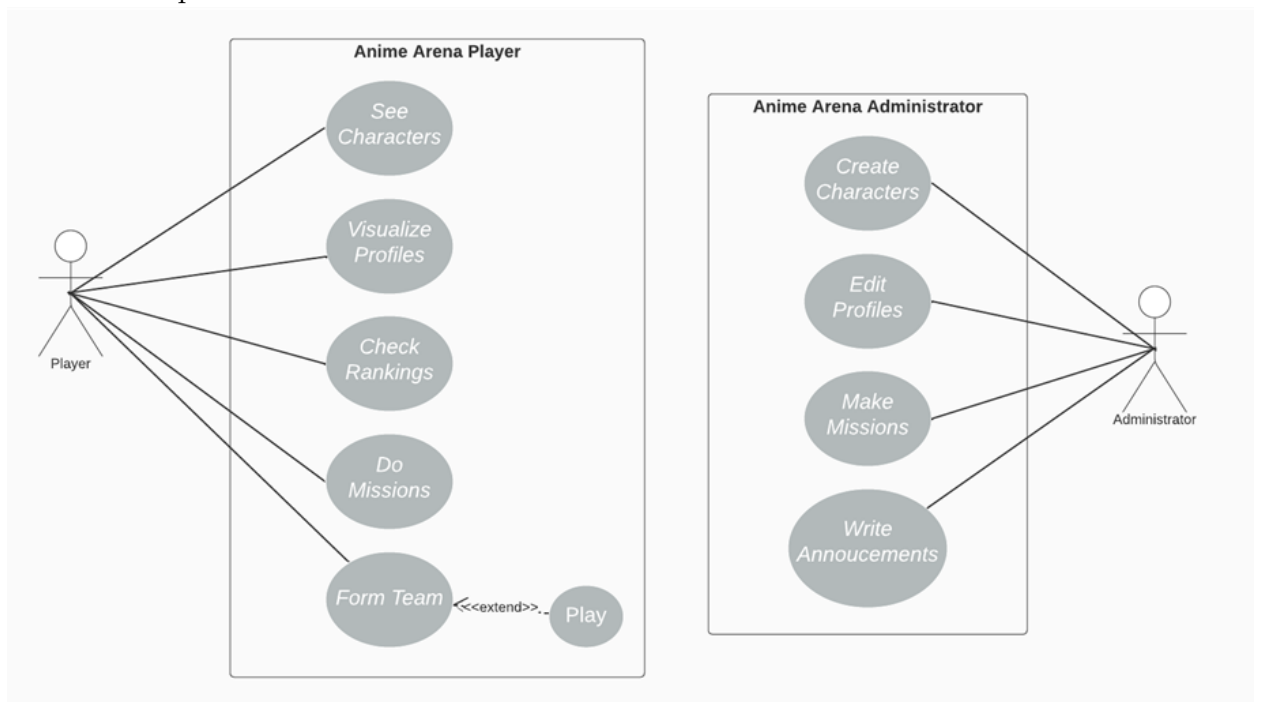
Capítulo 3

Projeto

Este capítulo explica a solução apresentada, detalhando as tecnologias utilizadas no desenvolvimento deste projeto.

3.1 Casos de Utilização

Neste projeto, existem dois atores. Apesar de todos os utilizadores serem *Players*, é possível serem também *Administrator*, tendo acesso a mais funcionalidades dos que não o são.



Um *Player* é o tipo de utilizador básico nos jogos, isto é, consegue visualizar o conteúdo disponível e jogar, neste caso contra um oponente. Ser *Administrator* mantém todas as características de um *Player*, no entanto este consegue também administrar o jogo, nomeadamente criar novo conteúdo como personagens, missões e anúncios.

3.2 Arquitetura

3.2.1 Definição de Personagens

3.2.2 Modelo EA

O modelo entidade-associação descreve os dados e a forma como estão relacionados entre si. De seguida será ilustrado o dito modelo dividido em diferentes partes, assim como as justificações que originaram o resultado final.

Começando por explicar as entidades possuidoras de atributos, contexto ao tema, temos os utilizadores na figura 3.1, missões em 3.2 e as habilidades das personagens em 3.3.

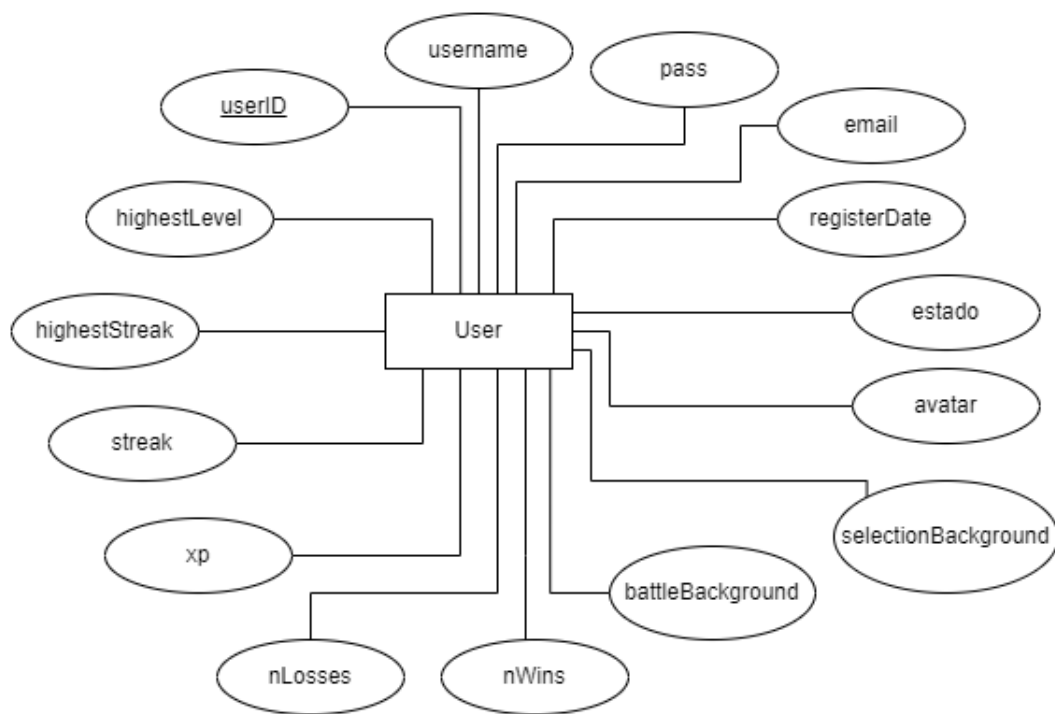


Figura 3.1: Entidade User

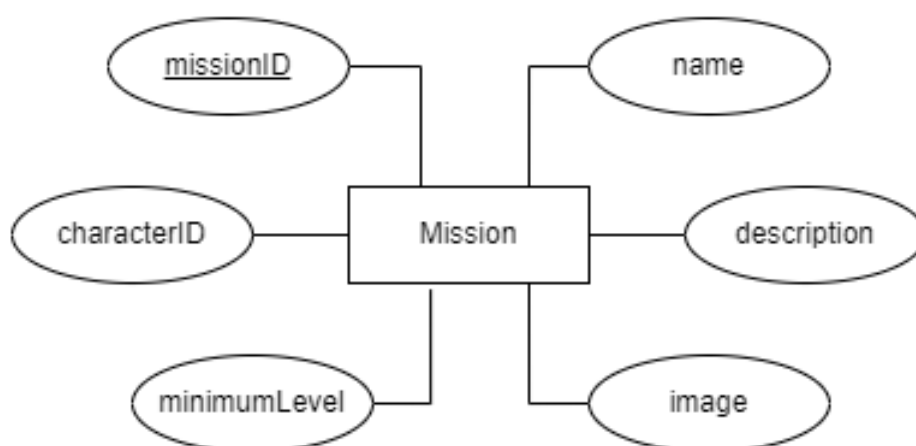


Figura 3.2: Entidade Mission

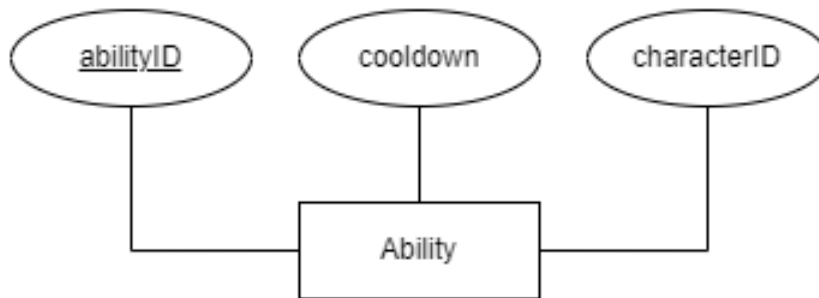


Figura 3.3: Entidade Ability

A entidade *User* trata toda a informação que diz respeito ao utilizador:

- userID - chave primária, número inteiro e serve como identificador do utilizador;
- username - corresponde ao nome de usuário, é único, tendo no máximo 16 caracteres;
- pass - senha do utilizador que todos têm de ter, e por motivos de segurança, TODO requer um mínimo de X caracteres, letras, números, etc TODO
- registerDate - data de registo do utilizador no site;
- xp - número inteiro que nunca pode ser negativo, diz respeito à experiência do utilizador. Ganhar e perder batalhas irá aumentar e diminuir este valor, respetivamente;
- avatar - um blob que representa uma imagem à escolha do utilizador;
- email - correio eletrónico associado à conta;
- nWins - número inteiro que nunca pode ser negativo. Começando com valor de zero, incrementa por um sempre que o jogador é vitorioso;
- nLosses - número inteiro que nunca desce de zero, aumenta sempre que o jogador é derrotado;
- selectionBackground - é um blob e corresponde á imagem de fundo na página de seleção da equipa;

- battleBackground - imagem de fundo da página de batalha (blob);
- streak - número inteiro sempre superior ou igual a zero, é o valor atualizado do número de vitórias consecutivas do jogador;
- highestStreak - número inteiro onde o valor mínimo que pode tomar é zero, indica o maior número de batalhas que o jogador venceu seguidas;
- estado - uma data que inicialmente é igual á *registerDate*. Esta pode ser alterada por um administrador, de forma a banir a conta em questão temporariamente ou permanentemente. Concretamente, o utilizador não consegue jogar se esta data disser respeito ao futuro.

3.2.3 Configuração de Habilidades

3.2.4 Configuração de Missões

3.3 Interface Humana

Capítulo 4

Demonstração

Capítulo 5

Conclusão

5.1 Sumário

5.2 Desenvolvimentos Futuros

Apêndice A

Um Detalhe Adicional

O “apêndice” utiliza-se para descrever aspectos que tendo sido desenvolvidos pelo autor constituem um complemento ao que já foi apresentado no corpo principal do documento.

Neste documento utilize o apêndice para explicar o processo usado na **gestão das versões** que foram sendo construídas ao longo do desenvolvimento do trabalho.

É especialmente importante explicar o objetivo de cada ramo (“branch”) definido no projeto (ou apenas dos ramos mais importantes) e indicar quais os ramos que participaram numa junção (“merge”).

É também importante explicar qual a arquitetura usada para interligar os vários repositórios (e.g., Git, GitHub, DropBox, GoogleDrive) que contêm as várias versões (e respetivos ramos) do projeto.

Notar a diferença essencial entre “apêndice” e “anexo”. O “apêndice” é um texto (ou documento) que descreve trabalho desenvolvido pelo autor (e.g., do relatório, monografia, tese). O “anexo” é um texto (ou documento) sobre trabalho que não foi desenvolvido pelo autor.

Para simplificar vamos apenas considerar a noção de “apêndice”. No entanto, pode sempre adicionar os anexos que entender como adequados.

Apêndice B

Outro Detalhe Adicional

Escrever aqui o detalhe adicional que melhor explique outro aspecto (diferente do que está no apêndice A) descrito no corpo principal do documento

...

Bibliografia

- [Bezos, 1994] Bezos, J. (1994). Amazon. <https://www.amazon.com/>.
- [Consortium, 1996] Consortium, W. W. W. (1996). Css. <https://www.w3schools.com/css/>.
- [Corporation, 1995] Corporation, O. (1995). Mysql. <https://www.mysql.com/>.
- [Corporation, 2015] Corporation, O. (2015). Mysql workbench. <https://www.mysql.com/products/workbench/>.
- [Crockford, 2002] Crockford, D. (2002). Json. <http://json.org/json-pt.html>.
- [Eich,] Eich, B. Javascript. <https://www.javascript.com/>.
- [Ek e Lorentzon, 2006] Ek, D. e Lorentzon, M. (2006). Spotify. <https://www.spotify.com/>.
- [Fleury,] Fleury, M. Jboss. <https://www.jboss.org/>.
- [Foundation, 1999] Foundation, A. S. (1999). Tomcat. <https://tomcat.apache.org/>.
- [Gosling,] Gosling, J. Java. <https://www.java.com/>.
- [Group, 1996] Group, P. G. D. (1996). Postgresql.
- [Hastings e Randolph, 1997] Hastings, R. e Randolph, M. (1997). Netflix. <https://www.netflix.com/>.
- [Hoffman, 2002] Hoffman, R. (2002). Linkedin. <https://www.linkedin.com/>.

- [Hurley et al., 2005] Hurley, C., Chen, S., e Karim, J. (2005). Youtube. <https://www.youtube.com/>.
- [IBM, 2001] IBM (2001). Eclipse. <https://www.eclipse.org/>.
- [Inc., 2009] Inc., M. (2009). MongoDB. <https://www.mongodb.com/>.
- [JetBrains, 2001] JetBrains (2001). IntelliJ. <https://www.jetbrains.com/>.
- [Kishimoto, 2002] Kishimoto, M. (2002). Naruto. <https://www.tv-tokyo.co.jp/anime/naruto2002/>.
- [Lerdorf,] Lerdorf, R. Php. <https://www.php.net/>.
- [McCool, 1995] McCool, R. (1995). Apache. <https://www.apache.org/>.
- [Microsoft,] Microsoft. C sharp. <https://docs.microsoft.com/en-us/dotnet/csharp/>.
- [Microsoft, 1989] Microsoft (1989). Microsoft sql server. <https://www.microsoft.com/en-us/sql-server/>.
- [Microsoft, 2002] Microsoft (2002). .net. <https://dotnet.microsoft.com/>.
- [Microsystems, 1999] Microsystems, S. (1999). Javasever pages.
- [Page e Brin, 1998] Page, L. e Brin, S. (1998). Google. <https://www.google.com/>.
- [Sittner, 1983] Sittner, J. (1983). Ancestry. <https://www.ancestry.com/>.
- [Stanek, 2000] Stanek, R. (2000). Netbeans. <https://netbeans.apache.org/>.
- [van Rossum, 1991] van Rossum, G. (1991). Python. <https://www.python.org/>.
- [W3C, 1998] W3C (1998). Xml. <https://www.w3schools.com/xml/>.
- [W3C e WHATWG,] W3C e WHATWG. Html. <https://html.com/>.
- [Wang, 2003] Wang, G. (2003). Litespeed. <https://www.litespeedtech.com/>.

-
- [Wespro, 2019] Wespro (2019). Naruto-arena. <http://naruto-arena.net/>.
- [Wirth,] Wirth, N. Object pascal. <https://objectpascalprogramming.com/>.
- [Young e Ewing, 1993] Young, B. e Ewing, M. (1993). Red hat. <https://www.redhat.com/>.
- [Zuckerberg, 2004] Zuckerberg, M. (2004). Facebook. <https://www.facebook.com/>.