



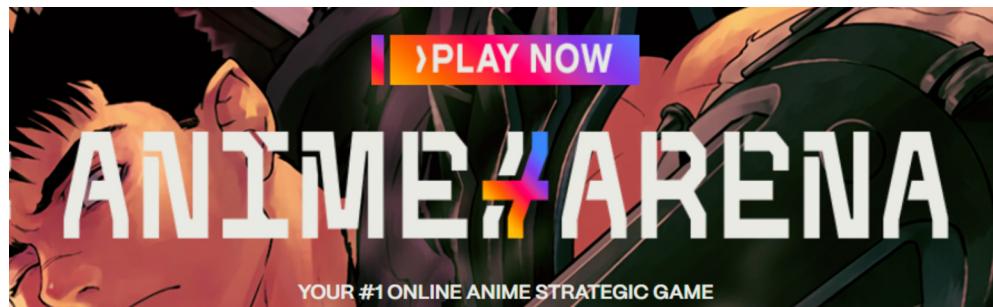
INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA  
DEPARTAMENTO DE ENGENHARIA ELETRÓNICA E DE  
TELECOMUNICAÇÕES E COMPUTADORES

---

LICENCIATURA EM ENGENHARIA INFORMÁTICA E MULTIMÉDIA  
UNIDADE CURRICULAR DE PROJETO

---

## Anime-Arena



*Autor*

---

Pedro Nuno Cruz Dias (45170)

---

*Orientador*

---

*Professor Doutor* Porfírio Pena Filipe

---

*setembro, 2022*



# Resumo

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

Neste âmbito, séries animadas japonesas (animes) possuem uma história rica e diversa. De forma a satisfazer o mercado, considera-se relevante a criação de um jogo que explore a evolução e adesão popular da animação japonesa, introduzindo novas personagens, habilidades e missões.

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

Esta plataforma é suportada pelo servidor aplicacional Tomcat e JavaServer Pages de forma a criar dinamicamente conteúdos baseados em HTML, CSS, JavaScript (AJAX) e XML, melhorando a experiência do utilizador e tornando possível aceder á case de dados suportada no sistema de gestão de dados MySQL.



# Abstract

The reissue of an old game called *Naruto-Arena* brought childhood memories to several people from all over the world. Having being part of the player's life for several years, they have shown interest for alternative versions of their favorite game.

In this scope, Japanese animation (anime) has a diverse and rich history. In order to satisfy the market, it's considered relevant the creation of a game that explores the evolution and popular rise of the Japanese animation, by introducing new characters, combinations, missions and rankings.

The goal of this project, denominated *Anime-Arena*, is to conceive and implement a multiplayer platform that is able to manage the interaction between a pair of players, who compete to improve their ranking and unlock characters.

This platform is supported by the applicational server Tomcat and JavaServer Pages in order to dynamically create content based on HTML, CSS, JavaScript (Ajax) and XML, improving the user experience and making it possible to access in the data management system MySQL.



# 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 responder a questões, como para dar a sua opinião e sugestões, permitindo-me realizar este projeto de forma natural.

Às pessoas que visionam lançar e manter o jogo, pelo debate de ideias, incansável ajuda na organização e planeamento, por toda a componente gráfica e pela criação de personagens, habilidades e missões. Obrigado “Betrayer”, “Clover”, “Aby” e “Alerane”.

Aos professores da unidade curricular de projeto, Paulo Trigo e Hélder Bastos, pelas críticas construtivas, pela disponibilidade em responder a questões durante o semestre e pelos materiais pedagógicos fornecidos.



*Dedico este projeto aos meus pais ...*

*... Ana e Pedro*

*... por tudo o que fizeram por mim.*



# Índice

<b>Resumo</b>	<b>i</b>
<b>Abstract</b>	<b>iii</b>
<b>Agradecimentos</b>	<b>v</b>
<b>Índice</b>	<b>ix</b>
<b>Lista de Tabelas</b>	<b>xi</b>
<b>Lista de Figuras</b>	<b>xiii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Enquadramento . . . . .	1
1.2 Motivação . . . . .	2
1.3 Objetivo . . . . .	2
1.4 Requisitos . . . . .	2
1.4.1 Funções do Sistema . . . . .	3
1.4.2 Atributos do Sistema . . . . .	5
1.4.3 Atributos e Funções do Sistema . . . . .	6
1.5 Contribuição . . . . .	6
1.6 Organização do Documento . . . . .	6
<b>2 Tecnologias</b>	<b>9</b>
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

<b>3 Projeto</b>	<b>13</b>
3.1 Casos de Utilização . . . . .	13
3.2 Arquitetura . . . . .	14
3.2.1 Camada de Dados . . . . .	15
3.2.1.1 Modelo Entidade Associação . . . . .	15
3.2.1.2 Esquema XML das Habilidades . . . . .	17
3.2.1.3 Esquema XML das Missões . . . . .	20
3.2.2 Camada de Aplicação . . . . .	21
3.2.2.1 Servlets . . . . .	21
3.2.3 Camada de Apresentação . . . . .	23
3.2.3.1 Interface Humana . . . . .	23
<b>4 Demonstração</b>	<b>25</b>
4.1 Exemplo Emparelhamento . . . . .	25
4.2 Exemplo Batalha . . . . .	28
4.2.0.1 Cenário da Batalha - Seleção Alvo Incorreto . . . . .	35
4.2.0.2 Cenário da Batalha - Desistência . . . . .	36
4.2.0.3 Cenário da Batalha - Desistência do Oponente . . . . .	38
4.2.0.4 Cenário da Batalha - Cancelamento de Habilidade . . . . .	39
4.2.0.5 Cenário da Batalha - Esgotamento de Tempo . . . . .	40
4.2.0.6 Cenário da Batalha - Vitória . . . . .	41
4.3 Exemplo Anúncios . . . . .	42
4.4 Exemplo Criar Personagem e Missão . . . . .	45
4.5 Exemplo Visualização de Personagens . . . . .	48
4.6 Exemplo Consultar Missões . . . . .	49
4.7 Exemplo Quadro de Honra . . . . .	50
<b>5 Conclusão</b>	<b>53</b>
5.1 Resumo . . . . .	53
5.2 Desenvolvimentos Futuros . . . . .	53
<b>A Controlo de Versões</b>	<b>55</b>
<b>Referências</b>	<b>57</b>

# **Lista de Tabelas**

1.1	Funções do Sistema . . . . .	3
1.2	Requisitos Funcionais . . . . .	4
1.3	Atributos do Sistema . . . . .	5
1.4	Requisitos Não-Funcionais . . . . .	5
1.5	Atributos e Funções do Sistema . . . . .	6
3.1	Atributos Entidade Mission . . . . .	16
3.2	Atributos Entidade Player . . . . .	16
3.3	Atributos Entidade Announcements . . . . .	17
3.4	Atributos Entidade Ability . . . . .	17
3.5	Atributos da Classe Character . . . . .	18
3.6	Métodos da Classe Character . . . . .	18
3.7	Extensão de Dados das Habilidades . . . . .	19
3.8	Atributos do Elemento Mission . . . . .	21
3.9	Atributos dos Elementos Character e Anime . . . . .	21
3.10	Servlets . . . . .	22
4.1	Cenário Principal de Emparelhamento . . . . .	26
4.2	Cenário Alternativo de Emparelhamento . . . . .	28
4.3	Cenário Principal da Batalha . . . . .	32
4.4	Cenário da Batalha - Seleção Alvo Incorreto . . . . .	35
4.5	Cenário da Batalha - Desistência . . . . .	36
4.6	Cenário da Batalha - Desistência do Oponente . . . . .	38
4.7	Cenário da Batalha - Cancelamento de Habilidade . . . . .	39
4.8	Cenário da Batalha - Esgotamento de Tempo . . . . .	41
4.9	Cenário da Batalha - Vitória . . . . .	42
4.10	Cenário Anúncios . . . . .	44
4.11	Cenário Criar Personagem e Missão . . . . .	46
4.12	Cenário Consultar Personagens . . . . .	48
4.13	Cenário Consultar Missões . . . . .	49
4.14	Cenário Quadro de Honra . . . . .	50



# **Lista de Figuras**

1.1	Fim do Naruto-Arena antigo . . . . .	1
3.1	Casos de Utilização . . . . .	13
3.2	Arquitetura da Plataforma . . . . .	14
3.3	Modelo Entidade Associação . . . . .	15
3.4	Diagrama XSD das Habilidades . . . . .	19
3.5	Habilidade de Asuna Yuuki . . . . .	20
3.6	Diagrama XSD das Missões . . . . .	20
3.7	Exemplo Requisitos de uma Missão . . . . .	21
3.8	Navegação entre Ecrãs . . . . .	23
4.1	Ambiente de Seleção . . . . .	25
4.2	Equipa Formada . . . . .	26
4.3	Emparelhamento . . . . .	27
4.4	Início da Batalha . . . . .	27
4.5	Sinalizações na Batalha . . . . .	28
4.6	Descrição do Ambiente de Batalha . . . . .	29
4.7	Classificações . . . . .	30
4.8	Turno do Oponente . . . . .	31
4.9	Fim do Turno do Oponente . . . . .	31
4.10	Alvos Possíveis da Habilidade . . . . .	33
4.11	Selecionar Alvo . . . . .	33
4.12	Fila de Habilidades . . . . .	34
4.13	Fim do Turno . . . . .	34
4.14	Alvos Possíveis . . . . .	35
4.15	Alvo Selecionado Incorreto . . . . .	36
4.16	Menu de Desistência . . . . .	37
4.17	Rendição do Jogador . . . . .	37
4.18	Vitória . . . . .	38
4.19	Execução de Habilidade . . . . .	39
4.20	Cancelamento da Habilidade . . . . .	40

4.21 Estado da Batalha . . . . .	40
4.22 Esgotamento de Tempo . . . . .	41
4.23 Aplicação do Golpe Final . . . . .	42
4.24 Página Inicial - Anúncios . . . . .	43
4.25 Visualizaçāo Anúncio . . . . .	43
4.26 Escrita do Anúncio . . . . .	44
4.27 Pré-visualização do Anúncio . . . . .	45
4.28 Definição da Personagem . . . . .	46
4.29 Definição de uma Habilidade . . . . .	47
4.30 Definição da Missão . . . . .	47
4.31 Todas as Personagens . . . . .	48
4.32 Informação da Personagem Zenitsu Agatsuma . . . . .	48
4.33 Missões . . . . .	49
4.34 Informação da Missão de Kushiki Byakuya . . . . .	50
4.35 Classificação . . . . .	51
4.36 Procura de um Jogador . . . . .	51
A.1 Emparelhamento . . . . .	55

# Capítulo 1

## Introdução

A reedição de um jogo antigo chamado *Naruto-Arena*, trouxe nostalgia a jogadores de todo o mundo. Tendo sido parte da infância de muitos jogadores durante 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*. Devido ao facto de, nesta altura, a série animada japonesa (anime) *Naruto* [Kishimoto, 2002] ser bastante popular entre as crianças e os jovens, muitos procuravam jogos relacionados com este seu interesse. Inevitavelmente encontraram este jogo e jogaram-no até 2018. Infelizmente, no final de 2015 e início de 2016, os criadores abandonaram o projeto. A inexistência de atualizações fez diminuir consideravelmente o número de jogadores e o projeto terminou definitivamente em 2018. No final de maio desse ano, quando se acedia à plataforma, apenas aparecia a imagem da figura 1.1.

*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 os interessados tristes, mas também esperança que os criadores voltassem e/ou alguém recriasse o seu jogo de infância que tanto gostavam. Este desejo concretizou-se em 2019 quando “Wespro” (uma pessoa que não fazia parte da equipa do jogo original) “ressuscitou” 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* trouxe alegria a muitas pessoas, mas também desejo por mais. Desde o abandono do jogo original até à atualidade, os jogadores têm expressado o seu desejo por versões alternativas. Integrar diversos animes origina novas personagens, habilidades e missões, exatamente o que muitos jogadores ansiavam.

O projeto denominado *Anime-Arena* visa satisfazer este mercado, englobando neste estilo de jogo vários animes.

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

## 1.3 Objetivo

O objetivo do projeto *Anime-Arena* é conceber e implementar uma plataforma multijogador, que seja 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 *Naruto-Arena* que são familiares e apelativas para os jogadores, mas também fazer uma interpretação pessoal e melhorar funcionalidades existentes, privilegiando aspetos de extensibilidade com a existência de vários animes, possibilitando manutenção contínua e configurando novas personagens, habilidades e missões.

## 1.4 Requisitos

Os requisitos descrevem as necessidades e propósitos do projeto e servem para reunir as informações necessárias para alcançar o objetivo.

No contexto deste projeto, um jogo constitui duas equipas e uma personagem possui quatro habilidades. Uma equipa agrupa um conjunto de três personagens e é definida pelo jogador, de forma a maximizar as suas hipóteses de vitória. O tempo

do jogo é distribuído por turnos atribuídos aos oponentes que devem fazer a sua jogada o mais rapidamente possível, calculando o dano e os recursos necessários das habilidades.

Durante uma batalha, cada jogador consegue efetuar, no máximo, três habilidades novas, correspondente às três personagens que constituem a sua equipa. No entanto, é necessário que existam as condições necessárias como a personagem não tiver sido derrotada nem estar paralisada, a habilidade não estar em “cooldown” e o jogador possuir as energias necessárias.

Estas energias são denominadas naturezas (“natures”) e existem cinco tipos: “Taijutsu” para habilidades físicas, “Heart” quando uma personagem necessita ser de determinada linhagem, “Energy” para ataques energéticos, “Spirit” quando se trata de uma habilidade mental/espiritual e “Random” que pode ser uma qualquer das outras naturezas.

O jogador receberá, no seu turno, uma natureza por cada personagem “viva” da sua equipa, com exceção do turno inicial, onde apenas começa com uma natureza aleatória.

Vencer e perder batalhas irão afetar a experiência, o nível, a classificação e o progresso das missões do jogador. Aumentar o seu nível dará acesso a novas missões que, se concluídas com sucesso, retornarão ao jogador a personagem correspondente.

#### 1.4.1 Funções do Sistema

As funções do sistema, organizadas com as categorias da tabela 1.1, são os requisitos funcionais e servem para representar o que o sistema é suposto fazer.

Categoría	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

*Anime-Arena* permite que os jogadores consigam visualizar os conteúdos existentes, nomeadamente personagens, habilidades, missões e anúncios. Navegando na plataforma, o jogador consegue encontrar outras informações, como as regras do jogo e as classificações dos jogadores.

Com a utilização das categorias da tabela 1.1, são apresentadas, na tabela 1.2, todas as funções deste sistema.

Ref	Função	Categoría
R1.1	Administração de perfis dos jogadores	Evidente
R1.1.1	Iniciar sessão	Evidente
R1.1.2	Identificação do perfil do jogador	Evidente
R1.1.3	Consulta e alteração de perfis dos jogadores	Evidente
R1.1.4	Terminar sessão	Evidente
R1.2	Gestão de conteúdos	Evidente
R1.2.1	Criação de personagens	Evidente
R1.2.2	Consultar habilidades das personagens	Evidente
R1.2.3	Composição de missões	Evidente
R1.2.4	Associação entre personagem e missão	Evidente
R1.2.5	Visualização de requisitos e progressos das missões	Evidente
R1.2.6	Introdução de novos temas	Evidente
R1.2.7	Escrita e leitura de anúncios	Evidente
R1.3	Alteração de configurações	Evidente
R1.3.1	Escolher tema	Evidente
R1.3.2	Alterar imagens	Evidente
R1.3.3	Mudar correio eletrónico	Evidente
R1.3.4	Modificar senha	Evidente
R1.4	Destaque de classificações	Evidente
R1.5	Explicação das regras do jogo	Evidente
R1.6	Jogar	Evidente
R1.6.1	Formação de uma equipa	Evidente
R1.6.2	Emparelhamento de jogadores	Evidente
R1.6.3	Sincronização entre os oponentes	Evidente
R1.6.4	Existência de múltiplos jogos em simultâneo	Invisível
R1.6.5	Batalhar contra outros jogadores	Evidente
R1.6.6	Aplicação de habilidades em tempo real	Evidente
R1.6.7	Cancelamento de habilidades	Evidente
R1.6.8	Finalização do turno	Evidente
R1.6.9	Desistência da batalha	Evidente
R1.6.10	Reprodução de sons	Adorno
R1.6.11	Identificação e atualização do resultado do jogo	Evidente

Tabela 1.2: Requisitos Funcionais

É necessário que a plataforma identifique o perfil dos utilizadores, pois existem administradores que possuem funcionalidades adicionais aos jogadores, como a criação de conteúdos.

Todos os jogadores conseguem alterar as suas configurações, mais concretamente

a imagem de perfil, a senha, o correio eletrónico, o tema e as imagens de fundo do ambiente de seleção e batalha.

Jogar é o requisito funcional mais importante e o seu bom funcionamento depende do emparelhamento de jogadores e da sincronização entre os oponentes.

Para entrar em batalha, o jogador necessita formar uma equipa. Durante o combate, existem diversas ações que o jogador consegue executar, como a aplicação de uma habilidade, desistência, finalização do seu turno e cancelamento de uma habilidade.

### 1.4.2 Atributos do Sistema

Os atributos do sistema são os requisitos não-funcionais e representam as características ou dimensões do sistema. Estes foram definidos como sendo obrigatório ou desejável (tabela 1.3).

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

Tabela 1.3: Atributos do Sistema

Assim, na tabela 1.4 encontram-se os atributos do sistema.

Atributo	Detalhe / Restrição de Fronteira	Categoría
Facilidade de Utilização	Adaptação da interface a diferentes tamanhos de ecrã	Desejável
	Fácil aprendizagem das mecânicas do jogo	Desejável
	Utilização de símbolos para pessoas daltónicas	Desejável
	Existência de vários idiomas	Desejável
	Animações em tempo real que facilitem a compreensão do jogo	Desejável
	Alteração do volume	Desejável
	Mecanismos de procura	Desejável
Interacção Pessoa-Máquina	Utilização de rato e de teclado	Obrigatório

Tabela 1.4: Requisitos Não-Funcionais

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

### 1.4.3 Atributos e Funções do Sistema

As relações entre funções e atributos do sistema são importantes para descrever os atributos relacionados com funções específicas.

Atributo	Funções
Facilidade de Utilização	R1.1, R1.1.1, R1.1.3, R1.1.4, R1.2, R1.2.1, R1.2.2, R1.2.3, R1.2.4, R1.2.5, R1.2.6, R1.2.7, R1.3, R1.3.1, R1.3.2, R1.3.3, R1.3.4, R1.4, R1.5, R1.6, R1.6.1, R1.6.5, R1.6.6, R1.6.7, R1.6.8, R1.6.9
Interação Pessoa-Máquina	R1.1, R1.1.1, R1.1.3, R1.1.4, R1.2, R1.2.1, R1.2.2, R1.2.3, R1.2.4, R1.2.5, R1.2.6, R1.2.7, R1.3, R1.3.1, R1.3.2, R1.3.3, R1.3.4, R1.4, R1.5, R1.6, R1.6.1, R1.6.5, R1.6.6, R1.6.7, R1.6.8, R1.6.9

Tabela 1.5: Atributos e Funções do Sistema

Na tabela 1.5 é possível observar quais as funções do sistema associadas aos atributos “Facilidade de Utilização” e “Interação Pessoa-Máquina”.

## 1.5 Contribuição

A contribuição principal do *Anime-Arena* assenta na satisfação do mercado que deseja versões alternativas do *Naruto-Arena*, introduzindo novas personagens, habilidades e missões de diferentes animes.

Adicionalmente, no contexto do projeto, foi estabelecida uma comunidade, constituída atualmente por cinco membros, para partilha de ideias e recursos do jogo que visa a sua manutenção, contribuindo na identificação de requisitos inovadores e organização do desenvolvimento.

## 1.6 Organização do Documento

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

O capítulo atual começa por enquadrar o projeto, explicando a origem e as características do jogo implementado. Fala-se da motivação do projeto, nomeadamente a atualidade e o interesse do público. São referidos, concretamente, o objetivo e os requisitos. Por último, na contribuição são toados sobre o resultado final.

No segundo capítulo são descritas as tecnologias consideradas relevantes para a elaboração deste projeto, nomeadamente linguagens, sistemas para gestão de dados, servidores aplicacionais e ferramentas.

O terceiro capítulo apresenta a implementação do projeto, os casos de utilização, a arquitetura, os servidores 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 o resumo e desenvolvimentos futuros.



# Capítulo 2

## Tecnologias

Este capítulo destaca as tecnologias estudadas no âmbito do projeto, salientando as que são tendencialmente gratuitas.

### 2.1 Linguagens

A linguagem de programação Java [Gosling, 1996] é orientada a objetos, independente da plataforma e possuidora de uma biblioteca extensa que facilita a cooperação com protocolos TCP/IP. Com Java, é possível criar páginas dinâmicas com interfaces interativas e que sejam compatíveis com grande variedade de dispositivos, independentemente da dimensão dos ecrãs. Na aplicação Web, Java aparece em Java ServerPages (JSP) [Microsystems, 1999]. Existem sites bastante conhecidos que utilizam Java na sua programação, nomeadamente o LinkedIn [Hoffman, 2002] e a Amazon [Bezos, 1994].

A linguagem de programação PHP [Lerdorf, 1995] é interpretada sem custos, utilizada originalmente para o desenvolvimento de aplicações, capaz de gerar conteúdo dinâmico na *World Wide Web*, tipicamente abreviada por Web. Alguns sites famosos que utilizam PHP são a Wikipédia [Wales e Sanger, 2001] e o Facebook [Zuckerberg, 2004].

A linguagem de programação JavaScript [Eich, 1995] é interpretada estruturada, sendo utilizado no desenvolvimento de aplicações para a Web, principalmente na criação de sites mais interativos. O Google [Page e Brin, 1998] e o Youtube [Hurley et al., 2005] são exemplos de sites que utilizam esta linguagem.

A linguagem de programação Python [van Rossum, 1991] é interpretada, imperativa, orientada a objetos e funcional. Esta linguagem dispõe várias *frameworks* como o *web2py* [web2py, 2007] e o *CherryPy* [CherryPy, 2008]. Alguns exemplos de sites que utilizam Python são a Netflix [Hastings e Randolph, 1997] e o Spotify [Ek e Lorentzon, 2006].

A linguagem de programação C Sharp [Microsoft, 2001] é multi-paradigma desenvolvida pela empresa Microsoft [Gates e Allen, 1975] como parte da plataforma .NET [Microsoft, 2002]. A sua sintaxe orientada a objetos foi baseada no C++ [Stroustrup, 1979], mas inclui muitas influências de outras linguagens, nomeadamente de Object Pascal [Wirth, 1986] e principalmente de Java. C Sharp tem como vantagem o acesso a várias *frameworks* e bibliotecas .NET que ajudam na eficiência e otimização de aplicações Web, mantendo compatibilidade com códigos de outras linguagens. Entre os sites que utilizam C Sharp, cita-se a Ancestry [Sittner, 1983].

É difícil falar de desenvolvimento Web sem falar de “HyperText Markup Language” (HTML) [W3C e WHATWG, 1993]. É uma linguagem baseada em etiquetas, utilizada na estrutura de páginas para a Web que são interpretadas pelos navegadores. Na maioria das vezes tem associado “Cascading Style Sheets” (CSS) [Consortium, 1996] - linguagem utilizada para definir o estilo de uma página HTML.

## 2.2 Sistemas de Gestão de Dados

Relativamente aos sistemas de gestão de dados, existem vários concorrentes no mercado, organizados 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 fontes 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 são atributos que contêm valores. O sistema de gestão de dados mais popular é o MySQL [Corporation, 1995], mas existem também outras opções, como por exemplo o Microsoft SQL Server [Microsoft, 1989] e o PostgreSQL [Group, 1996].

O sistema de gestão de base de dados MySQL possui determinadas características, como portabilidade (suporta qualquer plataforma atual), excelente desempenho, pouco exigente em termos de *hardware*, fácil manuseamento e controle transacional.

A principal funcionalidade do Microsoft SQL Server é armazenar e recuperar dados solicitados por outras aplicações. Existem várias versões, destinadas a públicos e utilizações diferentes, variando de pequenas aplicações que leem e escrevem dados localmente, até milhões de utilizadores que acedem, ao mesmo tempo, a grandes quantidades de dados a partir da Internet.

O PostgreSQL foi desenvolvido como um projeto de código aberto. Atualmente é um dos sistemas de base de dados mais completo. Algumas das suas características

são: facilidade de acesso, controlo transacional e suporte de interrogações 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] ou XML [W3C, 1998].

O mais popular designa-se MongoDB [Inc., 2009], é orientado a documentos, de código aberto, multiplataforma e escrito na linguagem C++. Permite às aplicações modelarem dados mais naturalmente, organizando-os em hierarquias complexas.

## 2.3 Servidores Aplicacionais

Numa aplicação para a Web, é necessário um servidor que interprete os pedidos feitos no protocolo “Hypertext Transfer Protocol” (HTTP) e com capacidade para executar código, de modo a ser possível comunicar com clientes distribuídos pela Internet. Neste contexto, destacam-se os seguintes servidores aplicacionais: Tomcat [Foundation, 1999], LiteSpeed [Wang, 2003] e JBoss [Fleury, 1999].

O Tomcat, desenvolvido por Apache Software Foundation [Coar et al., 1999], é um servidor Web que suporta as tecnologias Java Servlet e JSP. 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.

O LiteSpeed Web Server (LSWS) 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.

O JBoss é um servidor aplicacional de código fonte aberto baseado na plataforma JEE, 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 adotar as ferramentas adequadas. Existem diversas opções, salientando-se: Eclipse [IBM, 2001], NetBeans [Stanek, 2000], MySQL Workbench [Corporation, 2015] e IntelliJ [JetBrains, 2001a].

O Eclipse é um excelente editor, muito utilizado no mercado, para desenvolvimento em Java, suportando também outras linguagens se forem instalados os respetivos *plugins*, como por exemplo C [Ritchie, 1972], PHP, Python e Kotlin [JetBrains, 2001b]. No âmbito das características, refere-se a possibilidade de incrementar as funcionalidades e criar aplicações gráficas.

O NetBeans é um editor gratuito e de código fonte aberto para desenvolvimento Java, sendo também extensível para outras linguagens como por exemplo: PHP, Python e JavaScript. É possível utilizar nos principais sistemas operativos (Windows [Microsoft, 1985], Linux [Torvalds, 1991] e macOS [Inc., 2001]) e possui suporte para a criação de interfaces para aplicações Web, “Graphical User Interface” (GUI) e móveis.

O IntelliJ 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 a escrita, facilitando a criação de aplicações.

O MySQL Workbench é uma das ferramentas que torna possível a implementação de sistemas de base de dados relacionais e permite ao utilizador administrar o estado do servidor. Tendo sido mantido pela Oracle [Ellison et al., 1977] (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 descreve a solução apresentada, detalhando as abordagens consideradas no projeto *Anime-Arena*.

### 3.1 Casos de Utilização

A figura 3.1 ilustra os casos de utilização deste projeto.

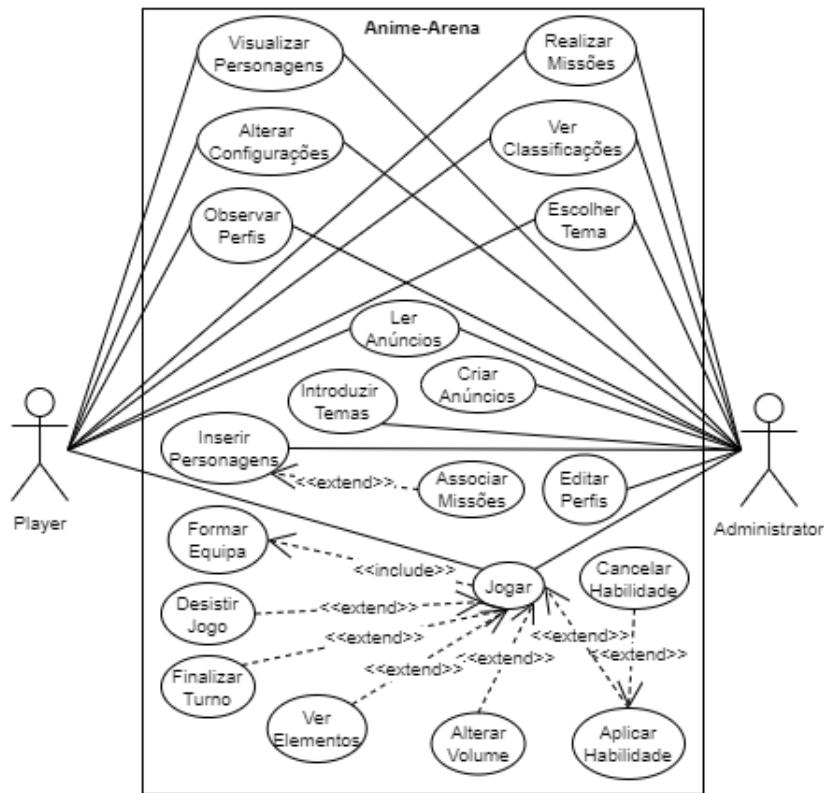


Figura 3.1: Casos de Utilização

Nos casos de utilização, são referidos dois atores: *Player* e *Administrator*.

O jogador (*Player*) é o tipo de utilizador comum dos jogos, isto é, consegue visualizar o conteúdo disponível, alterar configurações e jogar, neste caso contra um oponente.

O administrador (*Administrator*) é também um jogador que, adicionalmente, consegue administrar o jogo, nomeadamente criar conteúdo como personagens, temas, missões e anúncios.

De modo a ser possível iniciar um jogo, é necessário definir a sua equipa. Durante a batalha, o jogador pode realizar diversas ações como: aplicar uma habilidade (esta ação pode ser cancelada), ver elementos (informação dos jogadores, personagens ou habilidades presentes na batalha), alterar o volume do jogo e finalizar o turno.

## 3.2 Arquitetura

A arquitetura da plataforma está organizada em três camadas, nomeadamente: a interface para interação humana multi-dispositivo, o servidor aplicacional Tomcat e o sistema de gestão de dados suportado em MySQL. Esta arquitetura encontra-se ilustrada na figura 3.2.



Figura 3.2: Arquitetura da Plataforma

A camada de apresentação recorre a JavaServer Pages para criar conteúdos dinâmicos baseados em HTML. Para embelezar o aspeto visual, foi utilizado CSS e uma *framework* chamada Bootstrap [Otto e Thornton, 2011]. O uso de JavaScript combinado com a tecnologia “Asynchronous JavaScript And XML” (AJAX) permitiu melhorar a experiência do utilizador e enviar pedidos ao servidor, assim como recebe-los e mostrar a resposta ao utilizador.

Na camada de aplicação utilizou-se Java para implementar o processamento que envolve a comunicação com a base de dados.

Relativamente à camada de dados, foi definido o modelo de dados físicos, usando SQL [Chamberlin e Boyce, 1974], obtido a partir do modelo entidade associação. Também foram utilizados documentos XML para manter características das habilidades e missões.

### 3.2.1 Camada de Dados

#### 3.2.1.1 Modelo Entidade Associação

A figura 3.3 ilustra o modelo entidade associação do projeto *Anime-Arena*.

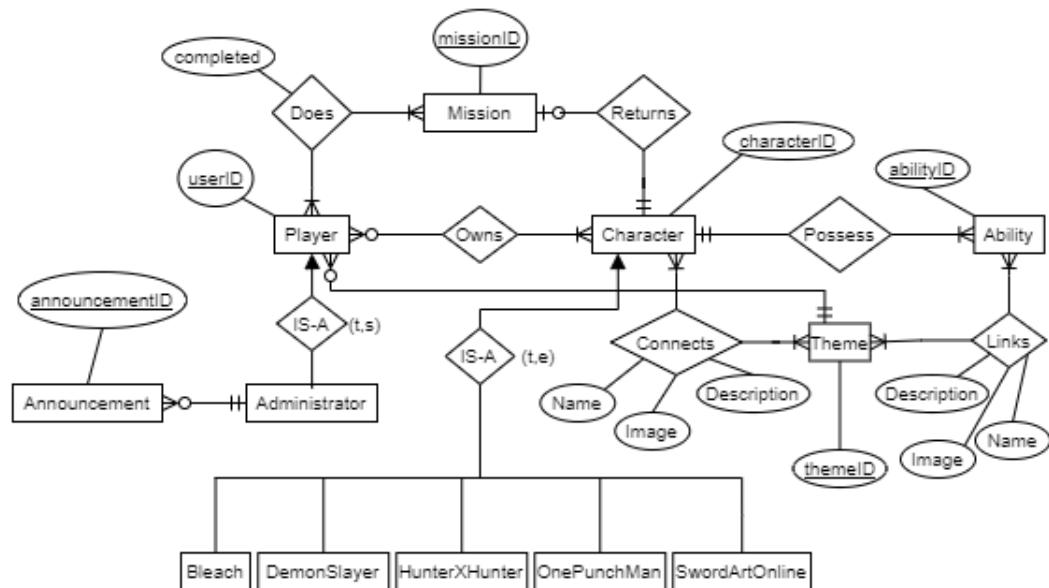


Figura 3.3: Modelo Entidade Associação

Cada personagem corresponde obrigatoriamente a um único anime. Neste modelo, as personagens podem ser dos seguintes tipos: *Bleach* [Kubo, 2001], *Demon Slayer* [Gotoge, 2016], *Hunter X Hunter* [Togashi, 1998], *One Punch-Man* [Murata, 2009] ou *Sword Art Online* [Kawahara, 2009].

Uma personagem (“Character”) pode ou não estar associada a uma missão (“Mission”), no entanto, cada missão retorna sempre uma personagem. A tabela 3.1 mostra os atributos da entidade “Mission”.

Atributo	Significado	Domínio
missionID	identificador	inteiro não negativo
name	nome	caracteres(32)
description	descrição	caracteres(5000)
image	imagem	tipo PNG até 16 MB
minimumLevel	nível mínimo necessário para a realizar	inteiro superior a 0

Tabela 3.1: Atributos Entidade Mission

As entidades “Player” e “Mission” estabelecem uma relação de “muitos-para-muitos” e a associação possui um atributo booleano, denominado *completed*, que indica se a missão foi concluída pelo jogador.

Um jogador (“Player”) possui, no mínimo, três personagens, no entanto uma personagem pode pertencer a nenhum ou muitos jogadores. A tabela 3.2 ilustra os atributos da entidade “Player”.

Atributo	Significado	Domínio
userID	identificador	inteiro não negativo
username	nome	caracteres(16)
pass	codificação da senha	caracteres(8) e padrão de pelo menos 1 letra maiúscula, minúscula e dígi
registerDate	data de registo na plataforma de jogo	data passada
xp	experiência	inteiro não negativo
avatar	imagem de perfil	tipo PNG até 64 KB
email	correio eletrónico	caracteres(128)
nWins	contador de vitórias em modo <i>ladder</i>	inteiro não negativo
nLosses	contador de derrotas em modo <i>ladder</i>	inteiro não negativo
selectionBackground	imagem de fundo do ecrã de seleção	tipo PNG até 16 MB
battleBackground	imagem de fundo do ecrã de batalha	tipo PNG até 16 MB
streak	contador de vitórias ou derrotas consecutivos em modo <i>ladder</i>	inteiro
highestStreak	maior número de vitórias consecutivas em modo <i>ladder</i>	inteiro não negativo
estado	indicador se pode jogar	booleano
isActive	indicador se o correio eletrónico foi verificado	booleano

Tabela 3.2: Atributos Entidade Player

Um jogador pode ser do tipo administrador (“Administrator”) e este, pode criar anúncios (“Announcement”) que estão representados na tabela 3.3.

Atributo	Significado	Domínio
announcementID	identificador	inteiro não negativo
title	título	caracteres(16)
category	categoria	caracteres(16)
smallDesc	resumo	caracteres(100)
fullDesc	texto completo	caracteres(5000)
postDate	data de publicação	data passada

Tabela 3.3: Atributos Entidade Announcements

Uma personagem possui quatro habilidades (“Ability”) e cada habilidade pertence a uma única personagem. A entidade “Ability” encontra-se ilustrada na tabela 3.4.

Atributo	Significado	Domínio
abilityID	identificador	inteiro não negativo
cooldown	tempo de espera até ser possível aplicada novamente	inteiro não negativo

Tabela 3.4: Atributos Entidade Ability

Neste modelo, existem temas representados pela entidade “Theme” que estão associados às personagens e habilidades em relações de “muitos-para-muitos”. Esta associação faz as personagens e habilidades possuírem um nome (“name”), imagem (“image”) e descrição (“description”) consoante o tema escolhido pelo jogador.

### 3.2.1.2 Esquema XML das Habilidades

A definição de habilidades é estruturada usando um esquema XML. As habilidades são definidas por chaves atributo-valor associadas a métodos da classe “Character”.

Todos os atributos da classe “Character” são privados e encontram-se descritos na tabela 3.5.

Os métodos que podem ser referidos na definição de habilidades são apresentados na tabela 3.6, com exceção dos respetivos “getters” e “setters”.

Atributo	Significado	Domínio
id	identificador	inteiro não negativo
hp	vida	inteiro entre 0 e 100, inclusive.
isStunned	indica se está paralisada	booleano
stunnedDuration	duração de turnos que está paralisada	inteiro não negativo
isInvul	indica se está invulnerável	booleano
invulDuration	duração da invulnerabilidade	inteiro não negativo
permanentDamageIncrease	dano adicional permanente	inteiro
temporaryDamageIncrease	dano adicional temporário	inteiro
dr	redução de dano	inteiro
natureGain	naturezas adicionais ganhas	inteiro
natureLoss	naturezas adicionais perdidas	inteiro

Tabela 3.5: Atributos da Classe Character

Adicionalmente, esta classe possui as habilidades pertencentes à personagem.

Método	Significado
Character	Construtor da classe que recebe o identificador como argumento
applyAbility	Método público sem retorno, que recebe uma habilidade, personagem e variável booleana como argumento
	Aplica a habilidade à personagem que recebeu nos argumentos
doDamage	Método privado que retorna um número inteiro e recebe uma habilidade e personagem como argumento
	Chamado por applyAbility e serve para calcular o dano
getAbilityDamageEnemyHP	Método privado que retorna um número inteiro e recebe uma habilidade e personagem como argumento.
	Chamado por doDamage se a habilidade fizer dano adicional consoante a vida da personagem recebida como argumento.
getAbilityDamageSelfHP	Método privado que retorna um número inteiro e recebe uma habilidade como argumento
	Chamado por doDamage se a habilidade fizer dano adicional consoante a vida da própria personagem

Tabela 3.6: Métodos da Classe Character

Quando um administrador introduz uma personagem, adicionalmente ao nome, descrição e imagem, necessita especificar as quatro habilidades que possui. Todos estes dados serão gravados num ficheiro XML, validados através de XSD e lidos pelo Java no início do combate.

A figura 3.4 apresenta o esquema XML que é usado para as habilidades.

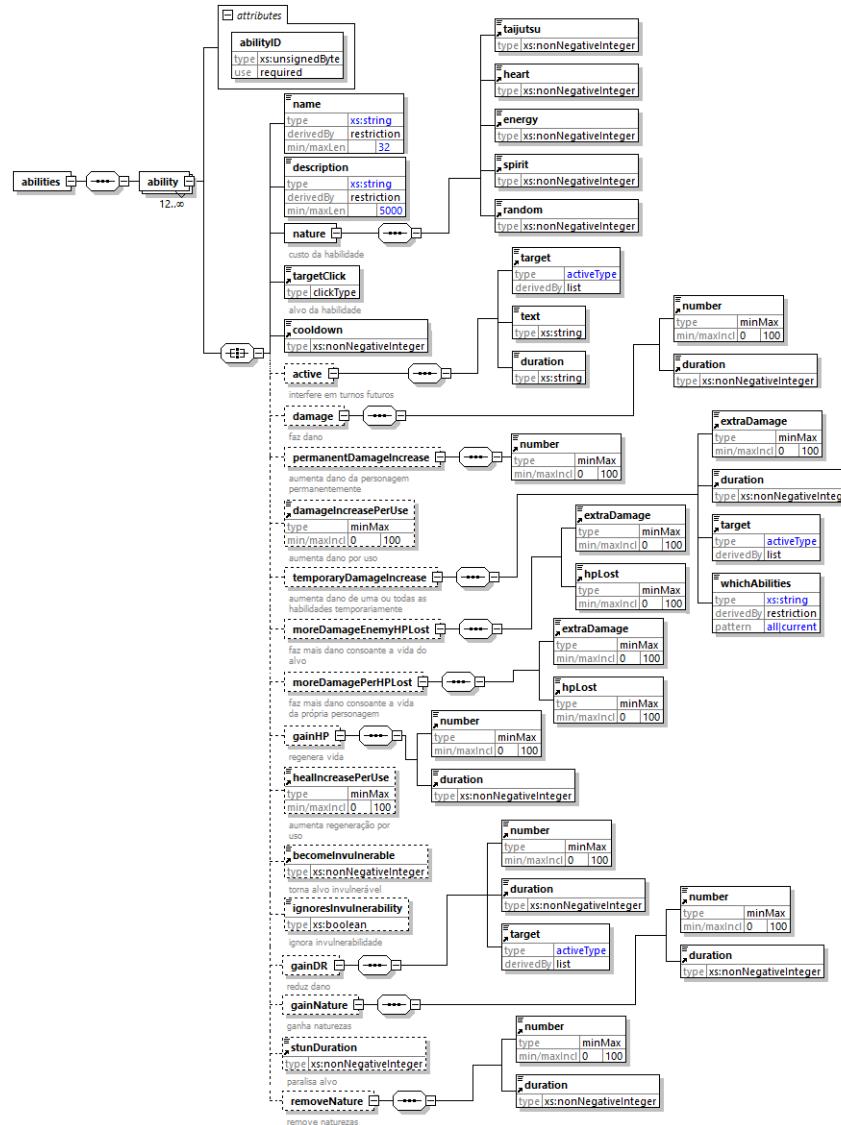


Figura 3.4: Diagrama XSD das Habilidades

Foram criados três “types”, para restringir os elementos de acordo com a aplicação, como se pode observar na tabela 3.7.

Type	Significado
activeType	é um enumerado: “self”, “enemy” ou “ally”
clickType	é um enumerado: “self”, “enemy”, “allEnemies”, “ally”, “allyORself” ou “allTeam”
minMax	número inteiro entre 0 e 100, inclusive

Tabela 3.7: Extensão de Dados das Habilidades

A título de exemplo, encontra-se ilustrado na figura 3.5 uma habilidade da personagem *Asuna Yuuki*.

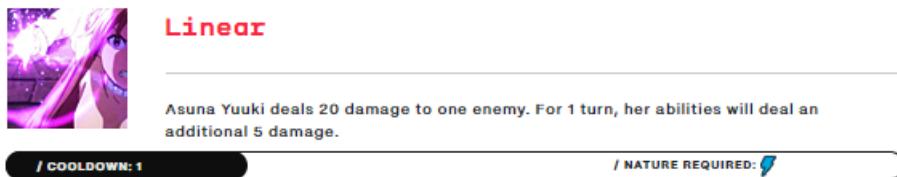


Figura 3.5: Habilidade de Asuna Yuuki

Nesta figura foi utilizado o tema por omissão em inglês.

### 3.2.1.3 Esquema XML das Missões

A figura 3.6 representa, num diagrama XML, os elementos e os atributos associados às missões.

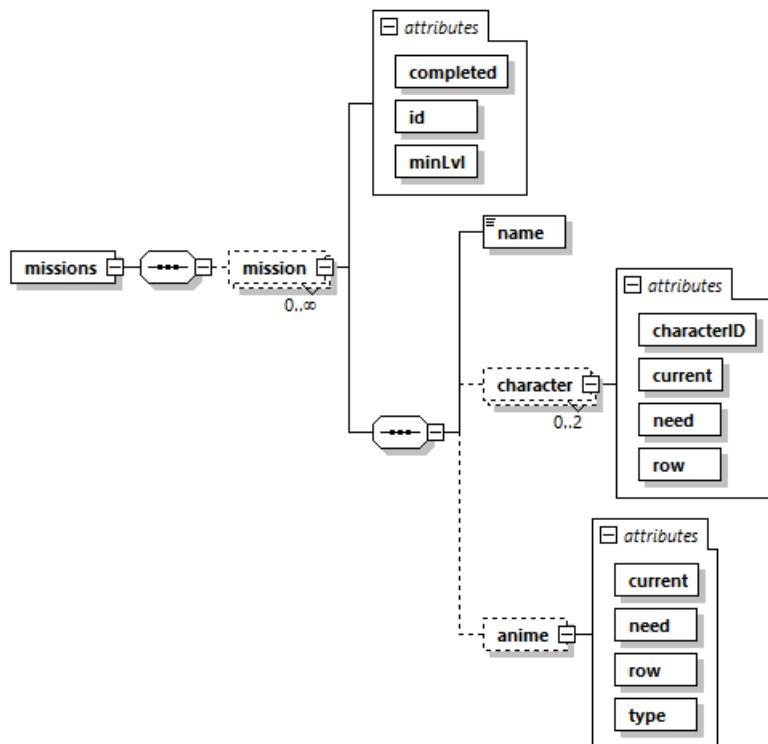


Figura 3.6: Diagrama XSD das Missões

O progresso dos jogadores é registado em documentos XML. Quando um jogador é vitorioso numa batalha em *Ladder* ou *Quick*, consoante os personagens utilizados, é verificado no seu documento se progrediu numa ou várias missões e atualiza-o. Em caso de derrota, o mesmo acontece nas missões onde é necessário vencer jogos consecutivos, fazendo o jogador ter que recomeçar.

Cada missão, no ficheiro XML, é um elemento denominado *mission* que possui três atributos, ilustrados na tabela 3.8.

Atributo	Significado	Domínio
completed	índice se a missão foi concluída	booleano
id	identificador	inteiro não negativo
minLvl	nível mínimo que o jogador necessita	inteiro maior do que 0

Tabela 3.8: Atributos do Elemento Mission

Dentro deste elemento *mission* existem os elementos do nome e requisitos da missão. Estes ocorrem entre uma a três vezes, e o elemento é chamado de *character* quando é necessário vencer com uma personagem em concreto, e *anime* para vitórias com qualquer personagem desse anime.

O elemento *character* contém um atributo denominado *characterID*, correspondente à personagem, enquanto que *anime* possui o atributo *type*, que indica o anime. Adicionalmente, ambos possuem três atributos iguais, representados na tabela 1.5.

Atributo	Significado	Domínio
current	progresso do jogador	inteiro não negativo
need	número de jogos necessário vencer	inteiro não negativo
row	indica se as vitórias necessitam ser consecutivas	booleano

Tabela 3.9: Atributos dos Elementos Character e Anime

A figura 3.7 mostra os requisitos de uma missão existente neste projeto.

Win 3 battles in a row with Kurasaki Ichigo (0/3)

Win 5 battles with Kushiki Rukia (0/5)

Win 20 battles with any character from Bleach (0/20)

Figura 3.7: Exemplo Requisitos de uma Missão

É possível observar que o jogador necessita vencer três batalhas seguidas com *Kurasaki Ichigo*, cinco jogos com *Kushiki Rukia* e vinte partidas com qualquer personagem do anime *Bleach*.

### 3.2.2 Camada de Aplicação

#### 3.2.2.1 Servlets

De modo a interligar a aplicação do lado do cliente e da base de dados, foi utilizado JavaScript (Ajax) para enviar pedidos ao servidor e receber as respostas.

<b>Servlet</b>	<b>Propósito</b>
Authentication	Validar os dados inseridos pelo utilizador (nome de utilizador e senha)
	Inserir utilizador na base de dados
	Enviar email de verificação
Announcement	Mostrar os quatro anúncios mais recentes
	Visualizar informação de um anúncio específico
ViewProfile	Criar a página de perfil do jogador
UpdateProfile	Editar perfil do jogador
ViewMissionInfo	Mostrar a informação completa de uma missão
ViewCharacterInfo	Mostrar informação de uma personagem
ViewBack	Inserir imagens de fundo
FindOpponent	Emparelhar jogador contra um adversário
	Cancelar procura de oponente
InGame	Redirecionar jogador para a batalha
	Adquirir e libertar semáforo
	Detetar resultado
	Remover jogo
ViewRanking	Inserir imagem da classificação dos jogadores nas suas personagens
Timer	Atualizar temporizador do turno
AbilityActions	Obter e redefinir as <i>natures</i> do jogador
	Atualizar a jogabilidade das habilidades
	Mostrar e guardar habilidades a ser aplicadas
	Animar personagens
	Calcular naturezas do tipo <i>random</i>
	Aplicar habilidade(s)
	Cancelar habilidade
Language	Selecionar tema e idioma
Settings	Alterar imagens, senha e/ou correio eletrónico
ViewAvatar	Mostrar imagem de perfil do utilizador
ViewCharacter	Ilustrar imagem da personagem
ViewAbility	Visualizar imagem da habilidade
VerifyEmail	Verificar correio eletrónico

Tabela 3.10: Servlets

Na tabela 3.10 encontra-se descrito as *Servlets* criadas e o seu propósito.

### 3.2.3 Camada de Apresentação

#### 3.2.3.1 Interface Humana

De modo ao utilizador conseguir navegar pelo site, foram construídas as páginas da figura 3.8. Adicionalmente, existem os ecrãs de entrada e registo do utilizador. Estes não se encontram na figura, pois a navegação mostrada assume que o utilizador esteja autenticado.

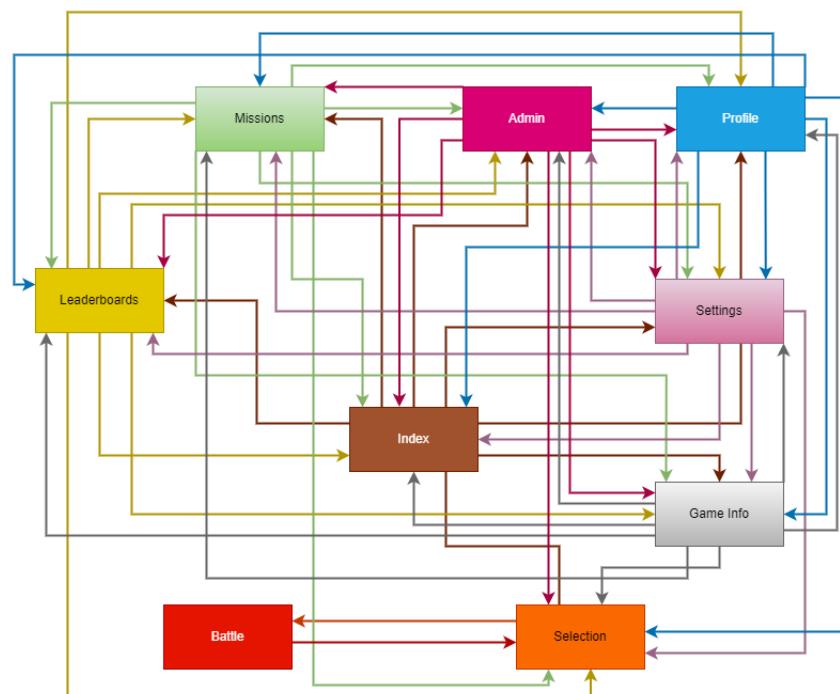


Figura 3.8: Navegação entre Ecrãs

A página **Login** é a porta de entrada para a plataforma, onde o utilizador insere as suas credenciais (nome e senha) para se autenticar.

No cenário onde o utilizador não possui conta na plataforma, ele pode registar-se na página **Register** preenchendo os campos necessários, nomeadamente o nome de utilizador, a senha e o correio eletrónico.

**Index** é a página inicial, onde os utilizadores conseguem visualizar a informação reduzida dos últimos quatro anúncios que, quando clicados, apresentam a sua informação completa.

De modo a incrementar a competitividade do jogo, está disponível em **Leaderboards** as melhores classificações dos jogadores. Se for desejável encontrar um jogador em particular, existe um mecanismo de procura que o permite fazer.

Os jogadores conseguem visualizar as missões existentes, na página **Missions**. Para facilitar a sua procura, estão separadas por *animes* e é possível omitir missões

completas, que não são possíveis aceder devido ao nível mínimo e/ou pesquisar por nome. Selecionar uma missão torna visível toda a sua informação.

O ecrã **Profile** mostra o perfil do jogador selecionado. No menu de navegação existe um “atalho” para o próprio perfil, no entanto é possível observar o de qualquer jogador.

**Settings** é uma página criada dinamicamente consoante a informação que o utilizador deseje ver e/ou alterar, nomeadamente a imagem de perfil, senha, correio eletrónico ou imagens de fundo.

Toda a informação necessária para compreender o jogo encontra-se em **Game Info**. Neste ecrã, é descrito as regras, o funcionamento e a informação de todas as personagens do jogo. É criada uma página dinamicamente que ilustra as habilidades da personagem selecionada.

**Selection** é o local onde os jogadores formam a sua equipa e seleccionam o modo de jogo: “ladder” afeta a classificação e as missões; “quick” impacta apenas o progresso das missões; “private” é um jogo amigável contra um jogador específico que não conta para classificações nem missões.

O ecrã **Battle** é onde o jogador utiliza a sua equipa para defrontar um oponente em tempo real, com o objetivo de derrotar todas as personagens do seu adversário. Existem efeitos sonoros em situações específicas, nomeadamente no início da batalha, quando uma personagem ou habilidade é selecionada, no fim do turno, quando uma personagem é derrotada e no fim da batalha (difere consoante o resultado).

Apenas administradores têm visibilidade e acesso a **Admin**. Nesta secção, é possível criar temas, personagens, missões e anúncios.

Salienta-se que a plataforma adapta-se à dimensão do dispositivo do utilizador e existem dois idiomas disponíveis: português e inglês.

# Capítulo 4

## Demonstração

Neste capítulo são apresentados exemplos que demonstram o funcionamento da plataforma implementada, visualizando sete cenários de exemplo.

### 4.1 Exemplo Emparelhamento

O primeiro passo para entrar em batalha é formar uma equipa. Entrando na página de seleção, os jogadores encontram um ecrã semelhante ao da figura 4.1.

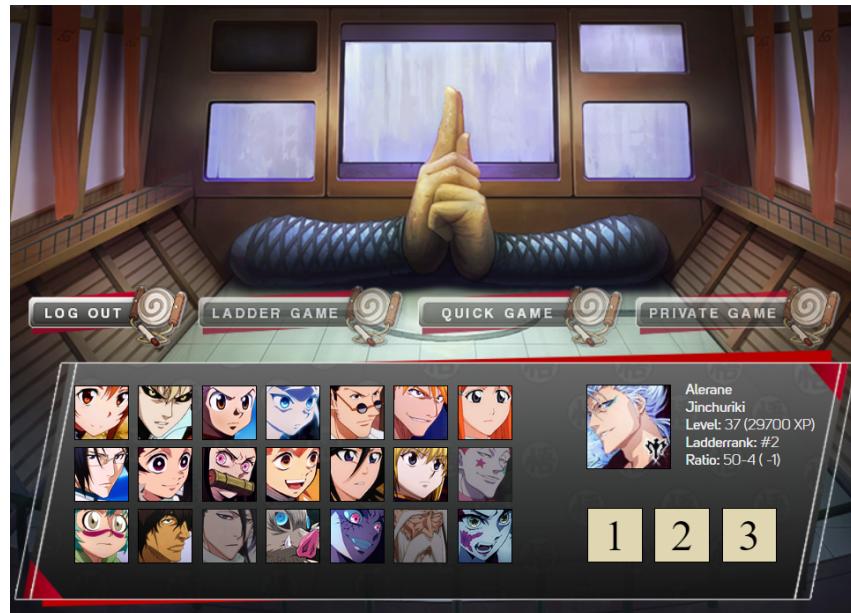


Figura 4.1: Ambiente de Seleção

No canto inferior direito, por cima dos números 1, 2 e 3 (onde se constitui a equipa), encontra-se a informação do utilizador.

A seleção de qualquer personagem fará aparecer um menu na parte superior do

ecrã com a sua informação. Existem personagens com opacidade reduzida, significando que o jogador ainda não completou as respetivas missões, não sendo possível utilizá-las na sua equipa.

Observa-se que os botões dos modos de jogo (“ladder”, “quick” e “private”) não são clicáveis, pois o jogador não definiu a sua equipa. A figura 4.2 mostra o ambiente de seleção quando o jogador escolhe as três personagens.

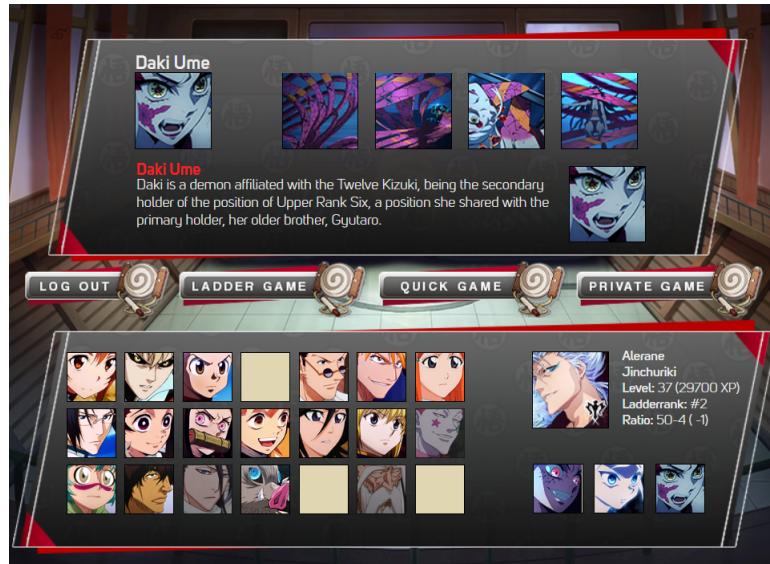


Figura 4.2: Equipa Formada

Nesta situação, é possível iniciar o cenário principal de emparelhamento, representado na tabela 4.1. Para simplificar, é assumido que o jogador pretende jogar o modo “quick”.

Cenário Principal (fluxo típico de eventos)	
Ação do Ator	Resposta do Sistema
1 O caso de utilização inicia quando o jogador forma uma equipa	
2 Jogador prime o botão “Quick Game”	
	3 Servidor coloca o jogador na lista de espera
	4 Servidor dá indicação ao jogador
	5 Servidor procura adversário
	6 Servidor encontra oponente e informa o jogador
	7 Servidor redireciona o jogador para o modo de batalha
8 Jogador entra em jogo	

Tabela 4.1: Cenário Principal de Emparelhamento

A figura 4.3 mostra o ecrã entre os números de sequência 2 e 6, inclusive.

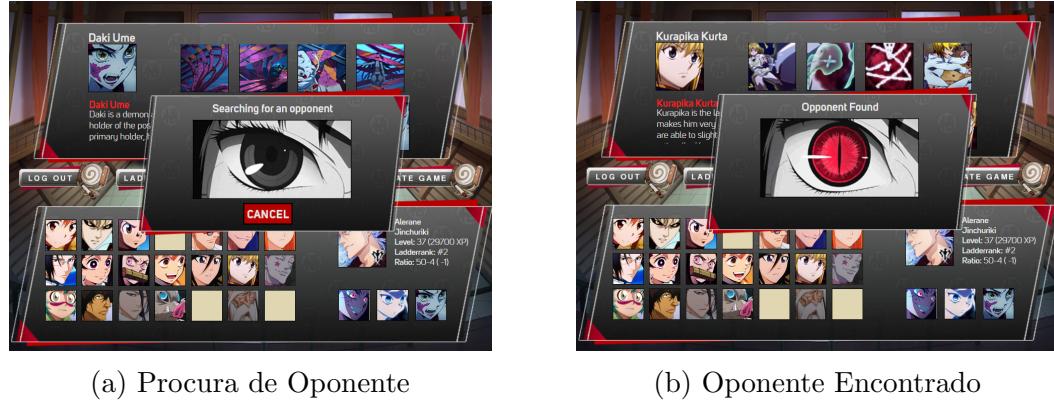
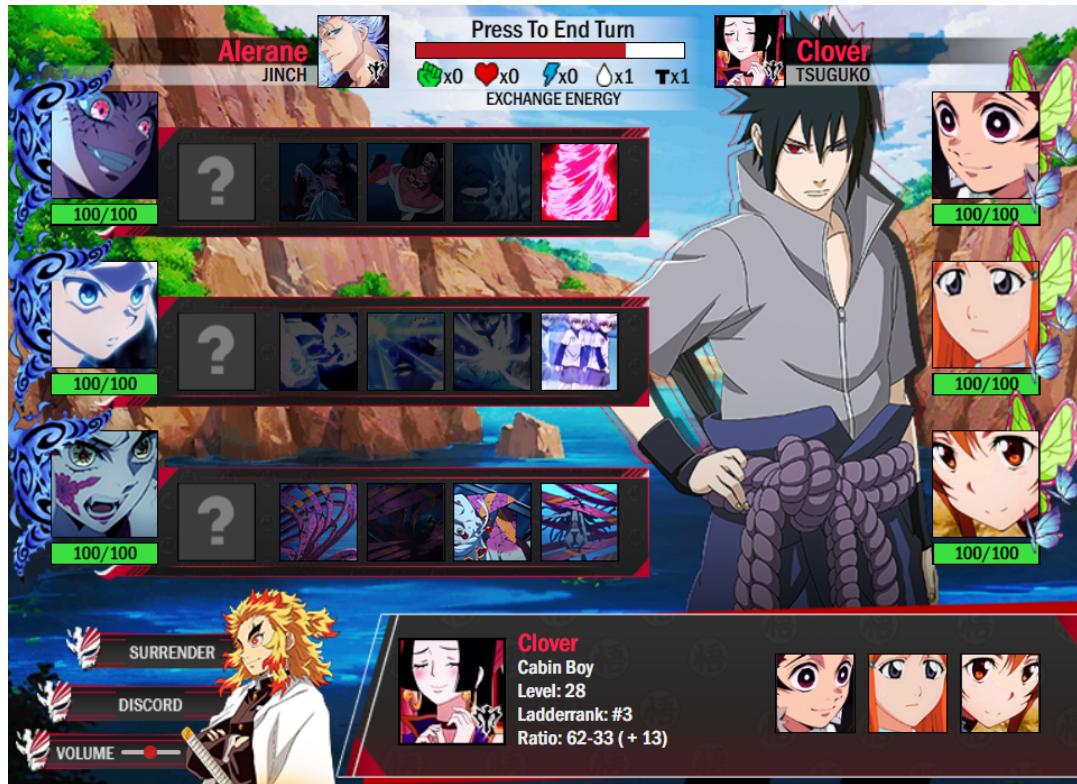


Figura 4.3: Emparelhamento

Premir o botão “Quick Game” dá uma indicação ao jogador que foi corretamente colocado na fila de espera e que o servidor está à procura de um oponente. Quando os jogadores são emparelhados com sucesso, ambos recebem um aviso e são redirecionados para o modo de batalha, como se observa na figura 4.4



encontrado, é possível cancelar esta ação, premindo o botão “Cancel” da figura 4.3, como se encontra no cenário alternativo da tabela 4.2.

Cenário Alternativo 1	
Número da Sequência	Alternativa
6	Jogador cancela a procura Servidor remove jogador da fila de espera. 7 e 8 sem efeito.

Tabela 4.2: Cenário Alternativo de Emparelhamento

Nesta situação, o ecrã do jogador voltará ao da figura 4.3.

## 4.2 Exemplo Batalha

Numa batalha, existem diversos fatores a ter em consideração, nomeadamente as naturezas que o jogador possui, o custo e “cooldown” das habilidades, tempo do turno e ações do adversário. Para facilitar a compreensão do ecrã, em “Game Info” foi descrito graficamente este ambiente, como mostram as figuras 4.5 e 4.6.

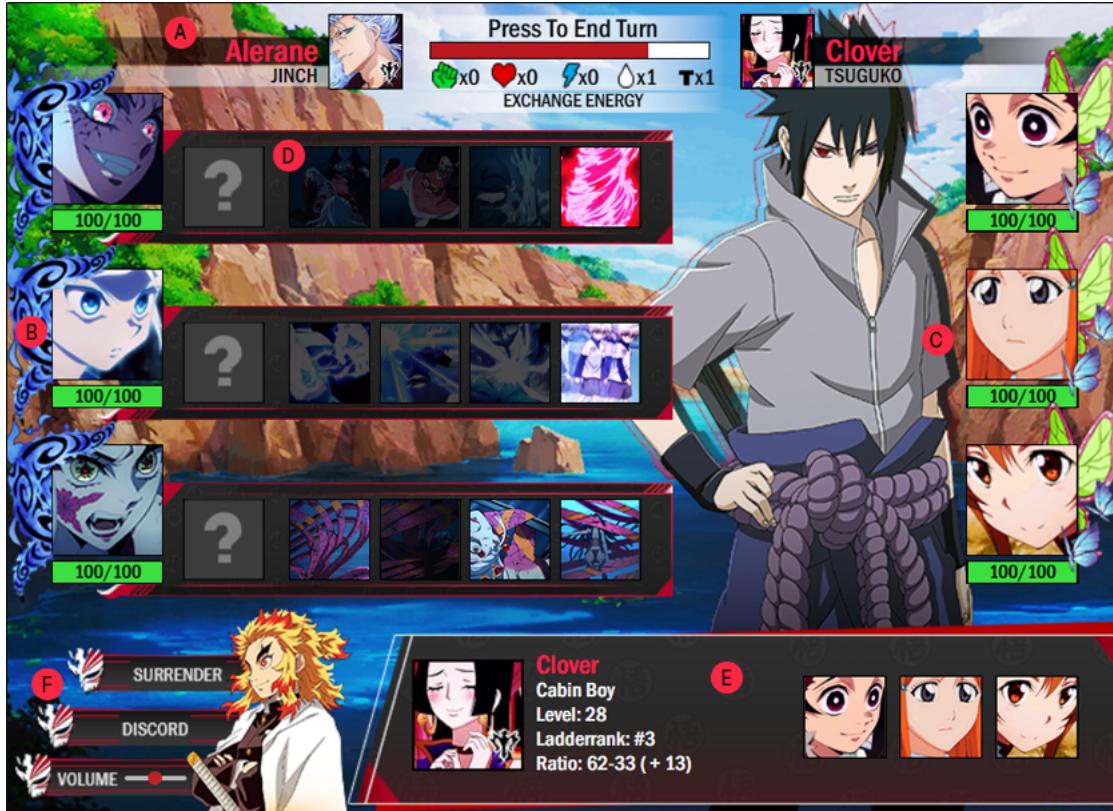


Figura 4.5: Sinalizações na Batalha

A	B	C
<p>In the top of the page, you can see your avatar, username and ranking on the left part, just like the opponent you are facing. In between them, it shows you the amount of natures you currently have, as well as how you can end your turn. You can exchange 5 natures for 1 specific as a last resort.</p>	<p>On the left part of the screen, your characters are displayed, along with a visual representation of your ranking. Each ranking has their own image, and will change as often as your ranking.</p>	<p>Your opponent's team that you must defeat in order to be victorious. Just like in B, the image in the back of the characters represents your opponent's ranking.</p>
D	E	F
<p>Where you can apply your character's abilities. If they have lower opacity, it means you cannot use them that turn. This can happen for several reasons like lack of natures, be on cooldown or the character is stunned. Each character can only do 1 new ability per turn. If you do decide to apply a ability, it will be displayed on the "?" holder. If you wish to undo this action, you can do it by double clicking on the current ability.</p>	<p>The information of the clicked element. This will change according to what you wish to see, be it a player, character or ability.</p>	<p>With this buttons, you can forfeit the match, visit our official discord and change the volume of the sound effects of the game.</p>

Figura 4.6: Descrição do Ambiente de Batalha

Na parte superior do ecrã (ponto A) encontram-se os participantes da batalha e as naturezas do jogador. É indicado que o turno lhe pertence e como o termina.

Em relação ao ponto B, as imagens e respetivas classificações diferem consoante a classificação do jogador. Na figura 4.7 é ilustrado a representação destas dez classificações, assim como o respetivo nível mínimo necessário.

Em C encontra-se a equipa do oponente que o jogador visa derrotar para se tornar vitorioso. Uma personagem é derrotada quando a sua vida (representada pela barra inferior) é menor ou igual a zero. As personagens iniciam o jogo com a vida máxima (100).

As habilidades que cada personagem possui encontram-se no ponto D. Menor opacidade significa que não é possível aplicar a habilidade.

Em E observa-se a informação do elemento selecionado. No início da batalha é mostrado a informação do seu oponente.

O jogador consegue desistir ou alterar o volume do som, no ponto F.



(a) Nível 1 - Cabin Boy



(b) Nível 6 - Demon Slayer



(c) Nível 11 - Ryoka



(d) Nível 16 - Missing-nin



(e) Nível 21 - Lieutenant



(f) Nível 26 - Tsuguko



(g) Nível 31 - Cipher Pol



(h) Nível 36 - Jinch



(i) Nível 41 - Admiral



(j) Nível 46 - Hashira

Figura 4.7: Classificações

Para esta demonstração será utilizado o ambiente da figura 4.8.

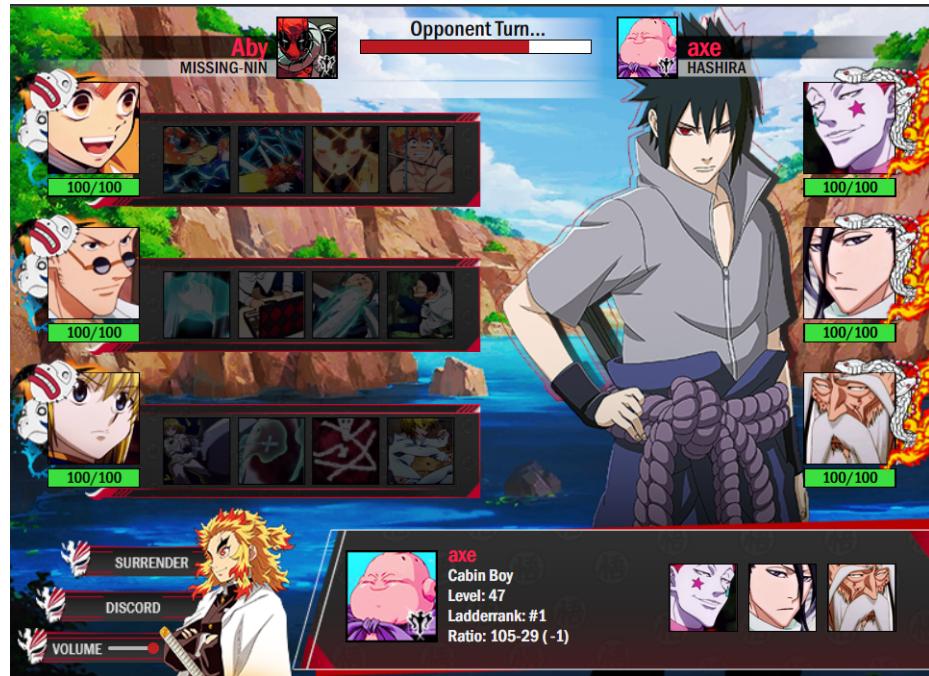


Figura 4.8: Turno do Oponente

Observa-se que o turno inicial foi atribuído ao oponente. A figura 4.9 ilustra o fim e alternância do turno para o jogador.

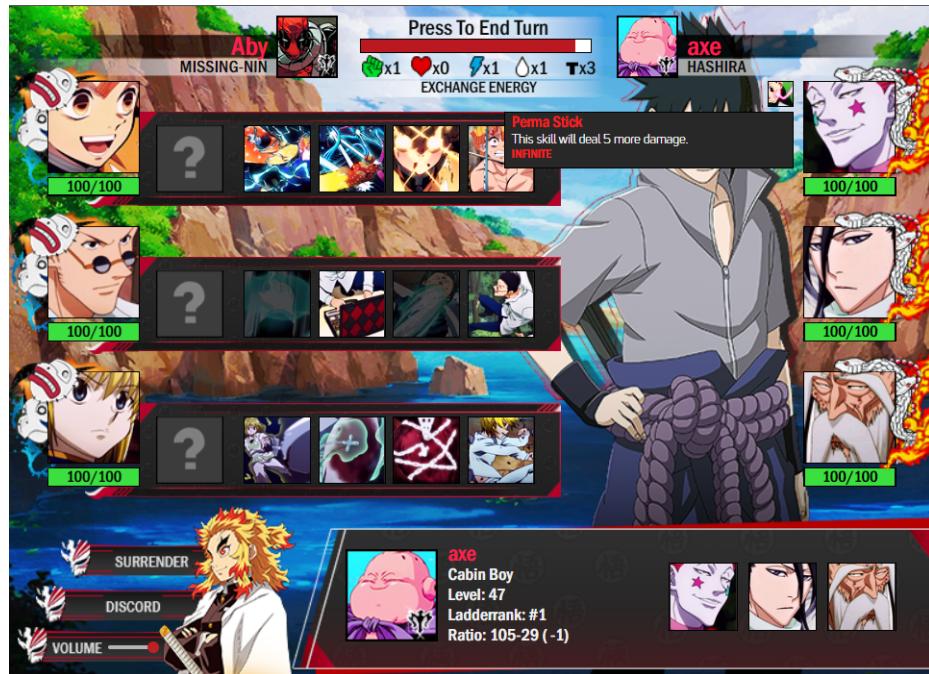


Figura 4.9: Fim do Turno do Oponente

Nesta situação, o jogador não consegue aplicar duas habilidades, porque não possui as naturezas suficientes.

Habilidades que afetam turnos futuros são inseridos no ambiente de batalha e é possível observar o seu efeito colocando o “rato” em cima da habilidade.

Agora que o jogador possui o turno, é possível inicial o cenário principal da tabela 4.3.

Cenário Principal (fluxo típico de eventos)			
Ação do Ator		Resposta do Sistema	
1	O caso de utilização inicia quando o turno da batalha pertence ao jogador		
2	Jogador prime uma habilidade possível de ser executada		
		3	Servidor insere uma animação na(s) personagem(ns) que são possíveis de aplicar a habilidade
4	Jogador clica numa personagem animada		
		5	Servidor desativa as habilidades da personagem que efetuou a habilidade
		6	Servidor desativa as animações existentes nas personagens
		7	Servidor atualiza as <i>natures</i> do jogador e as habilidades do jogo
		8	Servidor insere no alvo a habilidade a ser executada
9	Jogador pressiona para terminar o turno		
		10	Servidor mostra ao jogador a fila de habilidades
11	Jogador confirma o fim do seu turno		
		12	Servidor aplica a habilidade
		13	Servidor termina o turno do jogador
14	Jogador espera o fim do turno do seu oponente		

Tabela 4.3: Cenário Principal da Batalha

Quando o jogador prime uma habilidade possível de ser executada, é inserido uma animação na(s) personagem(ns) que são possíveis de aplicar a habilidade. Estes números de sequência (2 e 3) são ilustrados na figura 4.10.



Figura 4.10: Alvos Possíveis da Habilidade

Observa-se que a animação foi inserida em todas as personagens do oponente, pois trata-se de uma habilidade ofensiva. Selecionar a primeira personagem do oponente realiza o número de sequência 4, como mostra a figura 4.11



Figura 4.11: Selecionar Alvo

É possível ver que os números 5, 6, 7 e 8 da sequência foram executados, nomeadamente: a personagem que efetuou a habilidade teve as suas habilidades desativas, é indicado o alvo e a habilidade, as animações desapareceram e as naturezas do jogador foram atualizadas.

Pressionar “Press To End Turn” torna visível o ecrã da figura 4.12.

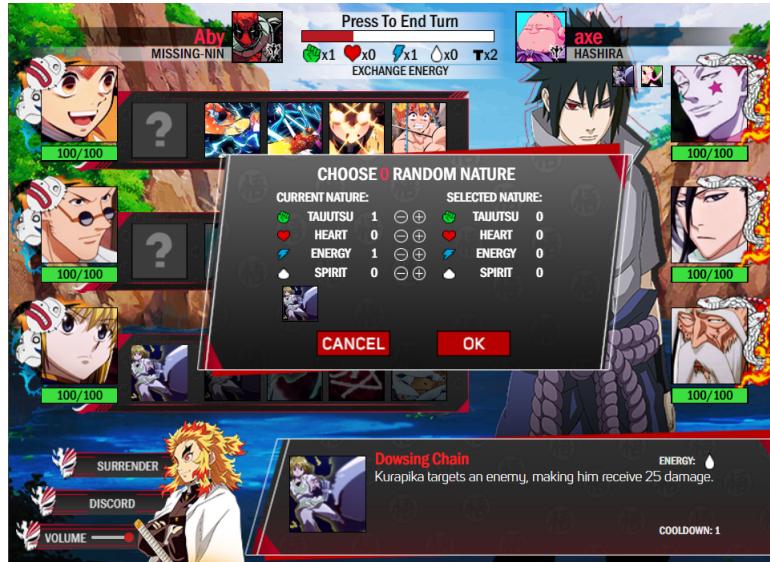


Figura 4.12: Fila de Habilidades

É representado a fila de habilidades que indica as ações que serão feitas se o jogador terminar o seu turno. O resultado desta confirmação encontra-se visualizado na figura 4.13.



Figura 4.13: Fim do Turno

Foi aplicado a habilidade, reduzindo a vida do alvo em 25. O turno do jogador terminou e este aguarda as ações do seu oponente, terminando o cenário principal da batalha.

De seguida serão exemplificados cenários alternativos que poderiam ter acontecido no cenário principal da batalha. Devido a não ser possível refazer o jogo, serão

utilizados novos exemplos onde estes cenários acontecem.

#### 4.2.0.1 Cenário da Batalha - Seleção Alvo Incorreto

Este cenário alternativo acontece quando o jogador seleciona uma habilidade possível de ser executada, mas não escolhe um alvo correto.

A situação da habilidade selecionada encontra-se na figura 4.14.



Figura 4.14: Alvos Possíveis

Esta figura ilustra a situação anterior ao número 4 da sequência, onde acontece o cenário alternativo da tabela 4.4.

Cenário Alternativo 1		
Número da Sequência	Alternativa	
4	Jogador clica numa personagem que não esteja animada	Servidor desativa animações existentes nas personagens. 5,7 sem efeito

Tabela 4.4: Cenário da Batalha - Seleção Alvo Incorreto

A figura 4.15 mostra o ecrã deste cenário alternativo.



Figura 4.15: Alvo Selecionado Incorreto

No menu inferior do ecrã é indicado que a personagem *Kushiki Byakuya* (alvo incorreto) foi selecionada pelo jogador e salienta-se que as animações desapareceram.

#### 4.2.0.2 Cenário da Batalha - Desistência

O jogador, a qualquer momento da partida, consegue render-se, dando a vitória ao oponente. Este cenário alternativo encontra-se na tabela 4.5.

Cenário Alternativo 2		
Número da Sequência		Alternativa
2	Jogador prima o botão para desistir da batalha	Servidor apresenta confirmação. 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 sem efeito
3	Jogador confirma desistência	Fim da batalha

Tabela 4.5: Cenário da Batalha - Desistência

Premir o botão de desistência (“Surrender”) origina este cenário alternativo e torna visível o menu da figura 4.16.

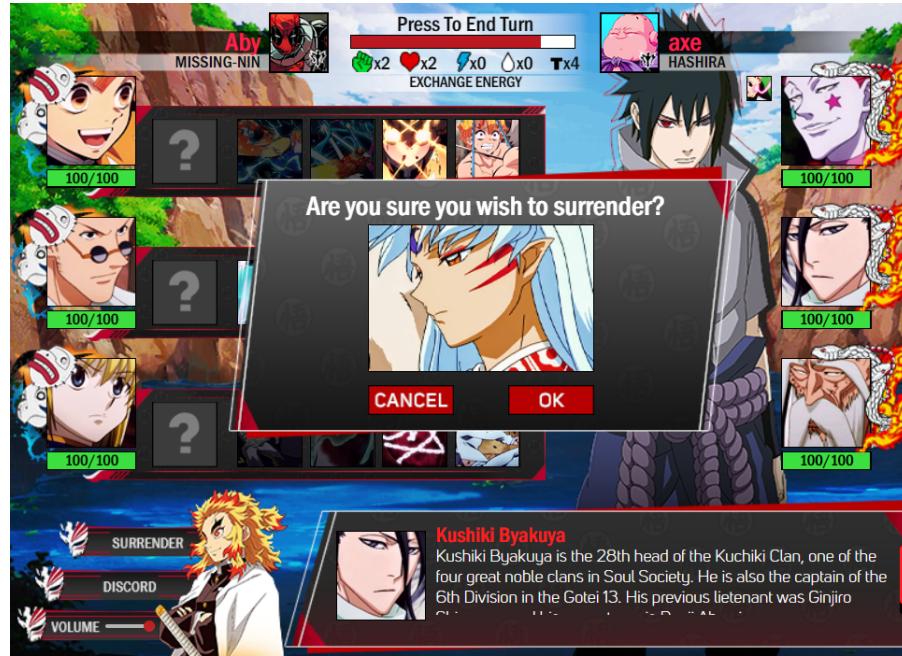


Figura 4.16: Menu de Desistência

C cancelar esta ação faz desaparecer este menu, no entanto a sua confirmação (número da sequência 3) termina a batalha, como mostra a figura 4.17.

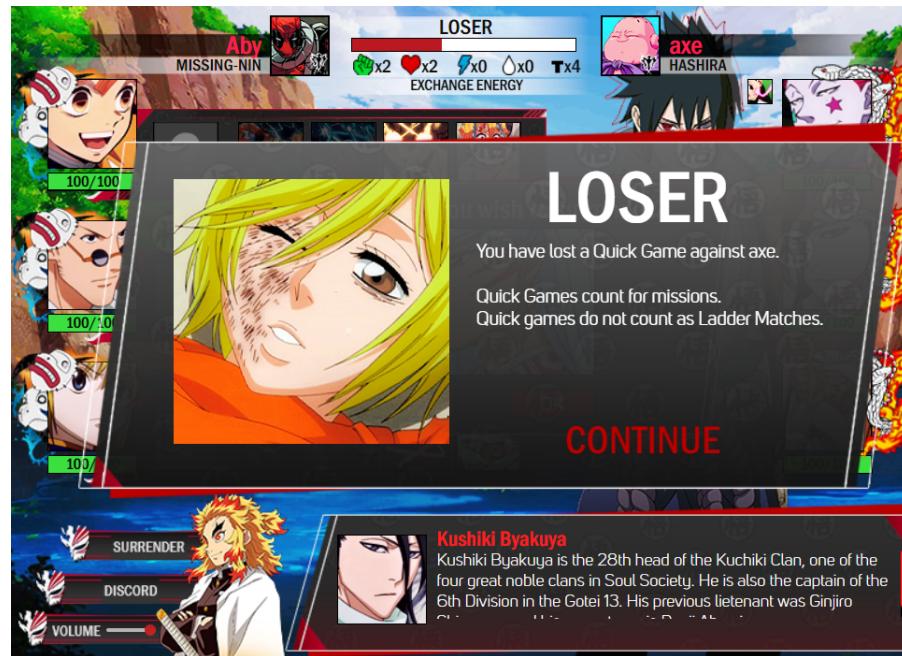


Figura 4.17: Rendição do Jogador

O fim da batalha atualiza os resultados de ambos os jogadores. Como se trata de uma batalha em modo *quick*, apenas afeta o progresso das missões.

#### 4.2.0.3 Cenário da Batalha - Desistência do Oponente

O jogador apenas tem conhecimento da desistência do oponente quando finalizar o seu turno. Assim, a seguir ao número da sequência 11, pode acontecer o cenário alternativo da tabela 4.6.

Cenário Alternativo 3	
Número da Sequência	Alternativa
12   Oponente desiste	Servidor mostra o menu de vencedor. 13 e 14 sem efeito.

Tabela 4.6: Cenário da Batalha - Desistência do Oponente

O menu de vencedor, resultante da desistência do oponente, é ilustrado na figura 4.18.

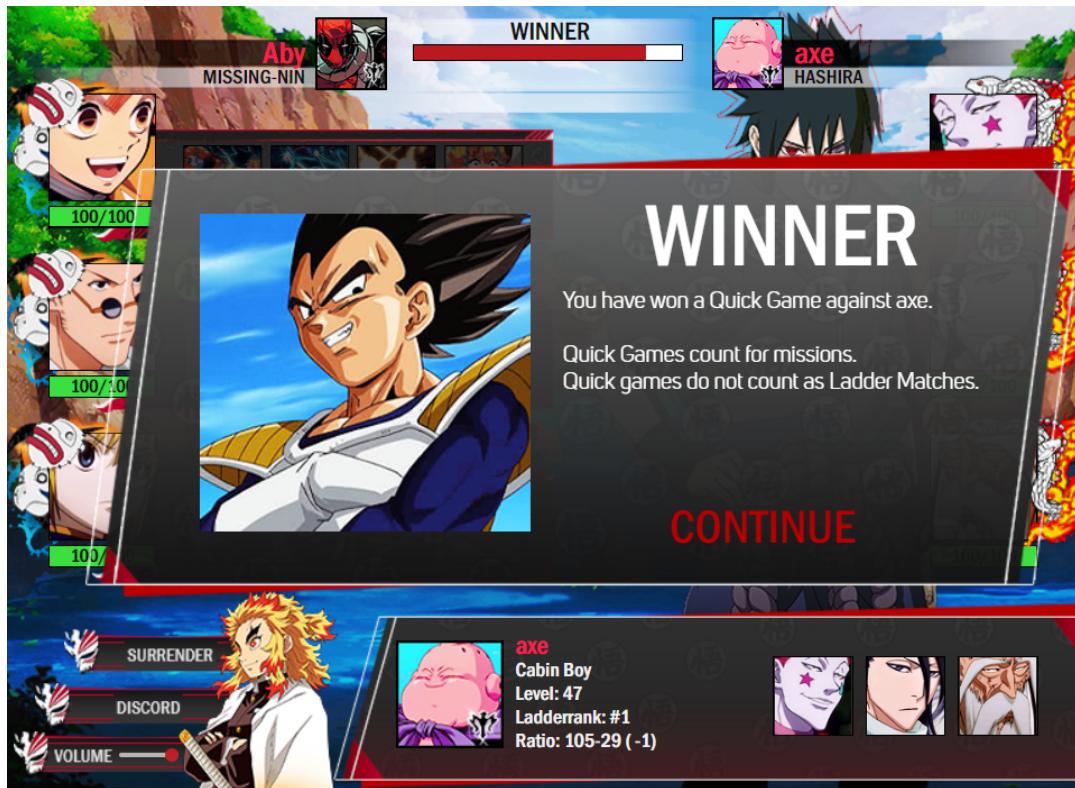


Figura 4.18: Vitória

O jogador é vitorioso e progride nas suas missões. Premir o botão “Continue” redirecionará o jogador para o ecrã da seleção de equipa.

#### 4.2.0.4 Cenário da Batalha - Cancelamento de Habilidade

O jogador pode cancelar a execução de habilidades antes de finalizar o seu turno, recuperando as naturezas gastas nessa ação.

A figura 4.19 exemplifica uma situação onde o jogador definiu corretamente uma habilidade.

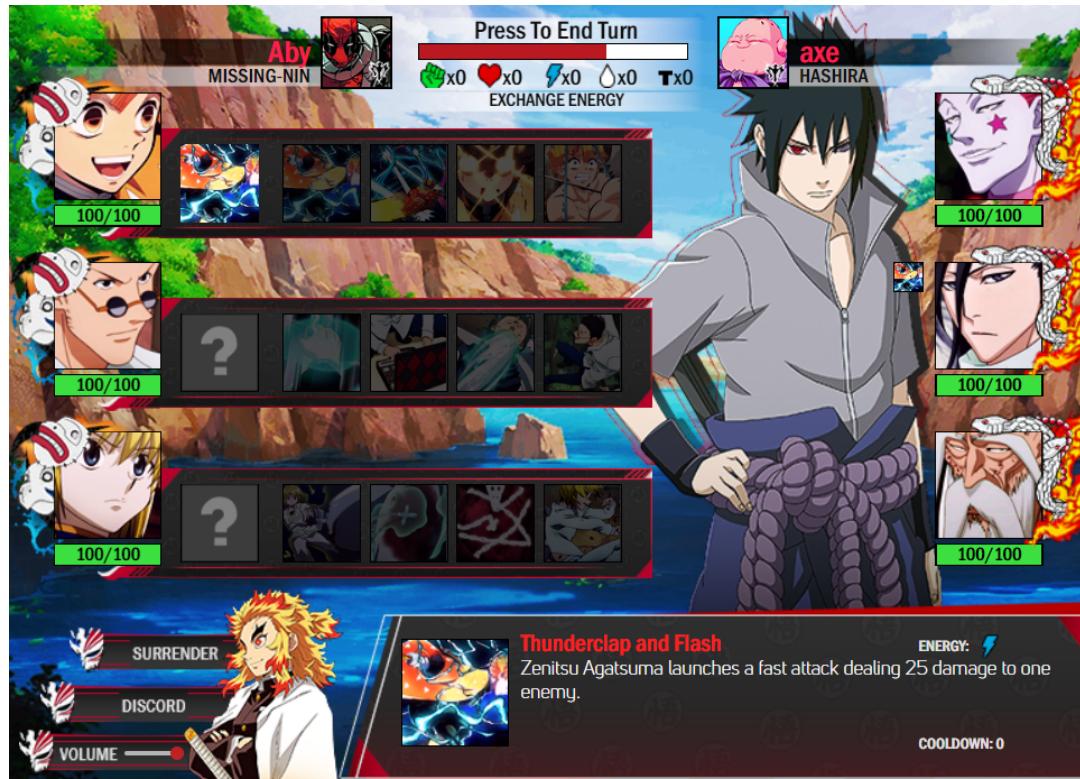


Figura 4.19: Execução de Habilidade

Nesta situação, é possível o cenário alternativo da tabela 4.7 acontecer, se o jogador decidir cancelar a habilidade da sua primeira personagem.

Cenário Alternativo 4		
Número da Sequência		Alternativa
10	Jogador cancela a habilidade	Servidor atualiza as naturezas do jogador e reavalia a jogabilidade das habilidades. 11, 12, 13 sem efeito.

Tabela 4.7: Cenário da Batalha - Cancelamento de Habilidade

O jogador consegue facilmente cancelar uma habilidade, realizando dois cliques consecutivos nela. O resultado desta ação é ilustrado na figura 4.20.

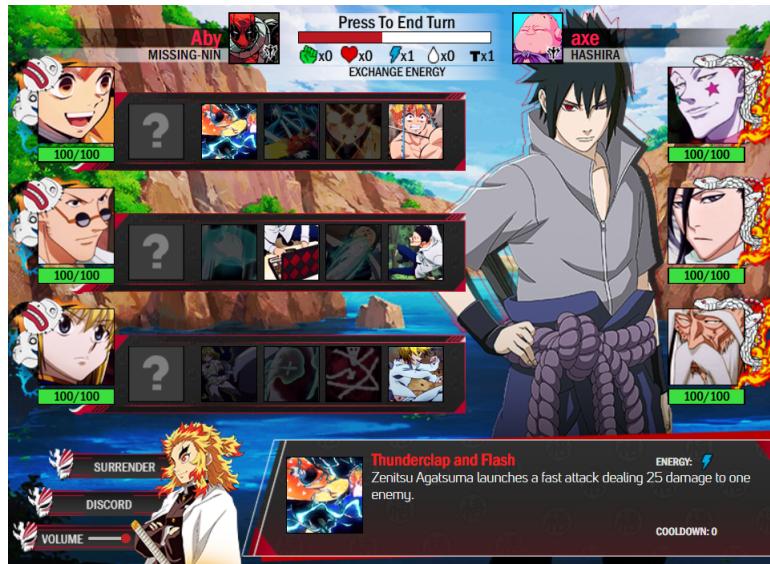


Figura 4.20: Cancelamento da Habilidade

Observa-se que a habilidade foi cancelada, as naturezas restauradas e o jogador consegue, novamente, realizar ações com a personagem.

#### 4.2.0.5 Cenário da Batalha - Esgotamento de Tempo

O tempo é um fator que o jogador necessita ter em consideração. Quando se esgota, finaliza o turno do jogador e nenhuma ação realizada nesse turno é aplicada. A figura 4.21 representa o ponto de situação antes do tempo chegar ao fim.

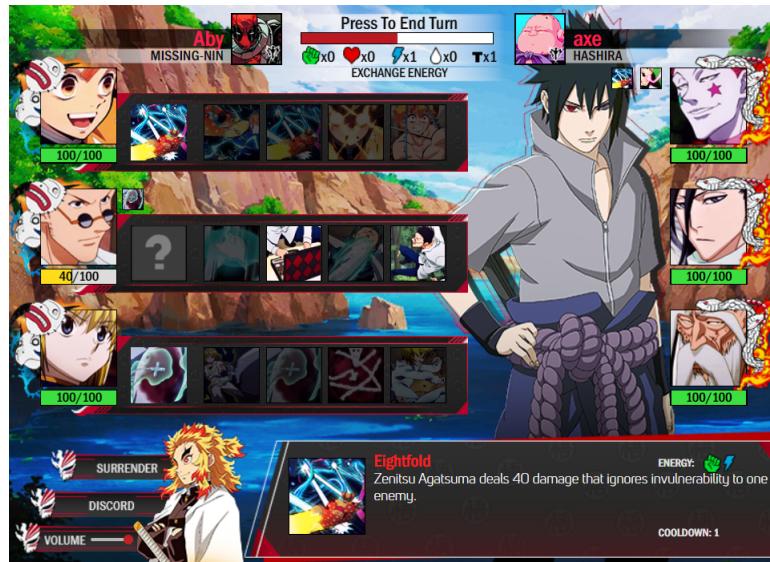


Figura 4.21: Estado da Batalha

Salienta-se que o jogador definiu duas habilidades. O cenário alternativo do esgotamento de tempo é representado na tabela 4.8.

Cenário Alternativo 5		
Número da Sequência		Alternativa
8	Termina o tempo do turno	Servidor termina o turno do jogador e passa-o ao oponente. 9, 10, 11 sem efeito.

Tabela 4.8: Cenário da Batalha - Esgotamento de Tempo

A figura 4.22 ilustra este cenário.

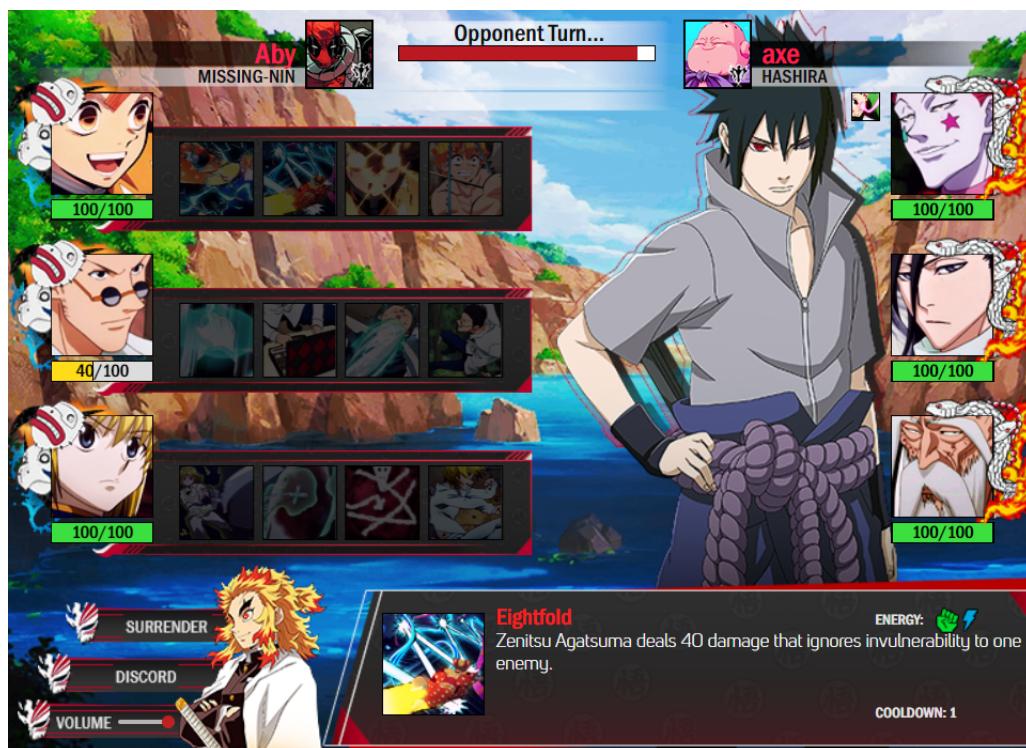


Figura 4.22: Esgotamento de Tempo

Confirma-se que não houve nenhuma alteração na batalha e que o jogador aguarda a ação do seu oponente.

#### 4.2.0.6 Cenário da Batalha - Vitória

Derrotar as três personagens do oponente tornam o jogador vitorioso. Esta situação termina a batalha e torna visível o menu de vitória. A figura 4.23 ilustra o jogador prestes a aplicar o “golpe final”.



Figura 4.23: Aplicação do Golpe Final

Premir o botão “OK” finaliza o turno e dá início ao cenário alternativo da tabela 4.9.

Cenário Alternativo 6		
Número da Sequência	Alternativa	
13	Jogador derrotou todas as personagens do seu adversário	Servidor indica ao jogador que é o vencedor.

Tabela 4.9: Cenário da Batalha - Vitória

A batalha termina e o jogador recebe indicação que foi vitorioso.

### 4.3 Exemplo Anúncios

Na página principal encontram-se os anúncios publicados pelos administradores, informando os jogadores de novidades como novas personagens, missões, mudanças de habilidades existentes e qualquer outra alteração ou nova funcionalidade.

Como se pode observar na figura 4.24, em “Announcement” é ilustrado o anúncio mais recente e em “Past News” encontram-se os três anúncios anteriores.

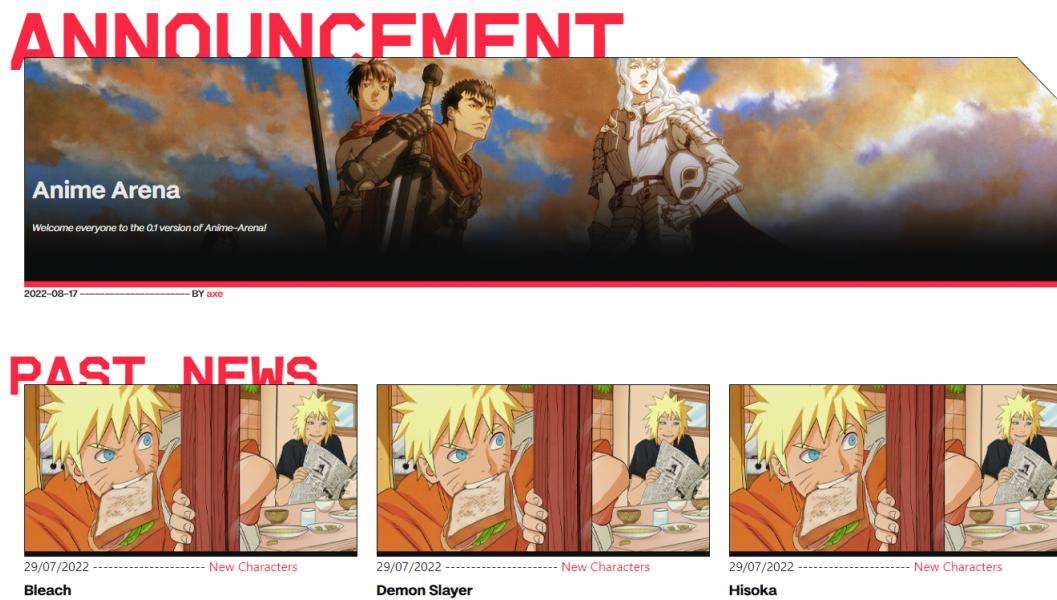


Figura 4.24: Página Inicial - Anúncios

É possível observar a informação detalhada de um anúncio selecionando-o. Na figura 4.25 ilustra-se o anúncio principal.

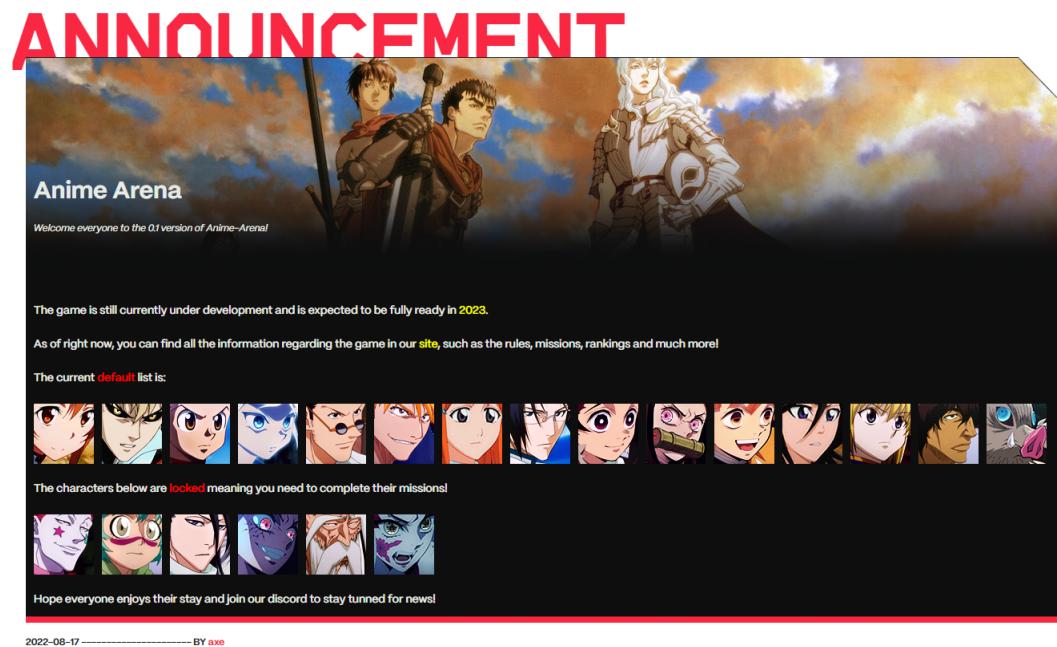


Figura 4.25: Visualização Anúncio

Na tabela 4.10 encontra-se representado o caso de utilização, em formato resumido, da criação de anúncios.

<b>Cabeçalho</b>	
Nome:	Anúncios
Resumo:	Administrador escreve e pré-visualiza um anúncio
Referências	R1.1.2, R1.2, R1.2.7

Tabela 4.10: Cenário Anúncios

De modo a criar um anúncio, é necessário que o administrador preencha determinadas informações, nomeadamente o título, a categoria, o resumo e o texto completo, como se observa na figura 4.26

The screenshot shows a web-based form for creating an announcement. At the top, there are two input fields: 'Title' and 'Category'. Below these is a section labeled 'Small description:' with a text area containing placeholder text 'Small description'. Underneath is a larger text area labeled 'Write Announcement'. Above this text area is a 'Full Announcement:' label with a toolbar featuring icons for bold (B), italic (I), and various colors (red, yellow, blue, green, purple). At the bottom of the form are two red buttons: 'PREVIEW' on the left and 'CONFIRM' on the right.

Figura 4.26: Escrita do Anúncio

É possível observar que existem botões debaixo de “Full Announcement:”, permitindo adicionar estilos ao texto e inserir imagens. Na figura 4.27 encontra-se um exemplo ilustrativo destes botões, assim como a opção de pré-visualizar o anúncio antes de ser submetido.

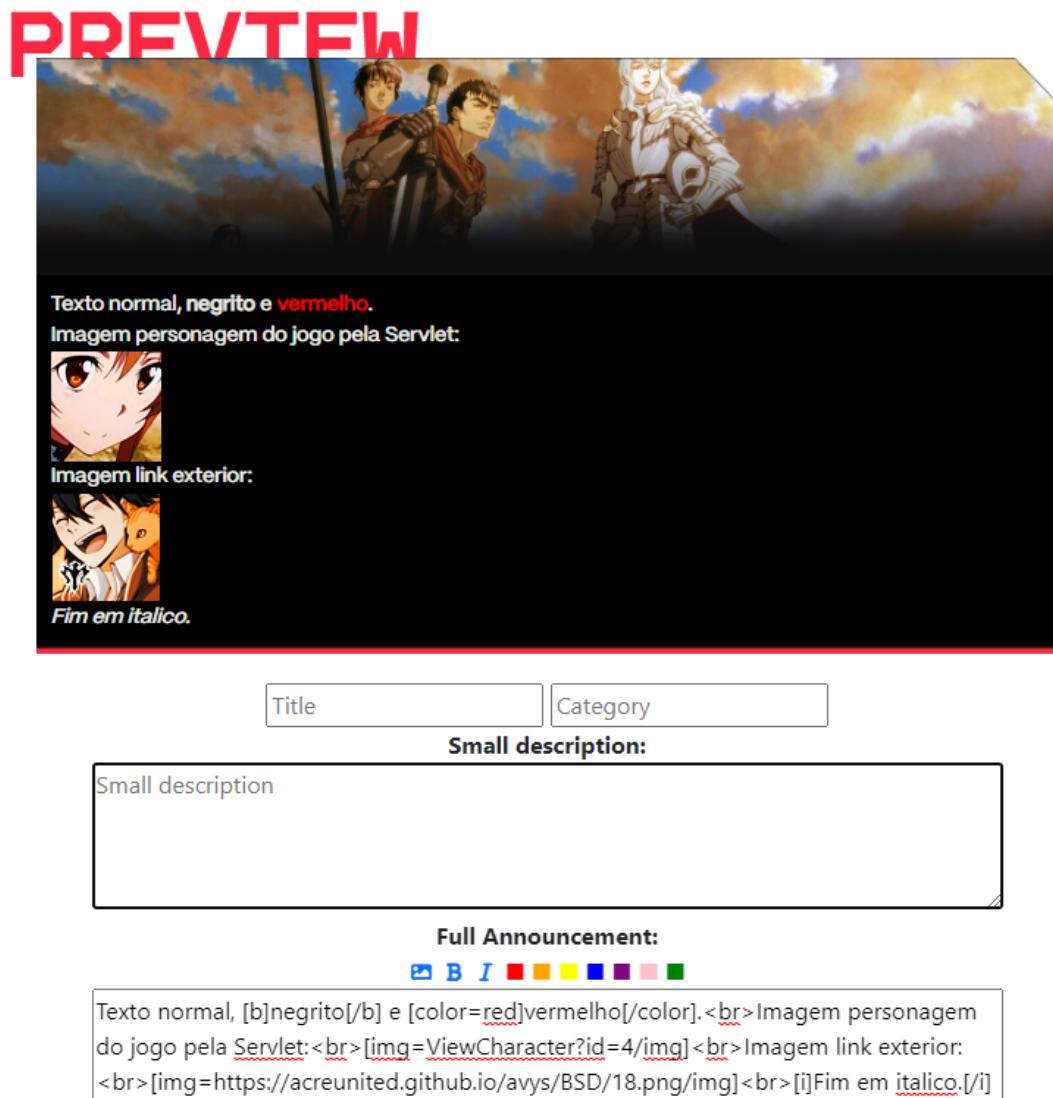


Figura 4.27: Pré-visualização do Anúncio

Tendo em consideração estas marcações, foi criado uma função que converte o texto inserido pelo utilizador em formato HTML. Pressionar o botão “Confirm” (figura 4.26) submete o anúncio e atualiza, automaticamente, a página principal da plataforma.

## 4.4 Exemplo Criar Personagem e Missão

De modo a ser possível jogar com personagens e realizar missões, é necessário a sua criação por parte de um administrador, originando a necessidade do caso de utilização da tabela 4.11.

<b>Cabeçalho</b>	
Nome:	Criação de Personagem e Missão
Resumo:	Administrador cria uma personagem e associa-a a uma missão
Referências	R1.1.2, R1.2, R1.2.1, R1.2.3, R1.2.4

Tabela 4.11: Cenário Criar Personagem e Missão

As figuras 4.28, 4.29 e 4.30 exemplificam o processo da criação de uma personagem e missão, nomeadamente através do preenchimento de um formulário.

**Character Name** nome exemplo

**Character Description**

Descrição personagem exemplo

**Character Picture** Escolher ficheiro KiritsuguEmiya.png

Figura 4.28: Definição da Personagem

Primeiramente, é necessário inserir a informação da personagem, nomeadamente o nome, a descrição e a imagem, como se encontra na figura 4.28. De seguida, o administrador necessita completar a informação das suas quatro habilidades: o nome, a descrição, a imagem e como se desempenha na batalha.

Para título de exemplo, apenas é ilustrado o formulário para uma habilidade na figura 4.29, pois este processo é repetitivo para as restantes.

**Ability1 Name:** Ability Name

**Ability1 Description:**  
Write Ability 1 Description (max 5000chars)

**Who is the target of this ability?** Self

**Does this ability do damage?** No

**Ability damage increase per use (0 if none)**

**Heal Increase Per Use (0 if none)**

**Permanent Character increase damage (0 if none)**

**Stun Duration (0 if none)**

**How many turns this ability makes character invulnerable (0 if none)**

**Does this ability ignore invulnerability?** No

**Does this ability remove Natures?** No

**Does this ability gain Natures?** No

**Does this ability gain Health?** No

**Does this ability gain Damage Reduction?** Yes

**How Many?:** Turn duration:

**Taijutsu (minimum 0):**

**Heart (minimum 0):**

**Energy (minimum 0):**

**Spirit (minimum 0):**

**Random (minimum 0):**

**Cooldown (minimum 0):**

**Ability1 Picture:** Escolher ficheiro Nenhum ficheiro selecionado

Figura 4.29: Definição de uma Habilidade

Observa-se que existem diversas opções booleanas. Uma resposta afirmativa torna visível os dados necessários para a determinada funcionalidade.

Para terminar, é necessário indicar se a personagem está associada a alguma missão. Em caso afirmativo, o formulário ilustrado na figura 4.30 ficará visível.

**Character Anime:** Bleach

**Default or Mission Character ?:** Mission

**Mission Name:** Mission Name

**Mission Description:**  
Write Mission Description (max 5000chars)

**Required Level (minimum 1):**

**Mission Picture:** Escolher ficheiro Nenhum ficheiro selecionado

**Requirements:**

**With which character?** None

**How many wins?**

**Are wins in a row?** Yes

**With which character?** None

**How many wins?**

**Are wins in a row?** Yes

**With which anime?** None

**How many wins?**

**Are wins in a row?** Yes

Figura 4.30: Definição da Missão

Na opção “Default or Mission Character”, se “Default” for selecionado, a personagem será oferecida a todos os jogadores.

## 4.5 Exemplo Visualização de Personagens

É importante que os jogadores tenham acesso à informação de todas as personagens do jogo, originando o caso de utilização da tabela 4.12.

Cabeçalho	
Nome:	Visualização de Personagens
Resumo:	Jogador consulta as personagens existentes e as suas respetivas habilidades
Referências	R1.2, R1.2.2, R1.3.1

Tabela 4.12: Cenário Consultar Personagens

Em “Game Info” encontram-se as personagens existentes do jogo, como ilustra a figura 4.31. De modo a facilitar a sua procura, é possível filtrar por anime e estão agrupadas em duas categorias: personagens por omissão ou com missão associada.

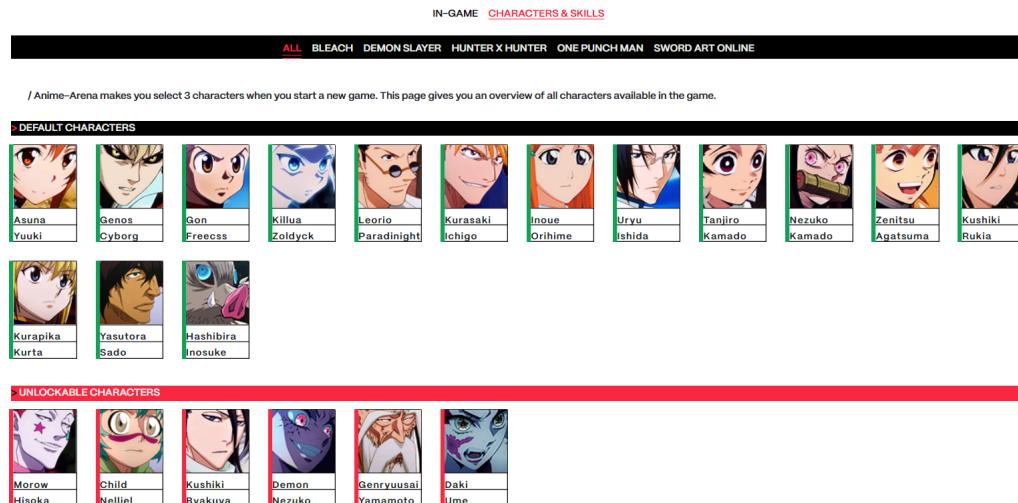


Figura 4.31: Todas as Personagens

É possível visualizar a informação das personagens selecionando-as.

Figura 4.32: Informação da Personagem Zenitsu Agatsuma

A figura 4.32 ilustra a informação da personagem *Zenitsu Agatsuma*.

## 4.6 Exemplo Consultar Missões

De modo a ser possível desbloquear personagens, é necessário que os jogadores consigam visualizar as missões correspondentes, originando o caso de utilização da tabela 4.13.

Cabeçalho	
Nome:	Consultar Missões
Resumo:	Jogador consulta uma missão, visualizando os requisitos e o seu progresso
Referências	R1.1.1, R1.2.5

Tabela 4.13: Cenário Consultar Missões

As missões encontram-se na página “Missions” e estão categorizadas em animes, de forma a facilitar a sua procura. Existem missões de três animes diferentes: *Hunter X Hunter*, *Bleach* e *Demon Slayer*. A figura 4.33 ilustra as missões do anime *Bleach*.

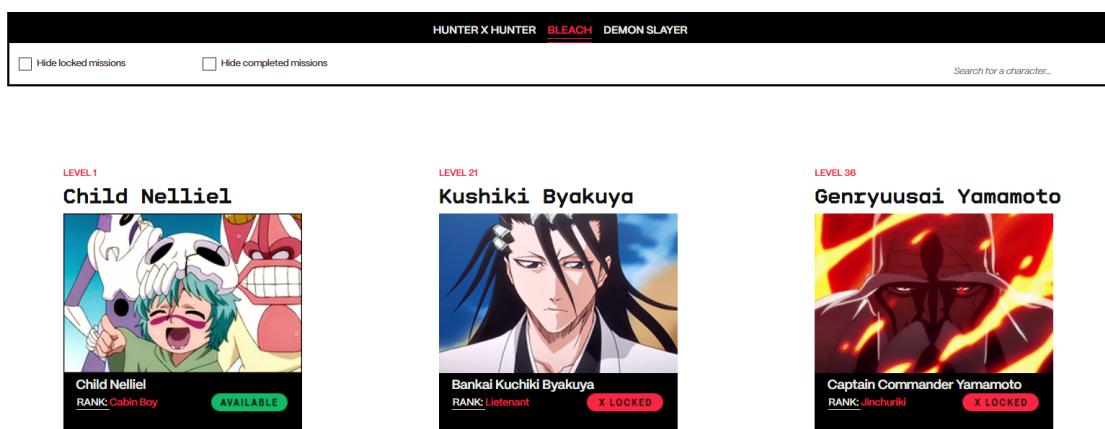


Figura 4.33: Missões

É possível esconder missões bloqueadas (o jogador não possui o nível mínimo) e/ou completas, assim como procurar pelo nome, facilitando a tarefa do utilizador e tornando a aplicação flexível para um número infinito de missões.

Na figura 4.34, é ilustrado a informação da missão de *Kushiki Byakuya*.

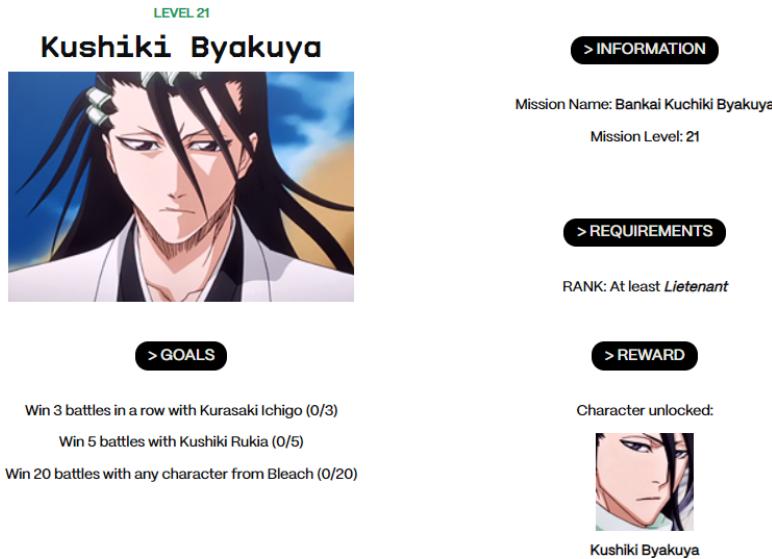


Figura 4.34: Informação da Missão de Kushiki Byakuya

Observa-se que, em adição ao nome, imagem e nível necessário para progredir na missão, o jogador encontra os requisitos, o seu progresso e a personagem que lhe é retornada quando a missão é concluída.

## 4.7 Exemplo Quadro de Honra

Sendo a competitividade uma componente importante do jogo, existe o caso de utilização da tabela 4.14, onde é possível visualizar o progresso dos jogadores no ecrã “Leaderboards”.

<b>Cabeçalho</b>	
Nome:	Destaque de classificações
Resumo:	Jogador observa as melhores classificações
Referências	R1.4

Tabela 4.14: Cenário Quadro de Honra

Existem três classificações:

- **Ladder Rank** - baseado na experiência dos jogadores;
- **Streak** - maior número de vitórias consecutivas;
- **Wins** - jogadores com o maior número de vitórias.

A figura 4.35 ilustra a classificação geral, denominada “Ladder Rank”.

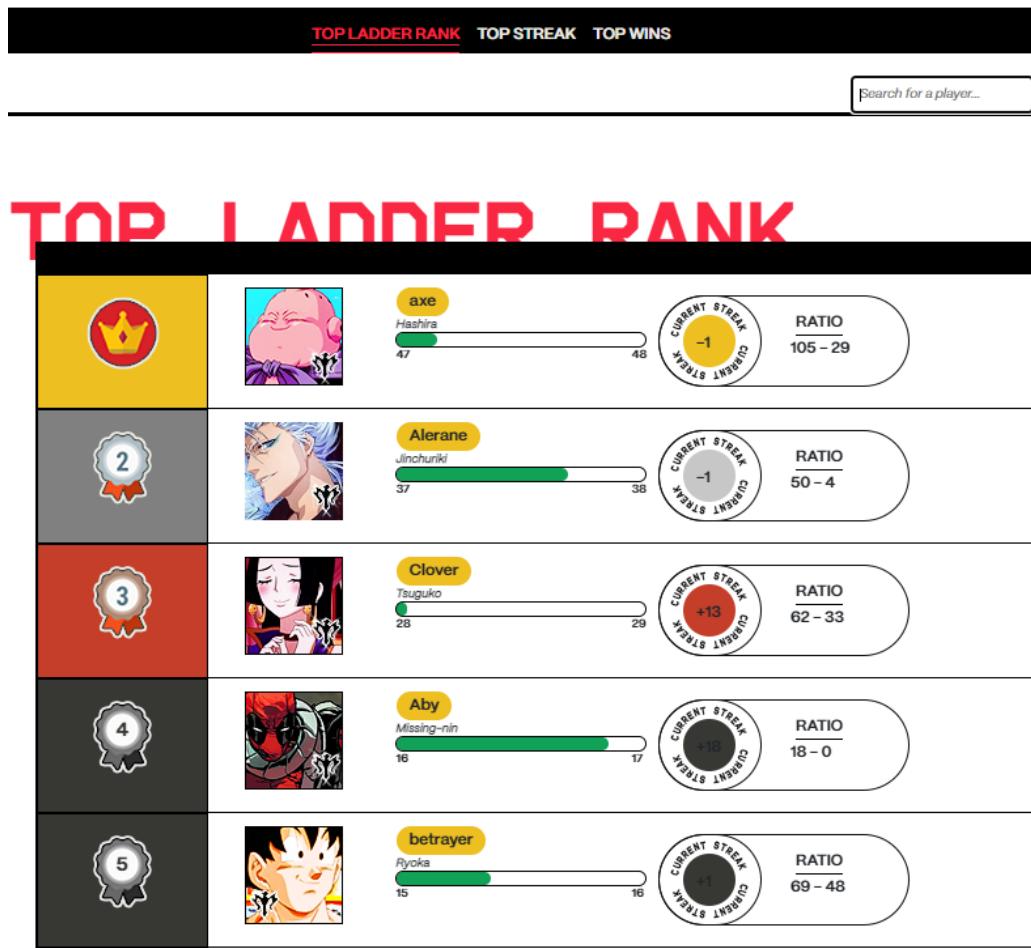


Figura 4.35: Classificação

É possível procurar um jogador pelo seu nome, como mostra a figura 4.36.

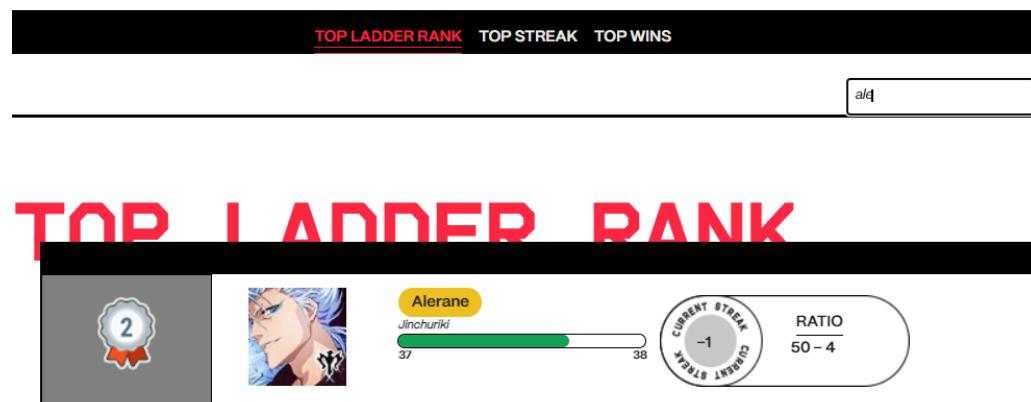


Figura 4.36: Procura de um Jogador

Observa-se que, pesquisando por “ale”, apenas o jogador “alerane” aparece na classificação.



# Capítulo 5

## Conclusão

Este último capítulo encerra o relatório, apresentando o resumo e desenvolvimentos futuros.

### 5.1 Resumo

Os objetivos do projeto foram atingidos. Foi implementada uma plataforma multi-jogador recorrendo à tecnologia JavaServer Pages suportada numa arquitetura que integra os servidores Tomcat e MySQL.

A realização deste projeto foi uma oportunidade para aprofundar conhecimentos lecionados ao longo do curso em diversas unidades curriculares, tais como: Modelação e Programação (MOP) com a introdução do Java; Tecnologias de Informação (TI) para as linguagens baseadas em etiquetas, como HTML e XML; Fundamentos de Sistemas Operativos (FSO) permitiu, com os mecanismos de sincronização, a existência de jogos em simultâneo e o sincronismo de turnos entre o par de jogadores; Produção de Conteúdos Multimédia (PCM) onde foi ensinado JavaScript e abrangeu os conhecimentos de CSS, nomeadamente com a utilização da *framework* Bootstrap; Infraestruturas Computacionais Distribuidas (ICD) no reforço das tecnologias associadas ao uso de XML com validações (XSD) e consulta (XPATH) recorrendo ao Java; Interacção Pessoa-Máquina (IPM) na constante preocupação com o utilizador; Sistemas de Base de Dados (SBD) onde foram desenvolvidos conhecimentos sobre modelos de dados relacionais (MySQL).

### 5.2 Desenvolvimentos Futuros

Criar maior variedade de personagens, habilidades e missões, explorando aspectos de flexibilidade já previstos. Este desenvolvimento está dependente do envolvimento dos membros da equipa criativa, nomeadamente nos aspectos gráficos.

Eventualmente os jogadores mais dedicados conseguirão, rapidamente, obter todas as personagens e/ou classificações, sendo necessário que os administradores continuem a desenvolver incrementalmente novos conteúdos.

Existem ideias sobre formas de financiamento necessário para alojamento do servidor de suporte ao jogo, nomeadamente publicidade, adição de novos temas, por exemplo com imagens animadas (.GIF) e novos cosméticos ilustrados nas classificações, conforme o gosto dos jogadores.

Seria igualmente interessante guardar os resultados estatísticos de batalhas das personagens, de modo aos administradores terem acesso a mais informação, facilitando o balanceamento do jogo.

Em termos de planeamento, existe trabalho a realizar na definição e distribuição de tarefas. Mais concretamente, para além dos artistas para a componente gráfica e dos programadores para implementar os requisitos, é necessário moderadores que ouçam as críticas dos jogadores, dedicando especial atenção ao balanceamento das habilidades.

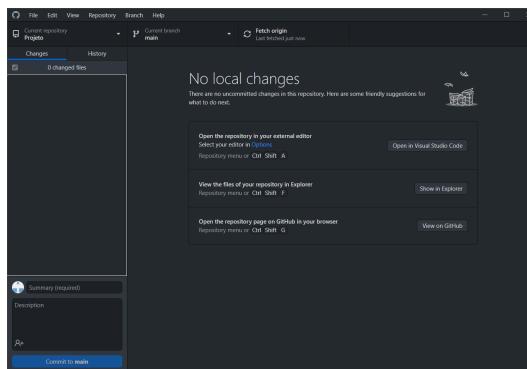
# Apêndice A

## Controlo de Versões

Foi utilizado o GitHub [Preston-Werner et al., 2008] para fazer o controlo de versões. É uma ferramenta bastante prática, útil e essencial para evitar perder trabalho, pois mantém o registo de todas as alterações e torna possível recuperar versões anteriores.

Como este projeto foi realizado individualmente, apenas foi utilizado o ramo principal por omissão.

Na execução dos comandos comuns, nomeadamente “push”, “pull”, “clone” e “commit”, foi utilizado a aplicação do GitHub para o computador. Quando houve necessidade de desfazer alterações e voltar a uma versão anterior, nomeadamente com o comando “reset”, utilizou-se o Git Bash.



(a) Aplicação do GitHub

A screenshot of a terminal window titled "MINGW64 /d/GitHub/Projeto/WebContent (main)". The window shows a single line of text: "\$ |".

(b) Git Bash

Figura A.1: Emparelhamento

A figura A.1 ilustra o ambiente das aplicações utilizadas no controlo de versões.



# Referências

- [Bezos, 1994] Bezos, J. (1994). Amazon. <https://www.amazon.com/>.
- [Chamberlin e Boyce, 1974] Chamberlin, D. D. e Boyce, R. F. (1974). Structured query language.
- [CherryPy, 2008] CherryPy (2008). Cherrypy. <https://docs.cherrypy.dev/en/latest/>.
- [Coar et al., 1999] Coar, K., Behlendorf, B., Cox, M., Eilebrecht, L., Engelschall, R. S., Fielding, R. T., Gaudet, D., Hyde, B., Jagielski, J., Kosut, A., Kraemer, M., Laurie, B., MacEachern, D., Mirzadeh, A., Parekh, S., Skolnick, C., Slemko, M., Stoddard, W., Sutton, P., Terbush, R., e van Gulik, D.-W. (1999). Apache software foundation.
- [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, 1995] Eich, B. (1995). Javascript. <https://www.javascript.com/>.
- [Ek e Lorentzon, 2006] Ek, D. e Lorentzon, M. (2006). Spotify. <https://www.spotify.com/>.
- [Ellison et al., 1977] Ellison, L., Miner, B., e Oates, E. (1977). Oracle corporation. <https://oracle.com/>.
- [Fleury, 1999] Fleury, M. (1999). Jboss. <https://www.jboss.org/>.
- [Foundation, 1999] Foundation, A. S. (1999). Tomcat. <https://tomcat.apache.org/>.

- [Gates e Allen, 1975] Gates, B. e Allen, P. (1975). Microsoft. <https://microsoft.com/>.
- [Gosling, 1996] Gosling, J. (1996). Java. <https://www.java.com/>.
- [Gotoge, 2016] Gotoge, K. (2016). Kimetsu no yaiba.
- [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., 2001] Inc., A. (2001). macos.
- [Inc., 2009] Inc., M. (2009). Mongodb. <https://www.mongodb.com/>.
- [JetBrains, 2001a] JetBrains (2001a). IntelliJ. <https://www.jetbrains.com/>.
- [JetBrains, 2001b] JetBrains (2001b). Kotlin. <https://kotlinlang.org/>.
- [Kawahara, 2009] Kawahara, R. (2009). Sword art online.
- [Kishimoto, 2002] Kishimoto, M. (2002). Naruto.
- [Kubo, 2001] Kubo, T. (2001). Bleach.
- [Lerdorf, 1995] Lerdorf, R. (1995). Php. <https://www.php.net/>.
- [McCool, 1995] McCool, R. (1995). Apache. <https://www.apache.org/>.
- [Microsoft, 1985] Microsoft (1985). Windows.
- [Microsoft, 1989] Microsoft (1989). Microsoft sql server. <https://www.microsoft.com/en-us/sql-server/>.
- [Microsoft, 2001] Microsoft (2001). C sharp. <https://docs.microsoft.com/en-us/dotnet/csharp/>.
- [Microsoft, 2002] Microsoft (2002). .net. <https://dotnet.microsoft.com/>.

- [Microsystems, 1999] Microsystems, S. (1999). Javaserver pages.
- [Murata, 2009] Murata, Y. (2009). One punch-man.
- [Otto e Thorntonn, 2011] Otto, M. e Thorntonn, J. (2011). Bootstrap. <https://getbootstrap.com/>.
- [Page e Brin, 1998] Page, L. e Brin, S. (1998). Google. <https://www.google.com/>.
- [Preston-Werner et al., 2008] Preston-Werner, T., Wanstrath, C., Chacon, S., e Hyett, P. J. (2008). Github. <https://github.com/>.
- [Ritchie, 1972] Ritchie, D. (1972). C.
- [Sittner, 1983] Sittner, J. (1983). Ancestry. <https://www.ancestry.com/>.
- [Stanek, 2000] Stanek, R. (2000). Netbeans. <https://netbeans.apache.org/>.
- [Stroustrup, 1979] Stroustrup, B. (1979). C++. <https://cplusplus.com/>.
- [Togashi, 1998] Togashi, Y. (1998). Hunter x hunter.
- [Torvalds, 1991] Torvalds, L. (1991). Linux.
- [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, 1993] W3C e WHATWG (1993). Html. <https://html.com/>.
- [Wales e Sanger, 2001] Wales, J. e Sanger, L. (2001). Wikipedia. <https://www.wikipedia.org/>.
- [Wang, 2003] Wang, G. (2003). Litespeed. <https://www.litespeedtech.com/>.
- [web2py, 2007] web2py (2007). web2py. <http://www.web2py.com/>.
- [Wirth, 1986] Wirth, N. (1986). 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/>.