Universidade da Beira Interior
Departamento de Informática

Programação de Dispositivos Móveis

Onceupona: O Escritor Surrealista

Elaborado por:
Alexandre Antão - a37416
Eduardo Paulos - a39409
Inês Lopes - a37559
João Domingos - a38023
Miguel Gregório - a35388

R-Team

Professor Doutor Pedro Inácio

Conteúdo

Lista de Figuras

Acrónimos

IDE *Integrated Development Environment*

UML *Unified Modeling Language*

MB *Megabytes*

UML *Unified Modeling Language*

RAM *Random Access Memory*

MHz *Megahertz*

XML *Extensible Markup Language*

Capítulo 1

Introdução

1.1 Descrição da Proposta

No âmbito da Unidade Curricular de Programação de Dispositivos Móveis foi solicitado a este grupo desenvolver uma aplicação *Android* que permita ilustrar virtualmente o clássico jogo francês *Cadavre-exquis*.
Este jogo precisará de dois ou mais jogadores onde, em conjunto, elaborarão um texto tendo apenas, cada um, acesso às duas últimas palavras que foram escritas pelo jogador anterior.

1.2 Constituição do Grupo

- Alexandre Antão - a37416;
- Eduardo Paulos - a38409;
- Inês Lopes - a37559;
- João Domingos - a38023;
- Miguel Gregório - a35388.

1.3 Organização do documento

De modo a refletir o trabalho que foi feito, este documento encontra-se estruturado da seguinte forma:

1. O primeiro capítulo – **Introdução** – apresenta o projeto, a constituição do grupo de trabalho e a respetiva organização do documento.

2. O segundo capítulo – **Engenharia de Software** procura mostrar o desenho do problema e o que foi usado para o resolver, nomeadamente as ferramentas e tecnologias usadas, os requisitos, diagramas de casos de uso, de atividades e de classe.

3. O terceiro capítulo – **Implementação** discute aquilo que foi efetivamente implementado, por que motivo foi escolhida essa implementação e não outra, uma breve introdução ao *layout* da aplicação para além de apresentar um breve manual de instalação e utilização.

4. O quarto capítulo – **Reflexão Crítica e Problemas Encontrados** debate o sucesso do projeto, nomeadamente através da comparação entre os objetivos propostos e aqueles que, efetivamente, foram realizados, passando pelos problemas que foram surgindo durante o desenvolvimento (refletindo sobre a sua resolução ou não). É também apresentada a distribuição de tarefas do projeto pelos membros do grupo;

5. O quinto capítulo – **Conclusões e Trabalho Futuro** retira a conclusão do projeto, levantando ainda questões a ser melhoradas/implementadas futuramente.

Capítulo 2

Engenharia de Software

2.1 Introdução

Neste capítulo serão apresentadas as ferramentas utilizadas, os requisitos de domínio do sistema, as funcionalidades, a estrutura e os comportamentos da aplicação. Estando organizado da seguinte forma:

- Ferramentas e Tecnologias Utilizadas (2.2) ;
- Requisitos (2.3);
- Casos de uso (2.4);
- Diagrama de classes (2.5);
- Diagramas de atividades (2.6).

2.2 Ferramentas e Tecnologias Utilizadas

Na execução deste projeto foi utilizado um *Integrated Development Environment* (IDE) para a escrita do código-fonte e elaboração do *design* gráfico. O IDE utilizado foi o *Android Studio*. Para a elaboração do relatório foi usado o *OverLeaf*, um editor de LaTeX *online*. Para a ilustração dos diagramas foi usado o *Astah*, uma ferramenta de *design* de *software* que suporta *Unified Modeling Language* (UML), e a extensão *simpleUML* do *Android Studio*.

2.3 Requisitos

Para a execução da aplicação ser bem sucedida o sistema deve conter, no mínimo, os seguintes requisitos:
– *Android* 1.6 *"Donut"* ;
– 32 *Megabytes* (MB) de *Random Access Memory* (RAM);

– 32 MB Memória *Flash*;
– 200 *Megahertz* (MHz) no processador.
http://www.mycomputeraid.com/computers/google-android-system-requirements/

2.4 Casos de Uso

Na aplicação desenvolvida existem várias funcionalidades.
O **Jogador Principal** pode a partir do menu inicial escolher **Ver Histórico** dos textos ou **Jogar**, sendo esta a funcionalidade principal.
Aquando a escolha da opção **Ver Histórico** o utilizador é capaz de **Selecionar Entrada**. Depois de selecionada é possível **Apagar** a mesma ou **Guardar Texto** no dispositivo.
É possível ao ator principal começar a **Jogar** e **Escrever** e **Ver Texto Final** sem personalizar o seu jogo, sendo que, por definição, o número de jogadores são dois, o número de rondas cinco e o número de caracteres cento e quarenta.
Quando o jogo termina é possível **Partilhar** o resultado ou **Guardar Texto** no aparelho.
O utilizador principal pode ainda **Personalizar Jogo** alterando o **Nome do Jogador**, **Cor do Jogador**, **Número de Jogadores**, **Número Máximo de Caracteres**, **Número de Rondas** podendo escolher um **Número de Rondas Aleatório**, ou pode selecionar a opção do **Próximo Jogador Aleatório**.
O(s) **Jogador(es) Participante(s)** apenas precisam de **Escrever**.

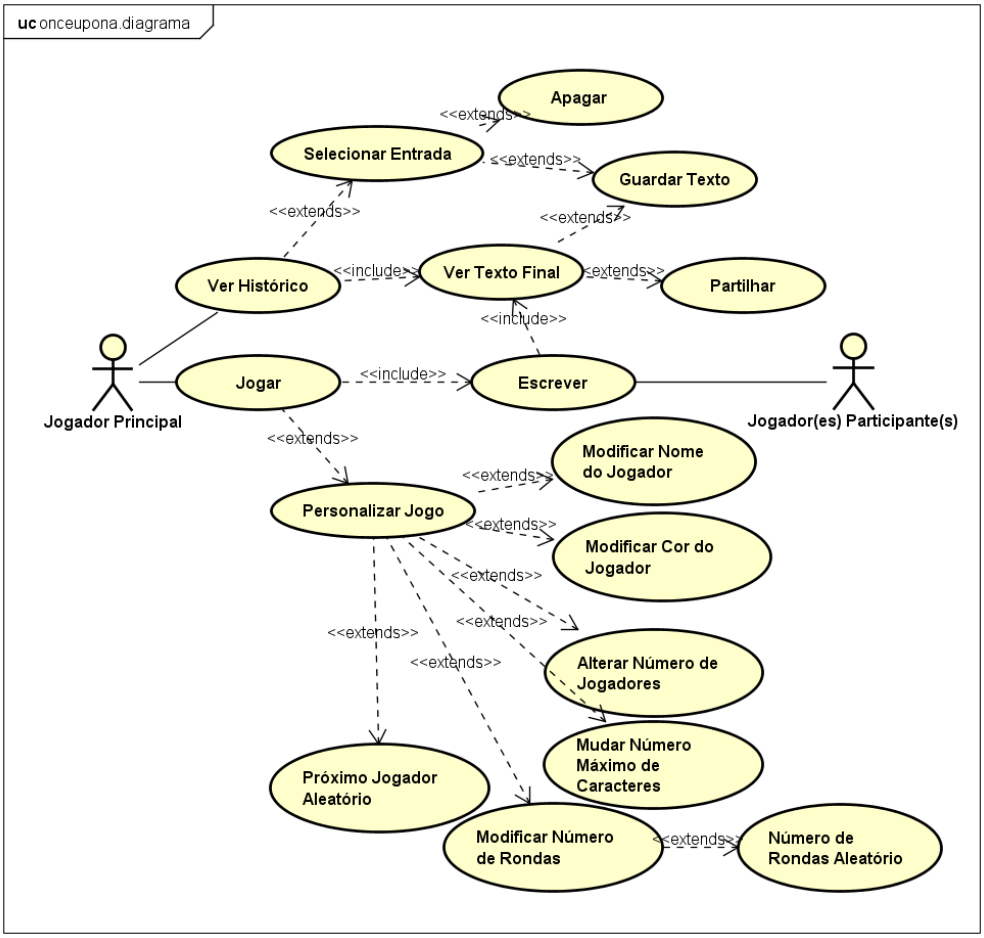

Figura 2.1: Diagrama de Casos de Uso do Sistema.

2.5 Diagrama de Classes

As classes abaixo são extensões da classe *Activity*.

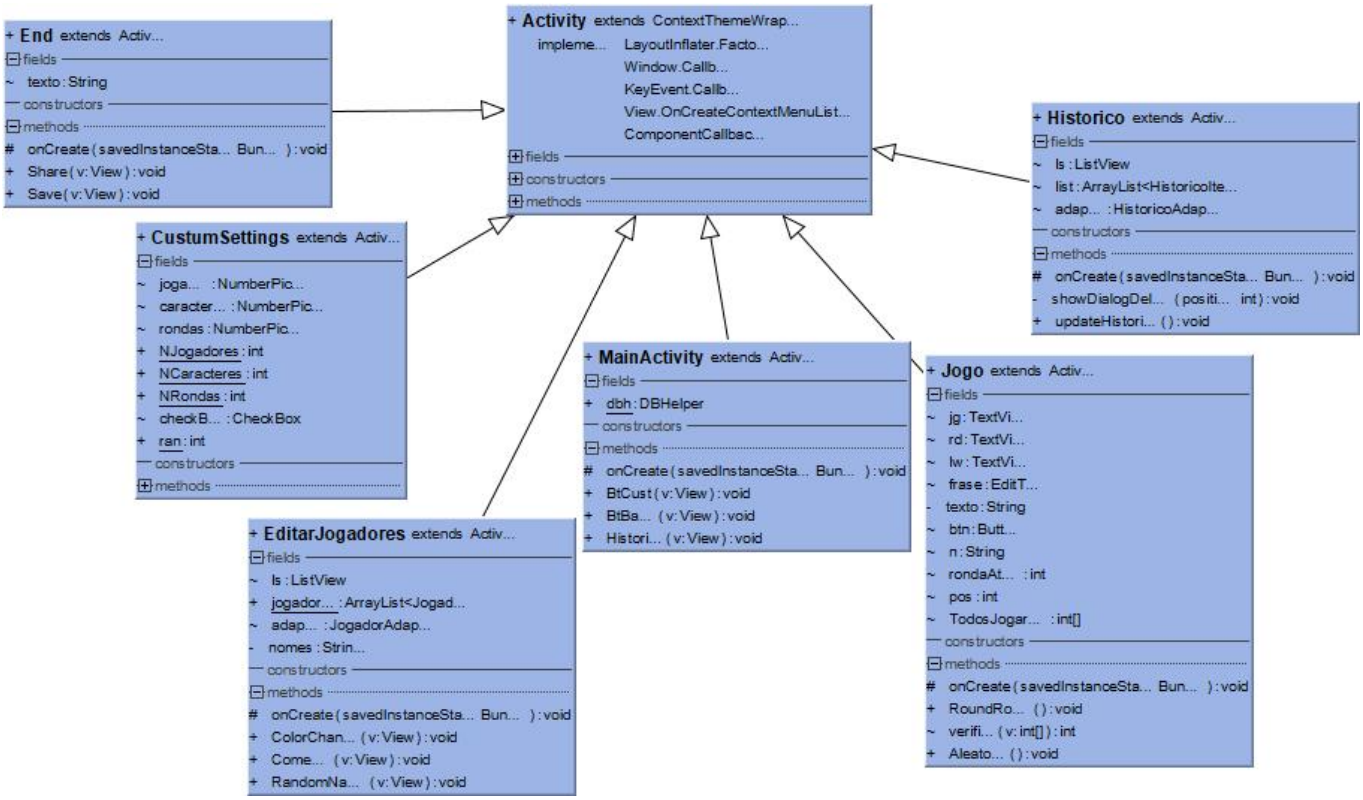

Figura 2.2: Extensões da Classe *Activity*.

Abaixo é visível o diagrama de classes da aplicação. Este representa como as classes existentes se relacionam entre si.

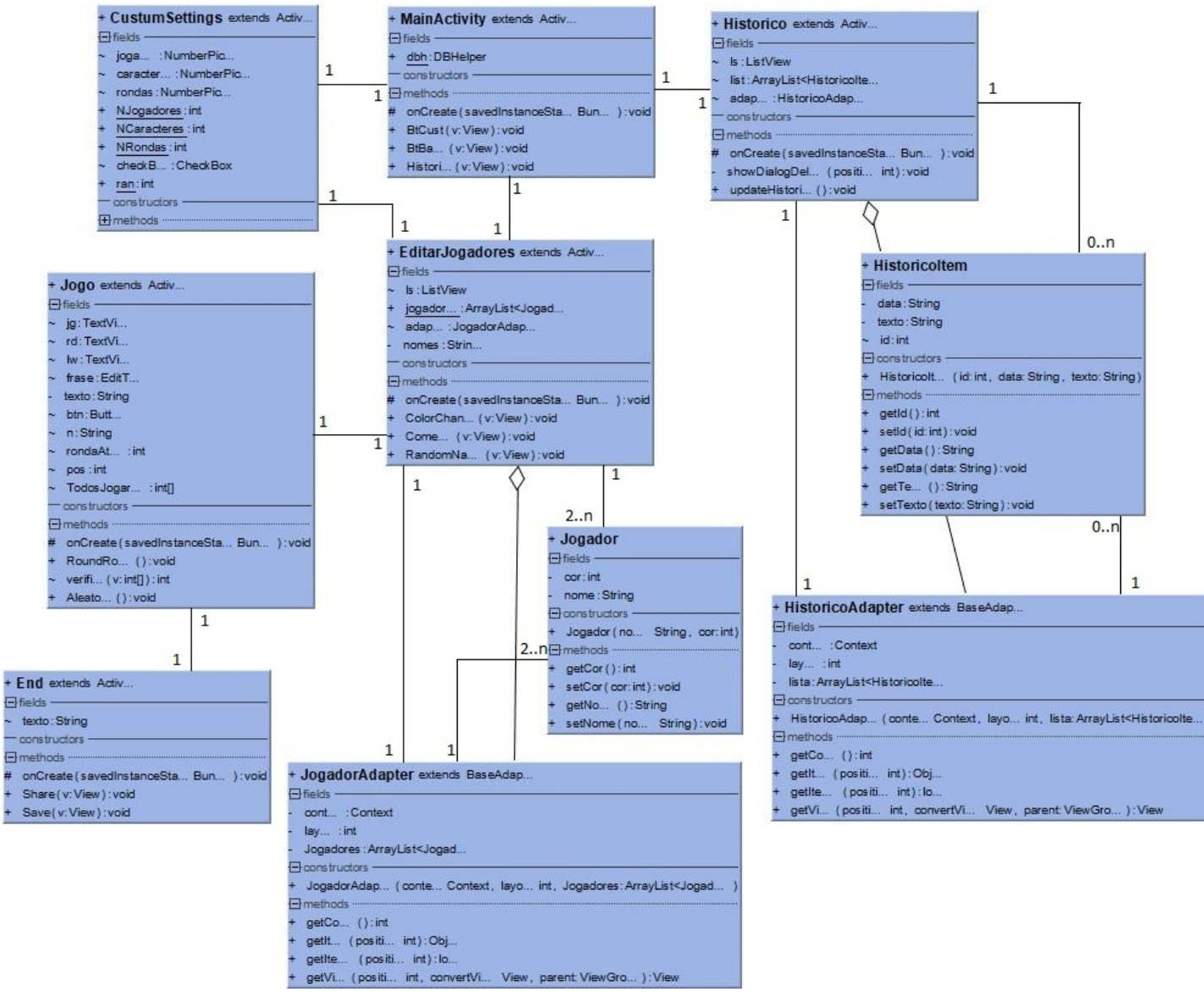

Figura 2.3: Diagrama de Classes da Aplicação.

No diagrama seguinte estão visíveis as classes que utilizam a base de dados.

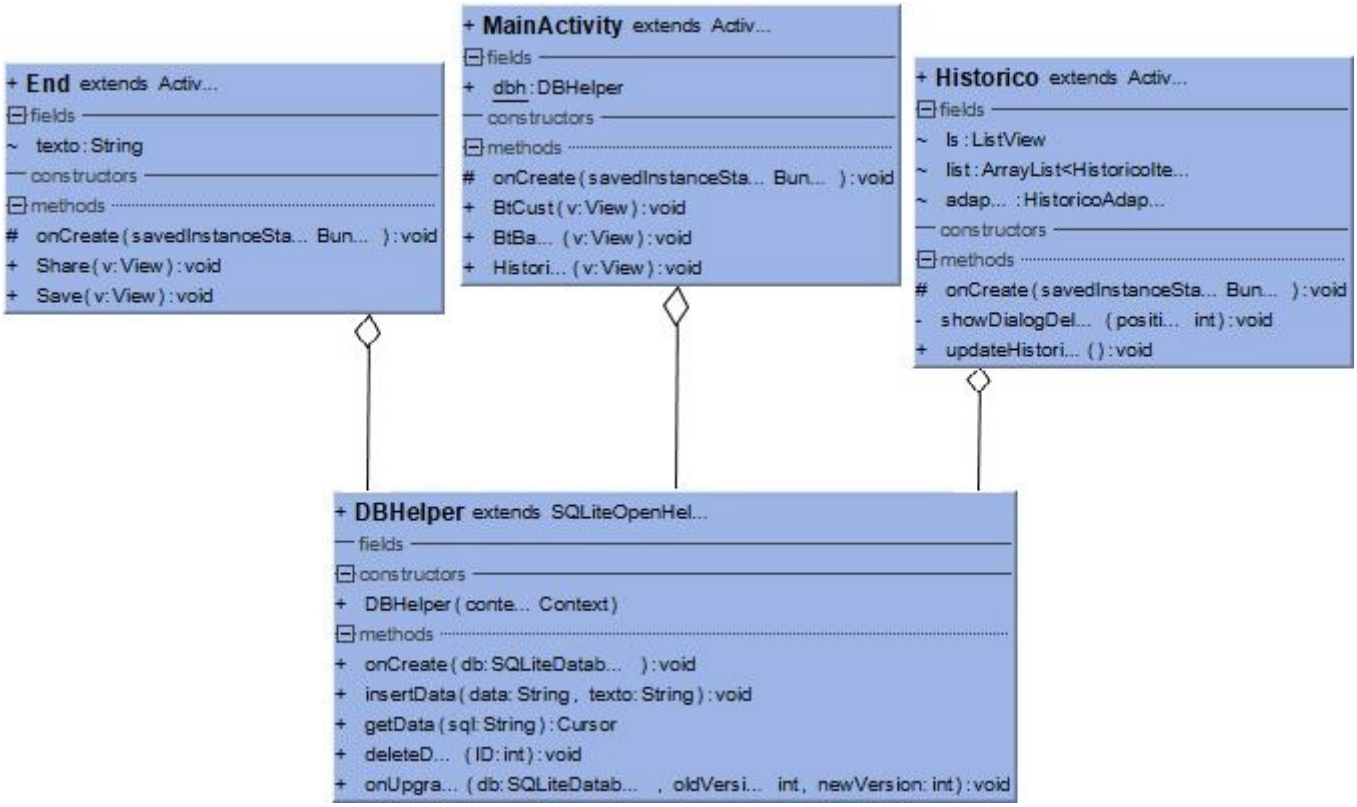

Figura 2.4: Diagrama de Associações com a Base de Dados.

2.6 Diagramas de Atividades

Nesta secção são apresentados os diagramas de atividades. Estes traduzem os fluxos das ações necessárias para chegar ao estado desejado. Estando este estado descrito na legenda e no título da figura.

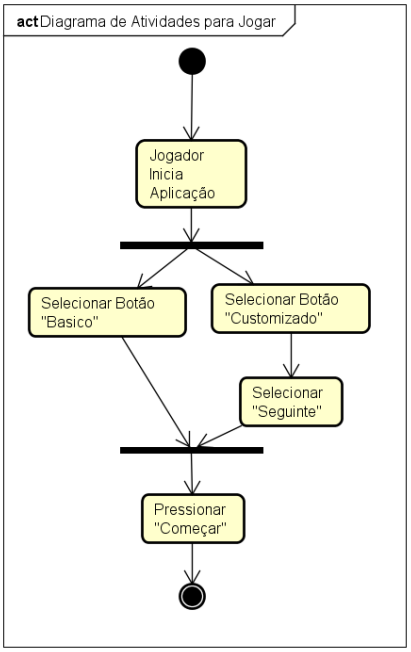

Figura 2.5: Fluxo de Atividades para Jogar.

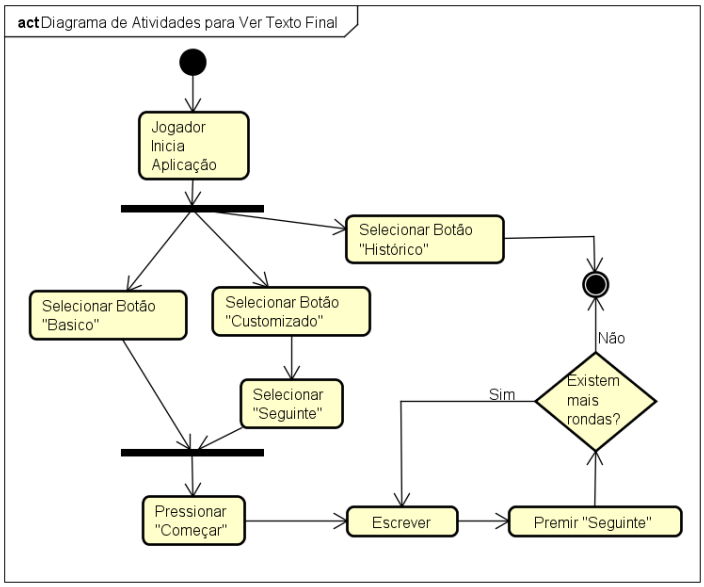

Figura 2.6: Fluxo de Atividades para Visualizar o Texto Final.

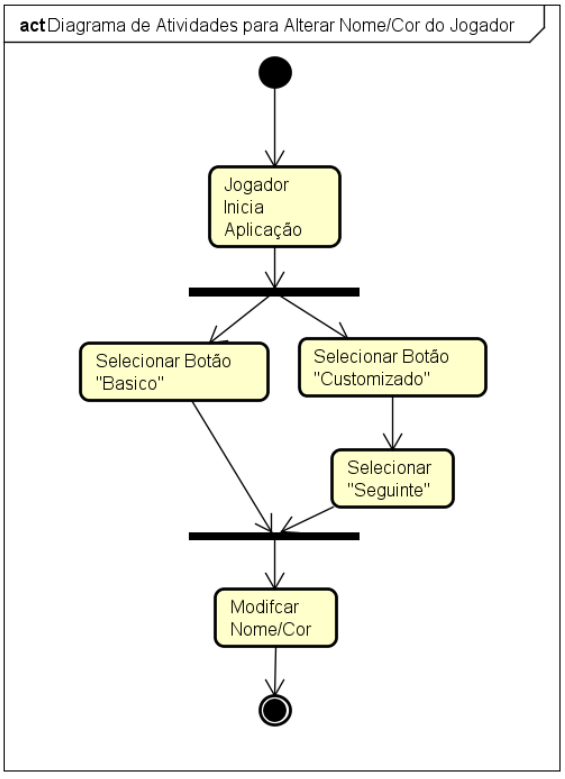

Figura 2.7: Fluxo de Atividades para Modificar a Nome ou a Cor do Jogador.

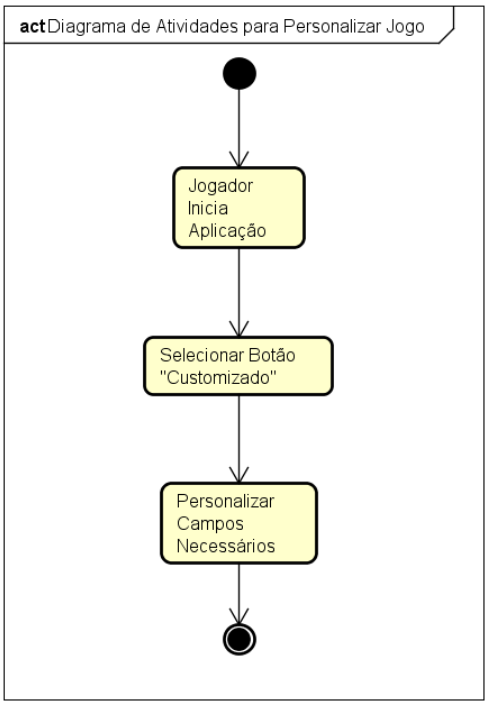

Figura 2.8: Fluxo de Atividades para Personalizar o Jogo.

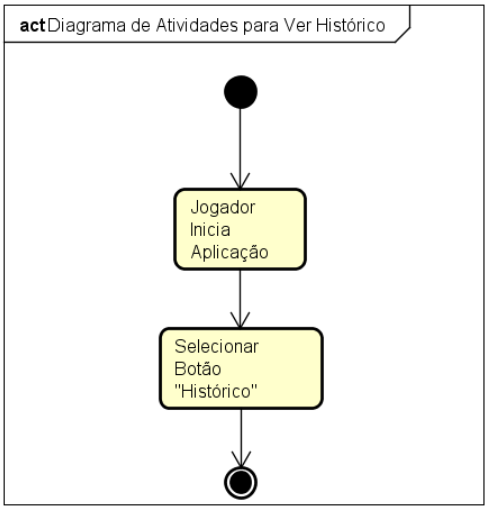

Figura 2.9: Fluxo de Atividades para Ver o Histórico de Jogos.

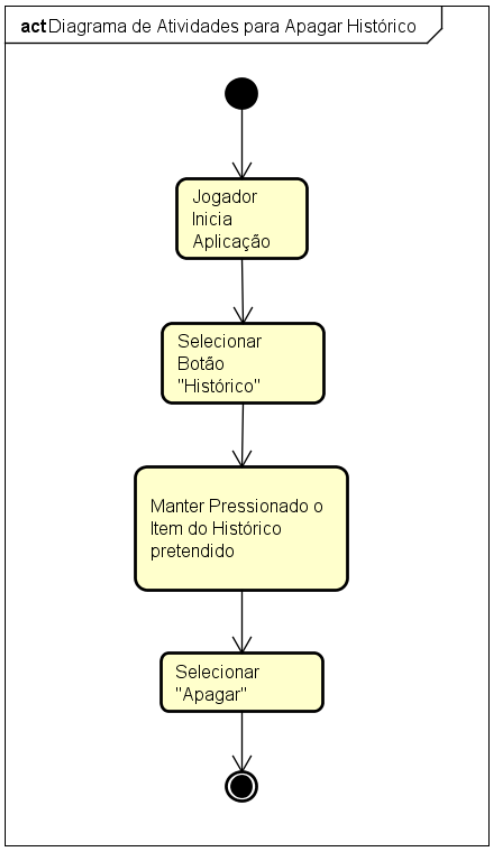

Figura 2.10: Fluxo de Atividades para Apagar determinada Entrada do Histórico.

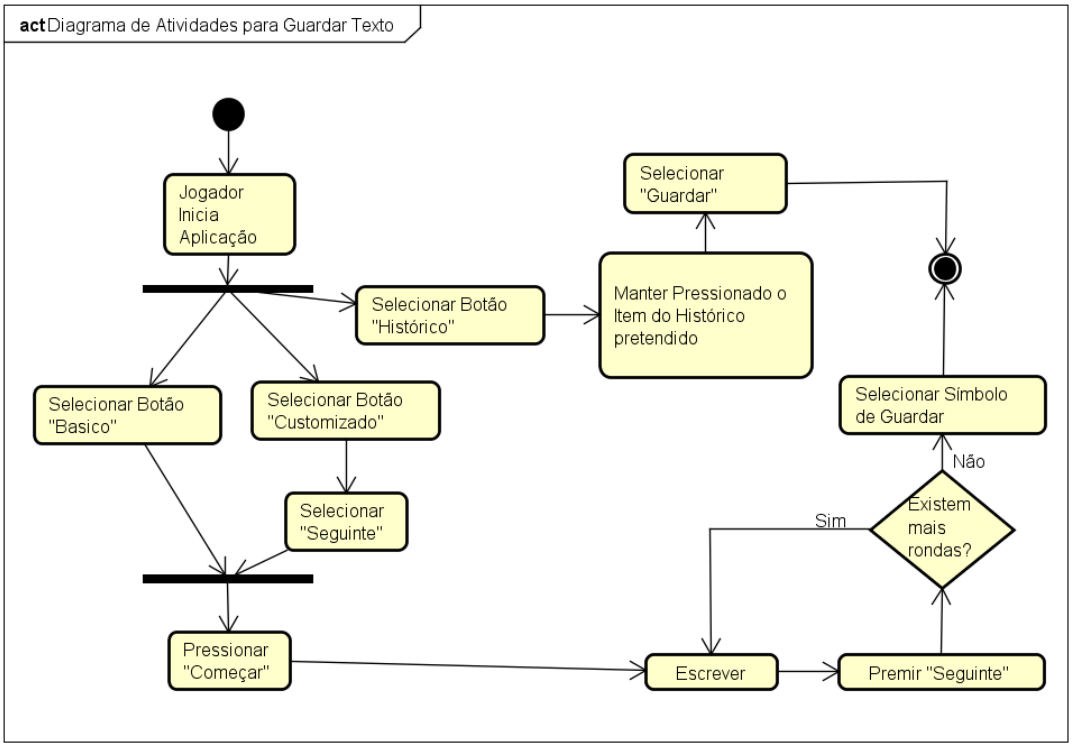

Figura 2.11: Fluxo de Atividades para Guardar o Texto Final.

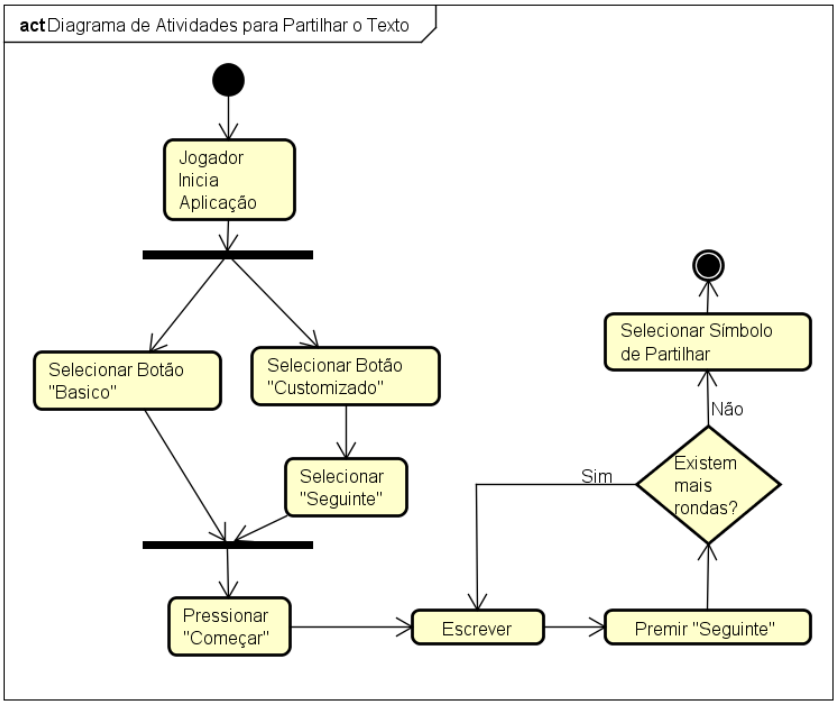

Figura 2.12: Fluxo de Atividades para partilhar o Texto Final.

Capítulo 3

Implementação

3.1 Introdução

Neste capítulo serão apresentadas as escolhas de implementação,um pequeno manual de instalação e utilização, bem como uma introdução ao *layout* da aplicação. Estando organizado da seguinte forma:

- Escolhas de Implementação (3.2) ;
- *Layout* (3.3);
- Manual de Instalação (3.4);
- Manual de Utilização (3.5).

3.2 Escolhas de Implementação

Inicialmente, o jogador terá de escolher entre o modo "básico"e "customizado".

Se optar pelo modo básico, saber-se-á à partida que serão dois jogadores, que serão 140 caracteres, e que serão feitas quatro rondas. Caso o utilizador opte por escolher o modo customizado, terá de ser o utilizador a escolher o número de jogadores, o número máximo de carateres, a ordem pela qual os jogadores jogarão (*Round Robin* ou aleatório) e o número de rondas.

```
public void BtBasic(View v){
        CustumSettings.NJogadores = 2;
        CustumSettings.NCaracteres = 140;
        CustumSettings.NRondas = 5;
        startActivity(new Intent(this,EditarJogadores.class));
    }
```

Listing 3.1: Implementação caso o utilizador escolha o modo básico

De seguida, no separador "Jogadores"são dadas ao utilizador as funcionalidades de alterar a cor do jogador e alterar o nome do jogador, podendo escolher nomes aleatórios já inseridos no código fonte, ou escrever o nome pretendido. Estes jogadores são inseridos de forma dinâmica na *ListView*, de acordo com o número de jogadores escolhidos anteriormente, ou, no caso do modo básico, por pré-definição são dois. Caso esta inserção fosse estática, teríamos de criar no *Extensible Markup Language* (XML), dez (ou mais) jogadores, e colocá-los visíveis ou invisíveis, de acordo com o número de jogadores que teria sido escolhido. Por este motivo, achámos que seria mais eficiente inserir os jogadores dinamicamente na *ListView*.

```
jogadores = new ArrayList<>(); //cria uma nova lista com Njogadores
        for(int i=0;i<CustumSettings.NJogadores;i++){
            Jogador j = new Jogador("Jogador " + i,R.color.colorAccent);
            jogadores.add(j);
        }

        ls = findViewById(R.id.ListView);

        adapter = new JogadorAdapter(this,R.layout.jogador,jogadores); //
            Pega na lista de jogadores e insere cada jogador no Layout do
            jogador
        ls.setAdapter(adapter); // insere cada elemento numa posi  o da
            ListView
        adapter.notifyDataSetChanged();
```

Listing 3.2: Inserção dinâmica de jogadores na *ListView*

De seguida, o utilizador será encaminhado para o jogo propriamente dito, onde estão implementadas as regras que definiu anteriormente. No modo customizado, o utilizador poderá escolher entre *round robin* ou aleatório. No modo *round robin* a ordem do jogo é simplesmente a ordem pela qual os jogadores foram inseridos, enquanto que no modo aleatório são gerados aleatoriamente números que definem os jogadores, e portanto, ditará a ordem do jogo.

Para além do jogo em si, foram implementados alguns métodos para facilitar e tornar mais agradável a interação entre os jogadores e a aplicação, nomeadamente a definição de uma cor para cada jogador onde, no texto final, será possível identificar quem escreveu o quê.

Foi também implementado um histórico de jogos, que mostra ao utilizador o texto gerado em jogos anteriores, bem como a data e hora a que esses jogos se realizaram.

3.3 *Layout*

Foi decidido que esta aplicação teria um design simples, limpo e objetivo, de modo a ser intuitiva e de fácil utilização.
Para elaboração dos *layouts* foram criados *Mockups*, que serão apresentados a seguir, bem como uma breve explicação de cada um.

Caraterísticas comuns entre layouts:

- *Padding* de 5pt;
- Família tipográfica: *Lato*.

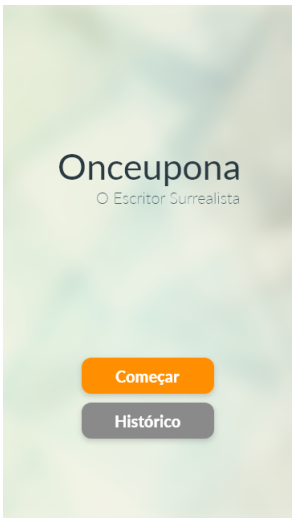

Figura 3.1: *Layout* Inicial

No ecrã inicial, a aplicação apresenta um *layout* simples apenas com os botões de "começar"e "histórico", que dão inicio ao jogo e acesso ao histórico da aplicação, respetivamente.

Ao começar o jogo, é dada ao utilizador a hipótese de escolher o modo de jogo, novamente com apenas duas opções, antecedidas de uma breve descrição de cada modo, de forma a ajudar o utilizador na escolha.

No modo customizado, as definições são geridas com a ajuda de *number pickers*, um *switch* e um *radio button*, pois achamos que serão as ferramentas mais intuitivas para o efeito pretendido. Através deste *layout* o utilizador poderá personalizar os diferentes aspetos do jogo.

Através deste *layout* pretende-se que o utilizador personalize os nomes dos jogadores, assim como as suas cores. Além destas opções, existe ainda uma outra opção que permite gerar nomes através do ícone de *shuffle*, o qual funciona como um botão. No fundo do *layout*, existe ainda um último botão com o ícone de *start* que permite a inicialização do jogo.

Este *layout* é referente ao jogo propriamente dito. É através dele que os jogadores poderão ver qual a vez de cada jogador, o número da ronda atual, número de rondas totais e as últimas

Figura 3.2: *Layout* de escolha do modo

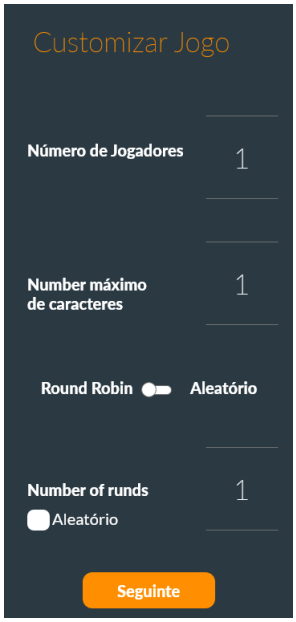

Figura 3.3: *Layout* de Customização do jogo

duas palavras escritas pelo jogador anterior. É também apresentado um botão que permite passar de utilizador em utilizador até ao até ao fim das rondas. Quando as rondas chegam ao fim, o utilizador é redirecionado para o próximo *layout*, tal como é representado na figura 3.6.

Neste *layout* será mostrado ao utilizador o texto gerado durante o jogo, bem como três botões:

- Botão de *share*, que permite ao utilizador fazer a partilha do texto gerado;
- Botão de Recomeçar, que permite recomeçar um novo jogo, voltando ao primeiro *layout*,

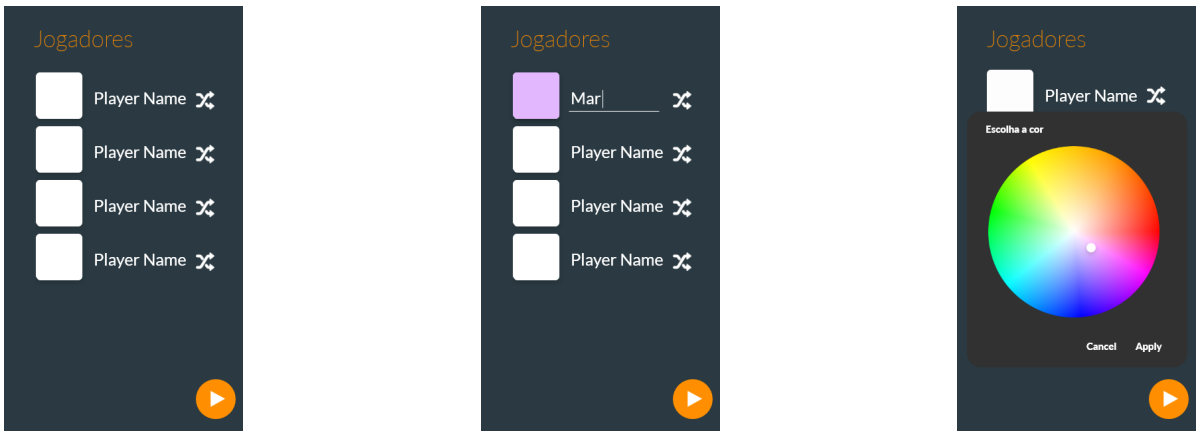

Figura 3.4: *Layouts* dos jogadores

Figura 3.5: *Layout* do jogo

tal como é apresentado na figura 3.1;

- Botão de guardar, que permite guardar o texto gerado num ficheiro *.txt*, com um nome à escolha.

O utilizador através destes *layouts* poderá ver o histórico de todos os textos gerados, bem como guardá-los ou apagá-los.

Todos as figuras/*mockups* apresentados anteriormente, são apenas uma propostas de como os *layouts* deveriam ser construídos. Na construção da aplicação, estes *layouts* sofreram algumas alterações pontuais, respeitando os *mockups* na maioria das situações.

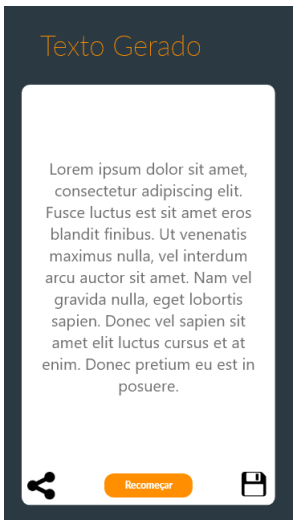

Figura 3.6: *Layout* do Texto Gerado

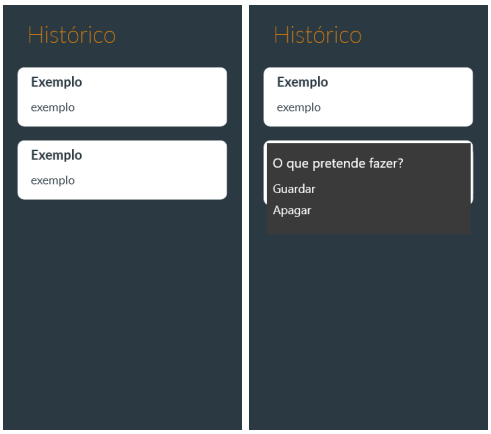

Figura 3.7: *Layouts* do Histórico

3.4 Manual de Instalação

Sendo que se trata de uma aplicação *android*, a mesma poderá ser adquirida na *Google Play Store* sendo que a instalação será semelhante a outra qualquer aplicação adquirida nessa plataforma.

3.5 Manual de Utilização

A aplicação começa com um *layout* inicial, onde é possível o utilizador consultar o histórico de jogos finalizados, ou começar um novo jogo.
Caso o utilizador aceda ao histórico, é possível consultar textos escritos anteriormente, bem como a data e hora de realização do jogo. Sendo possível eliminar ou guardar algum dos textos existentes.

Se se proceder para o começo de jogo o utilizador será redirecionado para um *layout* onde se escolhe o modo de jogo, podendo este ser o modo básico ou o modo customizado. Aquando esta escolha o utilizador será redirecionado para o *layout* onde se poderá dar início ao jogo, caso escolha o modo básico, ou, no caso de escolher o modo customizado, então será redirecionado para a página de customização do jogo.
No *layout* que permite a customização o utilizador deparar-se-á com diversas opções de customização. Sendo que nesta estarão presentes duas opções de customização de jogo e do jogador. Quanto ao jogo as opções que o utilizador pode personalizar são a quantidade máxima de carateres e a número de rondas que o jogo terá. Quanto ao jogador o conteúdo personalizável é o nome e a cor com a qual o seu texto será visualizado. A ordem do jogo, e o número de rondas podem ser escolhidas aleatoriamente, caso o utilizador escolha essa opção. Após a customização o utilizador será redirecionado para o *layout* do jogo assim que pressionar o botão que finaliza a personalização.
O *layout* do jogo é simples e intuitivo. Numa faixa superior deste, estará indicado o nome do jogador sobre o qual calha a vez de jogar naquela ronda, assim como o número atual da ronda. Após estas informações está presente uma caixa de texto branca onde o jogador precisará de introduzir o seu texto. A cima desta caixa estarão presentes as duas ultimas palavras escritas pelo jogador anterior.
No fim de todas as rondas, aparecerá o texto completo, escrito com as cores atribuidas a cada jogador. Aparecerá também a possibilidade de partilhar ou guardar o texto obtido, bem como o botão de recomeçar, encaminhando o utilizador para o *layout* inicial.

Capítulo 4

Reflexão Crítica e Problemas Encontrados

4.1 Introdução

Neste capítulo serão discutidos os objetivos propostos e os objetivos alcançados, bem como a divisão de trabalho pelos elementos do grupo e uma pequena reflexão crítica. Estando organizado da seguinte forma:

- Objetivos Propostos vs. Alcançados (4.2) ;
- Divisão de Trabalho pelos Elementos do Grupo (4.3);
- Problemas Encontrados (4.4);
- Reflexão Crítica (4.5).

4.2 Objetivos Propostos vs. Alcançados

Aquando a elaboração deste trabalho foi seguida a listagem de funcionalidades presente no enunciado do mesmo, sendo que tal listagem encontra-se subdividida numa enumeração das funcionalidades base da aplicação e das funcionalidades que deviam estar presentes numa versão mais elaborada da aplicação. As funcionalidades base são:

- Jogos de multi-jogadores em que será usado um único dispositivo móvel;
- Escolha do número de jogadores e da quantidade máxima de caracteres aceite;
- Registo do nome dos jogadores ou geração automática do nome do jogador;
- Modo de jogo pré-definido (jogo básico) com dois jogadores e número máximo 140 caracteres;
- Dois modo de jogo distintos quando existem dois ou mais jogadores, o modo *round robin*(a ordem dos jogadores nunca se altera) e modo aleatório (a ordem dos jogadores é alterada em cada ronda);

- Personalização do número de rondas;
- No fim do jogo o texto final é mostrado na sua totalidade e poderá ser guardado ou partilhado;
- Aceder ao histórico de texto.

Tendo todas elas sido implementadas com sucesso.
Das funcionalidades que foram propostas como trabalho adicional as seguintes foram concretizadas:

- Existência de cores diferentes para a melhor identificação dos jogadores.

4.3 Divisão de Trabalho pelos Elementos do Grupo

O trabalho possui quatro componentes distintas sendo elas o código da aplicação, a interface gráfica da aplicação, o teste da aplicação e o relatório do trabalho elaborado. Para a melhor gestão de tempo e para a realização dinâmica do projeto optou-se pela seguinte divisão:

- Código realizado por Alexandre Antão e João Domingos;
- Interface gráfica realizada por Eduardo Paulos;
- Testes à aplicação realizados por Miguel Gregório;
- Relatório realizado por Eduardo Paulos, Inês Lopes, João Domingos e Miguel Gregório.

4.4 Problemas Encontrados

Durante a implementação do código que constituí as funcionalidades base do trabalho o grupo não se deparou com grandes dificuldades, contudo houve uma dificuldade (problema) aquando a implementação do código que permite a edição dos jogadores. Este problema residia na implementação do código referente à adição de mais jogadores, onde não se consiga com que se pudesse adicionar mais do que sete jogadores e, quando este problema foi superado, a aplicação atribuía de maneira equívoca os dados associados a cada jogador (este problema foi, também, superado com sucesso). Na implantação de código referente às funcionalidades extra da aplicação o grupo não conseguiu implementar nenhuma para além da cor dos jogadores (como referido no 4.2), pelo que se pode considerar os outros pontos das funcionalidades extra como um problema que não se conseguiu superar.

4.5 Reflexão Crítica

Apesar de termos cumprido todas as funcionalidades obrigatórias faltaram fazer algumas funcionalidades adicionais, devido a má gestão de tempo, e falta de comunicação do grupo em período de pausa letiva. No entanto, as propostas adicionais não deixam de ser objetivos a melhorar e/ou desenvolver no futuro.

Capítulo 5

Conclusões e Trabalho Futuro

5.1 Conclusões Principais

Com a finalização deste trabalho conseguimos desenvolver a nossa capacidade de gerir tarefas e de trabalhar em equipa, acrescentamos ao nosso conhecimento tudo o que aprendemos ao desenvolver esta aplicação, somando assim à nossa experiência enquanto programadores mais conhecimento sobre o desenvolvimento de aplicações em ambiente *android*.

5.2 Trabalho Futuro

Como já foi referido no capítulo anterior, no nosso trabalho faltam algumas das funcionalidades opcionais, das quais gostaríamos de implementar num futuro próximo, essas são as seguintes:

- Permitir jogar em dispositivos diferentes através da rede de área local, servidor remoto ou outras tecnologias de comunicação;
- Implementar uma forma de eleger um jogador vencedor no final de cada jogo (e.g., o jogador que escreveu mais perto do número máximo de caracteres a cada ronda ou o jogador que, na opinião dos outros, teve mais piada na sua intervenção);
- Implementar um *ranking*;
- e fazer uma versão do jogo utilizando apenas voz (e som), ao invés de texto.

Bibliografia