

# binca

SEGURANÇA + ENTRETENIMENTO

## pre-entrega06

## ÍNDICE

Introdução -----	Pg1
Objetivos -----	Pg2
Diagrama de Gantt -----	Pg4
Requisitos Funcionais -----	Pg6
Lista de Bugs -----	Pg12

## INTRODUÇÃO

Um bom trabalho exige sempre um bom planeamento. Como tal, a fim de conseguirmos desenvolver da melhor forma o nosso protótipo, foi necessário delinear alguns aspectos, desde a definição de objetivos/metast a desenvolver até à atualização do diagrama de *gantt*.

Vamos então apresentar uma lista dos nossos objetivos a desenvolver; o diagrama de *gantt* atualizado com as tarefas já desenvolvidas e com as tarefas a desenvolver, tanto para esta entrega, como para a entrega final; uma lista com os requisitos funcionais atualizados e por fim a lista de *bugs*, incluindo os *bugs* já encontrados até ao momento e os *bugs* que esperamos encontrar no futuro.

## OBJETIVOS

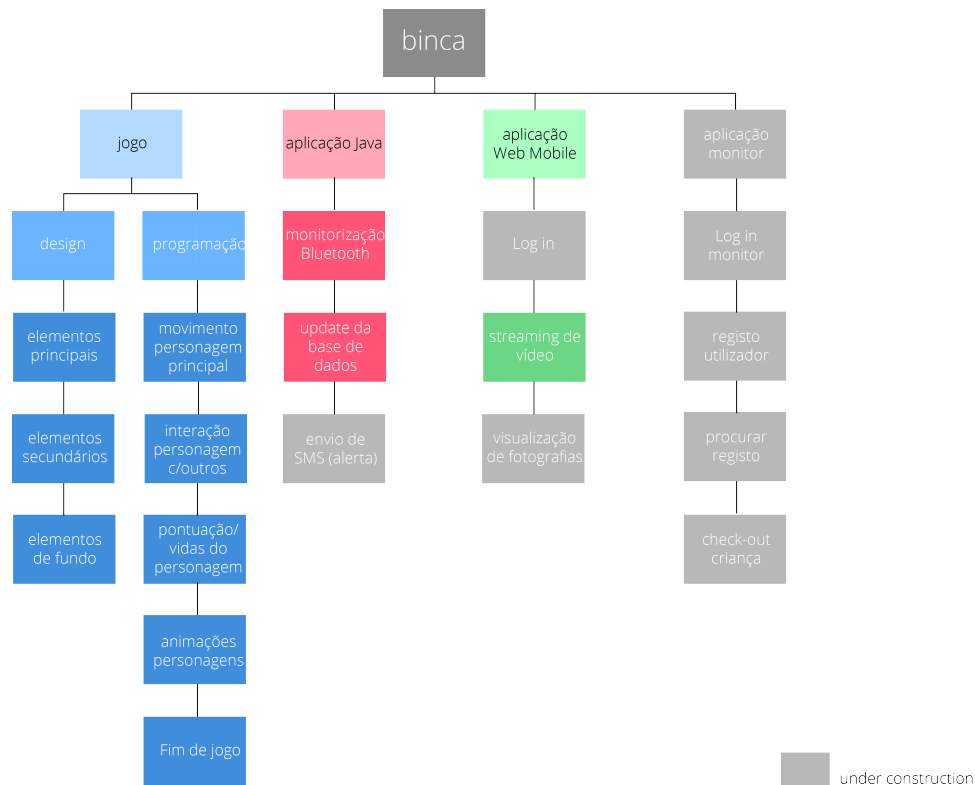
Para a entrega 06, achámos importante realçar quais os objetivos principais do projeto. Para isso, elaborámos um esquema com todas as funcionalidades do projeto, dividido por cada área e diferenciado por cores. A azul, está representada a área dos jogos, o cor de rosa a aplicação Java, a verde, a aplicação mobile e a amarelo a aplicação do monitor, tal como indica a seguinte imagem – *imagem 1*. As várias fases estão organizadas de uma forma hierárquica, onde o primeiro nível de informação (Jogo, aplicação *Java*, Aplicação *Web* para *mobile* e aplicação monitor) representam as principais áreas do projeto e as restantes, situadas em níveis inferiores a estas, são as tarefas pertencentes a essas áreas principais.



**Imagem 1:** Representação das funcionalidades do projeto.

A fim de planearmos a próxima entrega – dia 8 de Junho – definimos os objetivos/etapas que pretendemos atingir. Assim, e para ajudar a clarificar o que pretendemos fazer, optámos por fazer um esquema idêntico ao anterior, com as diferentes fases, diferenciando as já concluídas e as ainda por concluir – *imagem 2* – assinaladas a cor e a cinzento, respetivamente. Um dos objetivos que propomos são os seguintes: design do jogo *Encontra* e programação do mesmo, incluindo o sistema de ajuda de interação e sistema de ajuda sonoro. Optámos por concluir este jogo primeiro, devido ao seu elevado grau de interesse e motivação, pois um jogo que desafia mais a atenção do utilizador torna-se mais interessante. Outro objetivo, é a conclusão a 100% do jogo anteriormente já começado – *Gato Mergulhador* - onde falta apenas integrar o sistema de ajuda

de interação e o sistema de ajuda sonoro. Pretendemos também até a data da entrega, começar a desenvolver a captação de vídeo via *streaming* na aplicação *web* para *mobile* e também ter já terminada a aplicação *Java* para a monitorização *bluetooth* e para o *update* de dados na base de dados. Abaixo estão então representadas as funcionalidades que se pretendem implementar até à próxima entrega.



**Imagem 2:** Representação das funcionalidades para a #entrega06.

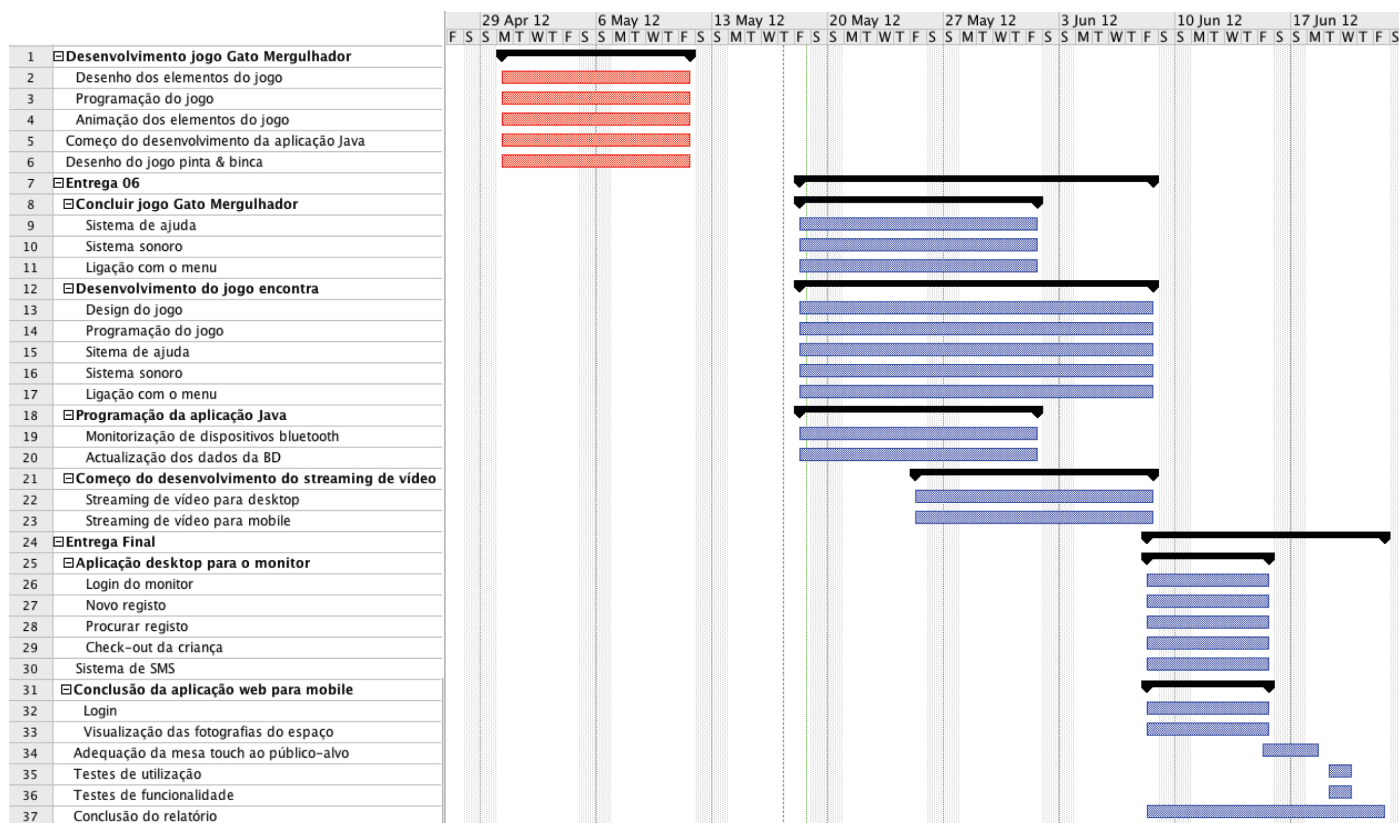
## DIAGRAMA DE GANTT

A fim de estruturar melhor as nossas tarefas e observar o calendário de realização das mesmas, desenhámos um diagrama de *gantt*. Neste estão representadas 3 categorias de tarefas: tarefas já desenvolvidas e concluídas; tarefas a desenvolver até à entrega 06; tarefas a desenvolver até à entrega final.

	Name	Duration	Start	Finish
1	<b>Desenvolvimento jogo Gato Mergulhador</b>	10 days?	4/30/12 8:00 AM	5/11/12 5:00 PM
2	Desenho dos elementos do jogo	10 days?	4/30/12 8:00 AM	5/11/12 5:00 PM
3	Programação do jogo	10 days?	4/30/12 8:00 AM	5/11/12 5:00 PM
4	Animação dos elementos do jogo	10 days?	4/30/12 8:00 AM	5/11/12 5:00 PM
5	Começo do desenvolvimento da aplicação Java	10 days?	4/30/12 8:00 AM	5/11/12 5:00 PM
6	Desenho do jogo pinta & binca	10 days?	4/30/12 8:00 AM	5/11/12 5:00 PM
7	<b>Entrega 06</b>	16 days?	5/18/12 8:00 AM	6/8/12 5:00 PM
8	<b>Concluir jogo Gato Mergulhador</b>	11 days?	5/18/12 8:00 AM	6/1/12 5:00 PM
9	Sistema de ajuda	11 days?	5/18/12 8:00 AM	6/1/12 5:00 PM
10	Sistema sonoro	11 days?	5/18/12 8:00 AM	6/1/12 5:00 PM
11	Ligação com o menu	11 days?	5/18/12 8:00 AM	6/1/12 5:00 PM
12	<b>Desenvolvimento do jogo encontra</b>	16 days?	5/18/12 8:00 AM	6/8/12 5:00 PM
13	Design do jogo	16 days?	5/18/12 8:00 AM	6/8/12 5:00 PM
14	Programação do jogo	16 days?	5/18/12 8:00 AM	6/8/12 5:00 PM
15	Sistema de ajuda	16 days?	5/18/12 8:00 AM	6/8/12 5:00 PM
16	Sistema sonoro	16 days?	5/18/12 8:00 AM	6/8/12 5:00 PM
17	Ligação com o menu	16 days?	5/18/12 8:00 AM	6/8/12 5:00 PM
18	<b>Programação da aplicação Java</b>	11 days?	5/18/12 8:00 AM	6/1/12 5:00 PM
19	Monitorização de dispositivos bluetooth	11 days?	5/18/12 8:00 AM	6/1/12 5:00 PM
20	Actualização dos dados da BD	11 days?	5/18/12 8:00 AM	6/1/12 5:00 PM
21	<b>Começo do desenvolvimento do streaming de vídeo</b>	11 days?	5/25/12 8:00 AM	6/8/12 5:00 PM
22	Streaming de vídeo para desktop	11 days?	5/25/12 8:00 AM	6/8/12 5:00 PM
23	Streaming de vídeo para mobile	11 days?	5/25/12 8:00 AM	6/8/12 5:00 PM
24	<b>Entrega Final</b>	11 days?	6/8/12 8:00 AM	6/22/12 5:00 PM
25	<b>Aplicação desktop para o monitor</b>	6 days?	6/8/12 8:00 AM	6/15/12 5:00 PM
26	Login do monitor	6 days?	6/8/12 8:00 AM	6/15/12 5:00 PM
27	Novo registo	6 days?	6/8/12 8:00 AM	6/15/12 5:00 PM
28	Procurar registo	6 days?	6/8/12 8:00 AM	6/15/12 5:00 PM
29	Check-out da criança	6 days?	6/8/12 8:00 AM	6/15/12 5:00 PM
30	Sistema de SMS	6 days?	6/8/12 8:00 AM	6/15/12 5:00 PM
31	<b>Conclusão da aplicação web para mobile</b>	6 days?	6/8/12 8:00 AM	6/15/12 5:00 PM
32	Login	6 days?	6/8/12 8:00 AM	6/15/12 5:00 PM
33	Visualização das fotografias do espaço	6 days?	6/8/12 8:00 AM	6/15/12 5:00 PM
34	Adequação da mesa touch ao público-alvo	2 days?	6/15/12 8:00 AM	6/18/12 5:00 PM
35	Testes de utilização	2 days?	6/19/12 8:00 AM	6/20/12 5:00 PM
36	Testes de funcionalidade	2 days?	6/19/12 8:00 AM	6/20/12 5:00 PM
37	Conclusão do relatório	11 days?	6/8/12 8:00 AM	6/22/12 5:00 PM

**Imagem 3** - Tarefas com respectiva calendarização e duração

Em cima estão então representadas as várias tarefas a desenvolver, estando algumas delas divididas em sub-tarefas de forma a serem representadas de forma mais estruturada. À frente de cada tarefa encontra-se a sua data de fim e de início, bem como a duração correspondente.



**Imagem 4** - Tarefas com respectiva calendarização gráfica

Tal como se pode observar, agora graficamente, em primeiro lugar temos as tarefas até agora concluídas (a vermelho) e de seguida todas as tarefas a realizar no futuro, para ambas as entregas (a azul).

## REQUISITOS FUNCIONAIS

Com o objetivo de uma melhor compreensão dos requisitos funcionais, optámos por dividi-los em categorias/partes do sistema e dentro destas distinguir entre *frontoffice* e *backoffice*. Para além desta divisão, optámos também por dividi-los em duas fases importantes para as entregas: dia 8 de Junho – entrega 06 - e dia 23 de Junho – data da entrega final do projeto.

### ENTREGA06 – 8 JUNHO

#### JOGO

##### *FrontOffice:*

- 1- O utilizador tem de poder escolher um jogo dentro da lista dos jogos apresentados.
- 2- Para que o utilizador entenda como deve jogar, vai existir um ecrã do tipo demo em que o jogador poderá ou não interagir com o sistema ao mesmo tempo que ouve as instruções.
  - 2.1- De forma a que o utilizador não seja obrigado a ouvir as instruções, é-lhe dito no início que caso não queira ouvir as instruções, poderá tocar num botão “avançar”.
- 3- A superfície vai ser táctil, de modo a que o utilizador possa interagir com o jogo através do toque.
- 4- Para que a aplicação seja mais dinâmica, irão existir sons de fundo e de toque.
- 5- Todos os jogadores irão jogar para a mesma pontuação, sendo que a pontuação irá incrementar à medida que os jogadores ganhem pontos.
- 6- Todos os jogos irão ter um botão “X”, onde o jogador poderá sair do jogo.

##### *BackOffice:*

- 1- Cada jogo irá ter um tipo de interação diferente: toque e arrastar.
- 2- Irá existir um computador para correr o jogo e monitorizar todo o processo de interação.
- 3- O jogo vai estar preparado para receber e reagir ao input do utilizador.
  - 3.1-Para receber o input do utilizador, vão estar ligadas ao computador as câmaras infravermelhas, de forma a atuar como sensor de toque.



3.2- Através do programa *CCV(Community Core Vision)*, vai ser possível identificar os pontos de toque na mesa *touch*.

3.3- Para reagir ao *input* do utilizador, o *ActionScript 3.0* irá processar os dados transmitidos pelo *CCV* e reagir.

#### CENTRAL DE PROCESSAMENTO (MAC MINI)

##### *Frontoffice:*

1- O computador vai mostrar ao utilizador (criança), através do projetor, a imagem do jogo na mesa *touch*.

##### *Backoffice:*

1- O computador vai servir de suporte ao jogo, correndo o *software* necessário à sua correta execução.

2- Vai receber os vários sinais *Bluetooth* emitidos pelos pontos de acesso (que por sua vez captaram o sinal emitido pelas pulseiras *Bluetooth*).

2.1- Através de uma aplicação *Java* vai monitorizar o sinal dos vários pontos de acesso, de forma a ter conhecimento da localização relativa da criança (como explicado no *post #reunião02*).

2.2- Após a análise da localização da criança, vai ser atualizado o seu estado na base de dados.

3- Vai servir de suporte físico para o servidor *web*, onde vai estar armazenada a base de dados.

4- Vai receber informação emitida pelas câmaras, tanto de vídeo como fotográfica, via *wi-fi*.

#### PULSEIRA BLUETOOTH

##### *Frontoffice:*

1- Cada criança vai possuir uma pulseira *Bluetooth*, sendo que esta vai estar numerada para diferenciação e identificação.

##### *Backoffice:*

1- Vai emitir sinal *Bluetooth* para os pontos de acesso.

## SISTEMA DE VIGILÂNCIA

### *Frontoffice:*

- 1- No local, vai existir uma webcam e uma câmara fotográfica.
- 2- Vão estar localizadas estrategicamente para conseguir dar cobertura a todo o espaço *Binca*.

### *Backoffice:*

- 1- A webcam vão transmitir imagem em direto, diretamente para o computador.
  - 2- De x em x tempo a câmara fotográfica vai captar uma imagem e transmiti-la, diretamente para o computador.
- 

## ENTREGA FINAL – 23 DE JUNHO

### JOGO

#### *FrontOffice:*

- 1- O utilizador tem de poder escolher um jogo dentro da lista dos jogos apresentados.
- 2- Para que o utilizador entenda como deve jogar, vai existir um ecrã do tipo demo em que o jogador poderá ou não interagir com o sistema ao mesmo tempo que ouve as instruções.
  - 2.1- De forma a que o utilizador não seja obrigado a ouvir as instruções, é-lhe dito no início que caso não queira ouvir as instruções, poderá tocar num botão “avançar”.
- 3- A superfície vai ser tátil, de modo a que o utilizador possa interagir com o jogo através do toque.
- 4- Durante o jogo, caso o utilizador tenha algum problema, o monitor responsável irá auxiliá-lo.
- 5- Para que a aplicação seja mais dinâmica, irão existir sons de fundo e de toque.
- 6- Todos os jogadores irão jogar para a mesma pontuação, sendo que a pontuação irá incrementar à medida que os jogadores ganhem pontos.

7- Todos os jogos irão ter um botão "X", onde o jogador poderá sair do jogo.

#### *BackOffice:*

1- Cada jogo irá ter um tipo de interação diferente.

2- Irá existir um computador para correr o jogo e monitorizar todo o processo de interação.

3- O jogo vai estar preparado para receber e reagir ao input do utilizador.

3.1- Para receber o input do utilizador, vão estar ligadas ao computador as câmaras infravermelhas, de forma a atuar como sensor de toque.

3.2- Através do programa CCV(*Community Core Vision*), vai ser possível identificar os pontos de toque na mesa *touch*.

3.3- Para reagir ao *input* do utilizador, o *software* escolhido para programar irá processar os dados transmitidos pelo CCV e reagir.

#### CENTRO DE PROCESSAMENTO (MAC MINI)

#### *Frontoffice:*

1- O computador vai mostrar ao utilizador (criança), através do projetor, a imagem do jogo na mesa *touch*.

2- Através de um ecrã externo, vai mostrar ao utilizador (pai/responsável) uma interface de registo.

#### *Backoffice:*

1- O computador vai servir de suporte ao jogo, correndo o *software* necessário à sua correta execução (como explicado em cima).

2- Vai receber os vários sinais *Bluetooth* emitidos pelos pontos de acesso (que por sua vez captaram o sinal emitido pelas pulseiras Bluetooth).

2.1- Através de uma aplicação Java vai monitorizar o sinal dos vários pontos de acesso, de forma a ter conhecimento da localização relativa da criança (como explicado no *post #reunião02*).

2.2- Após a análise da localização da criança, vai ser atualizado o seu estado na base de dados.

3- Vai servir de suporte físico para o servidor *web*, onde vai estar armazenada a base de dados.

4- Vai receber informação emitida pelas câmaras, tanto de vídeo como fotográfica.

4.1- Através de um servidor *Web*, vai emitir vídeo por *streaming*, para a aplicação.

#### PULSEIRA BLUETOOTH

##### *Frontoffice:*

1- Cada criança vai possuir uma pulseira *Bluetooth*, sendo que esta vai estar numerada para diferenciação e identificação.

##### *Backoffice:*

1- O *ID Bluetooth* de cada pulseira vai ser introduzido na base de dados.

2- Vai emitir sinal *Bluetooth* para os pontos de acesso.

#### BASE DE DADOS

##### *Frontoffice:*

1- Os pais/responsáveis das crianças vão poder introduzir os seus dados, nome e número de telefone na base de dados do sistema *Binca*.

2- Através de um *drop-down* menu vão poder escolher o número da pulseira da criança por que estão responsáveis, associando assim esta informação aos seus dados.

##### *Backoffice:*

1- A base de dados vai estar localizada num servidor *web* que estará fisicamente armazenado no *Mac mini* da mesa *touch*.

2- A base de dados vai guardar dois tipos de dados principais: os dados dos pais/responsáveis, nome e número de telefone; os dados relativos à identificação da criança, *id Bluetooth* da pulseira e número da pulseira (para ser mais simples a introdução de dados).

3- Vai possuir um parâmetro essencial para a notificação da localização da criança: o estado.

## SISTEMA DE SMS

### Frontoffice:

1- Os pais/responsáveis que não possuem sistema Android vão receber por SMS notificações sobre o estado da criança.

### Backoffice:

1- Envio de pacote *html* com o pedido e com os métodos a realizar para o *Web Service*.

2- Tratamento da resposta *xml* com os métodos realizados e a resposta a cada método.

## SISTEMA DE VIGILÂNCIA

### Frontoffice:

1- No local, vai existir uma webcam e uma câmara fotográfica.

2- Vão estar localizadas estrategicamente para conseguir dar cobertura a todo o espaço *Binca*.

### Backoffice:

1- A webcam vão transmitir imagem em direto, diretamente para o computador.

2- De x em x tempo a câmara fotográfica vai captar uma imagem e transmiti-la, diretamente para o computador.

De seguida, decidimos atualizar a nossa lista de funcionalidades, dividida nas várias tecnologias. Incluímos ainda os dois tipos de utilizadores do nosso sistema e que funcionalidades vão poder executar.

Tecnologia	Funcionalidade	Utilizador 1 - Criança	Utilizador 2 - Pais/responsáveis
Jogo	O utilizador vai poder escolher o jogo que deseja jogar	S	N
	O utilizador vai poder ouvir e ver as instruções do jogo	S	N
	O utilizador vai poder tocar em avançar e não visualizar as instruções	S	N
	O utilizador vai poder jogar e colecionar pontos	S	N
	O Utilizador vai poder tocar no "x" para sair do jogo	S	N
Aplicação Mobile	O Utilizador vai poder fazer login na aplicação	N	S
	O Utilizador vai poder visualizar streaming de vídeo do espaço Binca	N	S
	O Utilizador vai poder visualizar fotografias do espaço Binca	N	S
Computador	O Utilizador vai poder registar-se na base de dados Binca	N	S
Pulseira	O Utilizador vai possuir uma pulseira <i>Bluetooth</i> identificada	S	N
Sistema SMS	O Utilizador vai ser notificado por sms, do estado da criança	N	S

As funcionalidades assinaladas apenas constam nos objetivos para a entrega de dia 23 de Junho.

**Tabela 1** – Funcionalidades do sistema

## LISTA DE BUGS

### PROGRAMAÇÃO

#### JOGO – GATO MERGULHADOR

Durante a programação do jogo *touch* em AS3 (ActionScript 3.0), Gato Mergulhador, surgiram alguns problemas e erros, estando os mesmos apresentados em baixo.

**Tabela 1** - Lista de bugs de programação do jogo Gato Mergulhador

ID	Descrição	Solução	Terminado (S/N)	Total Erros por resolver	Grau (1-5)
1	Como obter uma intersecção fiel entre o personagem e os restantes elementos do jogo?	Definir vários pontos de intersecção e utilizar o <i>hitTestPoint</i> em vez de <i>Object</i>	S	0	2
2	Necessidade de colocar uma função a chamar uma segunda e essa mesma a chamar a primeira	Replicar a parte do código que é necessária para a segunda função executar algumas das funcionalidades da primeira	S	0	2
3	Erro de <i>output</i> – quando o personagem intersecta os elementos restantes	Não estava a ser feito o <i>break</i> nos vários <i>ifs</i> de intersecção logo estava a testar o <i>hitTestPoint</i> de elementos que já tinham sido removidos do <i>stage</i> e do <i>array</i> correspondente	S	0	3
4	Chamar um objeto de uma classe x para uma classe y	É necessário colocar essa variável do tipo <i>static</i>	S	0	3
5	Como saber a dimensão da mesa de forma a poder desenvolver o jogo de forma coerente em termos do tamanho dos objetos na mesa	Desenvolver uma aplicação <i>.air</i> para medir as coordenadas do rato (x,y) e assim conseguir calcular o tamanho do <i>stage</i> na mesa	S	0	1
6	Quando o personagem intersecta os inimigos a distância do touch/do rato ao personagem aumenta	Não estava a ser feito o <i>stopDrag()</i> na intersecção	S	0	3
7	O personagem principal, na mesa <i>touch</i> , não está a fazer a intersecção com os outros elementos	Na função <i>handleDrag</i> estava a utilizar o objeto <i>this</i> e não o personagem	S	0	5
8	Problema de criação de objetos dinamicamente através de classes numa <i>frame</i> que não a primeira	Avaliar a <i>currentFrame</i> e colocar uma condição que só quando estivesse na <i>frame</i> desejada é que chamava a função que iria criar os objetos	S	0	4
9	Incapacidade de utilizar o método <i>gotoAndPlay</i> numa classe que não a <i>main class</i>	Fazer o <i>gotoAndPlay</i> na <i>main class</i> porque é a única forma de aceder à <i>main timeline</i>	S	0	4
10	Não está a remover os objetos do jogo para passar para o ecrã <i>gameover</i>	Utilizar o método <i>removeChild()</i> , mas com um <i>if</i> primeiro de forma a verificar se esse elemento ainda pertence ao <i>stage</i>	S	0	4
11	Dificuldade em aumentar o tamanho do texto que está a ser colocado no <i>stage</i> dinamicamente	Criar um novo objeto do tipo <i>Format</i> e depois no <i>textField</i> desejado colocar <i>setFormat(nomeObjetoFormat)</i>	S	0	3
12	A bolha de proteção do personagem mexe-se de forma independentemente deste	Colocar a bolha sempre na mesma posição do gato, a realizar o <i>drag</i> e alterar o seu estado <i>visible</i> de <i>false</i> para <i>true</i> quando necessário	S	0	2

## APLICAÇÃO JAVA

Também durante a programação da aplicação *Java*, da parte que desenvolvemos até ao momento, tivemos alguns problemas de elevada importância, estando apresentados de seguida.

**Tabela 2** - Lista de bugs de programação da aplicação Java

ID	Descrição	Solução	Terminado (S/N)	Total Erros por resolver	Grau (1-5)
1	Como fazer o <i>disable</i> da <i>driver Bluetooth</i> no Mac ( <i>host</i> ) para que esta possa ser utilizada pela máquina virtual ( <i>guest</i> )	<code>sudo launchctl unload /System/Library/LaunchDaemons/com.apple.blued.plist</code> Faltava esta linha no código do terminal	S	0	4
2	Encontrar uma <i>driver</i> compatível com a <i>driver</i> existente <i>Bluetooth</i> e com a biblioteca utilizada	Download da <i>driver Microsoft Bluetooth Enumerator</i>	S	0	4

Ao nível da programação, no futuro, estamos à espera de encontrar alguns obstáculos em várias áreas, dentro dos objetivos propostos para a entrega em questão (entrega06), tais como:

- Avaliar a localização dos dispositivos *bluetooth*, porque os dados têm que ser constantemente atualizados e não só procurados;
- Fazer o *update* de um ou mais campo da base de dados, através da aplicação *Java*, pois envolve a junção de duas tecnologias: *Java* e *SQL*;
- Garantir o *refresh* da aplicação *web* para *mobile*;
- Introduzir o sistema de ajuda sonoro no jogo Gato Mergulhador;
- Programação do próximo jogo *touch*, encontra, devido à enorme quantidade de elementos tocáveis.

A fim de conseguir corrigir estes erros da melhor forma, elaborámos uma estratégia de correção. Esta baseia-se em 3 pontos fortes de pesquisa: pesquisa online em vários fóruns da respectiva linguagem de programação/tecnologia; pesquisa em livros da linguagem de programação/tecnologia em questão; pedido de auxílio junto de profissionais das linguagens de programação/tecnologias em questão, como por exemplos, professores.

## DESIGN

### JOGOS

Durante o desenho dos vários elementos do jogo, desde os personagens principais, aos restantes elementos ocorreram alguns erros, principalmente devido à inadequação do desenho do objeto face à funcionalidade ou ao cenário envolvente do jogo.

**Tabela 3** - Lista de bugs de design dos jogos

ID	Descrição	Solução	Terminado (S/N)	Total Erros por resolver	Grau (1-5)
1	Desenho do tubarão e da piranha pouco perceptível como algo mau	Colocação de dentes afiados, alteração da cores e inclusão de outros elementos	S	0	4
2	Cor do tubarão muito semelhante à cor do fundo	Alteração da cor do tubarão para cinzento escuro de forma a destacar-se face ao fundo azul	S	0	3
3	Desenho do balde de tinta confuso face à sua função	Modificámos a posição vertical do balde, ou seja, inclinámo-lo, de forma a ficar semelhante aos baldes utilizador em softwares de desenho	S	0	2

### APLICAÇÃO WEB PARA MOBILE

Na realização do manual de identidade da marca realizámos um estudo de design e de grelhas para os vários ecrãs da aplicação e durante esse mesmo estudo surgiram algumas incoerências.

**Tabela 4** - Lista de bugs de design da aplicação web para mobile

ID	Descrição	Solução	Terminado (S/N)	Total Erros por resolver	Grau (1-5)
1	Grelhas com alinhamento inadequado	-	N	1	3
2	Função confusa do botão circular azul	-	N	2	2

Ao nível do design, no futuro, estamos à espera de encontrar alguns obstáculos em várias áreas, dentro dos objetivos propostos para a entrega em questão (entrega06), tais como:

- Desenho do jogo encontra, pois todo o ambiente tem que funcionar como um todo;
- Desenho da aplicação web para mobile de forma a respeitar as grelhas.

A fim de conseguir corrigir estes erros da melhor forma, elaborámos uma estratégia de correção. Esta baseia-se em 2 aspetos: observar de que forma outras aplicações semelhantes ultrapassaram e implementaram estes elementos; pedido de auxílio junto de profissionais de design, como por exemplos, professores.



## CONTEÚDO

### JOGO – GATO MERGULHADOR

No desenvolvimento do jogo Gato Mergulhador tivemos problemas com a apresentação e o significado do ecrã game over, pois foi necessário refletir sobre o que representava de facto o fim de jogo: uma vitória ou uma derrota. Optámos por considerar o fim de jogo como uma vitória por pontuação máxima, de forma a trazer um ponto de vista positivo para algo que pode ser entendido como negativo: o fim do jogo. Consideramos este erro de gravidade 2, uma vez que é importante para a compreensão do utilizador mas não é essencial para a mesma nem para a sua interação com o jogo.

Ao nível do conteúdo, no futuro, estamos à espera de encontrar alguns obstáculos em várias áreas, dentro dos objetivos propostos para a entrega em questão (entrega06), tais como:

- Erros na gravação do som da narração dos jogos;
- Erros na edição do som da narração dos jogos;
- Erros textuais ou de formatação na apresentação dos conteúdos na aplicação web para mobile;
- Erros de codificação do streaming de vídeo, devido aos codecs específicos de cada browser mobile.

A fim de conseguir corrigir estes erros da melhor forma, elaborámos uma estratégia de correção. Esta baseia-se em 3 aspetos: planificar tudo ao detalhe de forma a tentar evitar que alguns destes erros ocorram; recorrer a fóruns online relacionados com o streaming de vídeo para mobile; recorrer ao auxílio de profissionais da área, nomeadamente professores.