

Documentação de Software



Visão geral

É um sistema que vai realizar o lançamento de dados.



Descrição do Produto:

Vai ser realizado e implementado pelo arduino UNO, com botões, led's e buzinas, para a interação com o jogador.



Requisitos Funcionais:

Uma lista de todos os requisitos funcionais do sistema, descrevendo as funcionalidades específicas que o sistema deve fornecer aos usuários.

Requisitos Funcionais	Impacto nos negócios	Prioridade	Descrição detalhada
O lançamento de dados.	Faz com que o jogo aconteça, da forma que planejamos.	ALTA	O primeiro jogador (que foi o primeiro sorteado) lança o dado e guarda o número que foi sorteado. E assim por diante.
Definição das pontuações	Para que os jogadores	MÉDIA	Após todos os jogadores jogar o dado, a dupla de heróis e a dupla

das partidas.	fiquem cientes da pontuação.		de vilões vão somar seus pontos entre si, e ao final da soma der um número maior ganha a partida, caso der empate fazer um novo lançamento de dados.
Ganhador da partida	Para que os jogadores vejam quem ganhou a partida.	MÉDIA	Caso os heróis ganharem acender o led verde e tocar a buzina em um tom grave. Caso os vilões ganhem acender a led vermelha e tocar a buzina em um tom agudo.
Mostrar o placar		MÉDIA	Guardar números de vitórias de cada lado, e ao final das 5 partidas quem tiver o números no placar maior ganha o jogo.
Vitória	Para que os jogadores vejam quem ganhou o jogo.	ALTA	Assim que placar ser mostrado, começar a tocar o led e buzina de acordo com os ganhadores: Heróis – piscar o led verde e tocar uma musica no tom grave. Vilões – piscar o led vermelho e tocar uma musica no tom agudo.



Requisitos Não Funcionais:

Uma lista de todos os requisitos não funcionais do sistema:

Requisitos Não Funcionais	Impacto nos negócios	Prioridade	Descrição detalhada
Os leds e a interação com os jogadores	Para deixar o jogo mais interativo.	BAIXA	Assim que for feita a somatória de pontos das duplas, será feita a comparação com os pontos dos adversários, ficando assim: VILÕES ganha o led vermelho acende - Heróis ganham o led verde acende (Isso será realizado a cada partida). No final do jogo (última partida)

			será analisado qual dos lados venceu mais partidas caso seja o lado dos heróis o led verde ira piscar, caso seja o dos vilões o led vermelho ira piscar..
A buzina e a interação com os jogadores	Para deixar o jogo mais interativo.	BAIXA	Assim que for feita a somatória de pontos das duplas, será feita a comparação com os pontos dos adversários, ficando assim: VILÕES ganha a buzina ira tocar em um tom agudo - Heróis ganham a buzina ira tocar em som grave (Isso será realizado a cada partida). No final do jogo (última partida) será analisado qual dos lados venceu mais partidas caso seja o lado dos heróis a buzina ira tocar um música no tom grave, caso seja o dos vilões a buzina ira tocar uma música no tom agudo.



Regras de negócio:

Uma lista com todas as regras de negócio, abordando políticas, restrições e procedimentos para garantir a eficácia e consistência das operações do sistema:

Regras de negócio	Impacto nos negócios	Prioridade	Descrição detalhada
Ter 5 partidas.	Uma estratégia, para que não aconteça tantos empates.	MÉDIA	O jogo terá 5 partidas ao todo, porém caso haja empate uma sexta partida será iniciada, isso vai acontecer até cada partida ter realmente um ganhador, sem empates.
1 dado de 6 faces.		MÉDIA	No jogo o número máximo que o jogador pode ter de pontuação

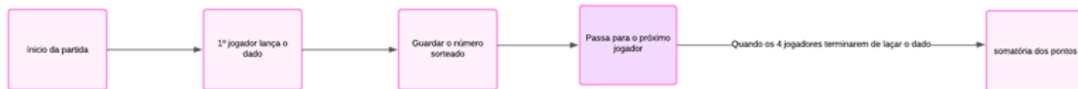
com a sua dupla é de 12.



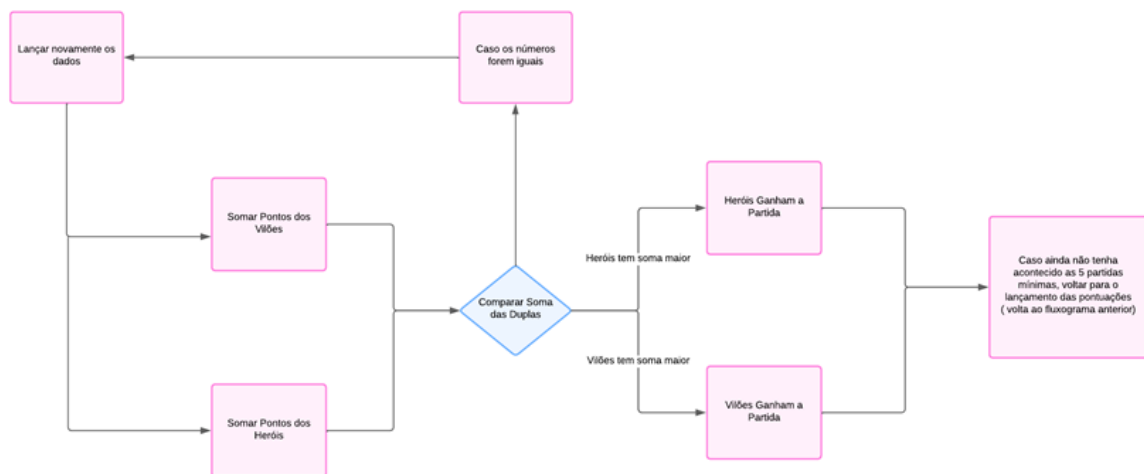
Fluxograma

Representação visual de processos, mostrando a sequência de etapas e decisões. Entidades principais do banco de dados, diagramas conceitual e lógico.

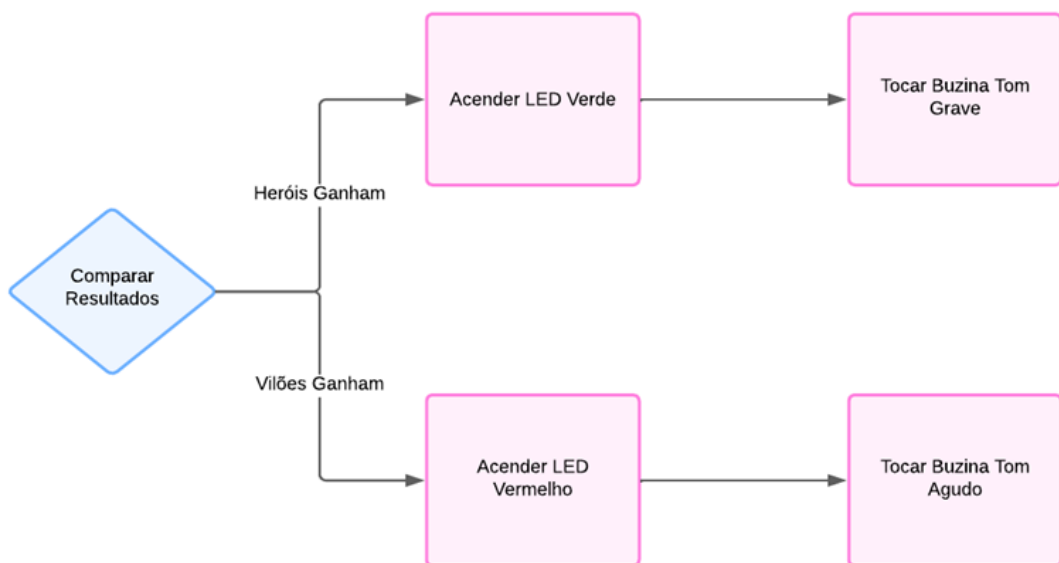
Lançamentos das pontuações



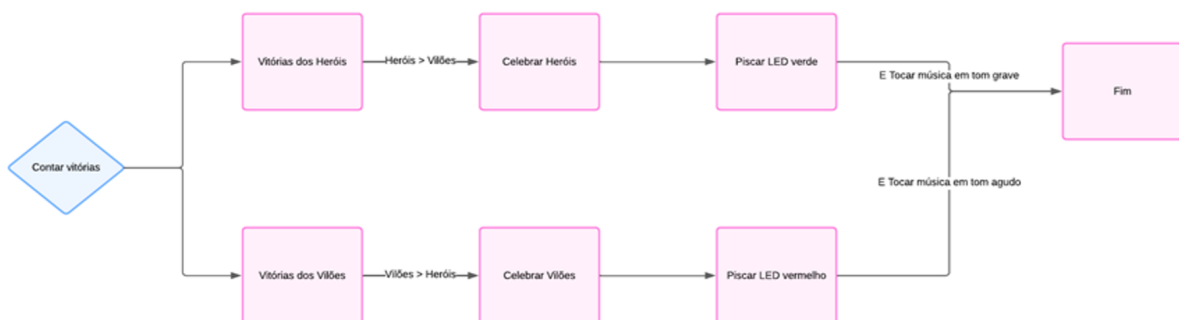
Soma e a comparação das pontuações das duplas



Interação dos leds e da buzina com os jogadores



Determinar os ganhadores



Descritivo dos endpoints

```

int PontosHerois = 0;
int PontosViloes = 0;
  
```

```

int Heroi1, Heroi2, Vilao1, Vilao2;
int NUMERO_DE_PARTIDAS = 5;
int partida = 0;
int GanhadorH = 0;
int GanhadorV = 0;
int Vitoria = 7;

bool botaoPressionado1 = false;
bool botaoPressionado2 = false;
bool botaoPressionado3 = false;
bool botaoPressionado4 = false;
//declarando os botões
#define BotaoHeroi1 6
#define BotaoHeroi2 8
#define BotaoVilao1 7
#define BotaoVilao2 9
//declarando os leds
#define ledVerde 3
#define ledVermelho 4
//declarando a buzzer
#define Buzzer 2

// Definição das notas musicais
#define NOTE_C4 262
#define NOTE_D4 294
#define NOTE_E4 330
#define NOTE_F4 349
#define NOTE_G4 392
#define NOTE_A4 440
#define NOTE_B4 494
#define NOTE_C5 523

// Define a estrutura da melodia
int melody[] = {
    NOTE_E4, NOTE_E4, NOTE_F4, NOTE_G4, NOTE_G4, NOTE_F4, NOTE_
    NOTE_D4, NOTE_C4, NOTE_C4, NOTE_D4, NOTE_E4,
    NOTE_E4, NOTE_D4, NOTE_D4, NOTE_C4, NOTE_C4,
    NOTE_E4, NOTE_E4, NOTE_F4, NOTE_G4, NOTE_G4, NOTE_F4, NOTE_

```

```

    NOTE_D4, NOTE_C4, NOTE_C4, NOTE_D4, NOTE_E4,
    NOTE_D4, NOTE_C4, NOTE_C4
};

// Define a duração de cada nota
int noteDurations[] = {
    4, 4, 4, 4, 4, 2, 4,
    4, 4, 4, 4, 2,
    4, 4, 4, 4, 2,
    4, 4, 4, 4, 4, 2, 4,
    4, 4, 4, 4, 2,
    4, 4, 4
};

void setup() {
    Serial.begin(9600);

    Serial.println(R"(
=====
|      Bem-vindo ao jogo,      |
|  Escolham seus personagens  |
|      para o inicio da      |
|      partida!!!            |
=====
)");

    pinMode(BotaoHeroi1, INPUT);
    pinMode(BotaoVilao1, INPUT);
    pinMode(BotaoHeroi2, INPUT);
    pinMode(BotaoVilao2, INPUT);
    pinMode(ledVerde, OUTPUT);
    pinMode(ledVermelho, OUTPUT);
    pinMode(Buzzer, OUTPUT);
    delay(1000);
}

void limparEstado() {

```

```

// Limpa o estado das variáveis para zero
Herói1 = 0;
Herói2 = 0;
Vilão1 = 0;
Vilão2 = 0;
PontosHeróis = 0;
PontosVilões = 0;
}

void limparEstadoBotao() {
    //Limpar o estado dos botoes
    botaoPressionado1 = false;
    botaoPressionado2 = false;
    botaoPressionado3 = false;
    botaoPressionado4 = false;
}

void loop() {
    //quem ganhou da partida
    String partida1 = "";
    String partida2 = "";
    String partida3 = "";
    String partida4 = "";
    String partida5 = "";

    for (byte partida = 0; partida < NUMERO_DE_PARTIDAS;) {
        //lançamento de dados
        if (!botaoPressionado1 && digitalRead(BotaoHerói1) == 1)
            Herói1 = random(1, 7);
        Serial.print("Pontos do Herói1: ");
        Serial.println(Herói1);
        delay(1000);
        botaoPressionado1 = true;
    }

    if (!botaoPressionado2 && digitalRead(BotaoHerói2) == 1)
        Herói2 = random(1, 7);
}

```



```

    Serial.print("Pontos do Heroi2: ");
    Serial.println(Heroi2);
    delay(1000);
    botaoPressionado2 = true;
}

if (!botaoPressionado3 && digitalRead(BotaoVilao1) == 1)
    Vilao1 = random(1, 7);
    Serial.print("Pontos do Vilao1: ");
    Serial.println(Vilao1);
    delay(1000);
    botaoPressionado3 = true;
}

if (!botaoPressionado4 && digitalRead(BotaoVilao2) == 1)
    Vilao2 = random(1, 7);
    Serial.print("Pontos do Vilao2: ");
    Serial.println(Vilao2);
    delay(1000);
    botaoPressionado4 = true;
}

PontosHerois = Heroi1 + Heroi2;
PontosViloes = Vilao1 + Vilao2;
if (PontosHerois > 0 && PontosViloes > 0 && (Heroi1 != 0 && Vilao1 != 0)) {
    if (PontosHerois > PontosViloes && partida < 5) {
        Serial.println("Os herois venceram a partida!!!");
        limparEstado();
        limparEstadoBotao();
        digitalWrite(ledVermelho, LOW);
        digitalWrite(ledVerde, HIGH);
        tone(Buzzer, 663);
        delay(400);
        tone(Buzzer, 701);
        delay(250);
        tone(Buzzer, 589);
        delay(400);
        tone(Buzzer, 663);
    }
}

```

```

    delay(250);
    tone(Buzzer, 526);
    delay(400);
    tone(Buzzer, 589);
    delay(250);
    tone(Buzzer, 440);
    delay(1000);
    noTone(Buzzer);
    delay(1000);
    digitalWrite(ledVerde, LOW);
    noTone(Buzzer);
    delay(1000);
    partida1 = "herois";
    GanhadorH += 1;
    partida2 = "herois";
    GanhadorH += 1;
    partida3 = "herois";
    GanhadorH += 1;
    partida4 = "herois";
    GanhadorH += 1;
    partida5 = "herois";
    GanhadorH += 1;
} else if (PontosViloes > PontosHerois && partida < 5)
    Serial.println("Os viloes venceram a partida!!!");
    limparEstado();
    limparEstadoBotao();
    digitalWrite(ledVerde, LOW);
    digitalWrite(ledVermelho, HIGH);
    tone(Buzzer, 392);
    delay(500);
    tone(Buzzer, 440);
    delay(250);
    tone(Buzzer, 468);
    delay(400);
    tone(Buzzer, 589);
    delay(400);
    tone(Buzzer, 468);
    delay(400);

```

```

    tone(Buzzer, 440);
    delay(250);
    tone(Buzzer, 392);
    delay(1000);
    noTone(Buzzer);
    digitalWrite(ledVermelho, LOW);
    delay(1000);
    partida1 = "viloes";
    GanhadorV += 1;
    partida2 = "viloes";
    GanhadorV += 1;
    partida3 = "viloes";
    GanhadorV += 1;
    partida4 = "viloes";
    GanhadorV += 1;
    partida5 = "viloes";
    GanhadorV += 1;
} else {
    Serial.println("Empate!!!");
    limparEstado();
    limparEstadoBotao();
    for (byte i = 0; i < Vitoria; i++) {
        tone(Buzzer, 750, 5000); // Frequência de 750Hz po
        digitalWrite(ledVermelho, HIGH);
        delay(100);
        digitalWrite(ledVerde, HIGH);
        noTone(Buzzer);
        delay(250);
        tone(Buzzer, 750);
        digitalWrite(ledVermelho, LOW);
        delay(100);
        digitalWrite(ledVerde, LOW);
        noTone(Buzzer);
        delay(250);
    }
}
partida += 1;
Serial.print("Partida ");

```

```

    Serial.println(partida);
}

if (partida == 5) {
    if (GanhadorH > GanhadorV) {
        Serial.println("Os heróis venceram o jogo");
        // Reproduz a melodia
        for (int i = 0; i < sizeof(melody) / sizeof(melody[0])
            int duration = 1000 / noteDurations[i];
            digitalWrite(ledVerde, HIGH);
            tone(Buzzer, melody[i], duration);
            delay(duration + 50); // Adiciona um pequeno atras
            digitalWrite(ledVerde, LOW);
        }
    }

    if (GanhadorV > GanhadorH) {
        Serial.println("Os vilões venceram o jogo");
        // Reproduz a melodia
        for (int i = 0; i < sizeof(melody) / sizeof(melody[0])
            int duration = 1000 / noteDurations[i];
            digitalWrite(ledVermelho, HIGH);
            tone(Buzzer, melody[i], duration);
            delay(duration + 50);
            digitalWrite(ledVermelho, LOW);

        }
    }

    else if (GanhadorV == GanhadorH) {
        Serial.println("O JOGO TERMINOU EMPATADO!!!");
    }
    Serial.println("Fim de jogo");
}
}
}
}

```



Divisão do projeto

Planejamento de funcionalidades divididas entre os participantes do grupo

Projeto	Descrição
- Definição das pontuações das partidas. - Ganhador da partida	Maria Clara de Paula
- O lançamento de dados.	Maria Fernanda
- Vitória - Mostrar o placar	Priscila

ALTERAÇÃO

Retiramos alguns requisitos funcionais, sendo eles:

- “Sorteio da ordem que os jogadores vão lançar o dado” - que optamos por retirar pois, não estava ficando do jeito que planejávamos (Esse requisito não afeta em nada).
- “Escolha dos personagens” - também optamos por que os próprios jogadores decidem seus personagens entre si.
- implementação de música ao final da partida - Pensamos nessa mudança para que o jogo fique mais dinâmico e atrativo.