

**UNIVERSIDADE DE SÃO PAULO**  
**ESCOLA POLITÉCNICA**  
**DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E**  
**SISTEMAS DIGITAIS**  
**PCS3635 - LABORATÓRIO DIGITAL I**



**ESPECIFICAÇÃO PRELIMINAR DO PROJETO**  
**ASTROGENIUS**

Felipe Luis Korbes - NUSP: 13682893

Henrique Eduardo dos Santos de Souza - NUSP: 13679972

João Felipe de Souza Melo - NUSP: 13682913

**Turma: 5**

**Professor: Reginaldo Arakaki**

**Data da aula: 28/02/2024**

São Paulo  
2024

# Sumário

<b>1. Introdução e Objetivos.....</b>	<b>1</b>
<b>2. Descrição da Proposta.....</b>	<b>1</b>
2.1 Hipóteses Simplificadoras.....	2
2.2 Montagem Física do Console de Arcade.....	3
2.3 Controle do Player.....	4
2.4 Circuito Digital.....	4
2.5 Interface Gráfica.....	5
<b>3 Requisitos Funcionais.....</b>	<b>5</b>
3.1 (RF1) Implementação Física em placa FPGA.....	6
3.2 (RF2) Controle da Nave Espacial.....	6
3.3 (RF3) Sistema de Vida.....	7
3.4 (RF4) Sistema de Pontuação.....	7
3.5 (RF5) Sistema de geração de asteroides.....	7
3.6 (RF6) Destruição do asteroide.....	8
3.7 (RF7) Tiro realizado.....	8
3.8 (RF8) Registro de pontuações dos players na memória.....	9
3.9 (RF9) Menu de interação.....	9
3.10 (RF10) Monitor.....	10
3.11 (RF11) Lógica do Genius.....	10
<b>4 Requisitos Não funcionais.....</b>	<b>11</b>
4.1 (RNF1) Performance.....	11
4.2 (RNF2) Usabilidade.....	12
4.3 (RNF3) Escalabilidade.....	12
4.4 (RNF4) Manutenção.....	12
4.5 (RNF5) Som atrativo.....	13
<b>5 Cronograma.....</b>	<b>13</b>

## 1. Introdução e Objetivos

O game *asteroids* é um dos games clássicos de tiro multidirecional, da era dos arcades e do console Atari, com temática espacial. Trata-se de um game em que o player deve controlar uma nave, constantemente ameaçada por asteroides, que devem ser destruídos por ataques de tiros para garantir a sobrevivência da nave e obter pontuação.



*Figura 1 - Tela de game do Asteroids do Atari*

Cada asteroide, ao receber um tiro, parte-se em alguns pedaços, que também podem ser destruídos e, a cada colisão com a nave, o jogador perde pontos de vida.

O projeto visa aplicar os conhecimentos adquiridos nas aulas experimentais do Genius. Ele utilizará a lógica do jogo implementada na experiência 6, conforme especificado. Além disso, o projeto tem como objetivo facilitar a implementação através do planejamento incremental das implementações e da definição de requisitos funcionais, não funcionais e opcionais. O projeto será desenvolvido na linguagem de descrição de hardware Verilog.

## 2. Descrição da Proposta

Propõe-se nesse projeto uma reprodução simplificada do jogo clássico *asteroids* num modelo de um console de arcade, com interação com player por meio de sensores físicos - botões e/ou joystick -, e visualização em monitor de tela digital, nesse caso, de um monitor de computador. Fisicamente, haverá uma série de botões e um joystick, para controle da nave, que estarão conectados a uma placa FPGA, que contém a lógica fundamental do game. Além disso, a placa terá um

canal de comunicação serial com um computador para que as informações do circuito do game possam ser traduzidas em informações gráficas no monitor. Assim, o console constitui-se de um equipamento completo, que engloba desde as entradas do player até a visualização do game no monitor.

Para tornar viável a reprodução do game dentro do prazo da disciplina, o grupo optou por adotar um conjunto de simplificações úteis, que permitem ao menos, na pior das hipóteses, cobrir as funcionalidades essenciais do game, como a mecânica de tiros, a movimentação da nave, o sistema de vidas e de pontuação, dentre outros. Uma vez que o projeto já alcançou o patamar de suprir as necessidades básicas do projeto, acrescenta-se então novas *features*, com complexidades adicionais à lógica básica e mais mecânicas de jogo.

## 2.1 Hipóteses Simplificadoras

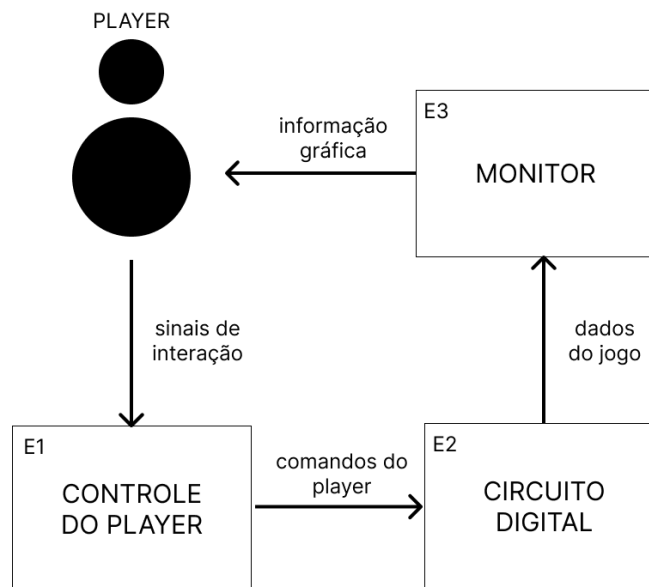
Nessa reprodução do asteroids clássico, emprega-se a seguinte série de hipóteses e simplificações, identificadas pelos códigos pseudônimos H1, H2, H3, H4 e H5, para melhor referência nesse documento:

- (H1) A nave ficará fixa no centro da tela e deve alterar apenas seu ângulo, enquanto que os asteroides vêm de direções diversas.
- (H2) O game habita um espaço discreto de uma matriz de pixels  $N \times N$ , com  $N$  grande o suficiente para o movimento parecer suave visualmente, mas pequeno o suficiente para não causar problemas de processamento.
- (H3) Tanto a nave quanto os asteroides movimentam-se apenas em linhas verticais ou horizontais
- (H4) Os asteroides, ao receberem tiros da nave, não se partem em asteroides menores. Ao invés disso, asteroides atingidos são imediatamente destruídos e desaparecem do jogo.
- (H5) O jogo é projetado para um único player por vez, que inicia com três vidas, e com opção de escolher entre três modos de dificuldade.

Para a hipótese H2, considera-se  $N=1024$ , o que resulta numa grade discreta de  $1024 \times 1024$  pixels. Esse número intermediário entre o básico e o excessivo, pelas estimativas do grupo, permite uma visualização boa do game, sem comprometer o processamento ao nível de prejudicar a jogatina.

## 2.2 Montagem Física do Console de Arcade

O Console de Arcade, que constitui a montagem física do projeto, é constituído pelas três grandes entidades (E1) controle do player, (E2) circuito físico do console, (E3) monitor do console, dispostas tal qual na figura 2.



*Figura 2 - Arquitetura básica do Sistema*

A entidade E1 é um componente físico com sensores físicos para captação das entradas do player. Ela contará com botões e joystick, a partir dos quais o player poderá dar comandos. Esses estímulos aos sensores geram respostas na forma de sinais de entrada para o circuito digital, da entidade E2, do circuito digital, que se responsabiliza por processar essas informações sob a lógica de funcionamento game, atualizando o estado do sistema todo. A entidade E2 também encarrega-se de gerar sinais úteis para renderização do jogo na monitor, entidade E3. O monitor toma para si os sinais de saída do circuito e compila essa informação de forma gráfica, o que permite ao jogador ver o jogo em tempo real.

Por razões de simplicidade, entre o circuito digital e o monitor, há um computador intermediário, que irá realizar a comunicação e tradução dos sinais do circuito digital e do monitor, de forma a facilitar a renderização na tela. Esse módulo não é apresentado no diagrama de alto nível da figura 2 porque considera-se que a presença do computador só ocorre devido à simplicidade da placa FPGA em comparação ao monitor digital; ou seja, executar a conexão direta entre a placa e o

monitor daria ao projeto uma complexidade adicional irrelevante para seu propósito, que está concentrado na lógica do game.

### **2.3 Controle do Player**

Para essa aplicação, optou-se por implementar um controle de quatro botões B\_UP, B\_DOWN, B\_RIGHT e B\_LEFT, para controle da posição da nave, mais dois botões B\_ATTACK e B\_SPECIAL, para ataques. Estes botões serão também reaproveitados para navegação no menu do jogo. Para facilitar a identificação, os botões B\_UP, B\_DOWN, B\_RIGHT e B\_LEFT serão rotulados fisicamente como quatro setas, com adesivo sobre eles, e serão chamados semanticamente pelo game de “cima”, “baixo”, “direita”, “esquerda”, respectivamente. Além disso, os botões B\_ATTACK e B\_SPECIAL serão rotulados com as cores vermelho e verde, nessa ordem, e terão a semântica de “ataque” e “ataque especial”, no contexto da jogatina, e “cancela” e “confirma”, para navegação no menu. Dessa forma, verde e vermelho estão de acordo com a convenção cultural de cores para confirmar ou negar algo, o que facilita o aprendizado do usuário na interação com o sistema.

### **2.4 Circuito Digital**

A entidade E2, o circuito digital, é responsável por processar os sinais de controle do player, sob a lógica do game, e enviar os sinais de renderização para o monitor. Dessa forma, ele constitui-se como a engrenagem do game, que executa as atualizações do estado atual segundo os estímulos do player e gera a visualização gráfica do resultado. Para esse projeto, o circuito digital será implementado em uma placa FPGA DE0-CV. Haverá o desenvolvimento de várias lógicas de funcionalidade, no caso do projeto Asteroids:

- Ganhar pontos.
- Perder vidas.
- Destruir asteroides.
- Criar asteroides.
- Menu de interação.
- Salvar pontos.

## 2.5 Interface Gráfica

Visualmente, o monitor deve exibir em sua tela a nave espacial, como uma pequena seta no centro da tela cuja ponta principal é a frente da nave, e os asteroides, que podem surgir em formatos variados e diversos. O fundo deve ser preto, representando o espaço, e pode conter pequenos detalhes decorativos. Além disso, o game deve apresentar no canto superior esquerdo as informações de pontuação e quantidades de vidas, semelhantemente ao jogo original. Assim, o game ficará com aspecto gráfico parecido com o original, tal qual na figura 1.

## 3 Requisitos Funcionais

De acordo com as hipóteses simplificadoras de H1 a H5 e com a proposta idealizada pelo grupo, chegou-se ao seguinte conjunto de grandes requisitos funcionais para o projeto:

- **(RF1)** Implementação em placa FPGA
- **(RF2)** Controle da nave espacial
- **(RF3)** Sistema de vidas
- **(RF4)** Sistema de Pontuação\*
- **(RF5)** Sistema de geração de asteroides
- **(RF6)** Destruição do asteroide
- **(RF7)** Tiro realizado
- **(RF8)** Registro de pontuações dos players na memória
- **(RF9)** Menu de interação\*
- **(RF10)** Monitor\*
- **(RF11)** Hitbox

Os códigos **RFN** identificam o n-ésimo requisito funcional do projeto.

### 3.1 (RF1) Implementação Física em placa FPGA

Deve ser utilizada uma placa FPGA, com dispositivos sensores que permitam obter as entradas de controle do usuário. Sua tabela de requisitos é:

<b>Código: RF1</b>	<input checked="" type="checkbox"/> Funcional	<input type="checkbox"/> Não Funcional
<b>Requisito: Implementação Física em placa FPGA</b>		
<b>Descrição:</b> O circuito lógico deve ser sintetizado na placa FPGA DE0-CV		
<b>Prioridade:</b>	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Estabilidade:</b>	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Rationale:</b> Este requisito especifica a implementação física do Asteroids		
<b>Requisitos associados:</b> Não há		

### 3.2 (RF2) Controle da Nave Espacial

Inicialmente a primeira implementação será com botões da placa FPGA DE0-CV. Entretanto, caso seja possível, o grupo pretende adicionar a funcionalidade de se utilizar um joystick.

<b>Código: RF2</b>	<input checked="" type="checkbox"/> Funcional	<input type="checkbox"/> Não Funcional
<b>Requisito: Controle da Nave Espacial</b>		
<b>Descrição:</b> O game deve ter um controle com os botões de interação: <ul style="list-style-type: none"><li>- B_UP: a nave fica no sentido vertical ascendente.</li><li>- B_DOWN: a nave fica no sentido vertical descendente.</li><li>- B_RIGHT: a nave fica no sentido horizontal ascendente.</li><li>- B_LEFT: a nave fica no sentido horizontal descendente.</li><li>- B_ATTACK: a nave realiza o tiro na direção em que está.</li><li>- B_SPECIAL: a nave realiza o tiro especial na direção em que está.</li></ul>		
<b>Prioridade:</b>	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Estabilidade:</b>	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Rationale:</b> Este requisito especifica os botões a serem utilizados para jogar		
<b>Requisitos associados:</b> RF1		



### 3.3 (RF3) Sistema de Vida

O jogador deve possuir três vidas para jogar, antes de dar game over. Cada vez que a nave for atingida por um asteroide o jogador perde uma vida.

<b>Código: RF3</b>	<input checked="" type="checkbox"/> Funcional	<input type="checkbox"/> Não Funcional
<b>Requisito: Sistema de Vidas</b>		
<b>Descrição:</b> O player deve possuir exatamente três vidas, cada uma perdida em caso de colisão com um asteroide e que provocam o game over, caso acabem.		
<b>Prioridade:</b>	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Estabilidade:</b>	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Rationale:</b> Este requisito especifica o sistema de vidas e game over		
<b>Requisitos associados: RF1, RF5</b>		

### 3.4 (RF4) Sistema de Pontuação

O player possui uma barra de pontuação que aumenta a cada tiro dado em asteroides. A pontuação deve preservar-se, mesmo com a perda de uma das vidas, mas deve zerar a cada novo jogo.

<b>Código: RF4</b>	<input checked="" type="checkbox"/> Funcional	<input type="checkbox"/> Não Funcional
<b>Requisito: Sistema de Pontuação</b>		
<b>Descrição:</b> O player deve ter um contador de pontuação, que incrementa com o ataque e a destruição de cada asteroide.		
<b>Prioridade:</b>	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Estabilidade:</b>	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Rationale:</b> Este requisito determina quando o jogador deve ganhar pontos.		
<b>Requisitos associados: RF1, RF5, RF2, RF6</b>		

### 3.5 (RF5) Sistema de geração de asteroides

Os asteroides devem ser gerados aleatoriamente, com trajetórias em linhas retas, mas com velocidades variadas.

<b>Código: RF5</b>	<input checked="" type="checkbox"/> Funcional	<input type="checkbox"/> Não Funcional
<b>Requisito: Sistema de Geração de Asteroides</b>		
<b>Descrição:</b> A geração de asteroides deve ser aleatória, de forma que eles possam nascer na tela em posições e com velocidades randômicas.		
<b>Prioridade:</b>	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Estabilidade:</b>	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Rationale:</b> Este requisito determina como deve ser o processo de geração de asteroides		
<b>Requisitos associados: RF1</b>		

### 3.6 (RF6) Destruição do asteroide

Deve-se ter uma lógica para destruir os asteroides. Caso a bala chegue no asteroide então o asteroide deve “desaparecer” (o asteroide é destruído)

<b>Código: RF6</b>	<input checked="" type="checkbox"/> Funcional	<input type="checkbox"/> Não Funcional
<b>Requisito: Destruição do asteroide</b>		
<b>Descrição:</b> Quando um tiro disparado pela nave chegar a um asteroide então o asteroide deve desaparecer.		
<b>Prioridade:</b>	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Estabilidade:</b>	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Rationale:</b> Este requisito determina como o asteroide deve ser destruído.		
<b>Requisitos associados: RF7, RF2, RF5</b>		

### 3.7 (RF7) Tiro realizado

O tiro deve ser realizado quando o jogador apertar o botão B\_ATTACK e este deve ter uma velocidade.

<b>Código: RF7</b>	<input checked="" type="checkbox"/> Funcional	<input type="checkbox"/> Não Funcional
<b>Requisito: Tiro Realizado</b>		

<b>Descrição:</b> Quando o botão B_ATTACK for disparado então deve-se realizar o tiro com uma velocidade pré-definida		
<b>Prioridade:</b>	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Estabilidade:</b>	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Rationale:</b> Este requisito determina a condição para ocorrer o tiro.		
<b>Requisitos associados:</b> RF7, RF2		

### 3.8 (RF8) Registro de pontuações dos players na memória

Deve ocorrer o registro de pontuação assim que o player perder as três vidas.

<b>Código:</b> RF8	<input checked="" type="checkbox"/> Funcional <input type="checkbox"/> Não Funcional
<b>Requisito: Registro de Pontuações dos Players na Memória</b>	
<b>Descrição:</b> Quando o jogador perder as três vidas deve ser salvo o ponto em uma memória e ficar disponível para o usuário no menu.	
<b>Prioridade:</b>	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Estabilidade:</b>	<input type="checkbox"/> Alta <input type="checkbox"/> Média <input checked="" type="checkbox"/> Baixa
<b>Rationale:</b> Este requisito determina a condição para ocorrer o tiro.	
<b>Requisitos associados:</b> RF6, RF5	

### 3.9 (RF9) Menu de interação

O jogo a ser implementado pelo grupo terá o menu de interação onde será possível verificar as pontuações dos jogadores ou começar um novo jogo.

<b>Código:</b> RF9	<input checked="" type="checkbox"/> Funcional <input type="checkbox"/> Não Funcional
<b>Requisito: Menu de Interação</b>	
<b>Descrição:</b> Deve ter um menu de interação que deve mostrar uma opção para começar o jogo ou para verificar o ranking de pontos.	
<b>Prioridade:</b>	<input type="checkbox"/> Alta <input type="checkbox"/> Média <input checked="" type="checkbox"/> Baixa

<b>Estabilidade:</b>	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Média	<input type="checkbox"/> Baixa
<b>Rationale:</b> Este requisito determina a condição para ocorrer o tiro.			
<b>Requisitos associados:</b> RF8			

### 3.10 (RF10) Monitor

Deve haver uma interface pelo monitor com o jogador.

OBS: Devido ao pouco conhecimento do grupo em como realizar a conexão este requisito funcional estará genérico. Este requisito será especificado após marcarmos uma reunião com o monitor.

<b>Código:</b> RF10	<input checked="" type="checkbox"/> Funcional	<input type="checkbox"/> Não Funcional
<b>Requisito:</b> Monitor		
<b>Descrição:</b> Deve haver um monitor para interação com o usuário		
<b>Prioridade:</b>	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Estabilidade:</b>	<input type="checkbox"/> Alta <input type="checkbox"/> Média	<input checked="" type="checkbox"/> Baixa
<b>Rationale:</b> Este requisito determina a interação humano-jogador.		
<b>Requisitos associados:</b> RF9, RF6, RF5, RF4, RF3		

### 3.11 (RF11) Lógica do Genius

Deve-se comparar a posição do tiro com a posição do asteroide utilizando a lógica do Genius no estado de comparar a jogada para verificar quando o asteroide atinge a nave.

<b>Código:</b> RF11	<input checked="" type="checkbox"/> Funcional	<input type="checkbox"/> Não Funcional
<b>Requisito:</b> Hitbox		
<b>Descrição:</b> Deve-se comparar a posição do tiro com a posição do asteroide utilizando a lógica do Genius no estado de comparar a jogada para verificar quando o asteroide atinge a nave.		
<b>Prioridade:</b>	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média <input type="checkbox"/> Baixa

<b>Estabilidade:</b>	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Média	<input type="checkbox"/> Baixa
<b>Rationale:</b> Este requisito determina a lógica de comparação da posição do tiro e do asteroide.			
<b>Requisitos associados:</b> RF2, RF5, RF6, RF7			

#### 4 Requisitos Não funcionais

De acordo com as hipóteses simplificadoras de H1 a H5 e com a proposta idealizada pelo grupo, chegou-se ao seguinte conjunto de requisitos não funcionais para o projeto:

- **(RNF1)** Performance: Rapidez de processamento
- **(RNF2)** Usabilidade: Uso de Joystick
- **(RNF3)** Escalabilidade
- **(RNF4)** Manutenção
- **(RNF5)** Som atrativo\*
- Os códigos **RNFN** identificam o n-ésimo requisito funcional do projeto.

##### 4.1 (RNF1) Performance

O projeto deve ser rápido e ter um desempenho aceitável.

<b>Código:</b> RNF1	<input type="checkbox"/> Funcional	<input checked="" type="checkbox"/> Não Funcional
<b>Requisito:</b> Performance		
<b>Descrição:</b> O projeto deve ter rapidez de processamento, não podendo demorar muito entre o usuário pressionar um botão e a ação ser realizada.		
<b>Prioridade:</b>	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Estabilidade:</b>	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Rationale:</b> Determina a performance do projeto.		
<b>Requisitos associados:</b> RF1, RF10		

#### 4.2 (RNF2) Usabilidade

Os controles do jogo devem ser intuitivos e responsivos, permitindo que os jogadores se movam e atirem com facilidade. O objetivo do grupo é adicionar, na medida do possível, um joystick para controlar a direção da nave.

<b>Código: RNF2</b>	<input type="checkbox"/> Funcional	<input checked="" type="checkbox"/> Não Funcional
<b>Requisito:</b> Usabilidade		
<b>Descrição:</b> Deve-se fazer o uso de joystick para jogar Asteroide.		
<b>Prioridade:</b>	<input type="checkbox"/> Alta	<input type="checkbox"/> Média <input checked="" type="checkbox"/> Baixa
<b>Estabilidade:</b>	<input type="checkbox"/> Alta	<input type="checkbox"/> Média <input checked="" type="checkbox"/> Baixa
<b>Rationale:</b> Determina a utilização de uma nova interação do usuário.		
<b>Requisitos associados:</b> RF1, RF10, RF2, RF7, RNF1		

#### 4.3 (RNF3) Escalabilidade

O projeto deve ter alta escalabilidade a fim de ter um produto e um mercado disposto a pagar por.

<b>Código: RNF3</b>	<input type="checkbox"/> Funcional	<input checked="" type="checkbox"/> Não Funcional
<b>Requisito:</b> Escalabilidade		
<b>Descrição:</b> Deve-se fazer o jogo de modo a ser fácil a escalabilidade.		
<b>Prioridade:</b>	<input type="checkbox"/> Alta	<input type="checkbox"/> Média <input checked="" type="checkbox"/> Baixa
<b>Estabilidade:</b>	<input type="checkbox"/> Alta	<input type="checkbox"/> Média <input checked="" type="checkbox"/> Baixa
<b>Rationale:</b> Determina a escalabilidade do projeto		
<b>Requisitos associados:</b> RF1, RF10, RF2, RF7, RNF1, RNF2		

#### 4.4 (RNF4) Manutenção

O projeto deve ser fácil de manter e permitir a implementação de atualizações com facilidade.

<b>Código: RNF4</b>	<input type="checkbox"/> Funcional	<input checked="" type="checkbox"/> Não Funcional
---------------------	------------------------------------	---

<b>Requisito:</b> Manutenção		
<b>Descrição:</b> Deve-se fazer o jogo de modo a ser fácil implementação de atualizações.		
<b>Prioridade:</b>	<input type="checkbox"/> Alta	<input type="checkbox"/> Média <input checked="" type="checkbox"/> Baixa
<b>Estabilidade:</b>	<input type="checkbox"/> Alta <input type="checkbox"/> Média	<input checked="" type="checkbox"/> Baixa
<b>Rationale:</b> Determina a escalabilidade do projeto		
<b>Requisitos associados:</b> RF1, RF10, RF2, RF7, RNF1, RNF2, RNF2		

#### 4.5 (RNF5) Som atrativo

O projeto deve ser fácil de manter e permitir a implementação de atualizações com facilidade.

<b>Código:</b> RNF5	<input type="checkbox"/> Funcional <input checked="" type="checkbox"/> Não Funcional
<b>Requisito:</b> Som atrativo	
<b>Descrição:</b> Deve-se utilizar recursos de áudio (buzzer) a fim de dar uma melhor experiência ao jogador.	
<b>Prioridade:</b>	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Estabilidade:</b>	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Rationale:</b> Determina os recursos auditivos.	
<b>Requisitos associados:</b> RF1, RF10, RF2, RF7, RNF1, RNF2, RNF2	

### 5 Cronograma

O cronograma do projeto, resumido na tabela abaixo, foi organizado de acordo com uma lógica de implementação em que os requisitos mais básicos serão abordados nas primeiras semanas do projeto. Cada requisito a ser implementado depende diretamente do anterior, estabelecendo uma ordem sequencial essencial para o sucesso da implementação do projeto.

Conforme detalhado a seguir, a meta é concluir a implementação dos 11 requisitos funcionais do projeto nas três primeiras semanas. Considerando possíveis atrasos e desafios que possam surgir durante esse período, a semana 4 foi designada para ajustes e recuperação de eventuais atrasos. Qualquer tempo remanescente na semana 4 será dedicado à implementação dos requisitos não funcionais do projeto.

É importante destacar que ao longo de todas as semanas do projeto, simultaneamente à implementação, será elaborado um relatório técnico que documentará o processo de implementação e desenvolvimento. Este relatório será crucial para a entrega final do projeto.

Cronograma do projeto Resumido	
Atividade	Semana
<div>Não iniciado ▾ - RF1 (Implementação Física na placa FPGA)</div> <div>Não iniciado ▾ - RF2 (Controle da Nave Espacial)</div> <div>Não iniciado ▾ - RF5 (Sistema de Geração de Asteroides)</div> <div>Não iniciado ▾ - RF3 (Sistema de Vidas)</div>	06/03 a 13/03
<div>Não iniciado ▾ - RF7 (Tiro Realizado)</div> <div>Não iniciado ▾ - RF6 (Destruição do Asteroide)</div> <div>Não iniciado ▾ - RF4 (Sistema de Pontuação)</div> <div>Não iniciado ▾ - RF11 (Hitbox)</div>	13/03 a 20/03
<div>Não iniciado ▾ - RF8 (Registro de Pontuação dos Players na memória)</div> <div>Não iniciado ▾ - RF9 (Menu de Interação)</div> <div>Não iniciado ▾ - RF10 (Monitor)</div>	20/03 a 27/03
<div>Não iniciado ▾ - Correções de bugs e implementações finais</div> <div>Não iniciado ▾ - RFN5 (Som atrativo)</div> <div>Não iniciado ▾ - Montagem do protótipo para o produto final</div>	27/03 a 03/04