

**UNIVERSIDADE DO VALE DO ITAJAÍ
ESCOLA DO MAR, CIÊNCIA E TECNOLOGIA
CURSO DE ENGENHARIA DE COMPUTAÇÃO
MICROCONTROLADORES**

**IMPLEMENTAÇÃO DO JOGO ASTEIRÓIDE EM UM
MICROCONTROLADOR PIC18F4520**

por

Diogo Agenor Marchi
George de Borba Nardes

Itajaí (SC), Junho de 2020

**UNIVERSIDADE DO VALE DO ITAJAÍ
ESCOLA DO MAR, CIÊNCIA E TECNOLOGIA
CURSO DE ENGENHARIA DE COMPUTAÇÃO
MICROCONTROLADORES**

**IMPLEMENTAÇÃO DO JOGO ASTEIRÓIDE EM UM
MICROCONTROLADOR PIC18F4520**

por

Diogo Agenor Marchi
George de Borba Nardes

Relatório apresentado como requisito parcial da disciplina Microcontroladores do Curso de Engenharia de Computação para análise e aprovação.
Professor Responsável: Paulo Roberto Valim
(obs.: modelo baseado no usado na disciplina de Projeto de Sistemas Embarcados)

Itajaí (SC), Junho de 2020

SUMÁRIO

1 INTRODUÇÃO.....	4
1.1 Descrição do projeto	4
1.2 Objetivo geral.....	4
1.3 Delimitação do escopo	4
1.4 Solução Proposta.....	4
1.5 ANÁLISE DE VIABILIDADE	5
 2 PROJETO	 6
2.1 VISÃO GERAL.....	6
2.2 ANÁLISE DE REQUISITOS.....	6
2.2.1 Requisitos funcionais.....	6
2.2.2 Requisitos não funcionais.....	7
2.2.3 Regras de negócio	7
2.3 ARQUITETURA DE HARDWARE.....	8
2.4 ARQUITETURA DE SOFTWARE.....	9
2.4.1 Diagrama de Caso de Uso	10
2.4.2 Diagrama de Classe	10
2.4.3 Diagrama de Estado.....	12
2.4.4 Fluxograma	12
2.5 PLANEJAMENTO	13
2.6 CRONOGRAMA	14
 3 DESENVOLVIMENTO	 15
3.1 IMPLEMENTAÇÃO	15
3.2 VERIFICAÇÃO.....	15
3.3 RESULTADOS.....	15
 4 CONSIDERAÇÕES FINAIS / CONCLUSÕES	 16

1 INTRODUÇÃO

1.1 DESCRIÇÃO DO PROJETO

O projeto escolhido envolve o desenvolvimento do jogo de asteroide para um microcontrolador PIC. O objetivo do jogo é fazer com que a nave desvie dos asteroides que estiverem vindo em sua direção. Caso ocorra uma colisão com asteroide o jogo é finalizado. O controle da nave é feito por um GAMEPAD e o jogo é exibido em um display LCD ambos controlado por um microcontrolador.

Como o jogo será executado em um microcontrolador, algumas limitações deverão ser consideradas, como a falta de um SO para gerenciamento de entradas e saídas. Além disso, existe a necessidade de que seja considerado o funcionamento do jogo como um sistema executando várias tarefas ao mesmo tempo. Será feito um chaveamento entre funções para que o usuário tenha a impressão de que todas as funções do jogo estão sendo executadas paralelamente.

1.2 OBJETIVO GERAL

Este projeto de conclusão da matéria microcontroladores, tem como objetivo aplicar alguns conhecimentos adquiridos durante o semestre. O tema escolhido pelos alunos foi motivado por englobar alguns dos principais assuntos tratados, como integração do PIC18F4520 com o display HD44780, módulo de tratamento de interrupções externas e interrupções internas do PIC. Além disso, o desenvolvimento de um sistema utilizando task scheduling tendo como base o curso ofertado pela Microchip[1].

1.3 DELIMITAÇÃO DO ESCOPO

Será usado um display de 16 x 4(LM041L) como saída, e como entrada um keypad alterado para as necessidades do jogo. O keypad terá setas no eixo vertical e horizontal, uma tecla “enter” e uma tecla “escape”. O microcontrolador que será usado é o PIC18f4520. Todos os componentes podem ser encontrados no ambiente de simulação Proteus.

1.4 SOLUÇÃO PROPOSTA

Foram feitas algumas pesquisas sobre jogo de Asteroide com o display LCD, e pelo que foi achado[2], existem jogos apenas com o microprocessador Arduino UNO ou MEGA,

então, o asteroide com o PIC18f4520 seria um diferencial, e também, este projeto movimentará a nave nos dois eixos, eixo x e eixo y, em um display LCD 16x4.

Tabela 1– Comparativo da solução proposta com soluções existentes

Nome	Microcontrolador	Display	movimento
<u>SanticN4N</u>	Arduino Uno	LCD16X2	Sensor movimento
Este projeto	PIC18F4520	LCD16X4	Controle com botão

1.5 ANÁLISE DE VIABILIDADE

O projeto escolhido é viável como prova de conceito pois engloba diversos assuntos discutidos durante o semestre. Pode-se citar os conceitos de interrupção, timer e integração com dispositivos externos como principais motivos de viabilidade. Além disso, outro fator que viabiliza é que o nível de dificuldade do projeto fica dentro do esperado levando em conta o tempo disponível para execução.

2 PROJETO

2.1 VISÃO GERAL

O projeto escolhido envolve o desenvolvimento do jogo de asteroide para um microcontrolador PIC. O objetivo do jogo é fazer com que a nave desvie dos asteroides que estiverem vindo em sua direção. O controle da nave é feito por um GAMEPAD e o jogo é exibido em um display LCD. Será usado um display de 16 x 4 (LM041L) como saída, e como entrada um keypad alterado para as necessidades do jogo. O keypad terá setas no eixo vertical e horizontal, uma tecla “enter” e uma tecla “escape”.

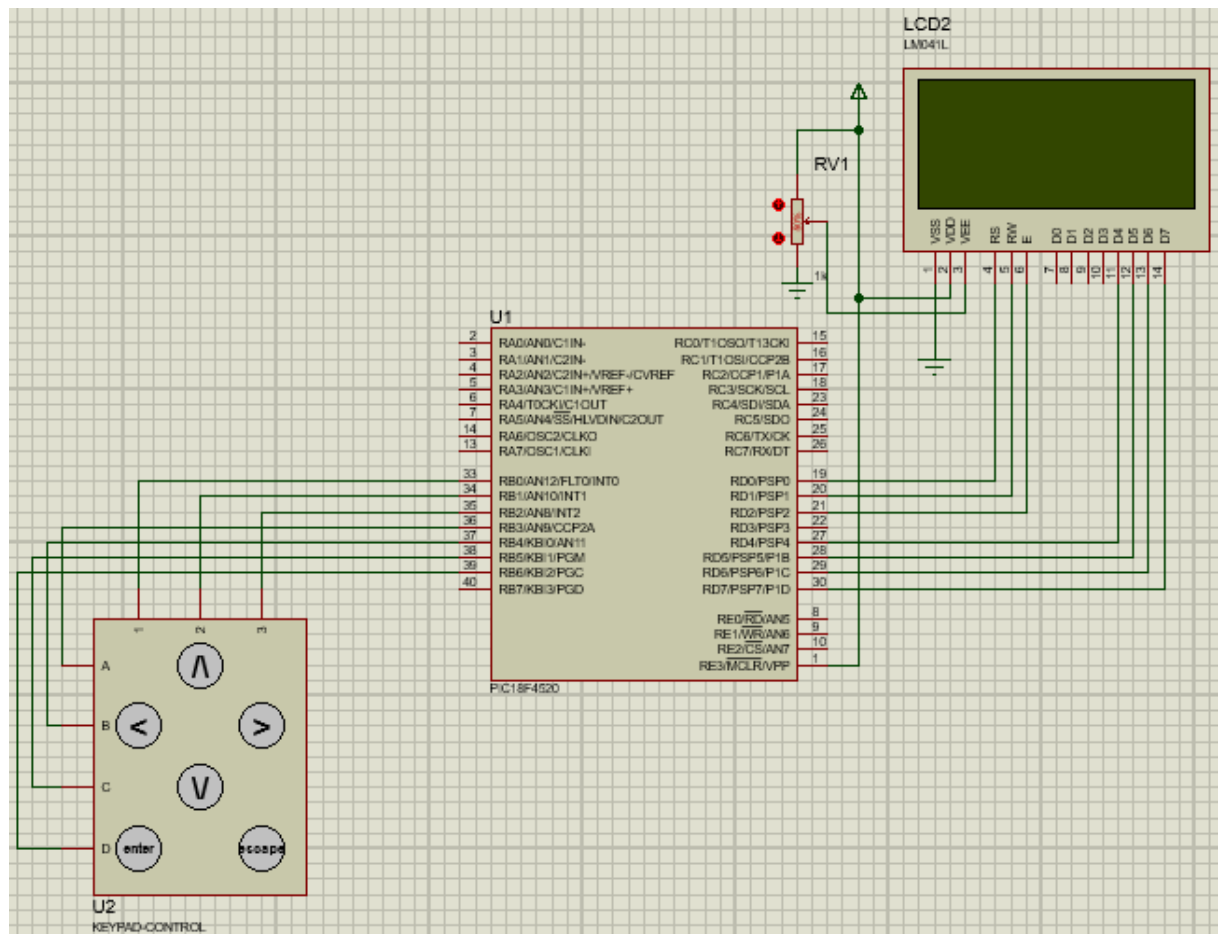


Figura 1- Componentes: Keypad, PIC e Display

2.2 ANÁLISE DE REQUISITOS

2.2.1 Requisitos funcionais

- RF01: O sistema deve permitir que o usuário navegue pelo menu principal;

- RF02: O sistema deve permitir que o usuário selecione uma opção do menu principal;
- RF03: O sistema deve permitir que o usuário retorne ao menu principal a qualquer momento;
- RF04: O sistema deve incrementar a pontuação durante o jogo.
- RF05: O sistema deve permitir que o usuário movimente a nave no espaço em todas as direções.
- RF06: O sistema deve identificar uma colisão da nave com um asteroide.

2.2.2 Requisitos não funcionais

- RNF01: o sistema será prototipado no microcontrolador PIC18F4520;
- RNF02: o sistema será simulado na ferramenta de simulação Proteus;
- RNF03: o código da aplicação será escrito em linguagem c;
- RNF04: será utilizado o ambiente de desenvolvimento MPLAB;
- RNF05: o protótipo deverá ter uma vazão de pelo menos 10 frames/segundo (conforme a aplicação);
- RNF06: o sistema deverá ser exibido no display LM041L 16x4
- RNF07: as entradas de usuário serão por meio de um teclado.
- RNF08: o teclado de entrada deve possuir os seguintes botões: CIMA, BAIXO, ESQUERDA, DIREITA, ENTER e ESCAPE.

2.2.3 Regras de negócio

- RN01: O menu principal deve ter três opções: jogar, instruções e créditos;
- RN02: A pontuação deve ser incrementada a cada asteroide desviado;
- RN03: Na ocorrência de uma colisão o jogo deve ser interrompido.
- RN04: Ao ser interrompido por uma colisão, o sistema deve informar a pontuação final.

- RN05: Ao ser interrompido por uma colisão e apresentar a pontuação final o sistema deve voltar ao menu principal.
- RN06: A nave poderá ser movimentada nos eixos XY.

2.3 ARQUITETURA DE HARDWARE

A figura 2 apresenta um esquemático do projeto de como será ligado os componentes e o que será usado.

A Figura 3 apresenta do diagrama de blocos do projeto. O bloco GAMEPAD é uma matriz de botões adaptada para as necessidades do jogo. O bloco DISPLAY LM041L será o display LCD no qual o jogo será exibido. Ambos os blocos terão as ações controladas pelo bloco Módulo Timer, que é interno ao PIC.

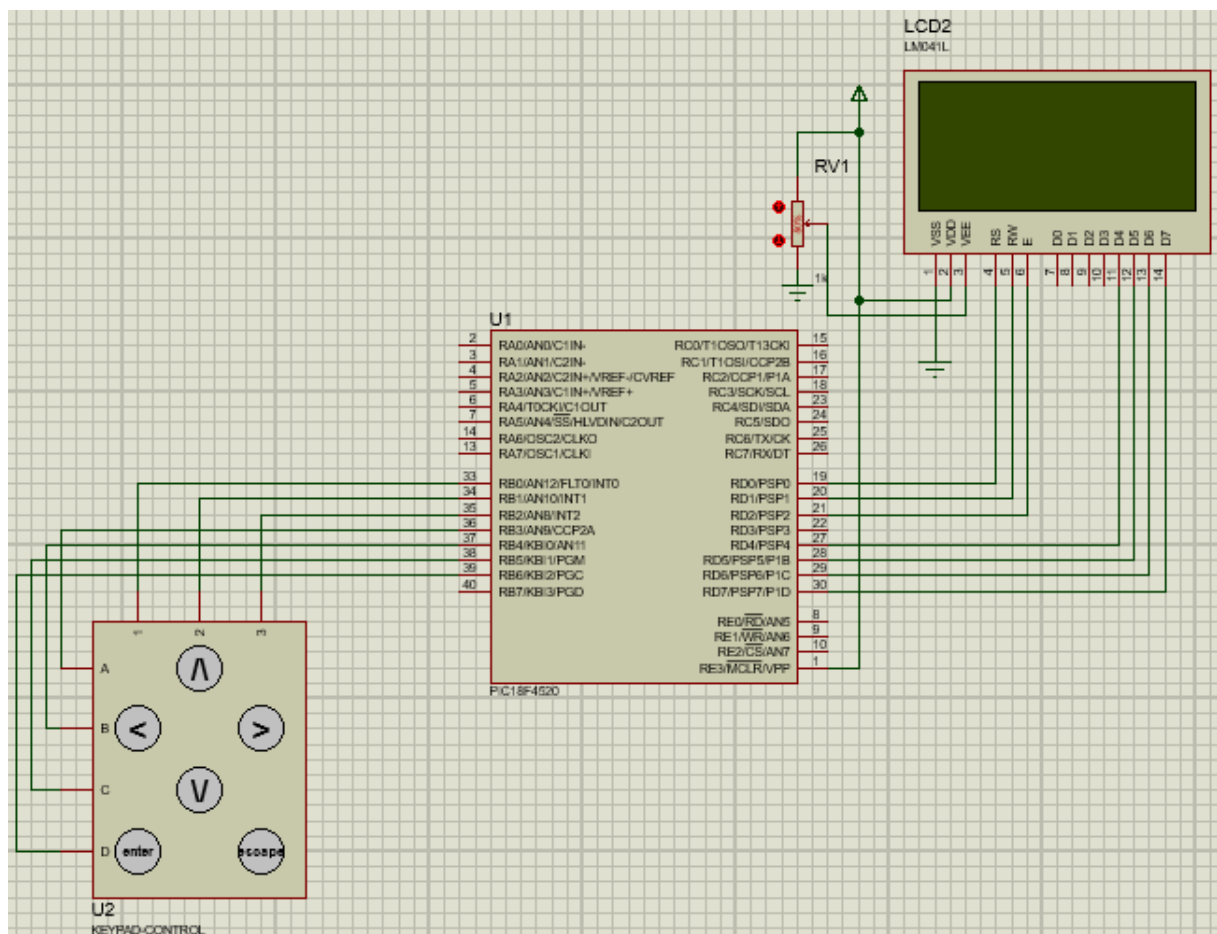


Figura 2 – Esquemático

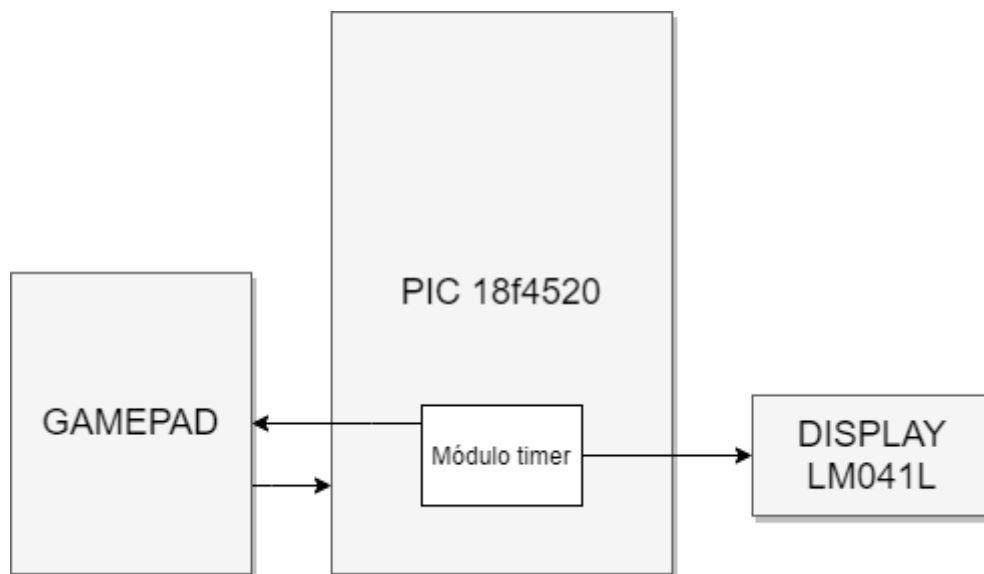


Figura 3- Diagrama de Bloco do Projeto

A saída do GAMEPAD é conectada as portas de entrada e saída do PIC. O estado das teclas será verificado de tempo em tempo.

2.4 ARQUITETURA DE SOFTWARE

Nesta seção será apresentado como planejamos construir o software para o projeto. Como ainda houve nenhuma implementação, pode ser que o que for descrito abaixo sofra algumas alterações conforme limitações ou novas ideias forem surgindo. Ao finalizar a implementação, a arquitetura descrita será revisada e ajustada conforme o resultado final, para que seja uma descrição com maior fidelidade.

O software será planejado da seguinte forma: algumas funções de print de tela, como menu, créditos e instruções, onde essas funções serão chamadas conforme for clicado no keypad control e isso trará ao usuário um software de tempo real. Terá a biblioteca keypad.h/keyppad.c onde nela estará descrita função que lerá as entradas. No arquivo task_maneger.h/task_maneger.c sera criado interrupções por timer, onde será programado entre 10-20ms para ficar verificando quais botões foram clicados.

Se algum botão foi pressionado, verificará onde o programa está, se está no menu, jogo, créditos. e tomará alguma decisão se no ambiente que estiver essa tecla tem algum funcionamento, por exemplo, se o código está no ambiente “créditos” e o usuário aperta a tecla “>” o código entende que isso não fará nada, pois neste ambiente o usuário poderá ler quem

produziu o projeto e logo após terminar a leitura, ele poderá apertar a tecla “escape” para voltar ao menu e nenhuma outra tecla fará movimentos pelo display LCD.

2.4.1 Diagrama de Caso de Uso

A figura 4 mostra os casos de uso envolvidos na interação do usuário com o sistema.

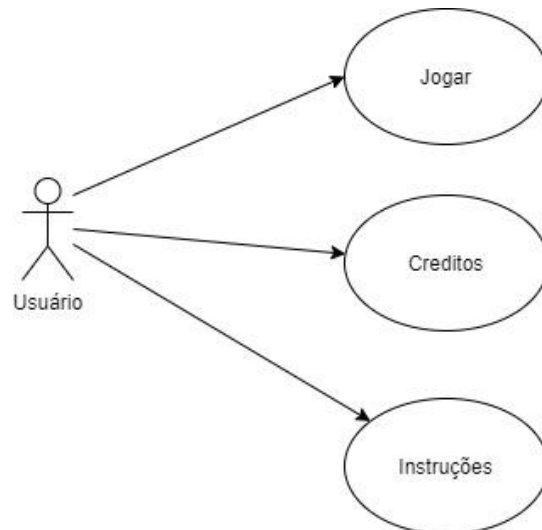


Figura 4 - Diagrama Casos de Uso

O usuário terá apenas 3 ações possíveis no sistema: jogar, visualizar créditos e visualizar instruções.

2.4.2 Diagrama de Classe

A Figura 5 apresenta o diagrama de classe, as classes são os arquivos do software. É importante ressaltar que ainda não foi especificado detalhadamente cada atributo dos arquivos, apenas a relação entre eles.

O arquivo principal é o “main.c”. É dele que parte todas as chamadas de funções e procedimentos. O arquivo “config.h” é onde será parametrizado as configurações do microcontrolador. O arquivo “task_maneger” é o que realizará o controle dos diferentes estados do sistema. Ele fará uso dos arquivos “lcd” e “keypad” para interagir com os componentes externos display e teclado respectivamente.

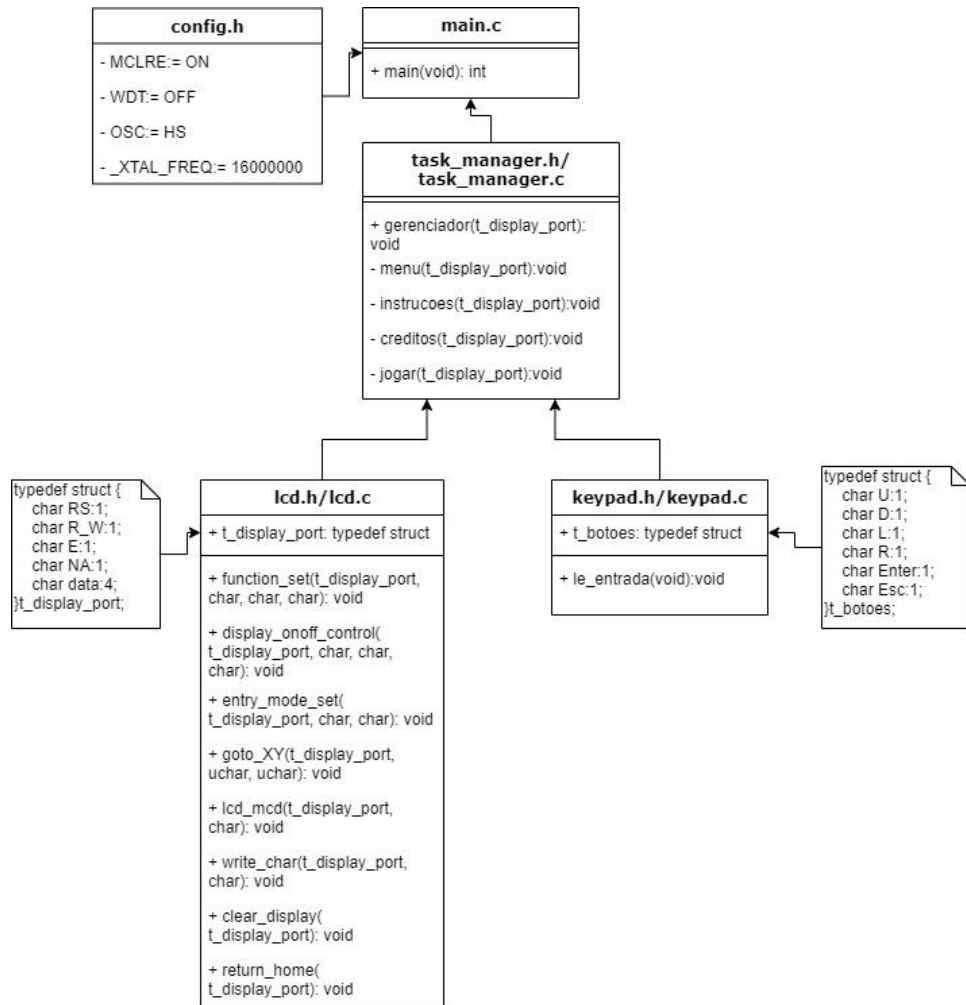


Figura 5 - Diagrama de Classe Completo

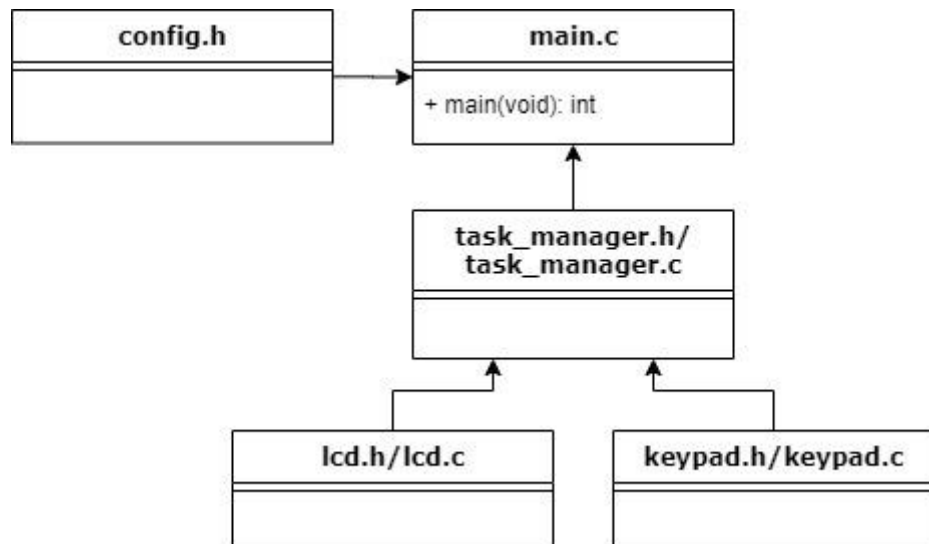


Figura 6 - Diagrama de Classe Simplificado

2.4.3 Diagrama de Estado

O diagrama de estado da Figura 6 apresenta todos os estados atingíveis durante a execução do jogo. Fica implícito que cada estado pode ser interrompido ao microcontrolador ser desenergizado.

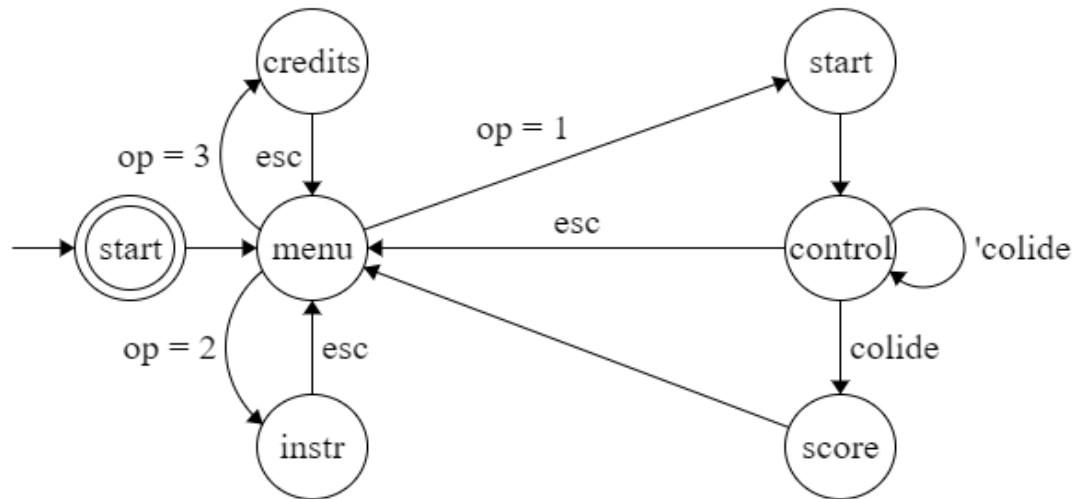


Figura 7- Diagrama de estados do jogo

2.4.4 Fluxograma

A fluxograma da figura 7 apresenta todos os passos que o usuário poderá dar ao iniciar o jogo. Ao início do programa, ele começa no menu, que tem 4 possibilidades para o usuário escolher: jogar, instruções, créditos ou finalizar.

Cada tela que ele entrar, se apertar a tecla “escape”, voltará ao menu. Ao entrar no jogo, o usuário comandará a nave através do keypad control, se bater a nave no asteroide, mostrará a pontuação na tela e voltará ao menu.

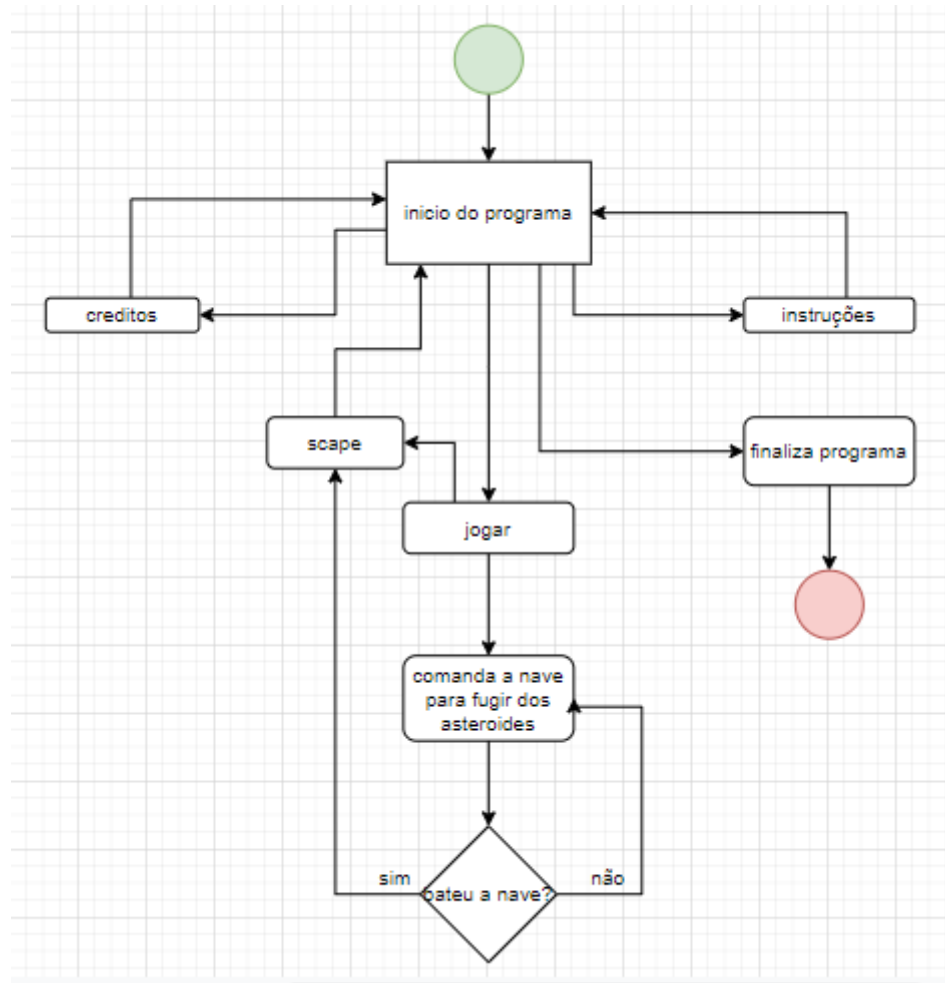


Figura 8 - Fluxograma

2.5 PLANEJAMENTO

O jogo será feito na linguagem C no MPLAB X com integração do Proteus com alguns requisitos descritos anteriormente. Para verificar se os requisitos poderão ser atendidos, foi feito uma tabela de testes abaixo:

Quadro 1 – Plano de verificação

Requisito	Procedimento de verificação/teste	Resultado esperado
RF01	Apertar nas setas do keypad control	Que o cursor modifique
RF02	Apertar enter em alguma das linhas do display	Que troque de tela
RF03	Apertar escape na tela do jogo	Volte ao menu
RF04	Ao passar por asteroides sem colidir	Incremente a pontuação
RF05	Apertar nas setas do keypad control	Nave se mova
RF06]	Ao colidir a nave com um asteroide	Acabe o jogo e mostre a pontuação final
RNF01	Verificar o microcontrolador	Ser o PIC18F4520
RNF02	Verificar a ferramenta de simulação	Ambiente Proteus
RNF03	Verificar linguagem de codificação	Linguagem C
RNF04	Verificar ambiente de desenvolvimento	MPLAB X
RNF05	Verificar vazão do protótipo	10 frames/segundo
RNF06	Verificar em qual display está sendo apresentado o jogo	LM041L 16x4
RNF07	Apertar nas teclas	Verificar se mudou algo no display
RNF08	Verificar as teclas do keypad control	Ter as teclas CIMA, BAIXO, DIREITA, ESQUERDA, ENTER e ESCAPE.

2.6 CRONOGRAMA

Para a primeira semana foi feito uma introdução do que será o projeto, mostrando o que será apresentado e quais componentes seriam usados.

Para a segunda semana será entregue este documento mostrando umas especificações a mais do projeto, como a arquitetura do software e como é pensado a implementação

Terceira, quarta e quinta semana será o tempo para a implementação do código. Quinta semana será a entrega do documento final com o código pronto.

Quadro 1 – Cronograma de execução

Atividade	Sem. 1	Sem. 2	Sem. 3	Sem. 4	Sem. 5	Entregável
Introdução do projeto	X					Documento
Especificação		X				Documento
Implementação			X	X	X	Código
Documentação final					X	Documento final e apresentação do projeto

3 DESENVOLVIMENTO

4 CONSIDERAÇÕES FINAIS / CONCLUSÕES

REFERÊNCIAS

[1] 20024 FRM4 - Interrupt and Task Scheduling - No RTOS Required

Disponível em: <https://www.youtube.com/watch?v=RnOzACq78dk>

[2] SanticN4N, (2011). Arduino Asteroid Game.

Disponível em: <https://www.instructables.com/id/Arduino-Asteroid-Game/>