



**Documento de Casos de Uso**

Core-MUSA

Universidade Estadual de Feira de Santana

**Build 3**

## Histórico de Revisões

Date	Descrição	Autor(s)
08/10/2014	Concepção do documento	<ul style="list-style-type: none"><li>• bezourokq;</li><li>• wsbittencourt;</li><li>• fmbboaventura;</li></ul>
13/10/2014	Build 2: Novo modelo de caso de uso	<ul style="list-style-type: none"><li>• wsbittencourt;</li><li>• jadsonfirmo;</li><li>• fmbboaventura;</li></ul>
16/10/2014	Build 3: Novo modelo de caso de uso	<ul style="list-style-type: none"><li>• wsbittencourt;</li></ul>
20/10/2014	Adição caso de uso LW e SW	<ul style="list-style-type: none"><li>• kelvincarmo;</li></ul>
23/10/2014	Revisão	<ul style="list-style-type: none"><li>• jadsonfirmo;</li></ul>
29/10/2014	Inclusão Casos de Uso: JPC	<ul style="list-style-type: none"><li>• di3goleite;</li></ul>
29/10/2014	Inclusão Casos de Uso: RET e NOP	<ul style="list-style-type: none"><li>• mtcastro;</li></ul>
30/10/2014	Refatoração do documento	<ul style="list-style-type: none"><li>• di3goleite;</li></ul>
30/10/2014	Inclusão Casos de Uso: HALT	<ul style="list-style-type: none"><li>• mtcastro;</li></ul>

## SUMÁRIO

<b>1</b>	<b>Introdução</b>	<b>4</b>
1.1	Objetivo . . . . .	4
1.2	Visão Geral do Documento . . . . .	4
1.3	Representação Simbólica . . . . .	4
1.4	Definições, Acrônimos e Abreviações . . . . .	5
<b>2</b>	<b>Atores do Sistema</b>	<b>5</b>
<b>3</b>	<b>Casos de Usos</b>	<b>5</b>
3.1	[UC 001] Execução de instruções . . . . .	5
3.1.1	Fluxo Principal de Eventos . . . . .	6
3.2	[UC 002] BRFL . . . . .	7
3.2.1	Fluxo Principal de Eventos . . . . .	7
3.3	[UC 003] Instrução LW . . . . .	8
3.3.1	Fluxo Principal de Eventos . . . . .	8
3.4	[UC 004] Instrução SW . . . . .	9
3.4.1	Fluxo Principal de Eventos . . . . .	9
3.5	[UC 005] CALL . . . . .	10
3.5.1	Fluxo Principal de Eventos . . . . .	11
3.6	[UC 006] JR . . . . .	11
3.6.1	Fluxo Principal de Eventos . . . . .	12
3.7	[UC 007] JPC . . . . .	12
3.7.1	Fluxo Principal de Eventos . . . . .	13
3.8	[UC 008] Instruções Lógicas e Aritméticas. . . . .	13
3.8.1	Fluxo Principal de Eventos . . . . .	14

3.9	[UC 009] RET . . . . .	14
3.9.1	Fluxo Principal de Eventos . . . . .	15
3.10	[UC 010] NOP . . . . .	15
3.10.1	Fluxo Principal de Eventos . . . . .	16
3.11	[UC 011] HALT . . . . .	16
3.11.1	Fluxo Principal de Eventos . . . . .	17

## 1. Introdução

Este documento tem como objetivo a especificação dos casos de uso do projeto Core Musa (concepção de um processador simples de propósito geral). O documento detalha cada caso de uso indicando os atores, os eventos (ações) e as condições de cada caso, além dos diagramas de casos de uso.

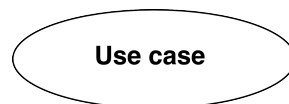
### 1.1. Objetivo

### 1.2. Visão Geral do Documento

- Sessão 2: Lista todos os possíveis atores do sistema.
- Sessão 3: Relata a lista dos casos de uso do projeto.

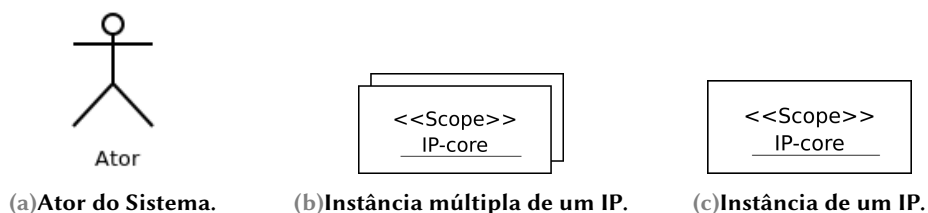
### 1.3. Representação Simbólica

A Figura 1 ilustra a simbologia utilizada para representar operações que devem ser realizadas pelo sistema. A Figura 2 apresenta os modelos de ilustração utilizados para representar os Atores do sistema. Um ator, dentro do escopo desta descrição, pode ser identificado como um módulo *top level*, ou como um elemento de entrada e saída (botoes, sensores, *displays*, etc).



**Figura 1: Exemplo de Caso de Uso.**

A simbologia usual para representação de um Ator é apresentada na Figura 2a, no entanto, para representar módulos incorporados, utiliza-se as representações ilustradas nas Figuras 2b e 2c, definidas por convenção. Este elemento, em geral, está associado aos módulos do sistema, ou IP cores de terceiros incorporados ao mesmo. Esta simbologia foi dividida, com o objetivo de representar instâncias únicas (Figura 2c), ou múltiplas (Figura 2b) de um determinado componente.



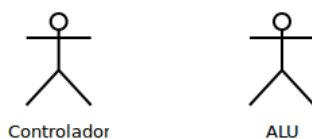
**Figura 2: Simbologia utilizada na implementação dos Casos de Uso.**

O projetista responsável por interpretar os diagramas não deve confundir-se no momento de analisar as simbologias de atores. A representação alternativa, não implica que o módulo será instanciado no subsistema em questão, mas sim que os recursos providos por este *core* são necessários para garantir o seu funcionamento.

#### 1.4. Definições, Acrônimos e Abreviações

Termo	Descrição
UC	Caso de Uso
ALU	Unidade Lógica e Aritmética
SB	Sub-fluxo
FS	Fluxo Secundário
NFR	Requisito Não Funcional
FR	Requisito Funcional
BT	Botão Direcional
PC	<i>Program Counter</i>

## 2. Atores do Sistema



**Controlador** – Unidade que controla a execução das operações.

**ALU** – Unidade Lógica e Aritmética.

## 3. Casos de Usos

Esta sessão apresenta o conjunto de UC realizados para a implementação do projeto *Core MUSA* (Núcleo de processamento de instruções do processador de propósito geral MUSA). As sessões a seguir foram divididas e nomeada utilizando a nomenclatura abreviada [UC (NÚMERO DO UC)] seguido de uma breve descrição em forma de título.

### 3.1. [UC 001] Execução de instruções

O controlador é responsável por decodificar instrução, solicitar operações na ALU e garantir o armazenamento dos resultados das operações no banco registradores.

*Atores*

**Controlador** – Unidade que controla a execução das operações.

**ALU** – Unidade Lógica e Aritmética.

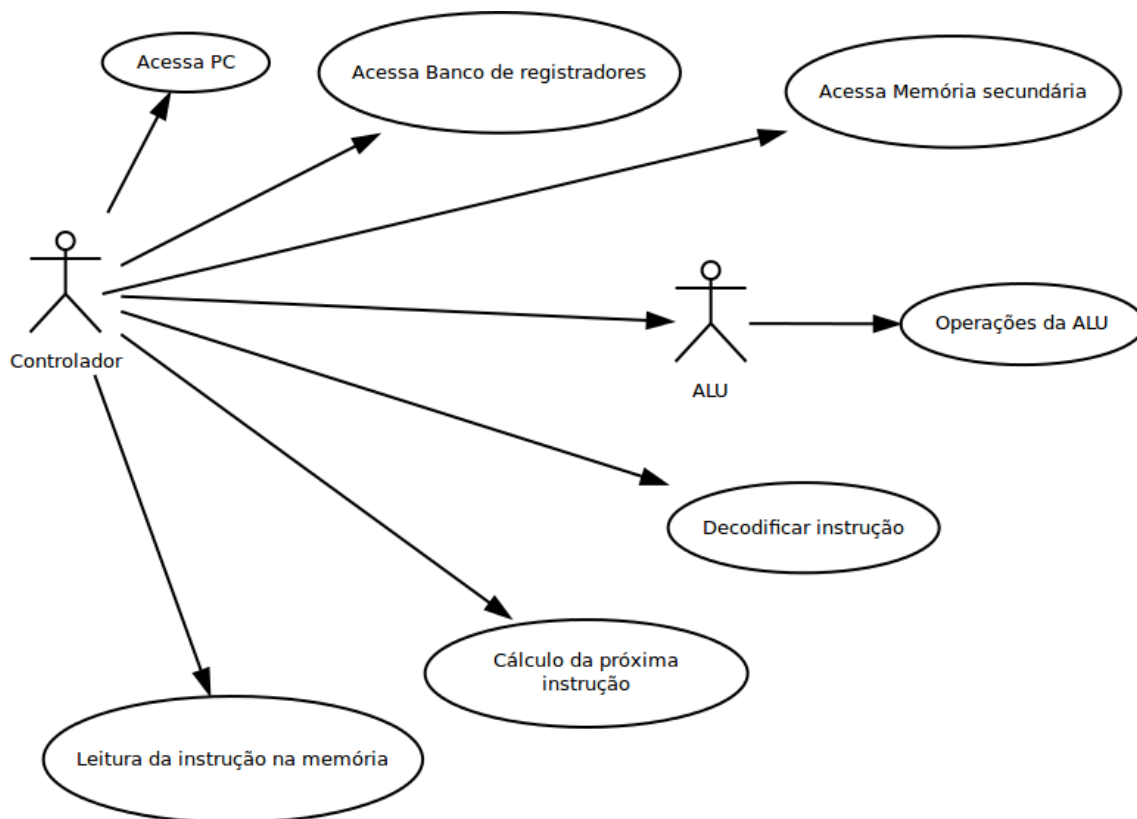
### Pré-condições

- Atender aos requisitos funcionais [FR01 e FR02];
- Leitura do PC;
- Realizar operações lógicas e aritméticas na ALU;

### Pós-condições

- Os resultados devem ser expressos nos registradores.

### Diagrama de Caso de Uso



#### 3.1.1. Fluxo Principal de Eventos

- P1. Acesso ao PC;
- P2. Leitura da instrução apontada por PC;
- P3. Acesso aos respectivos registradores;
- P4. Executa operações;
- P5. Atualiza registradores;
- P6. Atualiza valor do PC;

### 3.2. [UC 002] BRFL

O Processador tem a capacidade de fazer desvios condicionais através da utilização das flags do sistema.

*Atores*

**Controlador** – Unidade que controla a execução das operações.

**ALU** – Unidade Lógica e Aritmética.

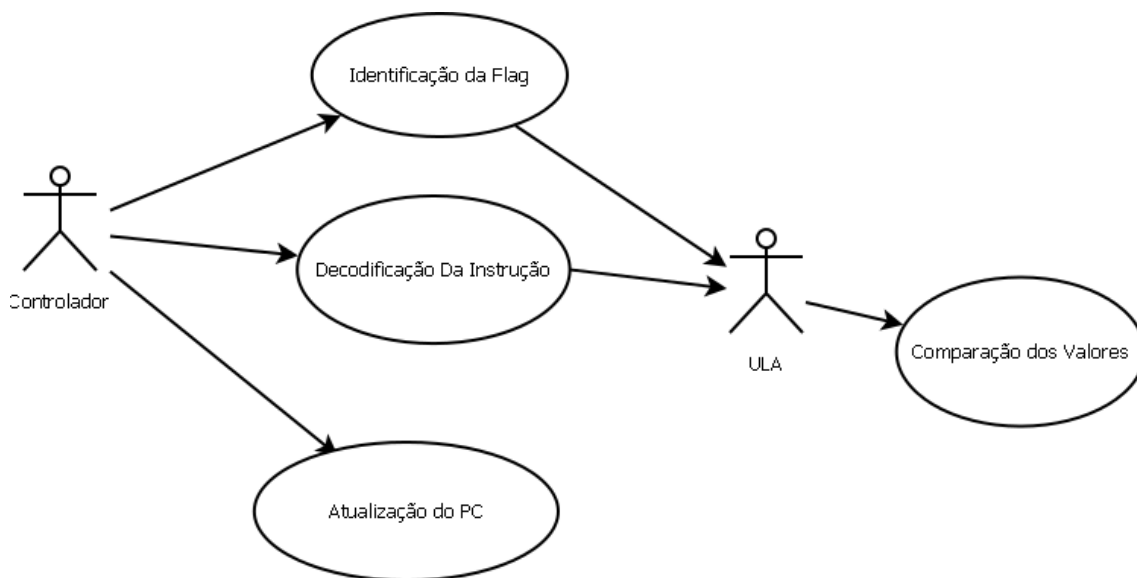
*Pré-condições*

- Atender ao requisito funcional [FR14];
- Leitura do PC;
- Realizar operações lógicas na ALU;

*Pós-condições*

- Alteração do PC caso verdadeira.

*Diagrama de Caso de Uso*



#### 3.2.1. Fluxo Principal de Eventos

- P1. Acesso ao PC;
- P2. Leitura da instrução apontada pelo PC;
- P3. Identificação e leitura da *Flag* ativa;
- P4. Executa a operação lógica;
- P5. Atualiza o valor do PC;



### 3.3. [UC 003] Instrução LW

O processador é capaz de carregar dados da memória pro banco de registradores.

*Atores*

**Controlador** – Unidade que controla a execução das operações.

**ALU** – Unidade Lógica e Aritmética.

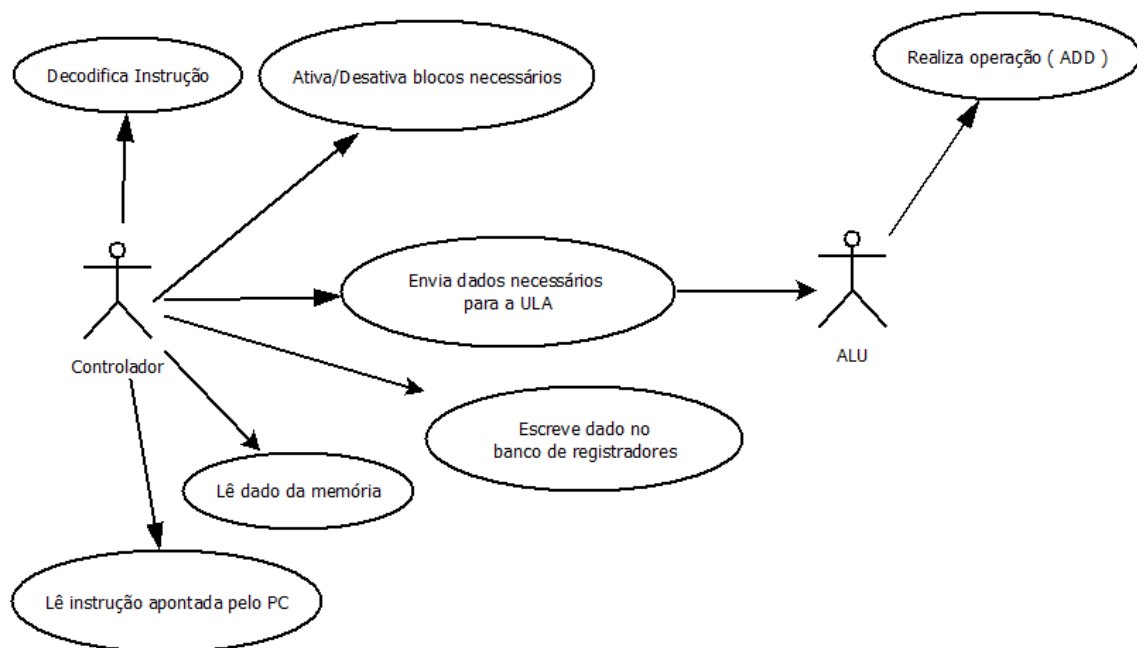
*Pré-condições*

- Leitura da Memória de instrução;
- A instrução necessita ter o *opcode* específico para a instrução LW;

*Pós-condições*

- O dado deverá ser salvo no registrador de escrita (RD) do banco de registradores;

*Diagrama de Caso de Uso*



#### 3.3.1. Fluxo Principal de Eventos

- P1. Decodificação da instrução lida na Memória de Programa;
- P2. Acesso aos respectivos registradores;
- P3. Leitura do endereço base, a partir do registrador RT;
- P4. Extensão do valor de 16 bits lido na instrução para 32 bits;

**P5.** Operação de soma com os dados de 32 bits, dado lido no registrador, com o valor de 16 bits estendido para 32 bits;

**P6.** Resultado da operação será lido pela memória de dados, que servirá como endereço de memória para ler o dado;

**P7.** O dado será enviado para o banco de registradores, e será escrito no registrador de escrita RD;

### 3.4. [UC 004] Instrução SW

O processador é capaz de escrever dados na memória.

*Atores*

**Controlador** – Unidade que controla a execução das operações.

**ALU** – Unidade Lógica e Aritmética.

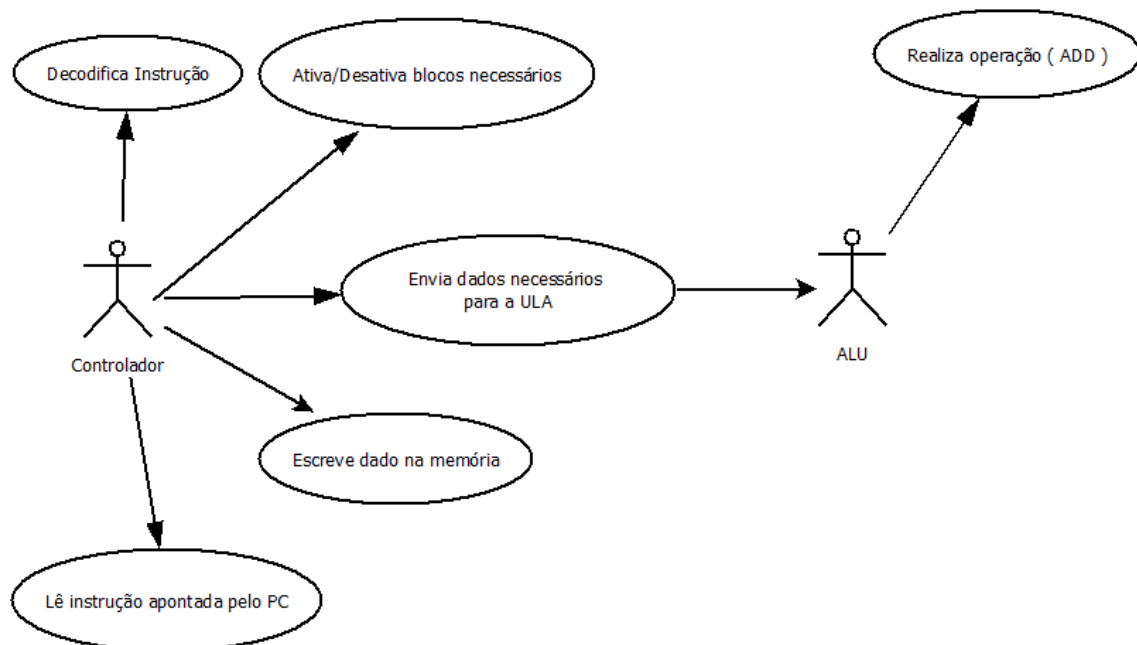
*Pré-condições*

- Leitura da Memória de instrução;
- A instrução necessita ter o *opcode* específico para a instrução SW;

*Pós-condições*

- O dado deverá ser escrito na memória de dados.

*Diagrama de Caso de Uso*



#### 3.4.1. Fluxo Principal de Eventos

**P1.** Decodificação da instrução lida na Memória de Programa;

- P2. Acesso aos respectivos registradores;
- P3. Leitura do endereço base, a partir do registrador RT;
- P4. Leitura do dado a ser escrito no registrador RS;
- P5. Extensão do valor de 16 bits, lido na instrução, para 32 bits;
- P6. Operação de soma com os dados de 32 bits com os seguintes valores: endereço lido do registrador RT e valor de 32 bits estendido;
- P7. Resultado da operação será lido pela Memória de Dados, que servirá como endereço de memória para escrever o dado.
- P8. O dado será escrito na Memória de Dados.

### 3.5. [UC 005] CALL

O Processador deve ser capaz de desviar um programa em execução para uma subrotina.

*Atores*

**Controlador** – Unidade que controla a execução das operações.

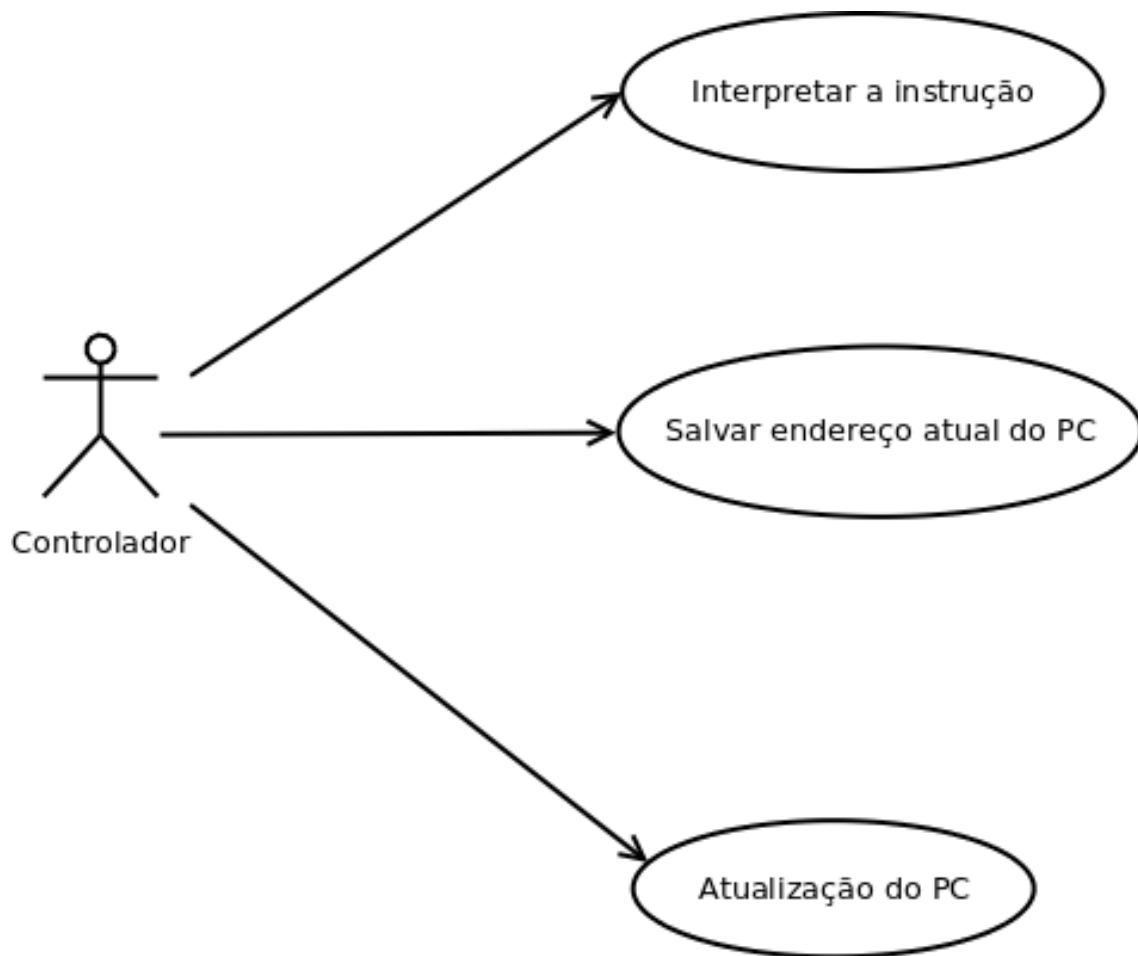
*Pré-condições*

- Atender ao requisito funcional [FR15];
- Leitura do PC;
- Ter espaço disponível na pilha de memória;

*Pós-condições*

- Alteração do PC.

### Diagrama de Caso de Uso



#### 3.5.1. Fluxo Principal de Eventos

- P1. Acesso ao PC;
- P2. Leitura da instrução apontada por PC;
- P3. Salvar o endereço atual do PC na pilha de memória;
- P4. Modifica o valor do PC para o endereço recebido pela instrução;

#### 3.6. [UC 006] JR

O Processador deve ser capaz de desviar um programa em execução para um endereço de destino.

#### Atores

**Controlador** – Unidade que controla a execução das operações.

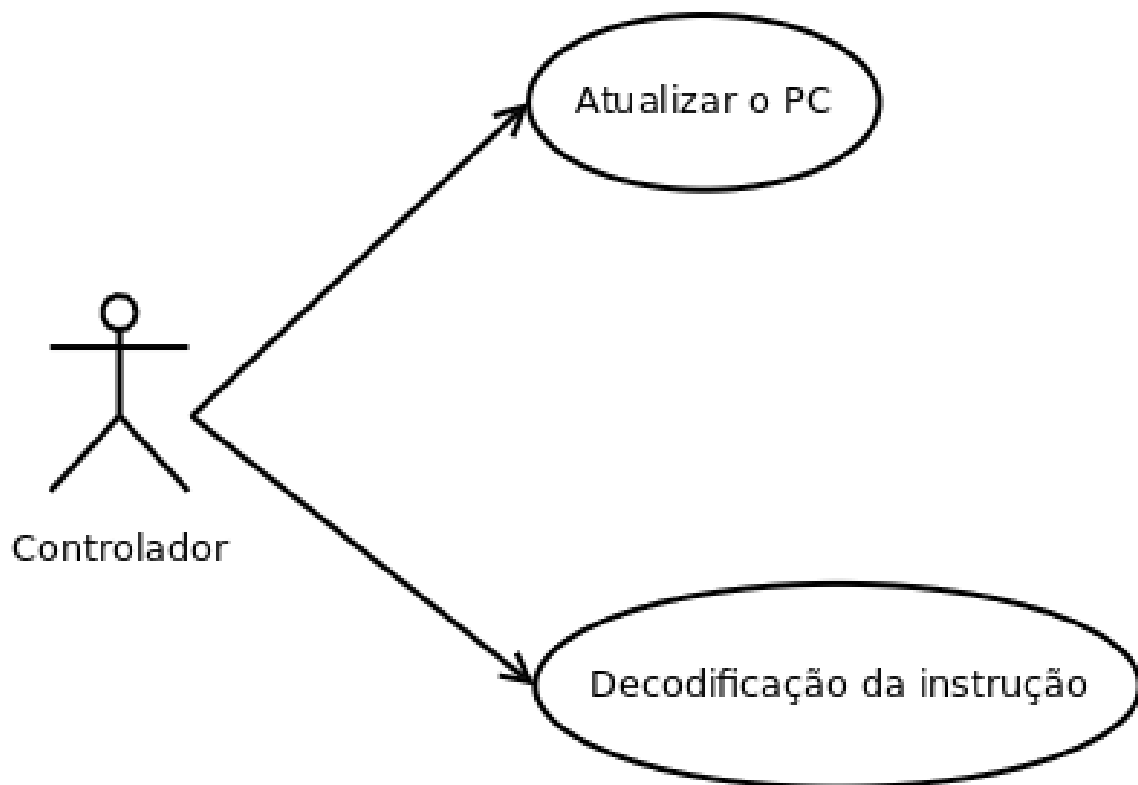
#### *Pré-condições*

- Atender ao requisito funcional [FR12];
- Leitura do PC;

#### *Pós-condições*

- Alteração do PC.

#### *Diagrama de Caso de Uso*



#### *3.6.1. Fluxo Principal de Eventos*

- P1.** Acesso ao PC;
- P2.** Leitura da instrução apontada pelo PC;
- P3.** Modifica o valor do PC pro endereço recebido na instrução;

#### **3.7. [UC 007] JPC**

O processador deve ser capaz de desviar um programa em execução para um endereço relativo ao PC.

#### *Atores*

**Controlador** – Unidade que controla a execução das operações.

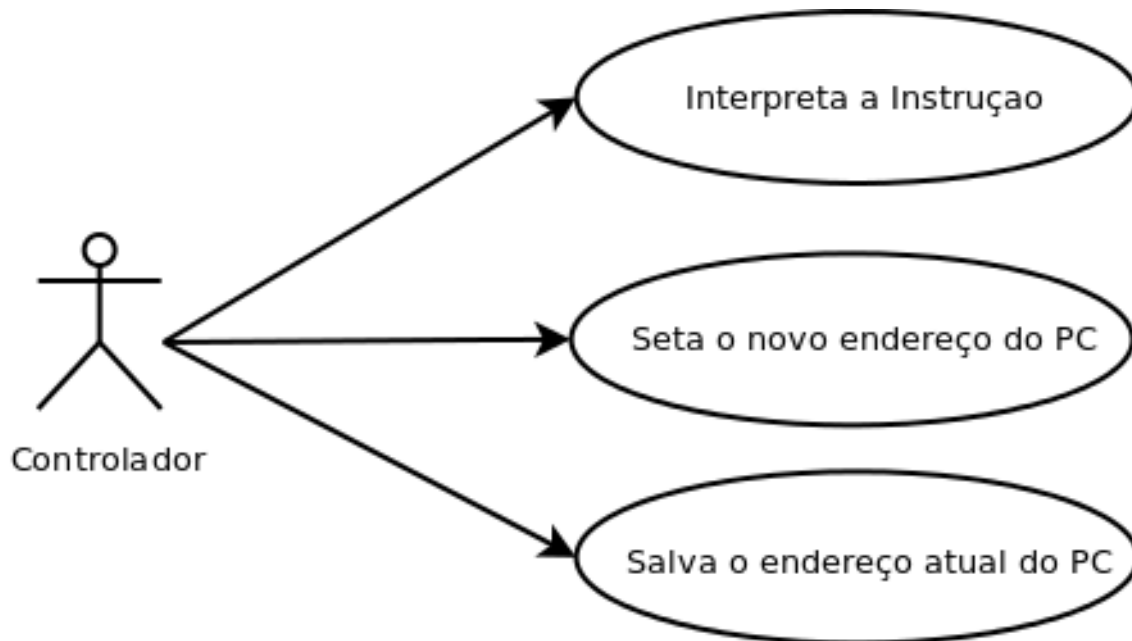
#### *Pré-condições*

- Leitura do PC;

#### *Pós-condições*

- Alteração do PC.

#### *Diagrama de Caso de Uso*



#### *3.7.1. Fluxo Principal de Eventos*

- P1.** Acessa o valor de PC;
- P2.** Faz a leitura da instrução apontada pelo PC;
- P3.** Modifica o valor do PC pro endereço recebido na instrução;

#### **3.8. [UC 008] Instruções Lógicas e Aritméticas.**

O controlador é responsável por decodificar instrução, solicitar operações na ALU e garantir o armazenamento dos resultados das operações no banco registradores.

#### *Atores*

**Controlador** – Unidade que controla a execução das operações.

**ALU** – Unidade Lógica e Aritmética.

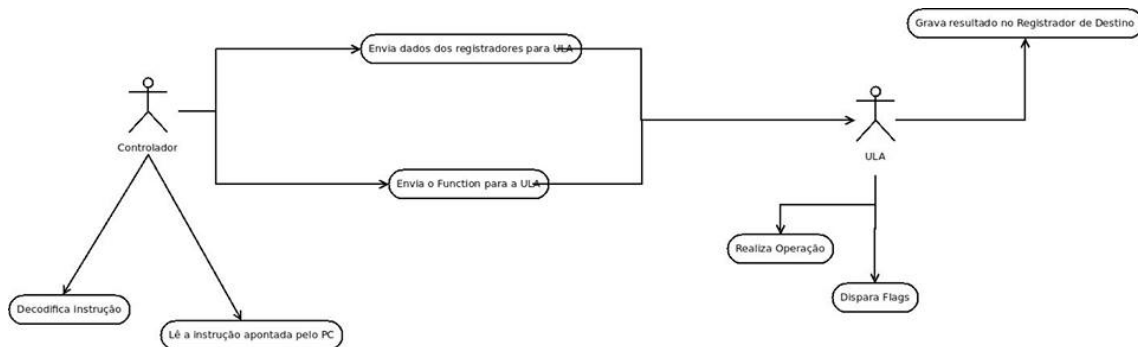
#### *Pré-condições*

- Atender aos requisitos funcionais [FR03 a FR10];
- Leitura do PC;
- Realizar operações lógicas e aritméticas na ALU;

### Pós-condições

- Os resultados devem ser expressos nos registradores.

### Diagrama de Caso de Uso



#### 3.8.1. Fluxo Principal de Eventos

- P1. Acesso ao PC;
- P2. Leitura da instrução apontada por PC;
- P3. Acesso aos respectivos registradores;
- P4. Controlador direciona os dados dos registradores para a entrada da ULA;
- P5. Controlador envia o *function* para ativar a operação desejada na ULA;
- P6. A ULA realiza as operações;
- P7. *Flags* são disparadas, caso seja necessário;
- P8. Envia o resultado pro registrador de destino;

### 3.9. [UC 009] RET

O processador deve ser capaz de retornar do último desvio tomado pelo programa.

#### Atores

**Controlador** – Unidade que controla a execução das operações.

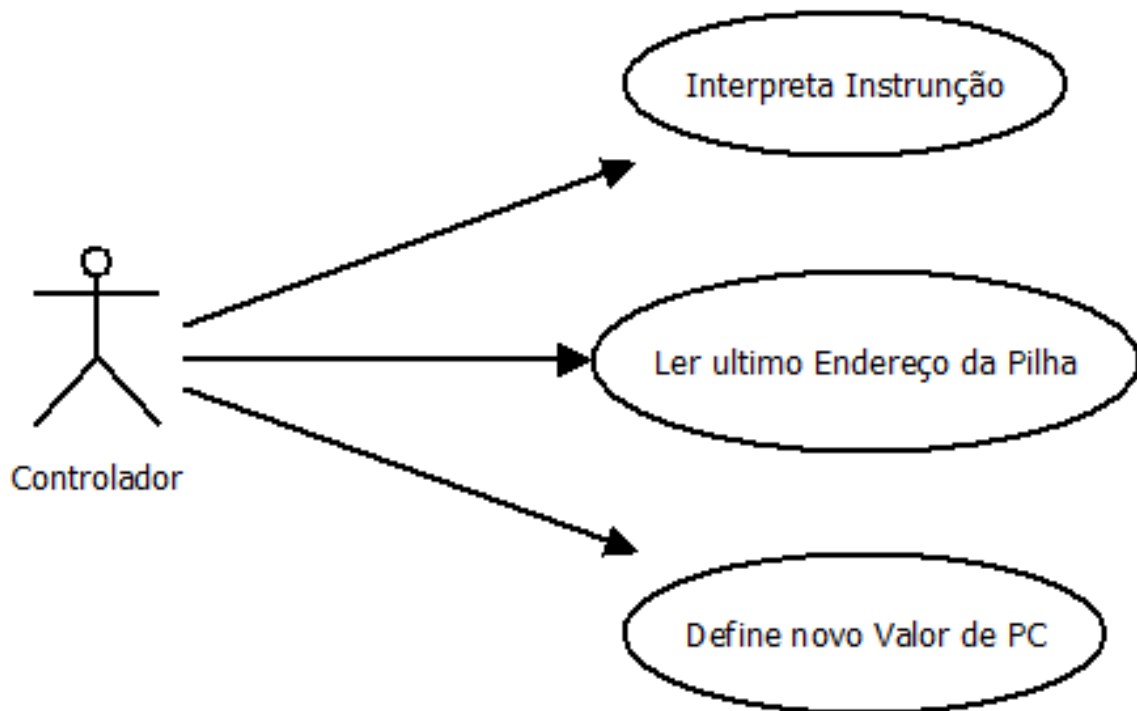
#### Pré-condições

- Atender ao requisito funcional [FR15];
- Leitura do PC;

#### Pós-condições

- Alteração do PC.

### Diagrama de Caso de Uso



#### 3.9.1. Fluxo Principal de Eventos

- P1. Acesso ao PC;
- P2. Leitura da instrução apontada pelo PC;
- P3. Leitura do endereço do topo da pilha;
- P4. Remoção do endereço do topo da pilha;
- P5. Incrementa o endereço removido do topo da pilha;
- P6. Modifica o valor do PC pro valor incrementado;

#### 3.10. [UC 010] NOP

O processador deve ser capaz de não realizar operações durante cinco ciclos de *clock*.

##### Atores

**Controlador** – Unidade que controla a execução das operações.

##### Pré-condições

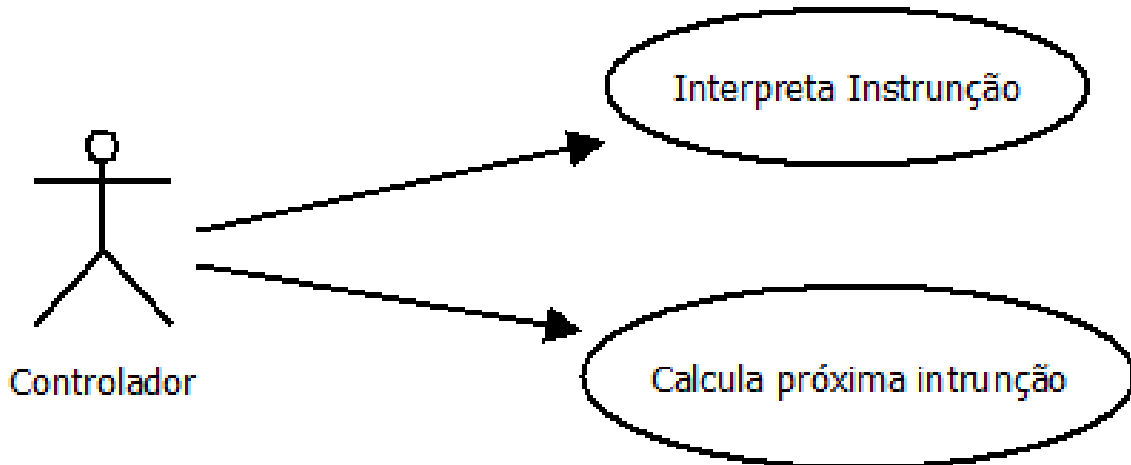
- Atender ao requisito funcional [FR15];
- Leitura do PC;



### *Pós-condições*

- Alteração do PC.

### *Diagrama de Caso de Uso*



#### *3.10.1. Fluxo Principal de Eventos*

- P1.** Acesso ao PC;
- P2.** Leitura da instrução apontada pelo PC;
- P3.** Modifica o valor do PC pro endereço da próxima instrução.

### **3.11. [UC 011] HALT**

O processador deve ser capaz de encerra o programa.

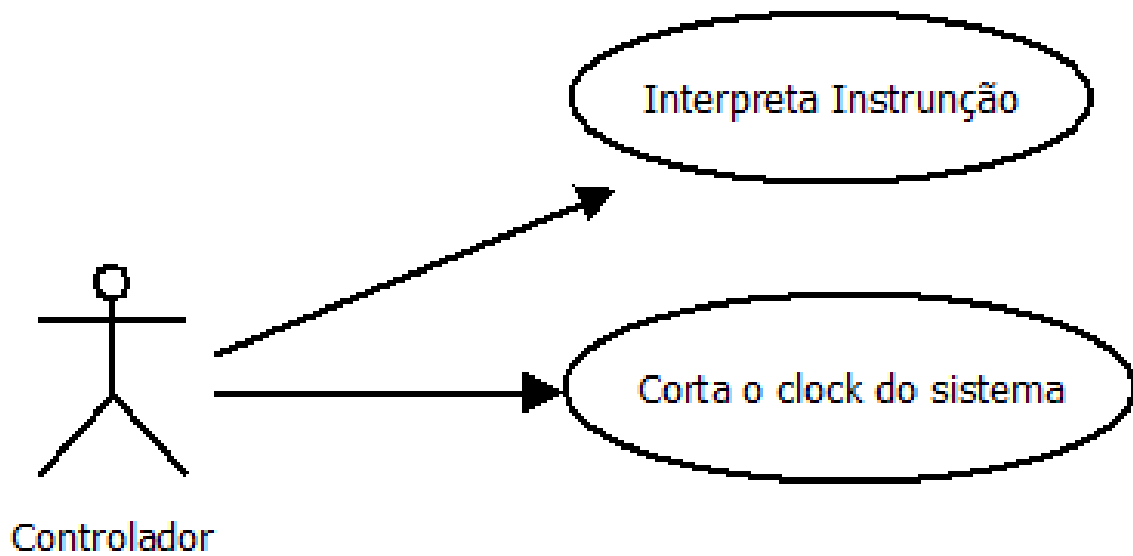
#### *Atores*

**Controlador** – Unidade que controla a execução das operações.

#### *Pré-condições*

- Atender ao requisito funcional [FR15];
- Leitura do PC;

### Diagrama de Caso de Uso



#### 3.11.1. Fluxo Principal de Eventos

- P1. Acesso ao PC;
- P2. Leitura da instrução apontada pelo PC;
- P3. Desliga o *clock* do processador.