



**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>
23/10/2014	Revisão	<ul style="list-style-type: none"><li>• jadsonfirmo;</li></ul>

## SUMÁRIO

<b>1</b>	<b>Introdução</b>	<b>3</b>
1.1	Objetivo . . . . .	3
1.2	Visão Geral do Documento . . . . .	3
1.3	Representação Simbólica . . . . .	3
1.4	Definições, Acrônimos e Abreviações . . . . .	4
<b>2</b>	<b>Atores do Sistema</b>	<b>4</b>
<b>3</b>	<b>Casos de Usos</b>	<b>4</b>
3.1	[UC 001] Execução de instruções . . . . .	4
3.1.1	Fluxo Principal de Eventos . . . . .	5
3.2	[UC 002] BRFL . . . . .	5
3.2.1	Fluxo Principal de Eventos . . . . .	6
3.3	[UC 003] Instrução LW . . . . .	6
3.3.1	Fluxo Principal de Eventos . . . . .	7
3.4	[UC 004] Instrução SW . . . . .	8
3.4.1	Fluxo Principal de Eventos . . . . .	8

## 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.

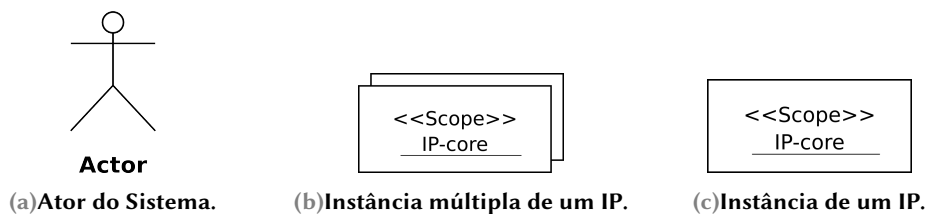
### 1.3. Representação Simbólica

A Figura ?? ilustra a simbologia utilizada para representar operações que devem ser realizadas pelo sistema. A Figura 2 ilustra as duas simbologias utilizadas 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 (botões, 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 que outrora deveriam utilizar a mesma simbologia, utiliza-se a representação ilustrada nas Figuras 2b e 2c, definida por convenção. Este elemento, em geral, está associado aos módulos do sistema, ou IP-cores de terceiros incorporados ao mesmo. Esta simbologia ainda foi dividida, tendo em vista representar instâncias únicas (Figura 2c), ou múltiplas (Figura 2b) de um determinado componente.



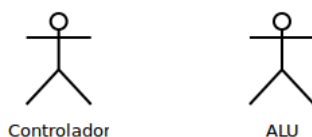
**Figura 2: Simbologia utilizada na implementação dos Casos de Uso.**

O projetista responsável por interpretar os diagramas não deve confundir-se no momento de interpretar 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
SB	Sub-fluxo
FS	Fluxo Secundário
NFR	Requisito Não Funcional
FR	Requisito Funcional
BT	Botão Direcional

## 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 por fim garantir o armazenamento dos resultados de operações no banco registradores.

*Atores*

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

**ALU** – Unidade Lógica e Aritmética.

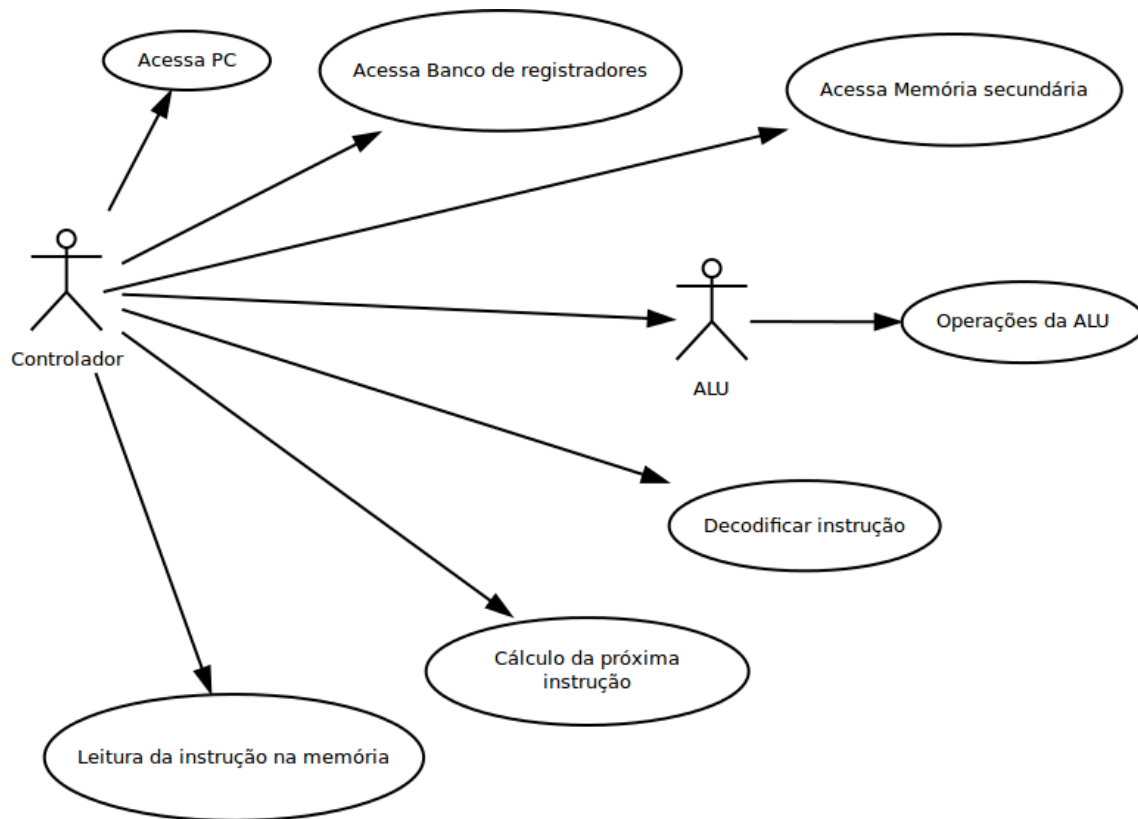
*Pré-condições*

- 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.

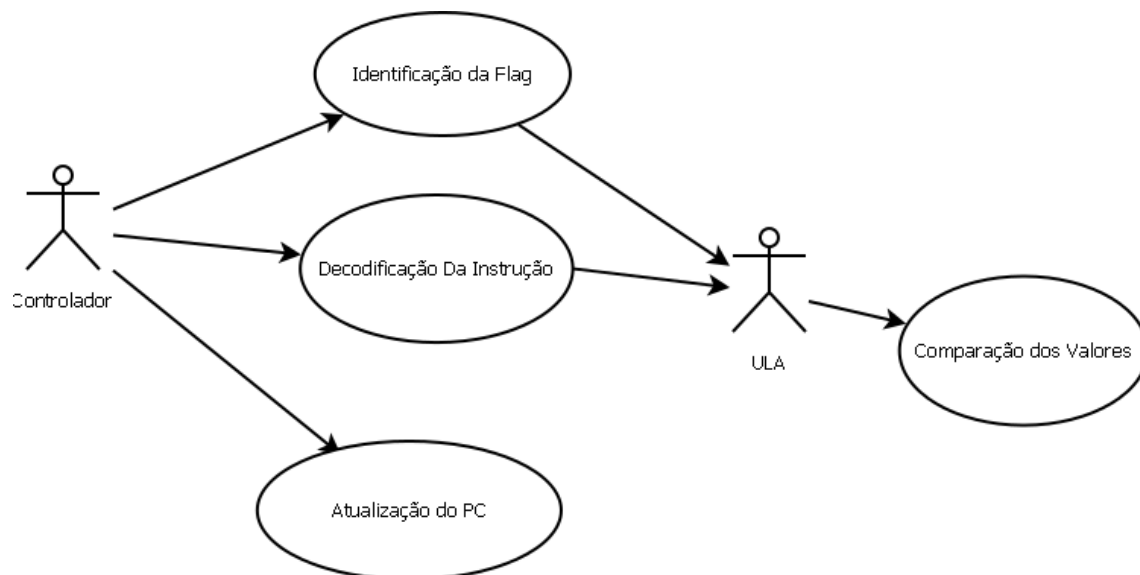
### Pré-condições

- 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 por PC;
- P3. Identificação e leitura da Flag ativa;
- P4. Executa operação Lógica;
- P5. Atualiza valor do PC;

### 3.3. [UC 003] Instrução LW

O processador é capaz de carregar dados da memória para o Banco de registradores.

### Atores

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

**ALU** – Unidade Lógica e Aritmética.

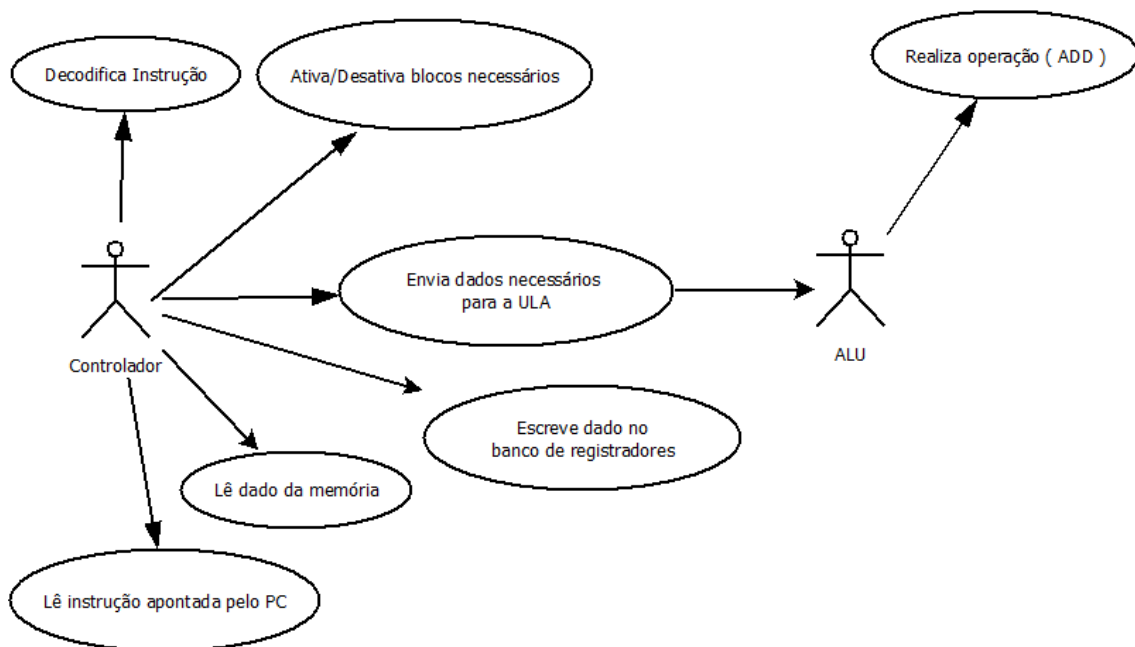
### Pré-condições

- Ler instrução da Memória de instrução;
- A instrução ter "opcode" específico para 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 extendido 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 o dado 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.

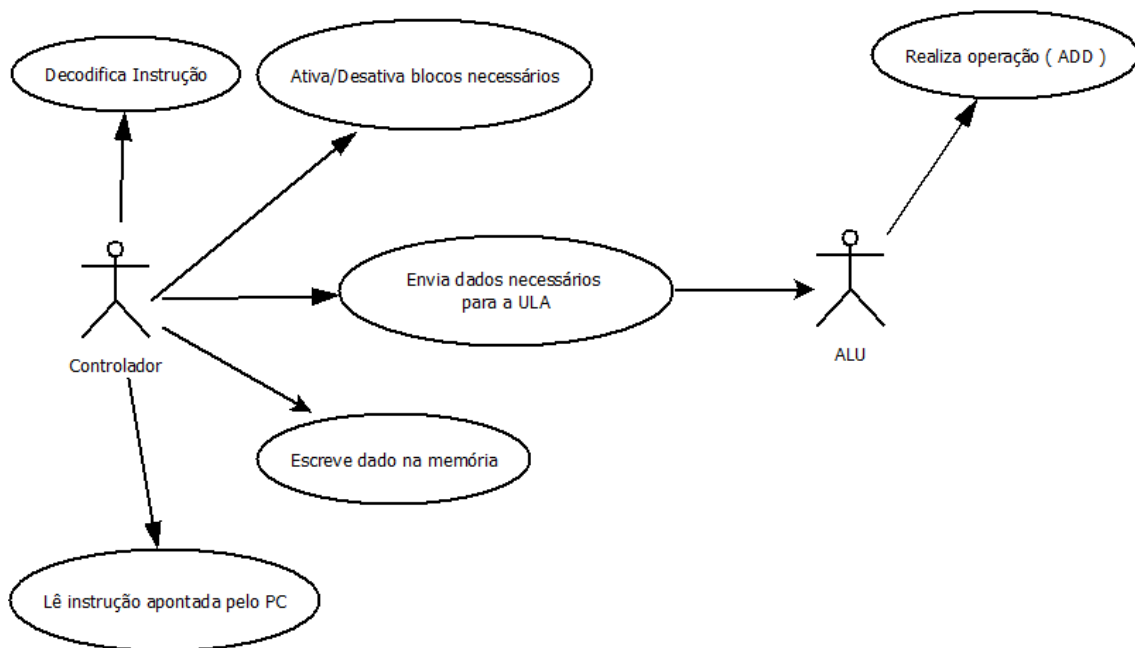
*Pré-condições*

- Ler instrução da Memória de instrução;
- A instrução ter opcode específico para 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 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.