

UNIVERSIDADE FEDERAL DE PELOTAS
CENTRO DE DESENVOLVIMENTO TECNOLÓGICO (CDTec)
CIÊNCIA DA COMPUTAÇÃO



ALYSSON NOGUEIRA RODRIGUES

RELATÓRIO DA CÉLULA DE PRODUÇÃO

PELOTAS

2016

O sistema para controlar a célula de produção foi desenvolvido em Java usando a máquina virtual Java Jamaica. O sistema se constitui de 1 controle e 8 plantas físicas, uma para cada equipamento presente na planta física. Ao todo são criados 14 threads, sendo 7 delas periódicas e 7 aperiódicas. E serão descritas ao longo do relatório.

O Controle

O controle é constituído da classe main. Na classe main estão os objetos da célula de produção(ou plantas físicas) e as funções periódicas que são os sensores. Na função main é executado as tarefas:

```
DepositoController();  
GuindasteController();  
EsteiraController();  
MesaController();  
Braco1Controller();  
PrensaController();  
Braco2Controller();
```

que servem para iniciar as threads das plantas físicas e um while que serve apenas para reiniciar o processo, quando este terminar.

A função DepositoController() controla o funcionamento da esteira de depósito, verificando se ela está “girando”(em funcionamento) e se há algum metal em cima dela. Se não está girando, inicia o funcionamento da esteira e sincroniza a posição do metal para parar a esteira quando o metal estiver em posição. A thread que se encarrega dessas funções é chamada de “depositoC” e é periódica, executando a cada 1 segundo e com um Dead Line de 10 segundos.

A função GuindasteController() controla o funcionamento da guindaste, verificando a posição do mesmo e se há algum metal em cima dela. Este controle aciona as funções de subir, descer e se mover do guindaste, assim como verificar se o metal foi solto na posição correta. A thread que se encarrega dessas funções é chamada de “guindasteC” e é periódica, executando a cada 1 segundo e com um Dead Line de 1 segundo.

A função EsteiraController() controla o funcionamento da esteira de alimentação, e possui basicamente os mesmos comandos da função de DepositoController() verificando se ela está “girando”(em funcionamento) e se há algum metal em cima dela. Se não está girando, inicia o funcionamento da esteira e sincroniza a posição do metal para parar a esteira quando o metal estiver

em posição. A thread que se encarrega dessas funções é chamada de “esteiraC” e é periódica, executando a cada 1 segundo e com um Dead Line de 10 segundos.

A função MesaController() controla o funcionamento da Mesa giratória, verifica se o metal está em posição na esteira de alimentação, se está em posição então dá os comandos para a Mesa se posicionar para receber o metal e depois os comandos necessários para a Mesa estar em posição para o braço 1 do robô poder pegar o Metal. Nesse momento a função envia o comando para a thread aguardar até o metal ser retirado. Quando isso de fato acontece, a o controle envia um sinal para a Mesa voltar a posição original. A thread que se encarrega dessas funções é chamada de “depositoC” e é periódica, executando a cada 1 segundo e com um Dead Line de 1 segundo.

A função Braco1Controller() controla o funcionamento do robô e do Braço 1. No início da execução ela verifica a posição do Metal na mesa, se caso a mesa estiver em posição o controle envia os sinais para fazer o braço 1 do robô pegar o metal de cima da mesa e o largar na prensa. Após largar o metal na prensa o controle comanda o robô para girar “x” graus até deixar o braco2 em posição. A thread que se encarrega dessas funções é chamada de “Braco1C” e é periódica, executando a cada 1 segundo e com um Dead Line de 1 segundo.

A função PrensaController() controla o funcionamento da planta física da Prensa. O controle da prensa é bem simplificado, começando com a verificação se existe um metal na prensa e se este já foi forjado, do contrario, envia o comando para a prensa fechar a tampa e forjar o metal, ao término envia o comando para abrir a prensa e ficar no aguardo até que o Braço 2 do robô retire o metal. A thread que se encarrega dessas funções é chamada de “PrensaC” e é periódica, executando a cada 1 segundo e com um Dead Line de 1 segundo.

Por fim, a função Braco2Controller() controla o funcionamento do robô e do Braço 2. No início da execução, o controle do braço 2 verifica se o metal já foi forjado na prensa, caso esta condição seja verdadeira, envia a ordem para o Braço 2 retirar o metal forjado da prensa, e os outros comandos necessários para posicionar o Braço 2 na esteira de depósito e soltar o metal. A thread que se encarrega dessas funções é chamada de “Braco2C” e é periódica, executando a cada 1 segundo e com um Dead Line de 1 segundo.

Esteira de Depósito

A Planta física da esteira de depósito esta implementada na classe Deposito que extends RealtimeThread e é uma thread aperiódica. Tem 3 funções principais além do construtor e 3 variáveis globais, a variável metal identifica que há um metal na planta física se metal diferente de NULL, a variável booleana positionMetal identifica se o metal está na posição final da execução na planta física e a variável girando identifica se a planta física está ou não em execução. A função run() simula o percurso do metal ao longo da esteira. E as

outras duas funções são os get's e set's da variável metal.

Guindaste

A Planta física do guindaste esta implementada na classe Guindaste que extends RealtimeThread e é uma thread aperiódica. Tem 6 funções principais além do construtor e 5 variáveis globais, a variável metal identifica que há um metal na planta física se metal diferente de NULL, as variáveis booleanas positionMetal, down, up e moving identificam em qual posição está o guindaste na planta física. A função down() simula a descida do guindaste, up() simula a subida, move() movimenta o guindaste em direção a esteira de alimentação e a função originalPosition faz com que o guindaste retorne a posição original. E as outras duas funções são os get's e set's da variável metal.

Esteira de Alimentação

A Planta física da esteira de depósito esta implementada na classe Esteira que extends RealtimeThread e é uma thread aperiódica. Como é uma esteira, sua implementação foi bem parecida. Tem 3 funções principais além do construtor e 3 variáveis globais, a variável metal identifica que há um metal na planta física se metal diferente de NULL, a variável booleana positionMetal identifica se o metal está na posição final da execução na planta física e a variável girando identifica se a planta física está ou não em execução. A função run() simula o percurso do metal ao longo da esteira de alimentação. E as outras duas funções são os get's e set's da variável metal.

Mesa Giratória

A Planta física da Mesa Giratória esta implementada na classe MesaGiratoria que extends RealtimeThread e é uma thread aperiódica. Tem 6 funções principais além do construtor e 4 variáveis globais, a variável metal identifica que há um metal na planta física se metal diferente de NULL, as variáveis booleanas positionMetal, down e gira90D identificam em qual posição está a Mesa Giratória na planta física. A função down() simula a descida da mesa giratória, gira90D() simula o movimento da planta física em 90 graus para a direita, a função waiting() faz a mesa giratória entrar em processo de espera enquanto o Braço 1 não retirar o metal de cima, e a função originalPosition faz com que a mesa giratória retorne a posição original. E as outras duas funções são os get's e set's da variável metal.

Robo

A Planta física do Robo está neste trabalho implementado em 3 classes, Robo que extends RealtimeThread, Braco1 e Braco2 que extends Robo. Na classe Robo existem apenas comandos para comandar o motor do robo e é formada por 3 variáveis globais e 5 funções além do construtor. a variável metal identifica que há um metal na planta física se metal diferente de NULL, a variável booleana positionMetal identifica se o metal está na posição final da execução na planta física e a variável position armazena em que posição atualmente está o robo. As funções Position45(), Position90(), Position180() servem para

posicionar o motor do robo. E as outras duas funções são os tradicionais get's e set's da variável metal.

Braço 1

O Braço 1, que é parte da planta física do robo e que possui as implementações das funções do Braço 1 esta implementada na classe Braco1 que extends Robo. Tem 4 funções principais além do construtor e 2 variáveis globais, a variável metal identifica que há um metal na planta física se metal diferente de NULL e a variável booleana positionMetal identifica se o metal está na posição final da execução. A função pegaMetal simula o tempo que um braço levaria para pegar um metal na mesa giratória, gira180() utiliza a função Position180() da classe mãe robo para se posicionar na frente da prensa. A função metalnaPrensa(), simula a colocação de um metal na prensa. A função posicionaBraco2(), utiliza as funções da classe mãe para já deixar o robo com o braço 2 posicionado para pegar o metal que sairá da prensa. E as outras duas funções são os get's e set's da variável metal.

Braço 2

O Braço 2, que também é parte da planta física do robo e que possui as implementações das funções do Braço 2 esta implementada na classe Braco2 que extends Robo. Tem 4 funções principais além do construtor e 2 variáveis globais, a variável metal identifica que há um metal na planta física se metal diferente de NULL e a variável booleana positionMetal identifica se o metal está na posição final da execução. A função setMetal neste caso, não faz só a passagem do metal de uma classe para a outra, mas também simula o braco2 do robo a pegar o metal, move180() utiliza a função Position180() da classe mãe robo para se posicionar na frente da esteira de depósito. A função posicionaBraco1(), utiliza as funções da classe mãe para já deixar o robo com o braço 1 posicionado para pegar o metal que chegará a mesa giratória. E a outra função é o tradicional get da variável metal.

Prensa

A Planta física da Mesa Giratória esta implementada na classe MesaGiratoria que extends RealtimeThread e é uma thread aperiódica. Tem 5 funções principais além do construtor e 3 variáveis globais, a variável metal identifica que há um metal na planta física se metal diferente de NULL, a variável booleana positionMetal identifica se o metal está na posição final da execução e a variavel open identifica se a prensa está aberta(true) ou fechada(false). A função AbrePrensa() simula a abertura da prensa, FechaPrensa() simula o fechamento da planta física, e a função ForjarMetal() simula o tempo em que um metal seria forjado. E as outras duas funções são os get's e set's da variável metal.

Diagrama de Classe

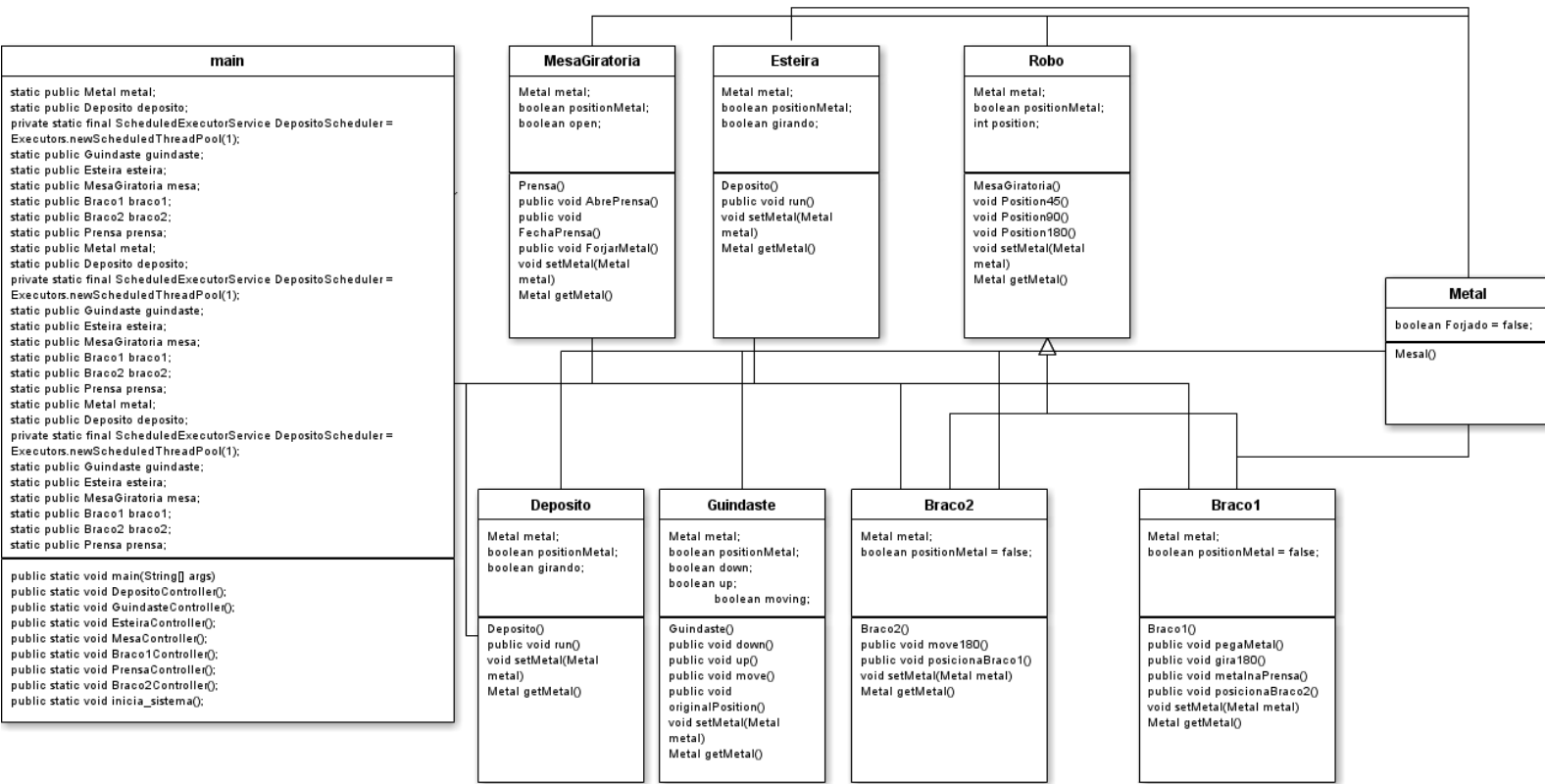


Diagrama de Sequência

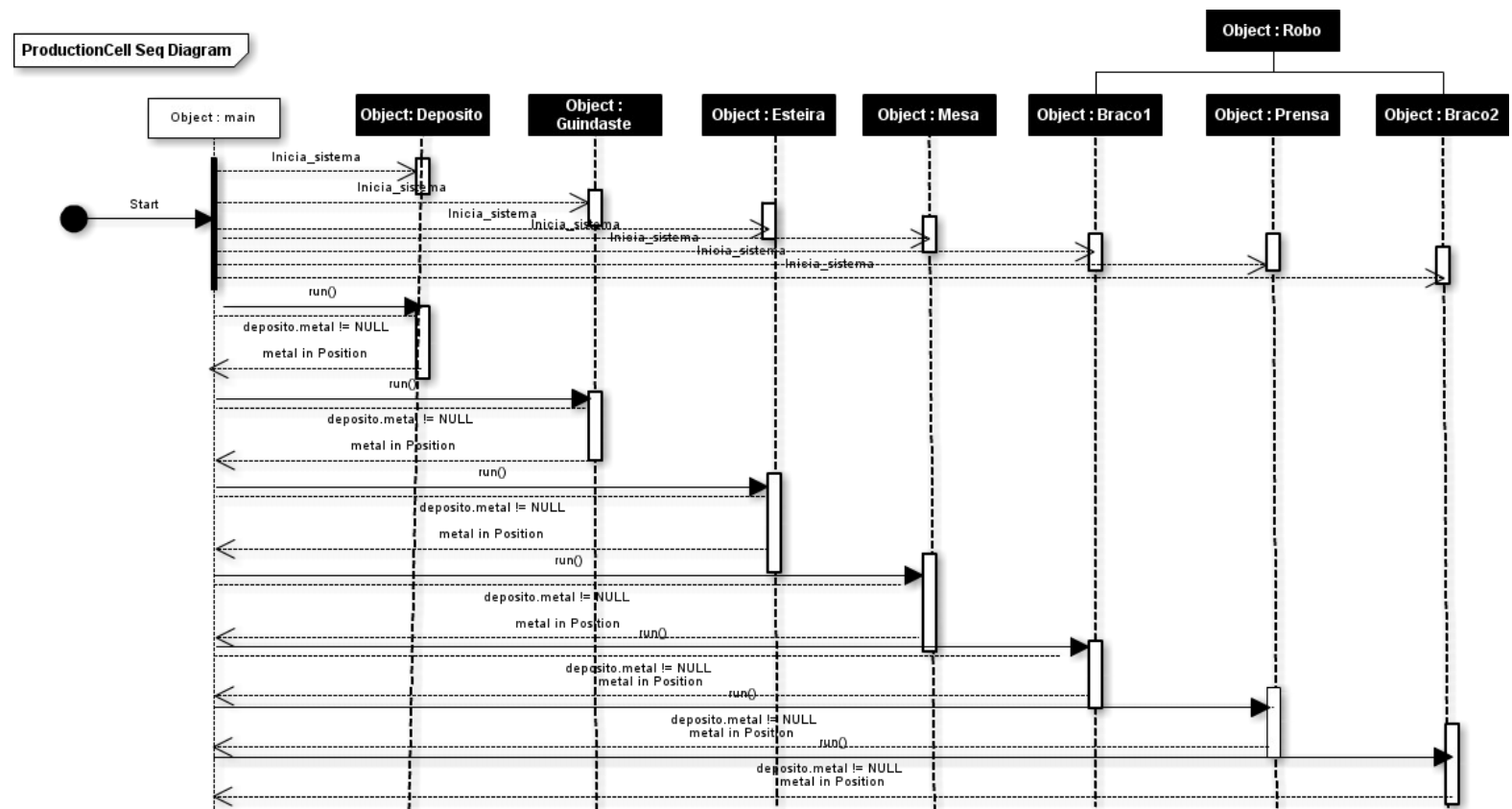


Tabela com tempo das threads

| Tarefa | Tempo | Duração | Dead Line | Prioridade |
|------------|------------|------------|-------------|------------|
| depositoC | 1 segundo | 6 segundos | 10 segundos | 1 |
| deposito | aperiódica | 5 segundos | | 1 |
| guindasteC | 1 segundo | 7 segundos | 1 segundo | 1 |
| guindaste | aperiódica | 7 segundos | | 1 |
| esteiraC | 1 segundo | 5 segundos | 10 segundos | 1 |
| esteira | aperiódica | 5 segundos | | 1 |
| mesaC | 1 segundo | 6 segundos | 1 segundo | 1 |
| mesa | aperiódica | 6 segundos | | 1 |
| Braco1C | 1 segundo | 6 segundos | 1 segundo | 1 |
| braco1 | aperiódica | 6 segundos | | 1 |
| PrensaC | 1 segundo | 8 segundos | 1 segundo | 1 |
| Prensa | aperiódica | 8 segundos | | 1 |
| Braco2C | 1 segundo | 6 segundos | 1 segundo | 1 |
| braco2 | aperiódica | 6 segundos | | 1 |