



INSTITUTO DE  
INFORMÁTICA  
UFG



ENGENHARIA  
DE SOFTWARE

**Atividade: Dia de Trabalho**  
Tarefa: Programação em Pares

**Versão 0.1**

Atividade: Dia de Trabalho	Versão: 0.1
Tarefa: Programação em Pares	Data: 18/12/2012
LMP-P03.2.4	

## Histórico da Revisão

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>
03/12/2012	0.1	Elaboração Inicial	Emerson José Porfírio
18/12/2012	0.1	Diagrama da tarefa	Emerson José Porfírio

Atividade: Dia de Trabalho	Versão: 0.1
Tarefa: Programação em Pares	Data: 18/12/2012
LMP-P03.2.4	

## Sumário

1.	Objetivos	4
1.1	Escopo	4
2.	Introdução	4
3.	Tarefa: Programação em Pares	4
4.	Metas	4
5.	<i>Input</i>	5
5.1	Pré-condições	5
5.2	Entradas	5
6.	<i>Output</i>	5
6.1	Pós-condições	5
6.2	Saídas	5
7.	Diagrama da tarefa: Programação em Pares	6
8.	Passos	6
9.	Template	6
10.	Papéis	7
11.	Padrões Relacionados	7
12.	Riscos	7
13.	Referências	8

Atividade: Dia de Trabalho	Versão: 0.1
Tarefa: Programação em Pares	Data: 18/12/2012
LMP-P03.2.4	

## **Tarefa: Programação em Pares**

### **1. Objetivo**

Apresentar e documentar a tarefa Programação em Pares da atividade Dia de Trabalho (Processo Produzir) que faz parte do LMP – Logiciel Mobile Process a ser utilizado pelo Grupo de Estudo Logiciel como trabalho prático para as disciplinas de Integração I e de Desenvolvimento de Software para Dispositivos Móveis do curso de Bacharelado em Engenharia de Software do INF - UFG.

#### **1.1 Escopo**

Tarefa 4 da atividade Dia de Trabalho do LMP – Logiciel Mobile Process (Processo Produzir).

### **2. Introdução**

O processo de desenvolvimento LMP – Logiciel Mobile Process abrange atividades acadêmicas referentes aos processos de engenharia de software do INF-UFG. Este oferecerá o apoio ao processo de desenvolvimento para dispositivos móveis, permitindo que o mesmo seja realizado de acordo com o planejamento de tempo e de recursos e com os requisitos funcionais e de qualidade definidos para os projetos propostos.

O modelo foi baseado no Processo MobileD e no RUP, além de seguir as orientações do Guia do MPS.Br 2011 (nível F).

### **3. Tarefa: Programação em Pares**

O objetivo da programação em pares é melhorar a comunicação, melhorar o processo de fidelidade e disseminação do conhecimento dentro da equipe, além de garantir a qualidade do código.

### **4. Metas**

Os objetivos de Programação em Pares são:

1. Evitar a geração de defeitos através do pensamento e discussões em par;
2. Encontrar defeitos tão cedo quanto possível através de uma revisão constante;
3. Disseminação de conhecimento do software e tecnologias relacionadas de um desenvolvedor para outro (gestão do conhecimento eficiente)
4. Reforçar a adesão dos desenvolvedores sobre as convenções acordadas para o processo (por exemplo, padrões de codificação de refatoração); e

Atividade: Dia de Trabalho	Versão: 0.1
Tarefa: Programação em Pares	Data: 18/12/2012
LMP-P03.2.4	

5. Fornecer aos desenvolvedores uma maneira eficiente de comunicação.

## **5. Inputs**

### **5.1 Pré-condições**

1. **Um par está disponível.** Naturalmente, para realizar a programação em par, a equipe do projeto tem de disponibilizar pelo menos dois desenvolvedores presentes.
2. **O Ambiente é adequado para programação em par.** O ambiente de trabalho tem de ser adequado para a programação em pares, isto é, ambos os desenvolvedores devem poder visualizar a tela do monitor sem precisarem trocar de lugar na inversão dos papéis Motorista/Navegador apenas passando o mouse e teclado um para o outro.
3. **As tarefas são definidas.** Pelo menos uma tarefa tem de ser definida (tarefas definidas no Dia de Planejamento), de modo que os desenvolvedores têm uma definição sobre o que cada um deve fazer.

### **5.2 Entradas**

1. Definição da Tarefa (cartão de tarefa). O trabalho é baseado na execução da tarefa definida o cartão de tarefas.
2. Manuais, especificações de API e material de apoio. O navegador pode procurar ajuda e informações adicionais no material de apoio, quando necessário.

## **6. Outputs**

### **6.1 Pós-condição**

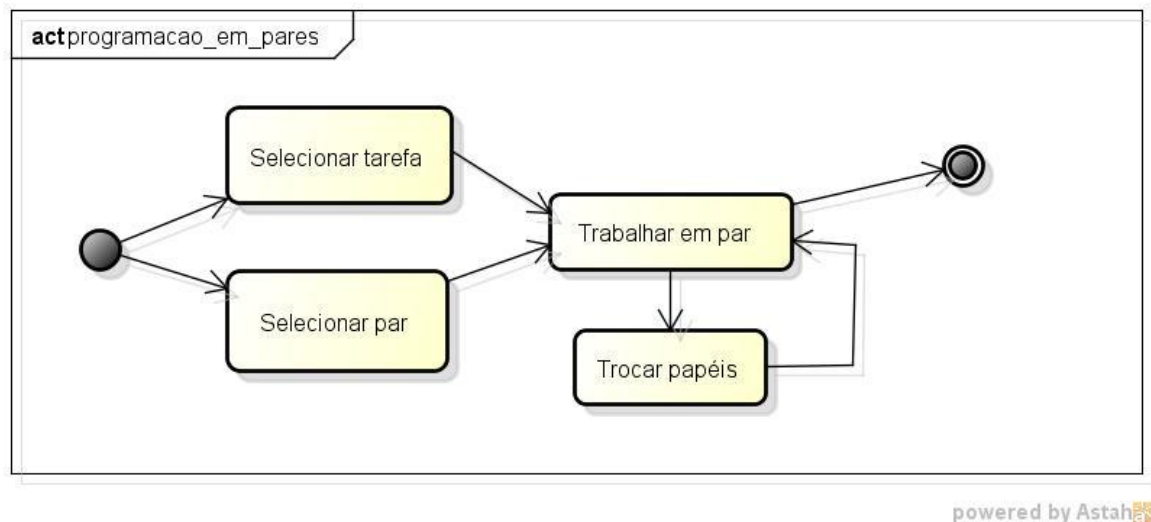
1. Uma sessão de programação em par pode ser encerrada sem qualquer critério explícito terminar. A Programação par em um projeto pode terminar, quando não há tarefas mais adequadas para serem programadas, ou se o projeto mudou em uma fase onde a implementação não é mais necessária.

### **6.2 Saída**

1. **Código fonte.** O código fonte (novo, reformulado ou refeito) produzido pelo par.

Atividade: Dia de Trabalho	Versão: 0.1
Tarefa: Programação em Pares	Data: 18/12/2012
LMP-P03.2.4	

## 7. Diagrama da tarefa: Programação em Pares



## 8. Passos

As etapas individuais de Programação em Pares são:

1. Selecione a tarefa a ser feita a partir dos cartões com as tarefas definidas para a iteração atual;
2. Selecione um par para executar a tarefa;
3. Trabalhar com o par. A maneira como esta etapa é realizada depende do tipo de atividade em par que está sendo feita, ou seja, codificação em par, refatoração em par ou retrabalho. Além disso, ambos os parceiros no par têm responsabilidades diferentes (ver Funções).
4. Alterar os papéis com a frequência suficiente para garantir sentimentos de igualdade e parceria, ou mesmo a carga de digitação, preservar um senso de controle em ambos os parceiros e impedir para que o trabalho se torne monótono.

## 9. Templates

1. **Padrão de codificação.** Um conjunto definido de padrões de codificação a serem seguidos, não só assegura a produção de código uniforme com diferentes parceiros e pares, mas também reduz o atrito entre os parceiros que podem de outra forma ser causados pela preferências de estilo e costumes diferentes de codificação.

Atividade: Dia de Trabalho	Versão: 0.1
Tarefa: Programação em Pares	Data: 18/12/2012
LMP-P03.2.4	

## 10. Papéis

As seguintes funções podem ser identificadas na execução de Programação em Pares:

1. O **Motorista** é a pessoa usar o teclado e o mouse durante uma sessão de programação em par. A principal responsabilidade do motorista é construir o código do programa ou de teste;
2. O **Navegador** é a pessoa sentada em frente ao computador ao lado o motorista. As responsabilidades do navegador incluem a observação e revisão do trabalho do Motorista constantemente para detectar “defeitos táticos”, como erros de digitação e de sintaxe, avaliar se a tarefa está sendo levada na direção certa, se o motorista tem uma visão clara sobre o que ele está fazendo, e se a peça em trabalho se encaixa no todo do software ou sistema em desenvolvimento, isto é, uma detecção estratégica dos defeitos. Muitas vezes o navegador pode fará isso, pedindo ao motorista para explicar o que está e por que ele está fazendo isso. Além de monitorar o trabalho do motorista, as responsabilidades do navegador incluem a facilitação do trabalho em encontrar as informações necessárias, por exemplo, documentos relacionados, e às vezes até mesmo explorar soluções alternativas para a tarefa, a fim de avaliar se o trabalho poderia ser feito de alguma outra forma. O navegador também contribui para o planejamento dos trabalhos futuros, fazendo anotações sobre as próximas tarefas por fazer.

## 11. Padrões Relacionados

Outros padrões que fazem parte deste ou são associados com a atividade são identificados abaixo:

- **Dia de Trabalho.** Programação em par faz parte do padrão Dia de Trabalho.
- **Dia de Planejamento.** Programação em par está associada ao padrão Dia de Planejamento, porque suas tarefas realizadas são definidas em Dia Planejamento.
- **Desenvolvimento Dirigido por Testes.** Programação par está associada ao padrão TDD, porque, além do código de produção o código de testes pode ser escrito utilizando a programação em par.
- **Refatoração.** Programação em par está associada ao padrão Refatoração, pois refatorações de código podem ser feitas durante a construção.

## 12. Riscos

Os possíveis riscos que podem resultar de Programação em Pares, bem como as soluções incluindo ações preventivas para evitá-los e medidas a tomar para minimizar seus efeitos são discutidos aqui:

- **O Par funciona em desacordo.** Como a programação em par envolve estreita colaboração e levando as opiniões da outra pessoa em consideração, discordâncias podem surgir. Solução: A questão essencial para os membros do par é ter respeito um pelo outro e serem compromissados com a tarefa. Se por exemplo durante um curto período de tempo, e, em seguida, o par pode decidir qual alternativa será a melhor entre as abordagens. Outra possibilidade é a que o motorista

Atividade: Dia de Trabalho	Versão: 0.1
Tarefa: Programação em Pares	Data: 18/12/2012
LMP-P03.2.4	

pode mostrar sua solução ao navegador por um tempo predefinido, e em seguida, o navegador pode decidir se aceita ou não sua sugestão. Deste jeito ambos os membros do par permanecem no controle das decisões tomadas e podem explicar seus pontos de vista um para o outro.

- **O outro sócio quer ser o motorista / navegador o tempo todo.** O outro sócio se sente mais confortável estar no papel de todos a todo o tempo, e não sugere ou concorda com as mudanças de papéis. Solução: Decidir sobre períodos fixos entre as mudanças de papel pode resolver o problema de imediato, mas, a fim de ajustar a atitude do “parceiro problema”, os conceitos básicos de programação em pares devem ser repassados pelo Gerente do Projeto explicando a importância das mudanças e a igualdade dos papéis. Também a rotação de pares pode aliviar a problema, porque impede que as pessoas fiquem muito acostumadas aos mesmos tipos de padrões de comportamento.
- **As pessoas tem medo de não conseguir o crédito para seu trabalho.** A motivação para a programação em pares, como a partilha de código ou a doação da própria experiência para beneficiar os outros pode ser baixa, já que na programação em pares é mais difícil diferenciar os esforços e conquistas de cada membro da equipe individualmente. Solução: estruturas de recompensa organizacional podem ser modificadas para recompensar as equipes em vez dos indivíduos. Se uma avaliação individual é necessária, pode ser feita usando avaliações de pares dentro a equipe.

### 13. Referências

- <http://agile.vtt.fi/mobiled.html>