

# **Pair Programming - Programação em Par**

## **O que é?**

A Programação em Par é uma técnica ágil de desenvolvimento de software originada na Extreme Programming (XP), onde dois programadores se unem em um computador. As duas pessoas trabalham juntas para projetar, codificar e testar o desenvolvimento de uma classe, de um método ou de um trecho de código.

Existem alguns estilos e Programação em Par, neste material iremos apresentar três estilos bastante utilizados pelas empresas. Dependendo do estilo, o ideal é que os dois programadores sejam igualmente habilidosos, porém, dependendo do estilo não é necessária a igualdade de habilidade entre os programadores, então, sim é possível praticar a Programação em Par com dois programadores com níveis de habilidade diferente e aproveitar a oportunidade para aumentar os conhecimentos do programador menos experiente.

A Programação em Par é um esforço colaborativo que envolve muita comunicação. A ideia é fazer com que os dois programadores se comuniquem, discutam abordagens, planejem a construção do código e resolvam problemas que podem ser de solução difíceis por um único desenvolvedor.

No entanto, o sucesso da Programação em Par depende do perfil dos programadores envolvidos, eles têm que ser colaborativos para que o desenvolvimento flua e tenha produtividade.

**“Programar em Par exige que as pessoas envolvidas sejam receptivas, compreensivas umas com as outras, engajadas e, sobretudo, humildes. É necessário aceitar que somos falíveis para que possamos programar em Par.”**

## **Como executar?**

Na Programação em Par dois programadores trabalham em uma estação de trabalho com um teclado e um mouse. A princípio enquanto um dos programadores está codificando, o outro acompanha seu trabalho para garantir a qualidade do código gerado, observando se os padrões de projeto estão sendo seguidos, se a declaração das variáveis e métodos estão seguindo a mesma nomenclatura padrão, se sintaxe utilizada está correta, se método que está sendo implementado não pode ser simplificado, inclusive, já observando as possibilidades de refatoração no mesmo, se os testes unitários ou de integração estão sendo seguidos à risca e de maneira geral.

O processo começa com os programadores recebendo uma tarefa bem definida. Eles acordam as etapas que serão cumpridas e estabelecem algumas metas. Eles discutem sobre as várias

técnicas e métodos de como cumprir cada etapa, geralmente o código resultante é provido de maior qualidade do que um código gerado por um programador trabalhando sozinho.

Os dois programadores se revezam na codificação (driver) e acompanhamento/análise (navegador), assim um verifica o trabalho do outro. Essa rotação de funções regularmente ajuda a manter os programadores em alerta e engajados.

Dependendo de como as duplas são coordenadas, programadores juniores e seniores podem trabalhar juntos, permitindo que programadores seniores compartilhem seus conhecimentos e hábitos de trabalho. Ele também ajuda os novos membros da equipe a se atualizarem em um projeto.

## **Benefícios da Programação em Par**

A Programação em Par inclui as seguintes vantagens:

- Redução de erros de codificação: Um programador (navegador) analisa o código construído pelo outro programador (driver), o que pode ajudar a reduzir erros e melhorar a qualidade do código. Com isso a necessidade de refatoração também diminui.
- Redução de erros e problemas encontrados nos testes unitários e testes de integração.
- O conhecimento é difundido entre os pares. Programadores juniores podem adquirir mais habilidades de programadores seniores. E aqueles que não estão familiarizados com um processo podem ser emparelhados com alguém que sabe mais sobre o processo.
- Há um aumento significativo na atenção depositada na codificação do software, pois o método mitiga fatores de desvio de atenção (e-mails, WhatsApp etc.).
- O rigor é maior no objetivo de seguir os padrões definidos para aquele projeto;
- Os programadores somam conhecimento para produzir códigos mais criativos e eficazes.

## **Desafios da Programação em Par**

Nem tudo é um mar de rosas, a Programação em Par também tem seus obstáculos:

- Aumento do custo de um projeto: A Programação em Par reduz o retrabalho, aumenta a produtividade e qualidade do código, mas ainda assim, estima-se que a soma da

produtividade de dois programadores trabalhando separados é maior que a produtividade de dois programadores trabalhando juntos.

- Quebra da dupla antes do encerramento da atividade: Diversos são os fatores que contribuem para a quebra da Programação em Par antes do término da atividade, por exemplo: faltas, alocação de um dos programadores em outra atividade, férias etc.
- Antes só do que mal acompanhado: Algumas pessoas têm dificuldade de trabalhar em duplas e quando essas pessoas são alocadas para a Programação em Par normalmente sua produtividade não é boa.
- Tempos diferentes: Trabalhando várias horas juntos é normal os programadores necessitem de pausas, porém cada pessoa tem necessidade de pausas em tempos diferentes, isso pode comprometer a produtividade da dupla.
- Envolvimento diferente. Quando o envolvimento dos programadores não é semelhante, faz com a participação na realização da tarefa seja muito diferente, o que compromete a produtividade da dupla.

## Estilos e abordagens de Programação em Par

A seguir são apresentados três estilos da Programação em Par, existem outros estilos e algumas variações desses estilos, porém estes são os mais utilizados.

### Driver / Navegador.

A abordagem driver/navegador também conhecida como piloto/co-piloto ou



motorista/navegador é um estilo de Programação em Par onde, um programador (driver) conduz a construção de uma parte do código e explica ao outro programador (navegador) o que está sendo construído. Por outro lado, o navegador fica responsável pela revisão do código, mitigação de erros e analisar a implementação de melhorias no código. O driver e o navegador trocam de função com

frequência. Este estilo funciona bem para um programador novato trabalhando junto com um

programador experiente, neste caso, quando o programador experiente assume a função de navegador é possível realizar mais intensamente a transmissão de conhecimentos para o programador novato.

## Não Estruturado.

A maioria dos relacionamentos de programação em Par se enquadra no estilo não estruturado,



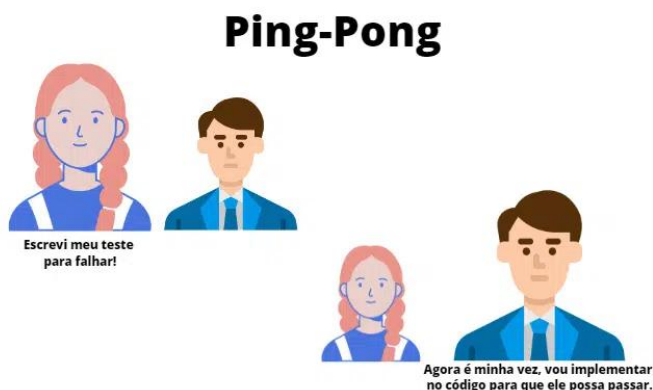
onde dois programadores trabalham e se ajudam simultaneamente. Este estilo não apresenta regra de papéis (quem codifica / quem analisa) e tão pouco estabelece uma ordem de atuação dos programadores. Ambos os programadores devem ter níveis de habilidade correspondentes. Este estilo é bastante utilizado para que os programadores tenham mais liberdade em trocar ideias sobre o que está sendo feito.

O estilo não estruturado não é o mais indicado para projeto muito longo.

Essa abordagem funciona bem quando os programadores têm que encontrar a melhor solução para um projeto.

## Pingue-Pongue.

O estilo de Programação em Par ping-pong é o mais diferente entre eles, geralmente utilizado



para desenvolver tarefas que são orientadas a testes — também conhecido como TDD (Test Driven Development).

Como o próprio nome já diz, “ping-pong” pode ser considerado “pulando de um programador para outro” isso quer dizer que um programador desenvolve o teste com o objetivo de evidenciar as falhas e o outro programador

implementa o código para que o teste que falhou a primeira vez passe. Os programadores alternam os papéis de evidenciar falha e fazer o código passar.

Observação: o principal objetivo do TDD que significa desenvolvimento guiado por testes é seguir um ciclo entre falhas e sucesso até que os programadores consigam atingir o seu objetivo final garantindo então uma qualidade melhor do software por ter sido desenvolvido com base em teste.

## Práticas recomendadas para programação em Par

Algumas práticas recomendadas a serem seguidas ao implementar a Programação em Par incluem:

- Comunicação consistente. Se a comunicação não for intensa entre a dupla, provavelmente os programadores não estão colaborando um com o outro.
- Troque de função de forma consistente. Isso promove um maior compartilhamento de habilidades entre os programadores e os mantém engajados.
- Emparelhe com cuidado. Certifique-se de que os dois programadores serão capazes de trabalhar bem juntos, sem “soluções”. Caso contrário, isso criará um ambiente de trabalho ruim.
- Use um ambiente de desenvolvimento familiar. Ambos os programadores devem estar familiarizados com o ambiente em que estão trabalhando.
- Peça esclarecimentos quando necessário. Particularmente quando um novato trabalha com um especialista, o novato deve aproveitar qualquer oportunidade para aprender.
- Faça pausas quando necessário. Trabalhe em um ritmo que se adapte a ambos os programadores.

## Considerações Finais

Esta introdução a Programação em Par é um compilado dos sites abaixo.

<https://www.techtarget.com/searchsoftwarequality/definition/Pair-programming#:~:text=Pair%20programming%20is%20an%20Agile,code%20and%20test%20use%20stories>.

<https://www.devmedia.com.br/implementando-pair-programming-em-sua-equipe/1694>

[Pair programming: como implementar a programação em par | Trybe \(betrybe.com\)](#)

