

# Metodologia RAD

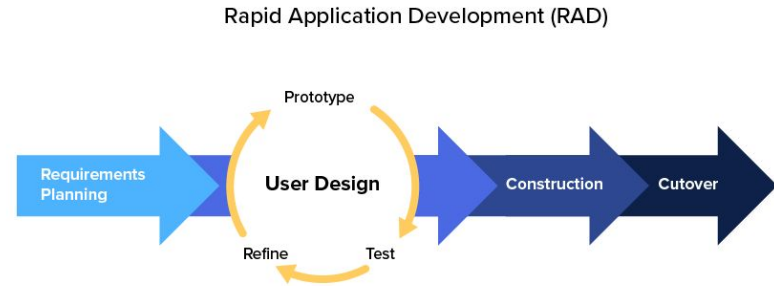
Grupo 3 - Receitalista

# Rapid Application Development (RAD)

Criada na década de 1970, a metodologia RAD possui foco no desenvolvimento rápido de aplicativos por meio de iterações e aprovações frequentes com comentários contínuos dos clientes. O RAD enfatiza a usabilidade do software, os comentários dos usuários, a entrega rápida em vez do planejamento de longo prazo e um único conjunto inicial de requisitos para a criação de itens como aplicativos personalizados.

# Rapid Application Development (RAD)

RAD se baseia no modelo de espiral, em que um ou mais modelos de desenvolvimento são usados para trabalhar em um projeto.



# Etapas do método RAD

1. Definir requisitos de projeto
2. Criar protótipos
3. Construção, teste e incorporação de comentários
4. Finalizar e implementar

# Definir requisitos de projeto

Todos os envolvidos (desenvolvedores, clientes e usuários) definem o escopo e os requisitos do projeto de forma geral, como metas e expectativas. Seja por meio de um lançamento, os usuários irão propor sua visão, e os clientes e os desenvolvedores de TI vão ajudar a finalizar todos esses requisitos.

Vale ressaltar que um dos benefícios do método RAD é que, embora tenham decidido sobre os requisitos, é possível facilmente mudar a qualquer momento durante o ciclo de desenvolvimento.

# Criar protótipos

Nos estágios iniciais da prototipagem, os desenvolvedores têm a oportunidade de criar soluções alternativas que geram um produto funcional sem sacrificar a qualidade. Assim, o objetivo é produzir **rapidamente** um modelo de trabalho para apresentar aos envolvidos, tendo como a experiência do usuário, o teste e os comentários, um papel crucial.

Comentários consistentes ajudam o trabalho em equipe em um sistema ativo, em vez de um design abstrato. Trabalhando de forma consistente ao lidar com soluções temporárias e bugs, é possível se ajustar para garantir que os requisitos sejam atendidos e em um modelo funcional.

Os erros são encontrados e depurados no início do processo, ajudando a manter a linha do tempo dos participantes e garantindo que o projeto seja mais bem estruturado para futuras adições.

# Construção, teste e incorporação de comentários

Etapa onde o protótipo funcional se transforma em um modelo funcional. Os desenvolvedores coletam comentários dos usuários e criam o produto. Com a codificação do aplicativo, o teste do sistema e a integração da unidade, o protótipo e os sistemas beta são convertidos em um modelo funcional. Uma vez que as equipes estão usando pouco código e ferramentas do método RAD, é possível lidar com qualquer mudança rapidamente.

O software é exaustivamente testado, e os usuários podem fornecer alterações ou novas ideias à medida que os problemas são descobertos. Não deve haver muitos erros, pois o benefício do método RAD é que você pode ver a maioria dos erros em tempo real na fase de prototipagem e, em seguida, fazer ajustes imediatamente. Assim que os participantes estiverem satisfeitos com seu produto, pode-se concluí-lo.

# Finalizar e implementar

O estágio final é fazer uma versão otimizada do produto final implementada em um ambiente de produção, em que seja estável e fácil de manter para garantir a longevidade. Recursos, funções e estética são finalizados e depois de passar para o sistema de produção, os usuários podem fazer testes ou treinamentos em grande escala. O time ainda gasta um longo tempo *debugando* o sistema e rodando as atualizações finais antes de ir para o sistema em tempo real.



# Vantagens x Desvantagens (RAD)

## Vantagens

- Requisitos podem ser mudados a qualquer momento
- Incentiva e prioriza o feedback do cliente
- *Reviews* são rápidas
- Tempo de *deploy* reduzido
- Mais produtividade com menos pessoas
- A integração não é um problema, pois integra desde o início do projeto

## Desvantagens

- Precisa de uma forte colaboração da equipe
- Não é viável para times muito grandes
- Time altamente qualificado
- Precisa dos requisitos do usuário durante todo o ciclo de vida do produto
- Adequado apenas para projetos que têm um pequeno tempo de desenvolvimento

# RAD x Ágeis

## RAD

- Cria o mais rápido possível aplicações funcionais
- Perfeito para projetos que exigem menor tempo para serem concluídos
- Pode se ajustar às mudanças de requisitos
- Envolve os clientes e usuários em todo o ciclo de desenvolvimento

## Ágeis

- Constrói o aplicativo dividindo grandes objetivos em *sprints* menores
- Cada sprint é uma entrega
- Pode ajustar-se facilmente às mudanças, mesmo em estágios posteriores
- Envolve os clientes e usuários em todo o ciclo de desenvolvimento

# Referências

**Rapid Application Development (RAD) | Definition, Steps & Full Guide.** Disponível em:  
<<https://kissflow.com/application-development/rad/rapid-application-development/>>.

**Ferramenta de desenvolvimento rápido de aplicativos (RAD) | Microsoft Power Apps.** Disponível em:  
<<https://powerapps.microsoft.com/pt-br/rapid-application-development-rad/>>.