

# DESCREVER A CONTEXTUALIZAÇÃO, OS CONCEITOS, PRINCÍPIOS, AS FERRAMENTAS E TÉCNICAS DA METODOLOGIA DE DESENVOLVIMENTO RÁPIDO DE SOFTWARE (RAD)

## CONTEXTUALIZAÇÃO

O desenvolvimento rápido de aplicações RAD (Rapid Application Development) é uma metodologia de desenvolvimento de software com foco na entrega em um período muito inferior ao do ciclo de desenvolvimento tradicional de software. Não se trata de uma entrega final, mas, sim, de um protótipo do software. Para que isso seja possível, é feito um planejamento mínimo para obter um protótipo rápido.

Na metodologia RAD, existe uma concentração no desenvolvimento dos principais módulos funcionais do sistema. Essa versão inicial, que, apesar de limitada, já é funcional, é chamada de protótipo.

---

É muito útil para a compreensão do sistema

---

**PROTÓTIPO** Serve de demonstração para os clientes

---

É mais flexível para mudanças

---

Quando está mais evoluído, pode ser integrado ao produto completo para uma entrega mais rápida da versão final

---

A RAD também pode ser aplicada para aperfeiçoar o treinamento prático de estudantes de computação, auxiliando-os em seus futuros empregos. Isso porque os estudantes podem aplicar o conhecimento adquirido nas aulas para desenvolver sistemas em etapas, conforme é proposto pela RAD. Como será mostrado mais adiante, o fator humano é um importante requisito para a aplicação dessa metodologia, então a sua aplicação para treinar recursos humanos pode acelerar a curva de aprendizado dentro de um curto período.



Os projetos RAD seguem o **modelo iterativo e incremental**. As equipes de desenvolvimento são pequenas, compostas por desenvolvedores, analistas de negócio e representantes de clientes. **Um dos aspectos mais importantes deste modelo é garantir que os protótipos desenvolvidos sejam reutilizáveis para o projeto do sistema, ou seja, a ideia não é criar unidades descartáveis.** Isso não contradiz o fato de o protótipo ser flexível.

O RAD foca no desenvolvimento rápido por meio de iterações frequentes e feedback contínuo.

---

### Você sabia

O modelo RAD foi introduzido pelo consultor de tecnologia e autor James Martin em 1991 (MARTIN, 1991). Surgiu como o reconhecimento da necessidade de atender o competitivo mercado de software, que tem uma demanda contínua por novas

aplicações. Uma característica que foi explorada para a formalização da RAD foi a flexibilidade do desenvolvimento de software para projetar modelos de desenvolvimento. Trata-se de uma combinação de sessões JAD, desenvolvimento de protótipos, equipes SWAT, entregas com prazo de entrega e ferramentas CASE.

---

---

É muito prática em diversos ambientes modernos de desenvolvimento.

#### **RAD**

Apresenta uma abordagem útil para criar aplicações de comércio eletrônico e aplicativos de dispositivos móveis.

Possui uma velocidade de entrega que pode determinar o posicionamento de uma empresa em um ambiente de mercado muito competitivo.

---

Portanto, trata-se de uma metodologia importante a ser empregada para que as empresas lancem suas aplicações antes de seus concorrentes.

Observe no fluxo como iniciar um projeto RAD:

Uma das formas de iniciar o projeto RAD é através da aplicação da metodologia Joint Application Development (JAD).

Trata-se de uma metodologia na qual usuários e analistas projetam o sistema juntos, sob uma liderança em oficinas de trabalho.

A ideia é potencializar o resultado do desenvolvimento através de dinâmicas de grupo.

Ou seja, definir os objetivos e as aplicações do sistema, desde a geração de telas até a geração de relatórios.

Tem como princípios: dinâmica de grupo; recursos audiovisuais; processo organizado e racional; a escolha do local; documentação com a abordagem WYSIWIG – “O que você vê é o que você obtém”.

A RAD foi a precursora do gerenciamento ágil de projetos. As características de prototipagem rápida e ciclos de liberação e iterações mais curtos fortaleceram o posicionamento da RAD como um método eficaz no desenvolvimento de software, tornando-se cada vez mais popular entre as empresas ágeis que procuram métodos que acompanhem o crescimento de suas necessidades comerciais e de clientes. Trata-se de uma metodologia orientada pelo feedback do usuário, e não por um planejamento detalhado e caro.

---

Os métodos tradicionais de desenvolvimento de software, como, por exemplo, a metodologia de desenvolvimento **cascata**, seguem modelos rígidos de processo. Isso significa que, nesses modelos tradicionais, os clientes são pressionados a estabelecer os requisitos antes do início do projeto. A iteração ao longo do projeto é baixa, o que complica o processo de mudança para novos requisitos e ajustes de viabilidade.



---

## CONCEITOS

Trata-se de uma abordagem interativa com o objetivo de produzir o desenvolvimento de software de alta qualidade. O resultado da aplicação da RAD é um software com menor custo, menos erros e menor tempo de desenvolvimento.

**A RAD pode ser considerada um tipo de técnica ágil.**

(NAZ & KHAN, 2015).

A metodologia RAD combina diversas técnicas para acelerar o desenvolvimento de aplicações de software. Outra forma pela qual a RAD é conhecida é como “Construção Rápida de Aplicações”, do inglês Rapid Application Building (RAB). Um dos principais elementos da RAD é o desenvolvimento de protótipos para chegar ao sistema final. Trata-se de um modelo adaptativo, uma vez que o desenvolvimento é feito em iterações em que mudanças podem ser realizadas a partir dos comentários do usuário. A ênfase está na criação rápida de um protótipo, em vez de um planejamento detalhado.

---

A metodologia RAD possui quatro elementos fundamentais:

Clique nas barras para ver as informações.

## **USO DE FERRAMENTAS PARA DAR SUPORTE AO DESENVOLVIMENTO**

O uso de ferramentas CASE facilita a automação no desenvolvimento de sistemas. Isso é obtido através de recursos como geração de código e verificação automática de erros de consistência. As ferramentas CASE, portanto, são usadas para gerar protótipos, dando, assim, suporte ao desenvolvimento iterativo, possibilitando que os usuários finais acompanhem a evolução do sistema à medida que ele está sendo construído.

## **METODOLOGIA BEM DEFINIDA**

É seguido um processo formal de desenvolvimento com atividades em etapas e entregas intermediárias. As tarefas são organizadas de modo a não negligenciar nenhum dos aspectos pré-acordados, e as técnicas são documentadas para garantir que uma tarefa seja executada da maneira correta.

## **PESSOAS**

Deve haver treinamento das pessoas tanto na metodologia de trabalho como no uso das ferramentas. As tarefas devem ser distribuídas por pequenas equipes, que devem trabalhar bem juntas.

## **GESTÃO**

O gerenciamento do projeto deve ser feito com rapidez. Isso é obtido através de oficinas de Planejamento de Requisitos e Design de Sistema para extrair rapidamente os requisitos dos usuários. Além disso, deve ser feita alocação de tempo fixo (**Timebox**) para entregar iterativamente o sistema para os usuários.

---

Além disso, existem dois tipos de projetos RAD:

### **Intensivo**

No tipo de projeto intensivo, uma equipe de desenvolvedores e usuários trabalham por um curto período (algumas semanas) e, ao final desse tempo, espera-se que produza um produto que seja utilizável.

### **Faseado**

Um projeto em fases é aquele distribuído por um longo período. Esses projetos são normalmente iniciados por um workshop JAD. As fases subsequentes do projeto são geralmente organizadas em termos de entrega e demonstração de protótipos incrementais. O objetivo é refinar continuamente o protótipo, tornando-o algo que seja entregue no final do timebox.

Como visto até aqui, está claro que a criação rápida de protótipo é a base da RAD.

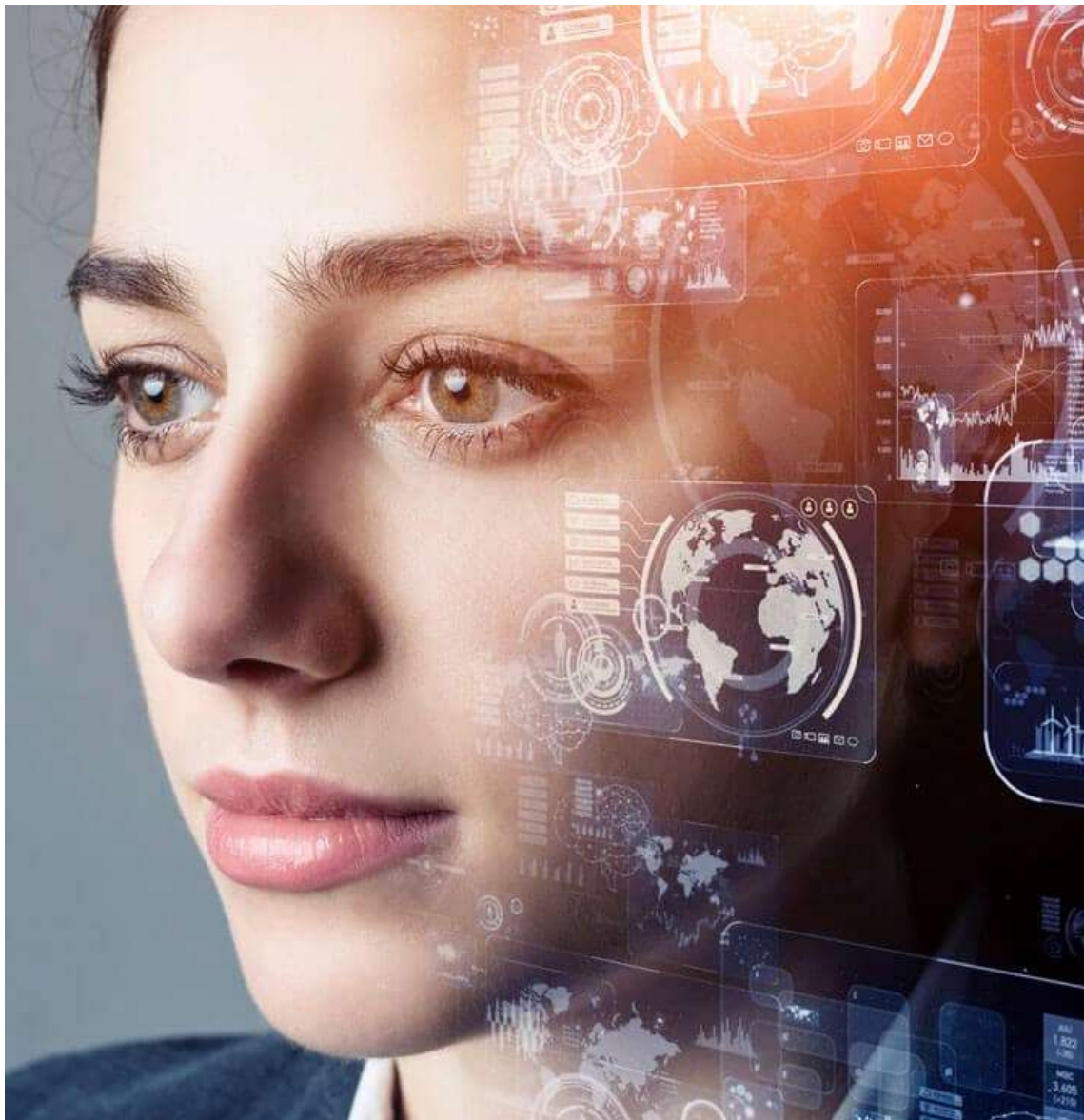
Nas situações em que os projetos são orientados por requisitos de interface do usuário, o desenvolvimento de protótipo é uma escolha muito adequada, pois é normal que o usuário crie a ideia de como a interface do sistema deve ficar ao longo do desenvolvimento do projeto. O desenvolvimento rápido de protótipos tem como pré-requisito o uso de ferramentas com suporte a componentes gráficos. No mercado, desde a década de 1990, existiam diversas ferramentas para esse fim, em que os programadores simplesmente podem selecionar um componente gráfico e arrastá-lo para um formulário. Desse modo, as interações com os usuários finais são mais produtivas, pois, constantemente, recebem um software operacional.

---



# PRINCÍPIOS

Constantemente, os programadores são pressionados a entregar as aplicações em prazos curtos e, muitas vezes, sabe-se com antecedência que o projeto terá de passar por modificações ao longo do desenvolvimento. Essas situações são exemplos em que o desenvolvimento rápido é bastante útil, pois ele está embasado exatamente na entrega rápida de protótipos que incorporam os comentários e as solicitações dos usuários a cada entrega. Para ser eficaz, no entanto, a RAD tem alguns requisitos que não são triviais. Alguns requisitos relacionados aos recursos humanos são os seguintes:



- Equipe de desenvolvedores qualificada e motivada.
- Usuários comprometidos com a participação ativa ao longo do projeto.
- Comprometimento para atingir o resultado satisfatório.

O desenvolvimento baseado na entrega de protótipos funcionais busca dar a oportunidade para que o usuário possa interagir com o projeto antes de receber o sistema final. Dessa forma, poderá fazer comentários e solicitações que guiarão os desenvolvedores na confecção do produto que atenda às suas expectativas sob o ponto de vista de funcionalidades, recursos, interatividade do sistema (experiência do usuário), relatórios, gráficos, entre outros.

---

O RAD é baseado em alguns princípios básicos, que são (FITZGERALD, 1998):

Clique nas barras para ver as informações.

### **ENVOLVIMENTO ATIVO DOS USUÁRIOS**

A metodologia RAD reconhece que o envolvimento do usuário é necessário para reduzir problemas caros de obtenção de requisitos. Além disso, os usuários podem rejeitar completamente os sistemas, se não estiverem suficientemente envolvidos no desenvolvimento. No centro da abordagem da RAD, estão as oficinas de design de aplicativos conjuntos (JAD) e planejamento de requisitos conjuntos.

### **EQUIPES PEQUENAS COM PODER DE DECISÃO**

As vantagens da elaboração de equipes pequenas estão na redução de ruídos de comunicação e na minimização de atrasos devido à burocracia que a hierarquia de uma metodologia tradicional impõe. Em relação aos ruídos de comunicação, os canais que tratam dessa área aumentam proporcionalmente ao tamanho da equipe, portanto equipes pequenas evitam a distorção e o conflito na comunicação. A respeito da redução do tempo, empoderar a equipe aumenta as chances de cumprir os prazos por causa da responsabilidade de tomada de decisão. As equipes têm o poder de tomar decisões sobre o design (embora as mudanças sejam reversíveis).

### **ENTREGA FREQUENTE DE PRODUTOS**

Diferentemente das metodologias de desenvolvimento tradicionais, em que os projetos podem levar muito tempo para serem concluídos, a RAD procura reduzir o tempo de desenvolvimento. Portanto, prazos mais curtos para o desenvolvimento são uma característica importante. Em vez de se concentrar no processo, a RAD tem como premissa a entrega de produtos que satisfazem os requisitos funcionais.

### **DESENVOLVIMENTO INCREMENTAL E ITERATIVO**

Outro princípio fundamental do RAD é que os sistemas evoluem de forma incremental em cada iteração. A cada nova iteração, surgem novos requisitos que são incorporados ao sistema. Desse modo, os sistemas evoluem através da prototipagem iterativa. Existe um entendimento no RAD que a especificação de requisitos é um processo não determinístico e que evolui à medida que desenvolvedores e usuários interagem com o protótipo do sistema.

### **ABORDAGEM TOP-DOWN**

Uma vez que, na metodologia RAD, os requisitos não precisam ser completamente definidos logo no início do projeto, eles são especificados em um nível apropriado ao conhecimento disponível no momento. Estes são então elaborados através de prototipagem incremental. Os sistemas são elaborados e confeccionados à medida que o conhecimento cresce. Além disso, como se trata de uma abordagem de “cima

para baixo” caracterizada por um curto período, todas as decisões são consideradas reversíveis rapidamente.

### UTILIZAÇÃO DE FERRAMENTAS DE AUTOMAÇÃO (CASE)

Trata-se de usar programas que facilitem a automação de processos, criação de diagramas, realização de testes e quaisquer tarefas que facilitem as entregas dentro dos prazos pré-estabelecidos e, obviamente, com qualidade. Além disso, essas ferramentas facilitam a reutilização de componentes que podem ser usados ao longo do projeto.

O ponto fundamental na metodologia RAD é que se trata de uma abordagem colaborativa entre todas as partes interessadas, que são: patrocinadores, desenvolvedores e usuários ao longo da vida de um projeto.

---

## FERRAMENTAS E TÉCNICAS

A RAD precisa ser suportada por ferramentas que auxiliem no desenvolvimento das aplicações rapidamente. Entre as categorias de ferramentas que dão suporte à RAD para desenvolver projetos de software estão:

- Integração de dados
- Ambientes de desenvolvimento
- Ferramentas de coleta de requisitos
- Ferramentas de modelagem de dados
- Ferramentas de geração de código

Desde que a RAD foi formalizada, foram desenvolvidas muitas técnicas para a sua utilização. Cada uma das técnicas tem suas particularidades, mas mantém a essência da RAD. No quadro a seguir, conheça algumas dessas técnicas (Naz; Khan, 2015):

TÉCNICA	PARTICULARIDADE
Modelo CBD	O método que descreve como componentes antigos podem ser reutilizados com os novos.
RepoGuard	É um framework para integração de ferramentas de desenvolvimento com repositórios de código-fonte.
Adição dinâmica ágil	Técnicas usadas para integração do ágil para tornar o projeto mais adaptável.
Método baseado em camadas para desenvolvimento rápido de software	Baseado em camadas que segue o <b>XP</b> .
Análise de projeto de sistema baseado em simulação	Desenvolvimento de ferramentas ágeis baseadas em simulação.
Uso de Ajax no RAD	Prototipagem rápida em aplicativos e ferramentas da Web.
Desenvolvimento de aplicativos multiusuário em ambiente distribuído rapidamente.	Middleware de comunicação.



A ideia do uso das técnicas de RAD é de otimizar os resultados obtidos dentro do tempo estimado, que, pela natureza da RAD, é curto. Essencialmente, um software é construído para atender a alguma demanda, ou seja, existe uma razão para que seja confeccionado. Portanto, a interação com os usuários auxilia o entendimento dos desenvolvedores para construir, agregar e incorporar esse entendimento em um protótipo através de técnicas e ferramentas **que acelerem a entrega e reduzam os desvios de compreensão**. A concordância sobre o propósito do sistema e a sua evolução é muito importante para o sucesso do projeto. Tanto desenvolvedores como clientes devem estar envolvidos em interações formais que fortaleçam o comprometimento de todos.

A pressão por soluções de software confiáveis e em curtos prazos favoreceu a criação da metodologia de desenvolvimento rápido de software (RAD). A ideia de entregar protótipos em um ciclo de desenvolvimento incremental e iterativo permite que o usuário possa ter rapidamente uma visão clara de como o sistema está progredindo e se existe alguma questão relacionada aos requisitos que precisa ser aperfeiçoada. Portanto, a colaboração entre desenvolvedores e usuários suporta o desenvolvimento de especificações mais precisas e validadas.