

<b>Verbos</b> GET {id} Detalhes do item GET Listagem POST Criar PUT Editar DELETE Remover	<b>RNF Monitoramento - Web</b> Quantidade de acessos Logging Tempo de resposta Origem dos acessos	<b>RNF Performance</b> Tempo de Resposta Tempo de Processamento Janela de processamento	<b>Princípios</b> Foco no cliente Evolução operacional Agilidade Novas tecnologias Essência competitiva
<b>Status Code</b> 200 OK 400 Requisição inválida 401 Não autorizado 403 Proibido 405 Método não permitido 408 Timeout da requisição 500 Erro interno 501 Não implementado 505 Versão HTTP não suportada	<b>RNF Monitoramento - Banco</b> Utilização Concorrências Locks	<b>RNF Integração</b> Evitar usar ETL (usar streaming/fila) APIs com REST (resposta em JSON) APIs devem ter Swagger documentado! Monitoramento das integrações Camadas anti-corrupção	<b>Engenharia de Software</b> DRY (não repetir código) KISS (manter curto e simples) Isoladamento Consistência (manter o padrão) Encapsulamento Group Related Functions Use Layering
<b>LOGs</b> INFO Funções críticas DEBUG Entrada de métodos WARN Erro tratado / RN ERROR Exceções	<b>RNF Monitoramento - Servidor</b> Uso de CPU Consumo de memória Falhas	<b>RNF Documentação</b> Doc da aplicação simples e objetiva Requisitos do ambiente Configurações específicas do ambiente Dependências	<b>RNF Segurança</b> Mecanismo de segurança (SysSec) Usuário de banco Single Sign-On (SSO) Dados só com usuários autenticados Infra, senhas, usuários protegidos Web-apps e recursos em HTTPS Bloqueio de buscadores
	<b>Requisitos desenvolvimento</b> Escalável Flexível Disponível Confiável Operável Interoperabilidade Recuperação rápida Manutenibilidade	<b>RNF Testes</b> Alto volume - fazer teste de Stress Pontos críticos - Teste de Regressão (ST)	<b>RNF Logging</b> Seguir padrão Sempre registrar exceções Classificação de criticidade Banco - gravar erros em tabela log
	<b>RNF Volumetria</b> Quantidade de acessos Período de pico Usuários simultâneos Volume de dados afetado Volume de dados trafegado	<b>Desenvolvimento</b> Feature Toggle Testes A/B Layout responsivo Front-end seguro e leve Back-end desacoplado e MS	<b>Monitoramento</b> Azure Insights



By **gcacars**  
[cheatography.com/gcacars/](https://cheatography.com/gcacars/)

Published 15th March, 2021.  
 Last updated 24th October, 2019.  
 Page 1 of 2.

Sponsored by **ApolloPad.com**  
 Everyone has a novel in them. Finish Yours!  
<https://apollopad.com>

### Teste unitário automatizado

Rotinas críticas

Regras de negócio

Queries

### Arquitetura técnica

Orientado à Serviços

Agnóstico à tecnologia

Escalável

Resiliente

Extensível

Infraestrutura leve

Distribuído

### CrITÉrios de aceite

Instrumentação (tracing e logging)

Testes unitários automatizados

Testes funcionais

Testes não-funcionais

Plano de execução (queries - custo)

Plano de implantação e rollback

Comunicação e capacitação

### RNF Infra - Cloud

Indicado para novas soluções

Evitar usar recursos exclusivos

Micro serviços com Docker ou PAAS

Ambiente certificado por Segurança

### RNF Infra - On Premise

Estrutura atual suporta?

Precisa liberar firewall?

Qual ambiente usar?



By **gcacars**  
[cheatography.com/gcacars/](https://cheatography.com/gcacars/)

Published 15th March, 2021.  
Last updated 24th October, 2019.  
Page 2 of 2.

Sponsored by **ApolloPad.com**  
Everyone has a novel in them. Finish Yours!  
<https://apollopad.com>