

# Estudo Dirigido

## Requisitos

**Nome:** Daniel Henrique Matos de Paiva

**RA:** 319226054

**Nome:** Josimar José

**RA:** 320137317

**Nome:** Pedro Henrique de Paula Oliveira

**RA:** 320136060

**Nome:** Rafael Ferreira Pedrosa

**RA:** 319228038

**Nome:** Herigson Felipe Braga Verciano

**RA:** 320136075

**Nome:** Douglas Ramos Rodrigues

**RA:** 329210212

- Cada integrante do grupo deverá postar uma cópia individual em pdf no ulife;
- Este estudo dirigido será avaliado como parte da Avaliação A3;

## Regra de Negócios

Uma Regra de Negócios é uma declaração de uma condição ou política que deve ser satisfeita dentro do ambiente corporativo. O objetivo delas é representar como os negócios, incluindo suas ferramentas de negócios, devem operar, por exemplo:



- O valor total de um pedido é igual à soma dos totais dos itens de venda acrescido de 10% de taxa de entrega;
- Um cliente do banco não pode sacar mais de R\$ 500,00 por dia em sua conta;
- Senhas devem ter, no mínimo, seis caracteres, entre números, letras e símbolos;
- Para alugar um carro, o cliente deve estar com a carteira de motorista válida;
- O número máximo de alunos por turma deve ser igual a 30.

As Regras de Negócios são revisadas por investidores, analistas de processo de negócios e designers de negócios para assegurar que as descrições do negócio estejam de acordo com o funcionamento do negócio. Elas também são utilizadas pelos analistas de sistemas e arquitetos de software ao definir e projetar um sistema que vai apoiar o negócio de uma empresa.

## Categorias de Regras de Negócios

As regras podem ser classificadas de várias maneiras, dentre elas a mais comum é dividi-las entre regras de limitação e de derivação. Essa classificação de regras de negócios é prática ao explicar o que são regras de negócios, como localizá-las e como trabalhar com elas. Logo, na prática não é necessário consultar sempre essas essas categorias e considerá-las como categorias fixas.

1. Regras de Limitação
  - 1.1. Regras de estímulo
  - 1.2. Regras de limite de operação
  - 1.3. Regras de limite de estrutura
2. Regras de derivação
  - 2.1. Regras de conclusão
  - 2.2. Regras de cálculo

---

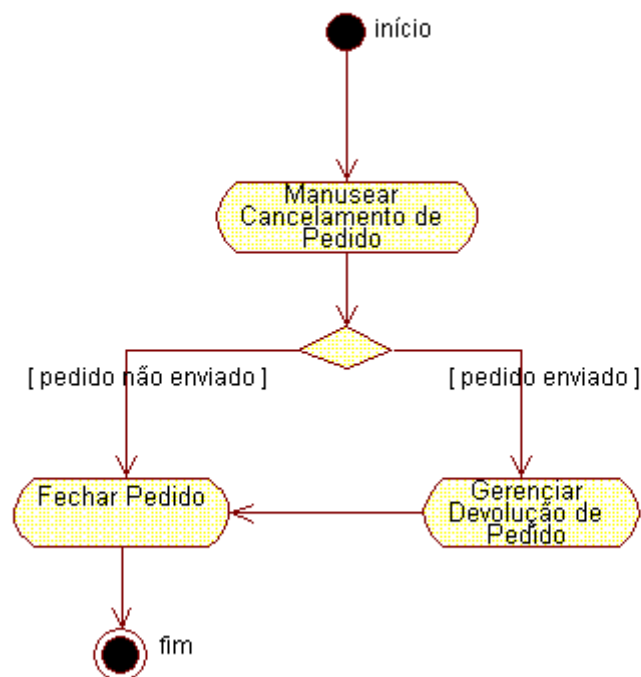
## Regras de Limitação

Especificam políticas ou condições que restringem a estrutura e o comportamento dos objetos. Regras de limite sempre podem ser aplicáveis ou aplicáveis somente sob determinadas condições. Os limites que sempre são aplicáveis são referidos como invariáveis.

## Regras de Estímulo

Limitam o comportamento especificando quando e se as condições devem ser verdadeiras para que o comportamento seja disparado. Examinar os eventos de negócios identificados é uma fonte muito útil para definir esses tipos de regras de negócios. Geralmente um evento de negócios é identificado porque alguém ou algo tem interesse em que o evento ocorra. Uma dica para descobrir esse tipo de regra de negócio é fazer a seguinte pergunta: *"Quais condições ou comportamento se aplicam depois que o evento ocorre?"*

**EXEMPLO:** Em um comércio qualquer, você poderá localizar a seguinte regra para encerramento de pedidos:



**QUANDO** um Pedido é cancelado

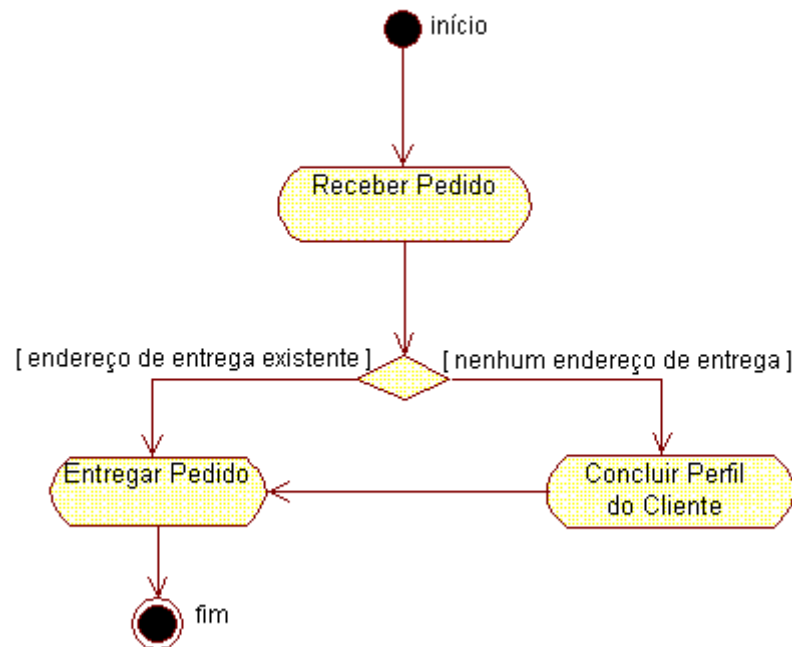
**SE** um Pedido não é enviado

**ENTÃO** fechar Pedido.

### Regras de Limite de Operação

Especificam as condições que devem ser verdadeiras antes e depois de uma operação para garantir que ela seja executada corretamente. Esse tipo de regra de negócios geralmente é convertida em pré-condições e pós-condições de um workflow ou em um caminho condicional ou alternativo em um workflow. Também pode ser uma meta de desempenho ou alguma outra regra não comportamental que deve ser rastreada nos casos de uso de negócios aos quais se aplica.

**EXEMPLO:** Em um comércio qualquer, você poderá localizar a seguinte regra:



Enviar Pedido ao Cliente

**SOMENTE SE** O cliente tiver um endereço de entrega.

No exemplo acima a regra de negócios é convertida em um caminho alternativo no workflow, ou seja, caso não haja nenhum endereço de entrega o cliente deverá concluir seu cadastro na loja.

## EXEMPLO 2

### DEVEMOS SEMPRE ESPERAR QUE

Todas as pesquisas do cliente devem ser respondidas dentro de 24 horas do seu recebimento

No exemplo acima a regra pode ser convertida em uma meta de desempenho

### Regras de Limite de Estrutura

Especificam políticas ou condições sobre classes, objetos e suas relações que não devem ser violados. Esse tipo de regra de negócio afeta as relações entre instâncias de entidades de negócios. São expressas pela existência de uma associação entre duas entidades de negócios, às vezes como uma multiplicidade na associação.



### DEVE SEMPRE ESPERAR QUE

**Um Pedido se refere a pelo menos 1 Produto.**

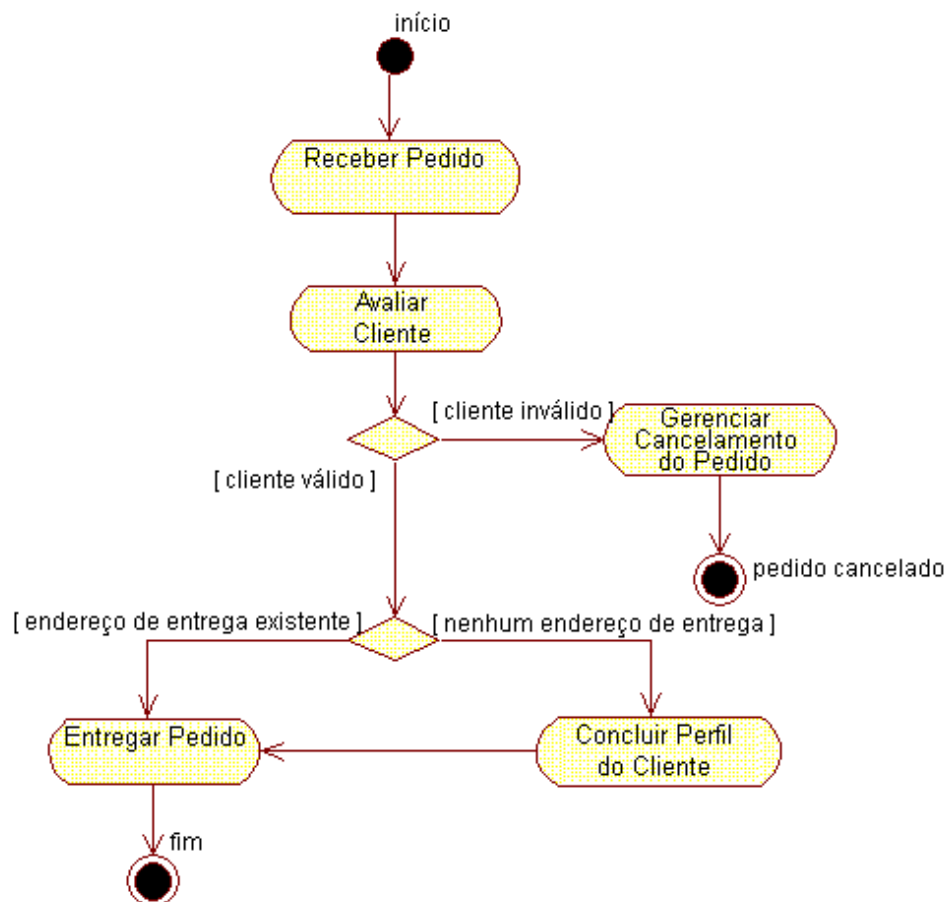
### Regras de Derivação

Especificam políticas ou condições para deduzir ou calcular fatos a partir de outros fatos.

### Regras de Conclusão

Especificam que se determinados fatos forem verdadeiros, uma conclusão pode ser deduzida. Geralmente, as regras de conclusão parecem semelhantes às de estímulo e resposta, de limite de operação ou de limite de estrutura. A diferença é a existência de algumas etapas que precisam ser atingidas para chegar à conclusão. A regra contém um método que precisa ser refletido em um estado de atividade do workflow e eventualmente em uma operação em um trabalhador de negócios ou entidade de negócios.

**EXEMPLO:** determinar o status de um cliente



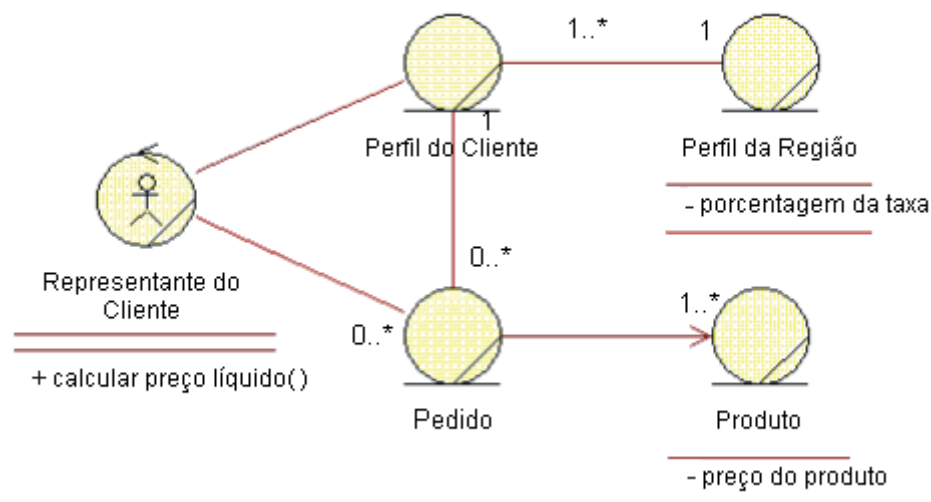
Um Cliente é um Bom Cliente **SE SOMENTE SE**

As faturas não pagas enviadas a esse Cliente têm menos de 30 dias.

## Regras de Cálculo

Deduzem seus resultados pela forma de processar algoritmos, uma variante mais sofisticada de regras de conclusão. As regras de cálculo são semelhantes às regras de conclusão. A diferença é que o método é mais formal e semelhante a um algoritmo. Assim como as regras

de conclusão, esse método precisa ser rastreado em uma tarefa no fluxo de trabalho e, eventualmente, em uma operação em um trabalhador de negócio ou uma entidade de negócio.



O preço líquido de um Produto **É COMPUTADO DA SEGUINTE MANEIRA**

$\text{preço do produto} * (1 + \text{porcentagem de imposto} / 100)$

## Exercícios

Considerando um restaurante que faz entrega de comidas para seus clientes, liste no mínimo 5 regras de negócios que podemos encontrar neste tipo de negócio.

Pensando num modelo de restaurante que atenda a clientela através de app's:

1. RN01: Enviar pedido para produção somente após confirmação de pagamento;
2. RN02: Aceitar pagamento somente através de cartão de débito e crédito;
3. RN03: Limitar o raio de entrega dos pedidos somente se a residência dos



clientes estiver ao alcance de 10km;

4. RN04: Confirmar o atendimento do restaurante somente se os pedidos feitos estiverem dentro do horário de 10:00 às 00:00;
5. RN05: Enviar pedido ao cliente somente se o cliente possuir endereço devidamente cadastrado e válido no sistema;

## Requisitos Funcionais

Um requisito é definido como *"uma condição ou um recurso com o qual um sistema deve estar em conformidade"*. Os **requisitos funcionais** especificam **ações que um sistema deve ser capaz de executar**, sem levar em consideração as restrições físicas.

- Registro de erro do sistema;
- Acesso Remoto;
- Entrar no Sistema (Logon)
- Inscrever em cursos;
- Cancelar matrícula;
- Visualizar notas dos alunos;
- Buscar cursos;
- Gerenciar usuários

- Visualizar informações do catálogo de cursos
- Compartilhar postagem
- Cadastrar notas;
- Visualizar Saldo;
- Enviar um PIX;
- Fazer Transferência;
- Fazer Download de arquivo

Os requisitos funcionais são detalhados em um Modelo de Caso de Uso e em casos de uso.

Os **requisitos funcionais especificam**, portanto, **o comportamento de entrada e saída de um sistema**.

## Exercícios

Considerando uma rede social (p.ex.: Instagram, Twitter, Facebook), liste no mínimo 7 requisitos funcionais que podemos encontrar neste tipo de negócio.

1. RF01: Cadastrar usuários
2. RF02: Compartilhar mídias como fotos, vídeos e áudios;
3. RF03: Realizar buscas
4. RF04: Gerenciar perfil

5. RF05: Deletar perfis
6. RF06: Autenticar usuários
7. RF07: Comentar fotos
8. RF08: Curtir fotos
9. RF09: Acompanhar páginas oficiais
10. RF10: Trocar mensagens entre usuários através de um sistema de chat

## Requisitos Não Funcionais

Requisitos que não são funcionais, ou seja, aqueles que não especificam ações que um sistema deve ser capaz de fazer, são chamados de requisitos não funcionais. Existem vários **requisitos não funcionais** e eles **descrevem apenas atributos do sistema** ou **atributos do ambiente do sistema**. Os Requisitos não-funcionais tratam tipicamente de problemas como os descritos abaixo:

- Funcionalidade: funcionalidades que não se relacionam com os casos de uso, por exemplo: auditoria, relatório, interoperabilidade e segurança;
- Usabilidade: fatores humanos, Experiência do Usuário (UX), estética, consistência na interface com o usuário, ajuda on-line e sensível ao contexto (tool tips), assistentes e agentes virtuais, documentação do usuário, materiais de treinamento;

- **Confiabilidade:** diz respeito à integridade, conformidade e interoperabilidade do software. Os requisitos a serem considerados são: frequência e gravidade de falha, capacidade de recuperação, possibilidade de previsão, precisão e tempo médio entre falhas (MTBF).
- **Desempenho:** descreve as características de desempenho do seu sistema, tais como: velocidade, eficiência, disponibilidade, precisão, produtividade, tempo de resposta, tempo de recuperação e uso de recursos.
- **Suportabilidade:** Descreva todos os requisitos que aprimorarão a suportabilidade ou manutenibilidade do seu sistema, tais como: padrões de codificação, convenções de nomeação, bibliotecas de classes, acesso à manutenção e utilitários de manutenção. Ainda, os requisitos de suportabilidade podem incluir as possibilidades de teste, adaptação, manutenção, compatibilidade, configuração, serviço, instalação e localização (internacionalização)

## Exercícios

Considerando uma loja online de jogos, por exemplo: Steam; liste no mínimo 5 requisitos não funcionais que podemos encontrar neste tipo de sistema

Remova este comentário e escreva a sua resposta aqui

1. RNF01: Compatibilidade de Sistemas Operacionais, sendo eles: Windows, Linux e MacOS
2. RNF02: Segurança nas transações comerciais
3. RNF03: Disponibilidade de 99% do sistema

4. RNF04: Suporte para 1000 requisições por minuto
5. RNF05: Design coerente e consistente na experiência do usuário

## Relacionando Regras e Requisitos

Agora que conhecemos de maneira prática e aprofundada o que são regras de negócio, requisitos funcionais e requisitos não funcionais; será que conseguimos relacionar cada um desses conceitos para diferentes tipos de projetos?

### Exercícios

1. Você está desenvolvendo um sistema para um restaurante que faz entregas para toda região metropolitana de Minas Gerais. Considerando este cenário, preencha a tabela abaixo relacionando cada regra de negócio encontrada por vocês, com um requisito funcional e não funcional correspondente.

Regra de Negócio	Requisito Funcional	Requisito Não Funcional
Enviar Pedido ao Cliente SOMENTE SE O cliente tiver um endereço de entrega.	Fazer Pedido	Fornecer certificado de segurança criptografado.
Gerar pedidos após confirmação de pagamento	Selecionar forma de pagamento	Segurança nas transações
Aceitar pagamento somente	Cadastrar cartão de débito e	Validar informações do

através de cartão de débito e crédito	crédito	cartão através de API's da entidade financeira
Limitar o raio de entrega dos pedidos na região metropolitana	Cadastrar endereço de entrega	Traçar rotas de entrega a partir de uma integração externa ao aplicativo
Limitar o horário de atendimento, entre 10:00 às 00:00;	Enviar notificações aos usuários	Todo dispositivo deverá utilizar um horário UTC baseado em Brasília
Enviar pedido ao cliente somente se o cliente possuir endereço devidamente cadastrado e válido no sistema	Mostrar andamento do pedido	Sincronização das informações através de servidores que utilizam um sistema distribuído

2. Você está desenvolvendo um sistema acadêmico para uma Universidade que recebe alunos de todos os países. Considerando este cenário, preencha a tabela abaixo relacionando cada regra de negócio encontrada por vocês, com um requisito funcional e não funcional correspondente.

Regra de Negócio	Requisito Funcional	Requisito Não Funcional
Um Aluno é aprovado SE SOMENTE SE Conseguir uma nota final acima de 70% em determinada disciplina.	Gerar Ata de Resultado Final	O sistema deve estar 98% disponível. Proporcionar serviços para proteção de acesso.
Um Aluno está em recuperação SE SOMENTE SE Conseguir uma nota final entre 60 e 69% em determinada disciplina.	Aplicar exame suplementar	Suportabilidade de alunos na plataforma do exame através da Internet

Um Aluno é aprovado SE SOMENTE SE Conseguir frequência acima de 80% em determinada disciplina.	Registrar frequência através de login	Ajuda online para retificação de faltas
Linguagem disponível conforme nacionalidade do usuário	Permitir seleção de idioma	Base de dados de idiomas
Um Aluno é aprovado SE SOMENTE SE Conseguir horas em estágio obrigatório em determinado curso.	Cadastro de Estágio	Barra de acompanhamento de horas concluídas

## FURPS

O documento de *Especificação Suplementar de Requisitos* captura os requisitos de sistema que não são capturados imediatamente nos casos de uso do modelo de casos de uso. Entre os requisitos estão incluídos os seguintes atributos de qualidade do sistema: usabilidade, confiabilidade, desempenho e suportabilidade. Tais requisitos fazem parte da FURPS, que é um modelo de classificação de atributos de qualidade de software, desenvolvido na Hewlett-Packard (HP) e publicado pela primeira vez por Grady e Caswell.

Considerando um sistema qualquer, preencha cada uma das seções abaixo do documento de Especificação Suplementar de Requisitos referente à FURPS.

Nome do Sistema

Super Xandão Fitness

**Descreva resumidamente o sistema e suas principais funcionalidades**

### Resumo da Aplicação

Este é um aplicativo para ajudar ao usuário que deseja trabalhar sua saúde e seu aspecto estético.

#### Funcionalidades:

- Monitorar a lotação da academia;
- Agendar horário para utilizar determinada área: natação, musculação, pilates, zumba ou arte marcial;
- Cadastrar, editar e visualizar a ficha de exercício pelo educador físico e através disso o marombeiro irá receber dicas de quais aparelhos conseguirão trabalhar determinada musculatura;
- Diferenciar perfis de usuário: aluno, educador físico;
- Realizar pagamento de mensalidade.

## 1. Funcionalidade

Descreva abaixo as funcionalidades que não se relacionam com os casos de uso, por exemplo: auditoria, relatório, interoperabilidade e segurança. Coloque o nome da funcionalidade e a descrição dela, por exemplo:

- **E-mail:** fornecer serviços que permitam que os usuários enviem e recebam mensagens;
- **Ajuda Online:** disponibilizar ajuda online para os usuários;
- **Segurança:** Proporcionar serviços para proteção de acesso a determinados recursos ou informações.

Resposta:



**Interfaces:** Fornecer interfaces adaptáveis para dispositivos mobile e desktop;  
**Segurança:** Oferecer método de autenticação seguro com criptografia, controle de acesso físico e lógico aos servidores do sistema, garantir redundância de dados em caso de desastre;  
**Ajuda Online:** permitir envio de mensagens de usuários com perfis de aluno para seu instrutor cadastrado.

## 2. Usabilidade

Descreva nesta seção todos os requisitos de qualidade relacionados a usabilidade, tais como: facilidade de uso, facilidade de aprendizado, padrões de usabilidade e localização; Por exemplo:

- Tempo de treinamento necessário para que usuários comuns ou avançados se tornem produtivos em operações específicas no sistema;
- Especifique períodos de tempo mensuráveis para tarefas típicas no sistema;
- UI/UX Design Patterns.

Os requisitos de usabilidade podem incluir as seguintes subcategorias: fatores humanos, estética, consistência na interface com o usuário, ajuda on-line sensível ao contexto, assistentes e agentes, documentação do usuário e materiais de treinamento.

Resposta:

- **Design minimalista:** menos informações por página, mais visuais, cores leves e atrativas;
- **Padronização:** consistência no formato dos menus, botões, cores e ilustrações / arte videográfica do aplicativo;
- **Aplicação com poucas telas:** utilização de poucos cliques para encontrar o que precisa, sem páginas subsequentes umas às outras, ajudando na memorização da funcionalidade das ferramentas disponíveis;
- **Ícones autoexplicativos:** ícones que transmitam a função apenas com seu desenho;
- **Botão Ajude-me:** breve tutorial explicando os elementos da página e suas funcionalidades;

### 3. Confiabilidade

Os requisitos de confiabilidade a serem considerados são: frequência e gravidade de falha, capacidade de recuperação, possibilidade de previsão, precisão e tempo médio entre falhas (MTBF). Por exemplo:

- A. Disponibilidade:** especifique a porcentagem de tempo disponível ( xx.xx%), as horas de uso, o acesso à manutenção, as operações de modo degradado, etc.
- B. Tempo Médio entre Falhas (MTBF):** normalmente especificado em horas, mas também poderá ser especificado em termos de dias, meses ou anos.
- C. Tempo Médio para Reparo (MTTR):** quanto tempo o sistema poderá ficar sem funcionar após uma falha?

- D. Exatidão:** especifique a precisão (resolução) e exatidão (através de algum padrão conhecido) necessárias na saída do sistema.
- E. Taxa máxima de erros ou defeitos:** geralmente expressa em termos de erros / KLOC (thousands of lines of code, milhares de linhas de código) ou de erros / ponto de função.
- F. Taxa de erros ou defeitos (categorizada em termos de erros de pouca, média ou muita importância):** os requisitos devem definir o que se entende por erro "crítico" (por exemplo, perda total de dados ou total incapacidade de usar determinadas partes da funcionalidade do sistema).

Descreva abaixo os requisitos de confiabilidade para seu sistema referente aos **itens A, B e C** da lista acima:

Resposta:

**Disponibilidade:** O sistema deve ter alta disponibilidade, visto que o aluno da academia pode realizar cadastro de perfil, ficha realização de pagamento da mensalidade e marcação de horário no momento em que desejar (98% de disponibilidade).

**Tempo Médio entre Falhas (MTBF):** 1 mês

**Tempo Médio para Reparo (MTTR):** Não pode ultrapassar 1 dia, visto que os meios de pagamento estão acoplados no sistema.

#### 4. Desempenho

Descreva as características de desempenho do seu sistema, tais como: velocidade, eficiência, disponibilidade, precisão, produtividade, tempo de resposta, tempo de recuperação e uso de recursos. Por exemplo:

- **Tempo de resposta de uma transação:** tempos médio e máximo;
- **Taxa de transferência:** quantidade de transações por segundo;

- **Capacidade:** o número de clientes ou de transações que o sistema pode acomodar;
- **Modos de degradação:** o modo aceitável de operação quando o sistema tiver sido degradado de alguma maneira;
- **Uso de recursos:** memória, disco, comunicações, etc

Resposta:

**Acessibilidade ao aplicativo:** modo off-line e on-line.

**Tempo de atualização de dados:** Atualizar a cada minuto a capacidade de lotação dos alunos no modo on-line.

**Modo de armazenamento:** Armazenar informações por 2 horas no modo off-line.

**Tempo de Timeout:** Obter tempo limite de resposta de 40 segundos.

## 5. Suportabilidade

Descreva todos os requisitos que aprimorarão a *suportabilidade* ou *manutenibilidade* do seu sistema, tais como: padrões de codificação, convenções de nomeação, bibliotecas de classes, acesso à manutenção e utilitários de manutenção. Ainda, os requisitos de suportabilidade podem incluir as possibilidades de teste, adaptação, manutenção, compatibilidade, configuração, serviço, instalação e localização (internacionalização)

Resposta:

**Convenção utilizada por determinada tecnologia:** Todo o código deve estar indentado e comentado;

**CleanCode:** Utilizar nomes objetivos para facilitar a manutenção;

**Padronização do Sistema:** O sistema deve utilizar o paradigma de orientação a objetos;

**Persistência de Dados:** O sistema deve ter ambientes para armazenamento de dados, sendo eles: Desenvolvimento, Homologação e Produção,;

**Ganho de mercado:** Aplicação com disponibilidade web e portabilidade móvel;

**Versionamento:** O sistema utilizará o Git como programa de versionamento e no padrão GitFlow.