

Professor:

Marco Antonio da Silveira Campos

mas.campos@sp.senai.br

cel: (19) 99768-1683



- São **PROIBIDOS** em sala de aula:
 - Celulares e tablets (devem estar sempre **desligados e na bolsa**);
 - Bonés, chapéus e correlatos;
 - Entrar sem uniforme (calça **jeans**, camiseta do Senai e sapato fechado);
 - Comer ou beber (aceitável garrafinha de água fechada);
 - Trazer notebook particular.
- Após o término da aula, o aluno deverá arrumar o seu ambiente de trabalho (teclado, mouse, monitor e cadeira) e manter a limpeza;
- Brincadeiras e conversas terão seus momentos;
- O aluno deverá ter no mínimo 75% de presença para ser aprovado;
- Meta para o SENAI: presença 95% e média 90;
- Atrasos – Registrados pelo professor e após o 3º será encaminhado a coordenação;
- Falta deverá ser justificada no dia seguinte, na entrada da aula com o professor. Esta é uma responsabilidade do ALUNO.

e. DESVANTAGENS

24. MATRIZ SWOT

25. GRÁFICO DE GANT

26. MATRIZ DA EQUIPE – PERFIL E FUNÇÃO

27. RECURSOS/TECNOLOGIAS/PLATAFORMAS
UTILIZADOS

28. RETORNO FINANCEIRO

29. ESTRATÉGIA DE ABORDAGEM

30. REQUISITOS

a. FUNCIONAIS

b. NÃO FUNCIONAIS

31. DOCUMENTAÇÃO TÉCNICA

a. BANCO DE DADOS

i. DER (DIAGRAMA DE
ENTIDADES E
RELACIONAMENTOS)ii. MER (MODELO DE
ENTIDADES E
RELACIONAMENTOS)

iii. DICIONÁRIO DE DADOS

iv. ENTIDADES CANDIDATAS

v. ENTIDADES EFETIVAS

b. DIAGRAMA DE CASO DE USO

32. PLANO DE TESTE

33. EVOLUÇÃO DO SISTEMA (futuro)

Requisitos Funcionais e Não Funcionais

Uma importante etapa durante o desenvolvimento de um programa é a definição dos requisitos, que podem ser de diferentes categorias.

De forma muito simplista, podemos considerar que um requisito de software é toda abstração de um *recurso*, *funcionalidade* ou *resultado* esperado de um sistema.

O processo de levantamento de requisitos é responsável por identificar necessidades e demandas.

“Na engenharia de software, mais precisamente na engenharia de requisitos (comumente chamada apenas de Análise de Requisitos ou Levantamento de Requisitos) é a disciplina que identifica a “dor” do cliente, faz um “diagnóstico” sobre sua origem e propõem um “tratamento terapêutico” para curá-lo.”

Chico Alff

FUNCIONAIS:

Requisitos funcionais são todas as necessidades, características ou funcionalidades esperadas em um processo que podem ser atendidos pelo software:

- inserir dados em um formulário;
- buscar pratos específicos em um cardápio;
- consultar o status de um pedido;
- realizar compras;
- alterar informações de um registro;
- elaborar relatórios.
- Manter Usuário.

O QUE É UM REQUISITO FUNCIONAL?



REQUISITO FUNCIONAL

Mas afinal, por que eles são chamados de **requisitos funcionais**? A categorização dos requisitos citados como requisitos funcionais se deve ao fato de que todos eles são ***funcionalidades atendidas*** através de uma ação do software ou comportamento específico do sistema.

Podemos dizer que **é considerado um requisito funcional**, todo cenário onde o usuário informa um dado, ou um sistema terceiro realiza uma solicitação qualquer durante uma interação com o sistema, que então, responde com determinada ação correspondente.

NÃO FUNCIONAIS:

Os requisitos não funcionais são todos aqueles relacionados à forma como o software tornará realidade os que está sendo planejado. Ou seja, enquanto os requisitos funcionais estão focados **no que será feito**, os não funcionais descrevem **como serão feitos**.

“Um REQUISITO FUNCIONAL é “O QUE” o sistema deve fazer Um REQUISITO NÃO FUNCIONAL é “COMO” o sistema deve fazer.”

Chico Alff

Dessa forma, requisitos não funcionais **devem sempre ser mensuráveis**, ou seja, deve ser **possível verificar** se ele está ou não sendo atendido pelo software.

Exemplos de requisitos não funcionais:

- O sistema deve ser multiplataforma – Windows, Linux e macOS
- O desenvolvimento deve ser em linguagem React Native.
- O programa deve funcionar offline.
- O sistema deve respeitar o tempo máximo de 160 segundos durante processamentos.

Estes indicam condições ou então características de **COMO** será executada determinada ação pelo sistema.
Lembre-se: requisitos não funcionais devem sempre ser mensuráveis!

REQUISITOS NÃO FUNCIONAIS



DESEMPENHO

QUAL O CRITÉRIO DE DESEMPENHO QUE O SISTEMA DEVERÁ ATENDER? O CRITÉRIO É MENSURÁVEL?

REQUISITO
NÃO FUNCIONAL



DISPONIBILIDADE

QUAL O DE DISPONIBILIDADE E ACESSIBILIDADE O SISTEMA DEVERÁ ATENDER? É POSSÍVEL VERIFICAR ESTE CRITÉRIO

REQUISITO
NÃO FUNCIONAL



SEGURANÇA

QUAIS AS CONDIÇÕES DE SEGURANÇA QUE O SISTEMA DEVE GARANTIR? COMO É POSSÍVEL MEDIR ESTAS CONDIÇÕES?

REQUISITO
NÃO FUNCIONAL



INTEGRABILIDADE

QUAIS MÉTODOS DE INTEGRAÇÃO E INTEROPERABILIDADE O SISTEMA DEVE ATENDER? COMO TESTAR ESTES MÉTODOS?

REQUISITO
NÃO FUNCIONAL

REQUISITOS FUNCIONAIS QUE COMPÕEM AS CARACTERÍSTICAS QUE O SISTEMA DEVE RESPEITAR AO ATENDER CADA UM DOS REQUISITOS FUNCIONAIS SOLICITADOS.

análise de
REQUISITOS



E-mail: SCOTTADAMS@AOL.COM



© 2004 Scott Adams, Inc. Dist. by UFS, Inc.



© 2004



www.dilbert.com



analise de REQUISITOS.com.br

Dilbert by Scott Adams, 2006

NÃO SÃO REQUISITOS NÃO FUNCIONAIS:

Esses abaixo, NÃO podem ser considerados como requisitos não funcionais. O motivo é simples, visto que nenhum deles é passível de verificação. Nenhum desses requisitos podem ser medidos, aferidos ou mensurados para verificar sua conformidade.

- O sistema deve ser rápido;
- Não deve corromper dados;

Identificando o tipo do requisito com 2 perguntas

Para não errar nunca na hora de categorizar um requisito como funcional ou não funcional basta responder duas perguntas obrigatórias para garantir a coerência da categorização dos requisitos:

- 1. A característica é sobre O QUE o sistema fará, ou COMO o sistema fará?**
- 2. É possível verificar se requisito não funcional está sendo atendido ou respeitado?**

São duas perguntas muito simples. Muitos analistas de requisitos comentem erros fundamentais durante o levantamento e especificação dos requisitos de um sistema.

Diagrama de Caso de Uso

<https://creately.com/blog/pt/diagrama/tutorial-de-diagrama-de-caso-de-uso/>

O diagrama de caso de uso é um tipo de diagrama comportamental e frequentemente usado para analisar vários sistemas. Eles permitem que você visualize os diferentes tipos de papéis em um sistema e como essas funções interagem com o sistema.

Importância do Diagrama de Caso de Uso

Os diagramas de caso de uso são usados para reunir um requisito de uso de um sistema. Dependendo da sua necessidade, você pode usar esses dados de diferentes maneiras. Abaixo estão algumas maneiras de usá-las.

- Para identificar funções e como os papéis interagem com elas – O propósito principal dos diagramas de caso de uso.

- Para uma visão de alto nível do sistema – Especialmente útil ao apresentar aos gestores ou partes interessadas. Você pode destacar os papéis que interagem com o sistema e a funcionalidade fornecida pelo sistema sem ir profundamente em funcionamento interno do sistema.
- Para identificar fatores internos e externos – Isto pode parecer simples, mas em projetos complexos de grande porte um sistema pode ser identificado como um papel externo em outro caso de uso.

Objetos do Diagrama de Caso de Uso

Os diagramas de caso de uso consistem em 4 objetos.

- Ator
- Caso de uso
- Sistema
- Pacote

Os objetos são explicados com mais detalhes a seguir.

Ator

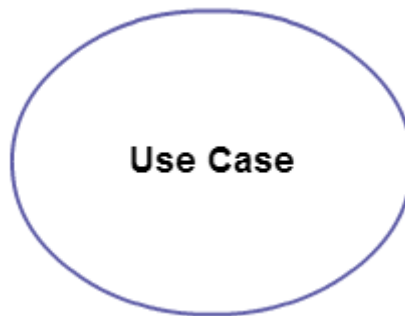
Ator em um diagrama de caso de uso é qualquer entidade que desempenha um papel em um determinado sistema. Pode ser uma pessoa, organização ou um sistema externo e normalmente desenhado como o esqueleto mostrado abaixo.



Actor

Caso de Uso

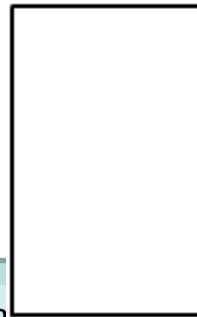
Um caso de uso representa uma função ou uma ação dentro do sistema. É desenhado como um oval e nomeado com a função.



Sistema

O sistema é usado para definir o escopo do caso de uso e desenhado como um retângulo. Este é um elemento opcional, mas útil quando se está a projetando sistemas grandes. Por exemplo, é possível criar todos do casos de uso e depois utilizar o objeto do sistema para definir o escopo coberto pelo projeto. Ou pode até usá-lo para mostrar as diferentes áreas cobertas em diferentes lançamentos.

System



Pacote

O pacote é outro elemento opcional que é extremamente útil em diagramas complexos. Semelhante aos diagramas de classes, os pacotes são usados para agrupar casos de uso. Eles são desenhados como a imagem mostrada abaixo.



Diretrizes do Diagrama de Caso de Uso

Embora os diagramas de caso de uso possam ser usados para vários fins, há algumas diretrizes comuns que você precisa seguir ao desenhar casos de uso.

Estes inclui padrões de nomear, direções de setas, colocação de casos de uso, uso de caixas de sistema e também uso adequado de relacionamentos.

Abordamos essas diretrizes em detalhes em um post de blog separado. Portanto, vá em frente e confira diagrama de caso de uso guidelines.

Relacionamentos do Diagrama de Caso de Uso

Há cinco tipos de relações em um diagrama de caso de uso.

- Associação entre um ator e um caso de uso
- Generalização de um ator
- Estenda o relacionamento entre dois casos de uso
- Incluir relação entre dois casos de uso
- Generalização de um caso de uso

Cobrimos todos esses relacionamentos em um post de blog separado que tem exemplos com imagens. Não entraremos em detalhes neste post, mas você pode verificar os relacionamentos nos diagramas de caso de uso.

Como criar um Diagrama de Caso de Uso

Até agora, você aprendeu sobre objetos, relacionamentos e diretrizes que são essenciais ao desenhar diagramas de caso de uso. Vou explicar os vários processos usando um sistema bancário como exemplo.

Identificação dos Atores

Atores são entidades externas que interagem com seu sistema. Pode ser uma pessoa, outro sistema ou uma organização. Em um sistema bancário, o ator mais óbvio é o cliente. Outros atores podem ser funcionários do banco ou caixa, dependendo do papel que você está tentando mostrar do caso de uso.

Um exemplo de uma organização externa pode ser a autoridade fiscal ou o banco central. O processador de empréstimos é um bom exemplo de um sistema externo associado como um ator.

Identificação dos Casos de Uso

Temos que identificar o que os atores precisam do sistema. Num sistema bancário, um cliente precisará abrir contas, depositar e levantar fundos, solicitar livros de cheques e funções semelhantes. Portanto, todos estes podem ser considerados como casos de uso.

Os casos de uso de alto nível devem sempre fornecer uma função completa exigida por um ator. Você pode estender ou incluir casos de uso, dependendo da complexidade do sistema. Uma vez que você identifica os atores e o caso de uso de nível superior você tem uma ideia básica do sistema. Agora você pode melhorá-lo e adicionar camadas extras de detalhes a ele.

Procure Funcionalidade Comum para usar Incluir

Procure por funcionalidades comuns que possam ser reutilizadas em todo o sistema. Se você encontrar dois ou mais casos de uso que compartilham funcionalidades comuns, você pode extrair as funções comuns e adicioná-las a um caso de uso separado. Então você pode conectá-lo através do relacionamento include para mostrar que ele é sempre chamado quando o caso de uso original é executado. (veja o diagrama para um exemplo).

É Possível Generalizar os Atores e Casos de Uso?

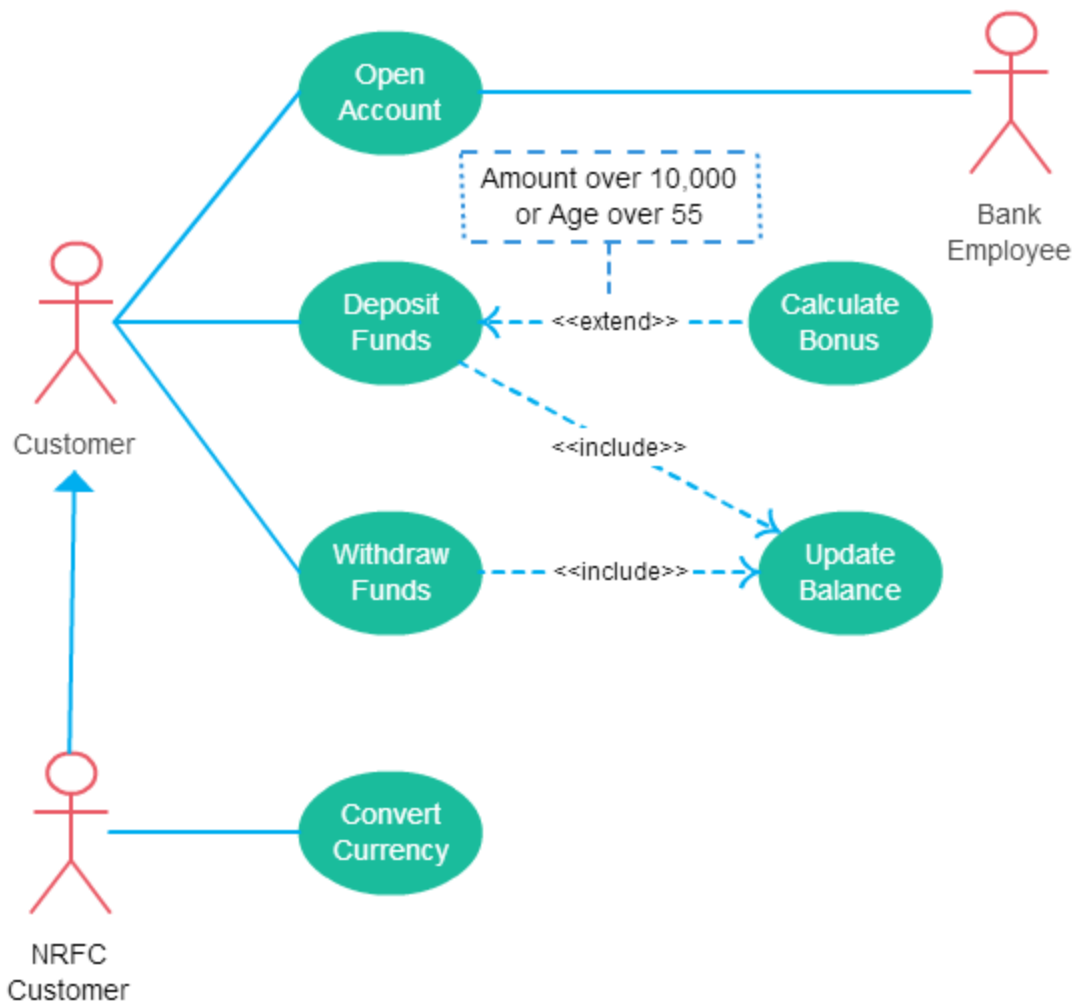
Pode haver instâncias em que os atores são associados a casos de uso semelhantes, enquanto acionam alguns casos de uso exclusivos apenas para eles. Em tais casos, você pode generalizar o ator para mostrar a herança de funções. Você pode fazer uma coisa semelhante para o caso de uso também. Um dos melhores exemplos disso é o caso de uso “Fazer Pagamento” em um sistema de pagamento. Pode ainda generalizá-lo para “Pagar com cartão de crédito”, “Pagar em dinheiro”, “Pagar com cheque”, etc. Todos eles têm os atributos e a funcionalidade de pagamento com cenários especiais exclusivos para eles.

Funções Opcionais ou Funções Adicionais

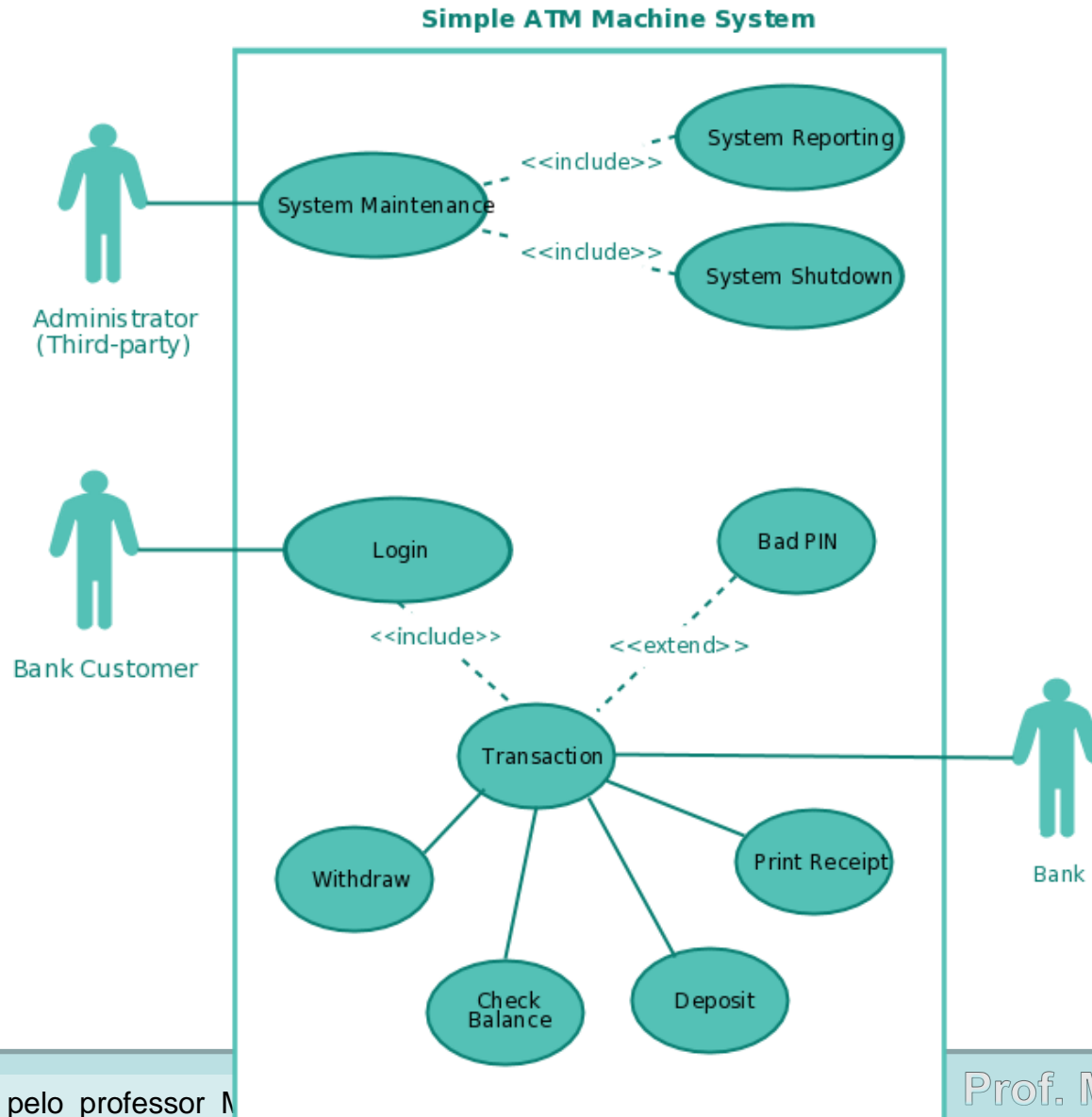
Há algumas funções que são acionadas opcionalmente. Em tais casos, você pode usar o relacionamento extend e anexar uma regra de extensão a ele. No exemplo do sistema bancário abaixo, “Calcular bônus” é opcional e só é acionado quando uma determinada condição é igualada.

A extensão nem sempre significa que é opcional. Às vezes, o caso de uso conectado através do alargamento pode complementar o caso de uso base. A única coisa a lembrar é que de caso de uso base deve ser capaz de executar uma função em seu próprio mesmo que de caso de uso de extensão não é chamado.

CASO DE USO



CASO DE USO



TUTORIAL DE DIAGRAMA DE CASO DE USO

<https://www.youtube.com/watch?v=ab6eDdwS3rA>

ESTRATÉGIAS DE ABORDAGEM

Existem duas principais estratégias de abordagem: **top down** e **bottom up**. Estes termos se referem a métodos utilizados para organizar as informações, ordenando fatos para serem utilizados em diversos campos que necessitem de uma forma de pensamento que facilite a administração de conhecimentos para um maior aproveitando destes, com maior facilidade de processamento e garantindo um sistema de prioridades e hierarquia.

TOP-DOWN:

é um termo cujo significado traduzido é “de cima para baixo” e em sua essência implica em dividir os processos de alguma coisa como a fabricação de um produto ou a criação de uma ideia para, desse modo, utilizar essa fragmentação (ou compartimentação) para tornar a compreensão de cada parte que compõem um todo muito mais fácil de ser realizada.

Cada parte dessa subdivisão, do maior ao menor (ou do mais alto para o mais baixo como sugere o nome) é profundamente estudado e detalhado da forma mais completa possível para que se obtenha um entendimento amplo do assunto que esteja sendo analisado.

BOTTOM-UP:

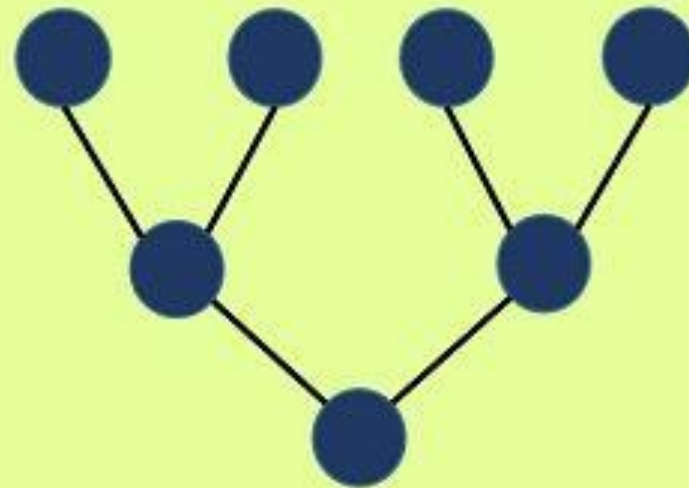
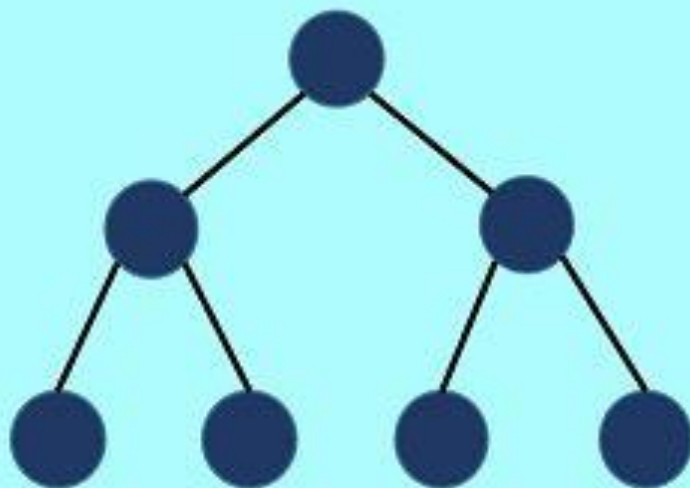
é um termo que traz o significado de uma abordagem realizada de baixo para cima. Ela consiste em um processo de análise e comportamento de informações que utiliza a compreensão de subdivisões dos assuntos para uma percepção mais completa e com uma nova interpretação das partes que formam o todo, ou seja, analisa e descreve os elementos mais básicos para formar um resultado maior.

Um exemplo desta forma de lidar com as informações é como o cérebro recebe somente detalhes de uma história, fragmentos e conforme vai recebendo mais e mais fatos, os junta de forma a obter uma história completa que faça sentido.

Seguindo esta linha de pensamento, as informações começam apenas como detalhes distintos e separados e conforme crescem se combinam para formar algo completo.

Diversas áreas do conhecimento se utilizam destes conceitos a fim de melhor realizar atividades, com mais facilidade e obtenho resultados melhores, sendo uma estratégia de organização valiosa.

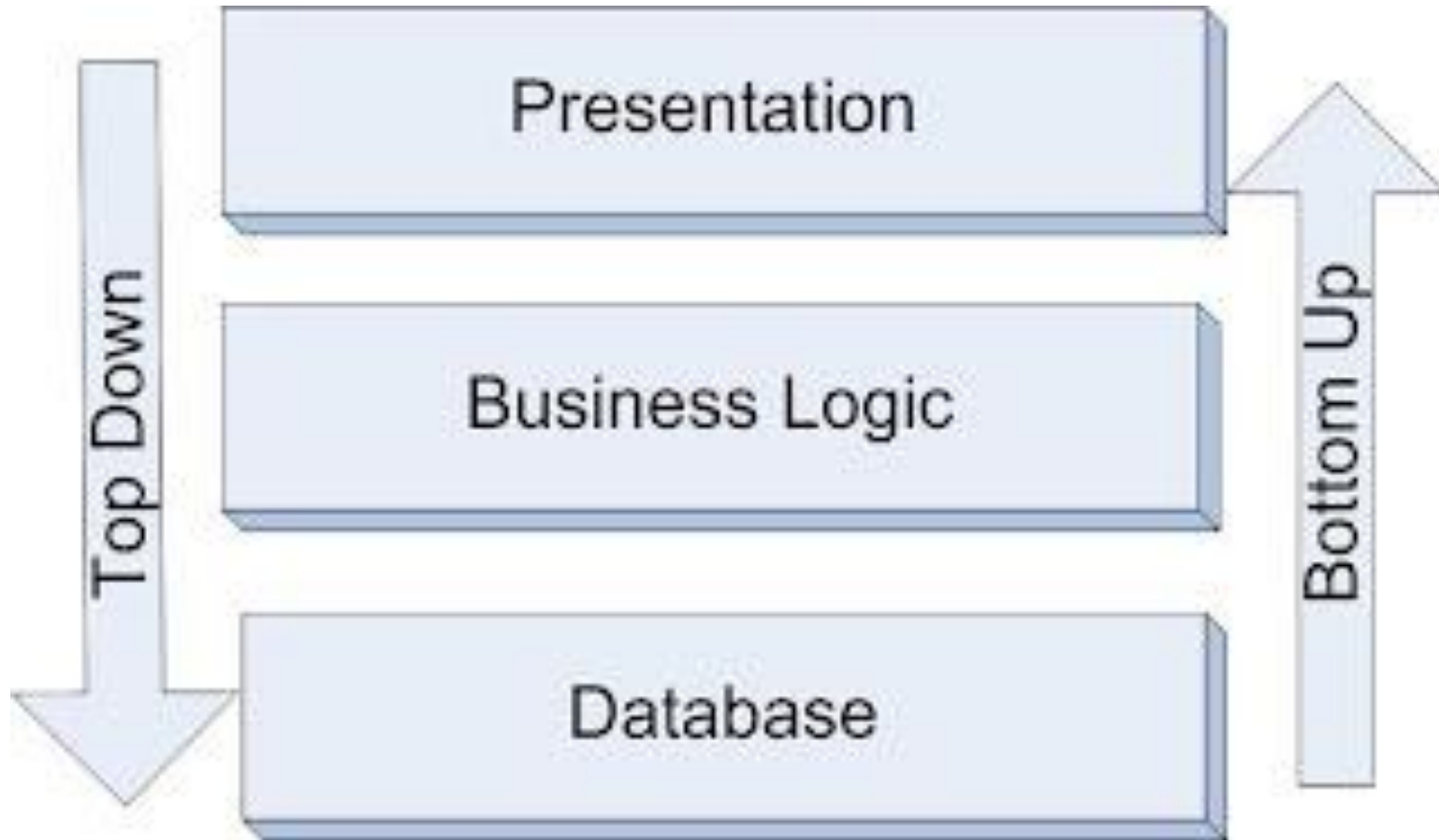
O desenvolvimento de softwares também utiliza os conceito de Top-down e Bottom-up, e são essenciais nesse processo. Eles ajudam no compreensão de sistemas complexos que necessitam de um conhecimento detalhado das partes para seu funcionamento, com testes, usos possíveis e como tudo se encaixa para formar um projeto final.



Top-down Approach

Vs

Bottom-up Approach



MATRIZ SWOT

A Análise SWOT ou Análise FOFA ou FFOA (Forças, Fraquezas, Oportunidades e Ameaças) (em Português) é uma ferramenta utilizada para fazer análise de cenário (ou análise de ambiente), sendo usada como base para Gestão e Planejamento Estratégico de uma corporação ou empresa.

O termo SWOT é uma sigla oriunda do idioma Inglês, e é um acrônimo de:

Forças → Strengths

Fraquezas → Weaknesses

Oportunidades → Opportunities

Ameaças → Threats

OBJETIVOS

- Efetuar uma síntese das análises internas e externas;
- Identificar elementos chave para a gestão da empresa, o que implica estabelecer prioridades de atuação;
- Preparar opções estratégicas: Riscos/Problemas a resolver;
- Fazer o diagnóstico da empresa;
- Fortalecer os pontos positivos, indica quais os pontos devem melhorar, mostra as chances de crescimento, aumentando as oportunidades;
- Deixar em alerta diante de riscos

APLICAÇÃO

Ambiente Interno (Forças e Fraquezas)

- Integração dos Processos
- Padronização dos Processos
- Eliminação de redundância
- Foco na atividade principal

Ambiente Externo (Oportunidades e Ameaças)

- Confiabilidade e Confiança nos dados
- Informação imediata de apoio à Gestão e Decisão estratégica
- Redução de erros

- ✓ As forças e fraquezas são determinadas pela posição atual da empresa e relacionam-se, quase sempre, a fatores internos. Estas são particularmente importantes para que a empresa rentabilize o que tem de positivo e reduza, através da aplicação de um plano de melhoria, os seus pontos fracos.
- ✓ Já as oportunidades e ameaças são antecipações do futuro e estão relacionadas a fatores externos, que permitem a identificação de aspectos que podem constituir constrangimentos (ameaças) à implementação de determinadas estratégias, e de outros que podem constituir-se como apoios (oportunidades) para alcançar os objetivos delineados para a organização.

DE MODO GERAL

- **AMBIENTE INTERNO**

- ✓ Strengths (Força) - Vantagens internas da empresa em relação às empresas concorrentes.
- ✓ Weaknesses (Fraquesa) - Desvantagens internas da empresa em relação às empresas concorrentes.

- **AMBIENTE EXTERNO**

- ✓ Opportunities (Oportunidades) - Aspectos positivos com potencial de fazer crescer a vantagem competitiva da empresa.
- ✓ Threats (Ameaças) - Aspectos negativos com potencial de comprometer a vantagem competitiva da empresa.

EXEMPLOS:

✓ Strengths (Força)

- A união da sua equipe para solucionar problemas
- Quantidade de ativos (imóveis, equipamento moderno, etc.):
- Capitalização barata.
- Localização privilegiada
- Modelo de cobrança com Preços mais competitivos

EXEMPLOS:

- ✓ Weaknesses (Fraquesa)
 - Produto altamente perecível
 - Matéria-Prima escassa
 - Equipe pouco qualificada
 - Tecnologia ultrapassada
 - Processo de entrega lento

EXEMPLOS:

- ✓ Opportunities (Oportunidades)
 - Vai sair uma nova Lei: Desenvolver um produto específico para atendê-la
 - Pode surgir um novo curso: Planejar para os funcionários terem acesso a ele
 - Minha concorrente precisa de ajuda: Podemos fazer uma fusão ou aquisição
 - Ter acesso à uma nova tecnologia: Planejar uma nova linha de produtos
 - Algum produto complementar ao meu ser lançado: Buscar parceria de marketing

EXEMPLOS:

✓ Threats (Ameaças)

- Entrada de um concorrente internacional no mercado
- Pirataria dos seus produtos
- Mudança na Legislação do seu setor
- Escassez de Mão-de-Obra
- Catástrofes naturais/guerras/etc

Figura 7 - Análise SWOT



DIAGRAMA DE GANTT

O diagrama de Gantt é um gráfico usado para ilustrar o avanço das diferentes etapas de um projeto.

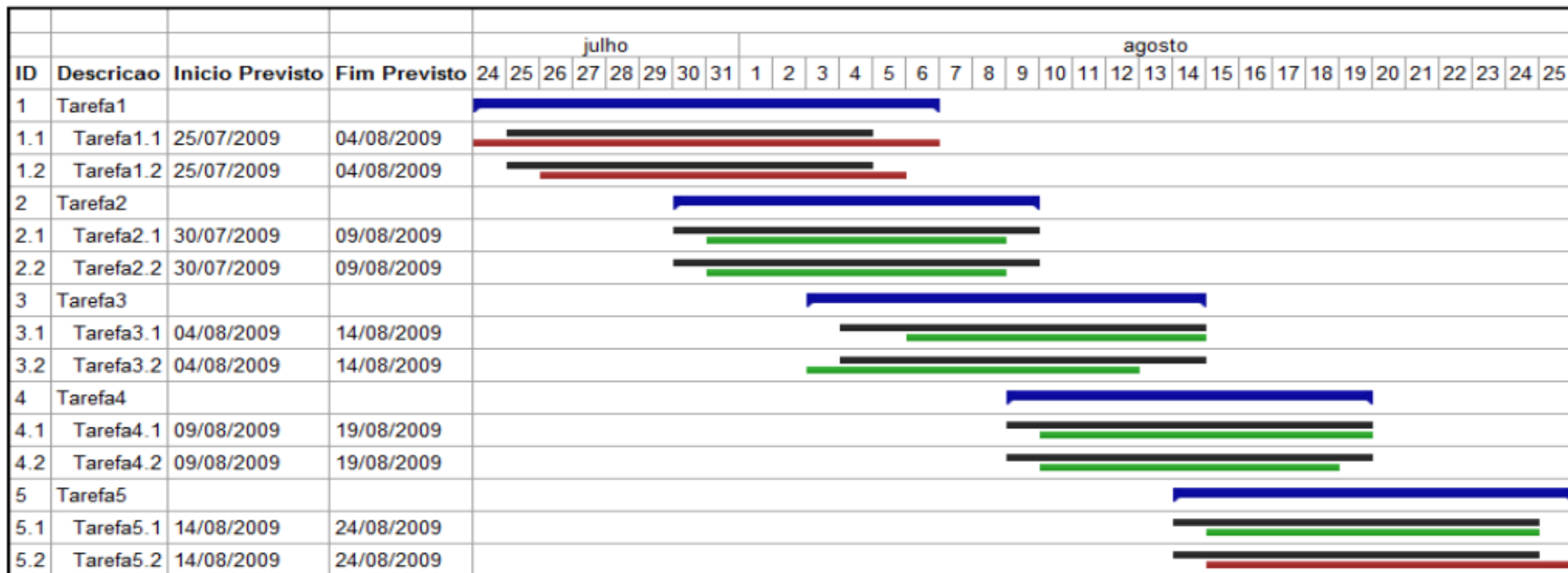
Os intervalos de tempo representando o início e fim de cada fase aparecem como barras coloridas sobre o eixo horizontal do gráfico.

Desenvolvido em 1917 pelo engenheiro mecânico Henry Gantt, esse gráfico é utilizado como uma ferramenta de controle de produção.

Nele podem ser visualizadas as tarefas de cada membro de uma equipe, bem como o tempo utilizado para cumpri-la. Assim, pode-se analisar o empenho de cada membro no grupo, desde que estejam associados, à tarefa, como um recurso necessário ao desempenho dela.

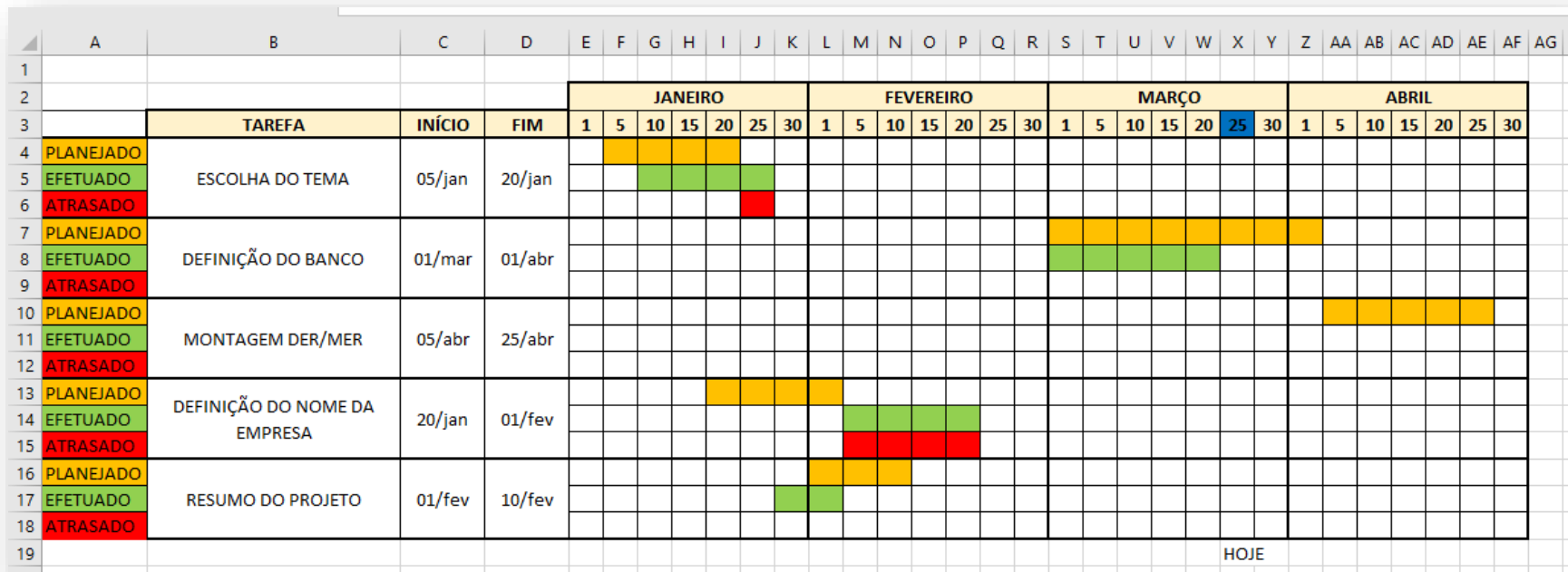


- Período planejado
- Período real de execução sem atrasos
- Período real de execução com atrasos

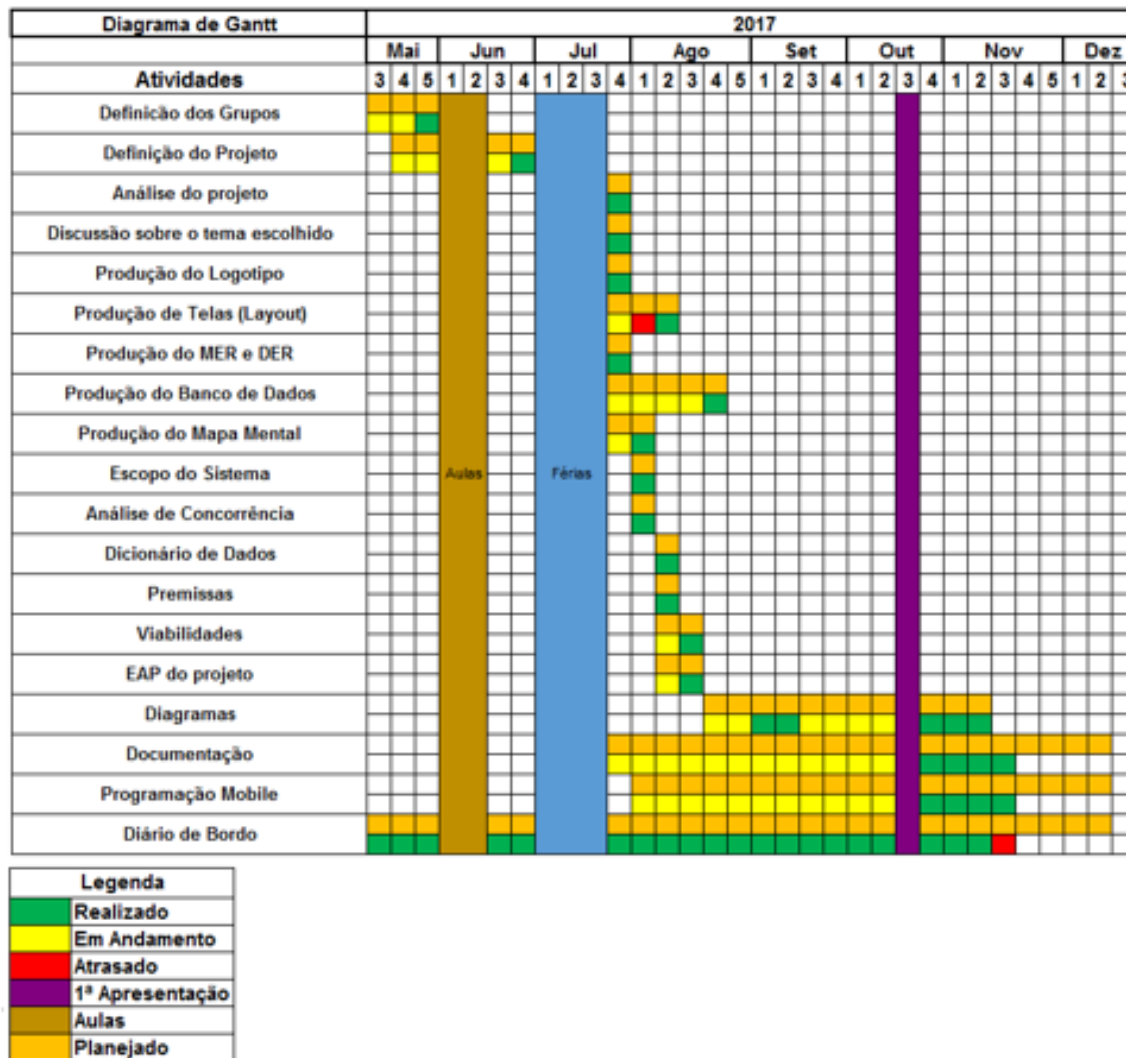


- Período planejado
- Período real de execução sem atrasos
- Período real de execução com atrasos

GANTT



GANTT



MATRIZ DA EQUIPE

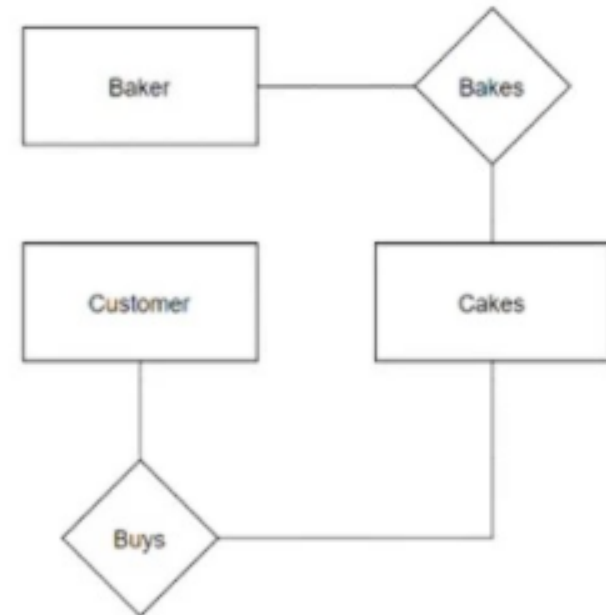
MATRIZ DA EQUIPE

NOME	PERFIL	FUNÇÃO
Asdrubal Pereira	Dinâmico, Pró-Ativo, Desinibido e Simpático	Programador
Iraildes Damásio	Tímida, recatada, simpática e sorridente	DBA
Astolpho Gramado	Falante, desatento e implicante	Documentador

DER/MER



DER - Completo



MER

PLANO DE TESTE

O Plano de Testes é a base para o início de qualquer teste de software. Neste documento são apresentadas as atividades de testes, escopo, tipo de teste objetivo e entre outros.

PLANO DE TESTE						
AUTOR	Ana Paula Bispo Mariano		PROJETO	Buffet Manager		
VERSÃO	1.0		DATA	10/11/2017		
Casos de Uso	Operação	Caso de Teste	Tipo de Teste	Objetivo	Critério de Êxito	Critério de Falha
Manter Funcionário	Cadastrar()	verificar funcionário cadastrado	Teste de Segurança	Verificar se um Funcionário esta cadastrado no sistema	Mensagem de usuário inválido	Tela de Eventos
Armazenar Dados	ConectarMySQL()	Conectar com o banco de dados	Teste de unidade	Verificar se o sistema consegue se conectar com o BD	O sistema é conectado e funciona	falhas na conexão
Manter Funcionário	ListarFuncionario()	Verificar lista de funcionario	Usabilidade	Verificar todos os funcionarios que estão disponives para trabalhar no evento	Funcionário não listado	Funcionário listado
Gerar Relatório	GerarRelatorio()	Verificar relatórios gerados	Caixa - Preta	Verificar se há falhas na geração de relatórios	Relatório gerado corretamente	Relatorio não gerado por falhas

DIFICULDADES ENCONTRADAS

- Quais as dificuldades encontradas na criação do projeto?

Exemplo:

- 1) Utilização de janelas Modal;
- 2) Utilização da ferramenta brModelo;
- 3) Falta de documentação sobre PDO;

QUEBRAS e NUMERAÇÃO

1. Para fazer uma QUEBRA DE PÁGINA, pressione CTRL-ENTER.
2. Para fazer uma QUEBRA DE SEÇÃO, escolha Layout, Quebras e escolha o tipo de quebra desejado.
3. Para inserir numeração na página, escolha Inserir e em seguida Número de Página.
4. Para desvincular da seção anterior, clique em “Vincular com o anterior” em Navegação.

SUMÁRIO

1. Para cada título que você deseja no índice de conteúdo, selecione o texto do título.
2. Vá para **Página Inicial > Estilos** e escolha **Título 1**.
3. Posicione o cursor onde você deseja adicionar o sumário.
4. Acesse Referências > Sumário. e escolha um estilo automático.
5. Se você fizer alterações no documento que afetam o sumário, atualize o sumário clicando com o botão direito no sumário e escolhendo Atualizar Campo.