

## **SOFTWARE DE GERENCIAMENTO DE FINANÇAS PESSOAIS**

Roberto de Vargas Mânica

Faculdades Integradas de Taquara – Faccat – Taquara – RS – Brasil  
manica@aluno.faccat.br

### **Resumo**

Muitas pessoas, enfrentam dificuldades em lidar com o dinheiro, e adequar o suas despesas à receita mensal que possuem. A ferramenta aqui apresentada é uma aplicação para gerenciamento de fianças pessoais, a ser disponibilizada, via WEB, por Instituições Financeiras Bancárias, tomando por base as movimentações da Conta Corrente de seus clientes. O descontrole das finanças pessoais pode levar a inadimplência, esta por sua vez impacta na concessão do credito pessoal. Isto baixa sensivelmente o risco na concessão de empréstimos o que reflete nos juros cobrados.

**Palavras-chave:** Web; livros; publicações.

## **PERSONAL FINANCE SOFTWARE MANAGEMENT**

### **Abstract**

*Many people face difficulties in dealing with money, and tailor your expenses to the monthly income they have. The tool presented here is an application for managing personal expenses , to be made available via WEB , by Banking Financial Institutions , based on the movements of the current account of its clients. The lack of personal finance can lead to defaults, this in turn impacts on the granting of personal credit . This lowers significantly the risk in lending which reflects the interest charged.*

**Keywords:** *Web; books; publications.*

### **1. Introdução**

Diante do acentuado comprometimento financeiro das famílias brasileiras, manter um controle sobre gastos é algo importante e que merece atenção. Anotações diárias sobre gastos e um olhar atento sobre o equilíbrio das despesas com as receitas é fundamental para saúde financeira pessoal e familiar.

Segundo pesquisa do SPC Brasil (set/2012) 44% dos brasileiros já estiveram ou estão impossibilitados de efetuarem compras a prazo por possuírem restrições cadastrais, ou mais popularmente, definido como tendo o nome “sujo”. Este cenário inclui pessoas das classes “A” a “

D”, sendo as duas últimas as que mais sofrem com a situação de inadimplência. Os principais motivos, apontados, são a falta de planejamento financeiro e a dificuldade em avaliar o quanto pagam efetivamente ao fazerem compras parceladas.

As anotações feitas em cadernos foram substituídas pelas facilidades no uso das Planilhas Eletrônicas. Digitando informações sobre nossos gastos diários obtemos uma visão mais ampla que possibilita mensurar nossas despesas e receitas. Na *Internet* encontramos uma infinidade de planilhas, cada qual com suas peculiaridades, que podem ser baixadas e utilizadas em diferentes dispositivos.

O presente trabalho de pesquisa e desenvolvimento tem por objetivo a construção de um software para Gestão de Finanças Pessoais.

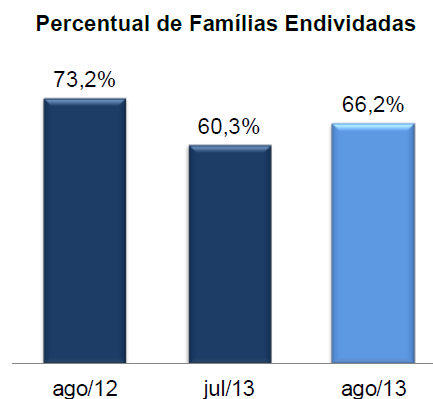
Esta ferramenta é direcionada às Instituições Financeiras e seus clientes. Interessadas na adimplência, este software torna-se importante auxílio na educação financeira de seus correntistas.

Utilizando-se das informações registradas no Banco de Dados, da Instituição esta aplicação disponibiliza ao cliente correntista, informações sobre seus gastos com base na movimentação de sua conta corrente e do cartão de crédito.

## 2. Referencial Teórico

Desde a antiguidade temos relatos sobre a preocupação com o controle financeiro pessoal, como demonstrado em **São Lucas 14, 28**, “**Quem de vós, querendo fazer uma construção, antes não se senta para calcular os gastos que são necessários, a fim de ver se tem com que acabá-la?**” (Bíblia Católica).

Nos dias de hoje podemos verificar o descontrole da população brasileira diante da oferta de crédito, como bem demonstra um estudo da Fecomercio-SP (2012), mesmo com o aumento sistemático da taxa de juros, há um crescimento significativo de novas operações para consumidores já endividados.



Fonte: CNC  
Elaboração: Assessoria Econômica/Fecomercio-RS

Figura 1 – Fonte Fecomercio RS (Set 2013)

## Realiza algum tipo de planejamento financeiro?

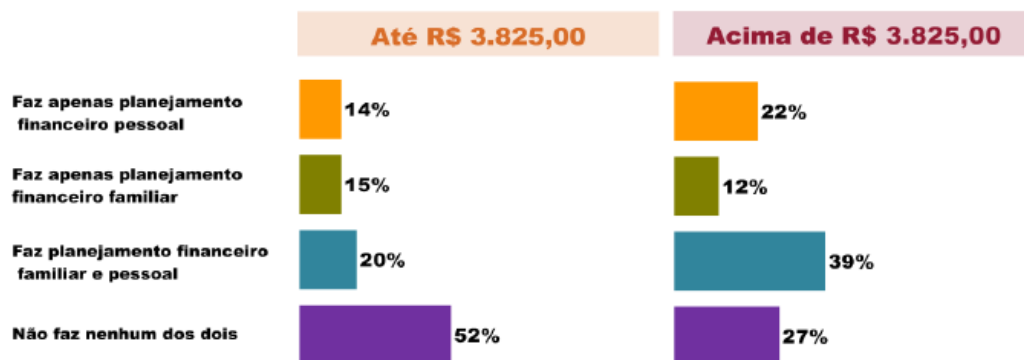


Figura 2 - Fonte: SPC Brasil (Set/2012)

A frase “**A humanidade é produto do desejo e não da necessidade**” (Gaston Bachelard – filósofo e ensaísta francês), segundo Macedo Junior (2007), parece muito bem externar o motivo do consumismo que estamos vivenciando na atualidade. As campanhas publicitárias fazem de tudo para transformar o desejo não só num simples querer mas em necessidade. As famílias têm desejos e necessidades, nelas são diferentes, em função da cultura do meio em que vivem, da sua renda etc.

O controle sobre seus gastos pode significar uma vida confortável e estável, comprar a casa própria, iniciar o próprio negócio, projetar a aposentadoria, uma educação melhor para os filhos e proteger seu patrimônio.

As despesas de consumo variam por vários fatores, área urbana, rural, ou orientação religiosa, segundo a Pesquisa de Orçamento Familiar - POF IBGE.



Distribuição percentual das despesas de consumo da área urbana, por composição da família, segundo grupos de despesa selecionados - Brasil - período 2008-2009

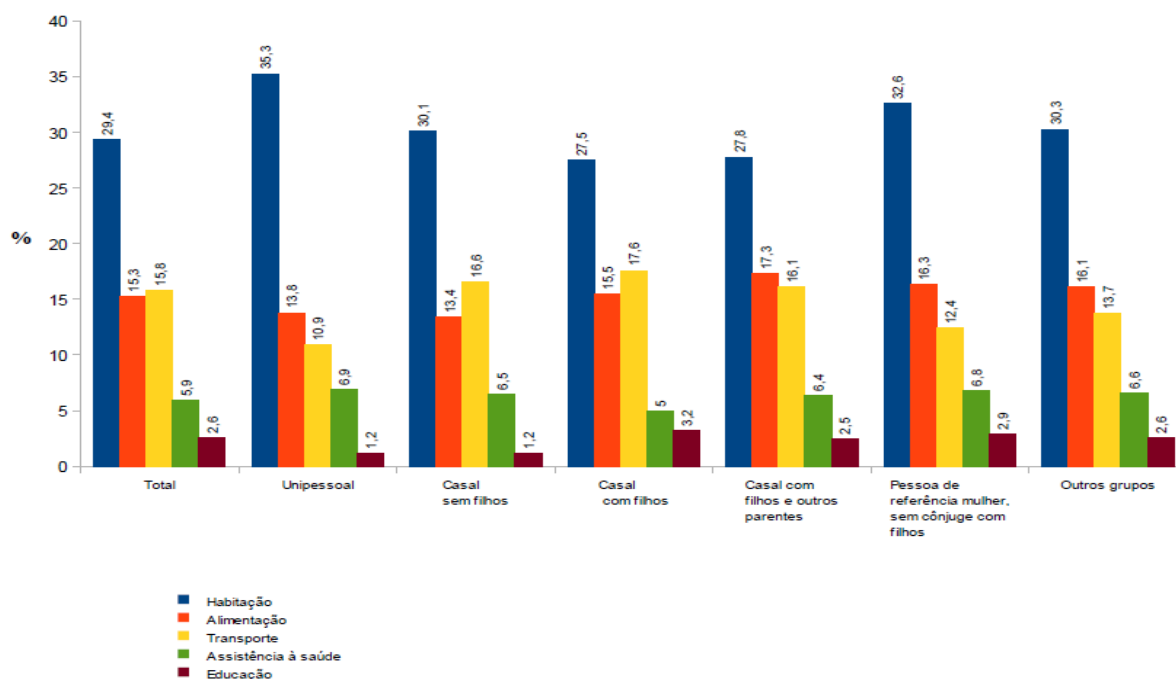


Figura 3 – Fonte: IBGE (2013)

A organização das despesas e receitas demonstra a saúde financeira e quais são os hábitos de consumo. Esta “fotografia” possibilita a correção de rumo a ser tomada a fim de que as despesas estejam condizentes com as receitas. Pode-se listar receitas e despesas por categorias, como o modelo proposto por Macedo Junior, em *A Árvore do Dinheiro* (2007).

<b>RECEITAS</b>	
Salários	RS.....
Receitas extraordinárias	RS.....
<b>Subtotal</b>	RS.....
<b>DESPESAS</b>	
<b>MORADIA</b>	
Aluguel/impostos	RS.....
Condomínio/prestação da casa	RS.....
Conta de luz/água/gás	RS.....
Telefone	RS.....
Consertos/manutenção	RS.....
<b>ALIMENTAÇÃO</b>	
Supermercado	RS.....
Feira/sacolão	RS.....
<b>TRANSPORTE</b>	
Prestação do carro/seguro	RS.....
Combustíveis/estacionamento	RS.....
Impostos	RS.....
Ônibus/metrô/trem	RS.....
<b>SAÚDE</b>	
Plano de saúde	RS.....
Médicos/dentistas	RS.....
Farmácia	RS.....
<b>EDUCAÇÃO</b>	
Mensalidades escolares	RS.....
Cursos extras – idiomas/computação	RS.....
<b>LAZER/INFORMAÇÃO</b>	
Academia/programas culturais	RS.....
Jornais/revistas	RS.....
TV por assinatura/Internet	RS.....
<b>OUTROS GASTOS</b>	
Vestuário	RS.....
Cuidados pessoais	RS.....
<b>RESERVA PARA GASTOS FUTUROS</b>	
Impostos	RS.....
Escala	RS.....
Viagem	RS.....
<b>Subtotal</b>	RS.....
<b>SALDO (Receita total – Despesas total)</b>	RS.....

Figura 4 - A Árvore do Dinheiro

Poucas pessoas têm o hábito de anotar em papel seus gastos diários, com o uso das planilhas eletrônicas estes controles ficaram mais fáceis. Mesmo assim, torna-se necessária a disposição sistemática de registrar despesas, anotar de alguma forma os valores e identificar a natureza dos gastos para poder então fazer os lançamentos devidos.

O uso dos dispositivos móveis, tais como *Notebooks*, *Tablets*, *Smartphones*, em muito auxilia na atividade de anotar e controlar os gastos diários para compor e controlar as despesas mensais.

Todos, de alguma forma, precisam da inserção manual de dados além de dependerem sempre da lembrança do usuário em fazê-los.

Diante de um cenário de aquecimento no consumo, e endividamento, pode-se verificar uma crescente disponibilização de livros sobre economia pessoal. Nota-se também, nestas publicações, que as mesmas discorrem de um modo geral sobre como poupar e investir em fundos e clubes de investimento, fundos de previdência, investimentos em renda fixa, imóveis etc. Mas muito pouco sobre economia doméstica ou pessoal.

Os bancos, por sua vez, têm direcionado suas atenções a esta situação de endividamento crescente de seus clientes demonstrado pela disponibilização, em seus sites, de informações sobre Educação Financeira. Estas informações chegam em forma de aulas, vídeos, exercícios, *podcasts* etc.

No caderno de Economia de O Globo (27/05/2013), segundo o BC (Banco Central), no final do primeiro trimestre deste ano o nível de endividamento chegou a 43,99%. Ou seja, a família brasileira deve aos bancos quase a metade do que ganha anualmente.

O Sistema Bancário em muito pode ser comparado ao comercial no que tange a busca de clientes e sua fidelização. O cliente, nos dias de hoje, assumiu um papel importante para a instituição haja vista a utilização cada vez maior de Sistemas de CRMs (*Customer Relationship Management*), BI (*Business Intelligence*) a fim de melhor compreender suas necessidades e novas formas de agregar rentabilidade no fortalecimento deste relacionamento.

Diante de uma concorrência acirrada, a disponibilização de uma aplicação que possa auxiliar o cliente na construção de um orçamento pessoal e familiar condizente com sua receita é mais uma ferramenta que poderá auxiliar a instituição na busca de um relacionamento com base na adimplência.

Tendo como pano de fundo uma situação, costumeira, de descontrole financeiro de uma boa parcela da população brasileira, conforme citado anteriormente, a disponibilização de ferramenta de gestão, automatizando os processos de atualização de despesas e receitas possibilitando a interação do usuário, se mostra uma ferramenta útil .

Este Sistema visa, sob este cenário, prover uma aplicação, de acesso em ambiente WEB, à clientes correntistas de instituições bancárias com a finalidade de controle financeiro pessoal .

Uma aplicação, com acesso por dispositivos móveis, para Gestão de Finanças Pessoais, a partir da movimentação financeira registradas nos Bancos de Dados e realizada por:

- (I) Cartão de Débito;
- (II) Cartão de Crédito;
- (III) Débitos autorizados por convênio;
- (IV) Juros Cheque Especial;
- (V) Tarifas, Empréstimos;
- (VI) Seguros, Previdência Privada e outros débitos;
- (VII) Créditos diversos;

### **3. Metodologia**

O software proposto, é voltado para um ambiente *WEB*, residente em um provedor através

do qual se desenvolverá simulações de acessos de clientes em atividades ativas e ou passivas.

A base de dados, armazenada no mesmo provedor, é construída pelo registro das movimentações financeiras, da conta corrente e do cartão de crédito, do cliente ao longo de um período.

Pelos critérios estabelecidos o software indicará se os valores de cada despesa estão dentro dos parâmetros (percentuais), os quais poderão ser customizados pelo cliente.

Contando com as informações disponibilizadas pelo sistema acrescidas de outras pelo cliente o software se tornará importante ferramenta de gerenciamento financeiro pessoal e familiar.

O desenvolvimento de *software* é um processo, este por sua vez pode ser entendido por um diálogo, um aprendizado, uma interação entre usuário e projetista. A materialização de um *software* é o resultado de um processo, que reuniu conhecimento interativo, organizado e filtrado.

O processo é roteiro, organização, controle, um conjunto de ações, tarefas alocadas dentro de um modelo ou metodologia.

Pressmann (2011), defende uma metodologia genérica para engenharia de *software* estabelecida por cinco atividades: comunicação, planejamento, modelagem, construção e entrega.

Estas cinco atividades são desenvolvidas sob a forma de fluxo, este são caracterizados em cinco outros tipos: linear, iterativo, evolucionário e paralelo.

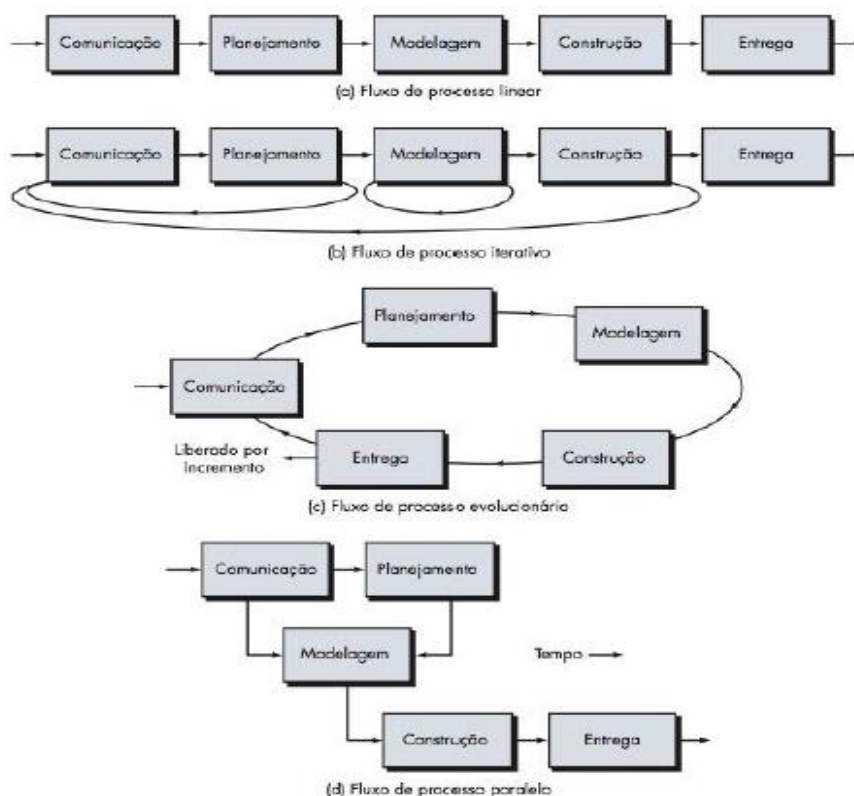


Figura 5 - Fonte: Pressmann (2011) 7ª Edição

Os modelos de processos não são abordagens definitivas de processo de *software*, mas sim uma representação abstrata deste. Representam um processo sob uma perspectiva particular, podem representar diferentes formas de desenvolvimento dando uma ênfase diferente ao desenvolvimento das atividades bem como ao fluxo de processo.

O Modelo Cascata, sugere uma abordagem sequencial, linear e sistemática, em um sentido único iniciando pelo levantamento de dados junto ao cliente e finalizando com a implantação e a manutenção do mesmo durante sua vida útil, Pressmann (2011).

No Modelo Incremental as atividades são desenvolvidas de forma escalonada, sequencial e linear, e com entregas antecipadas de funcionalidades ao usuário, possibilitando sugestões para implementações nas novas versões a serem apresentadas. Assim ao longo do processo de desenvolvimento o sistema recebe melhorias, estas frutos de requisições do cliente e novas percepções da equipe, Pressmann (2011).

No Modelo Evolucionário, utiliza-se dos paradigmas de Prototipação e Espiral, levando a uma atividade cíclica, que numa versão inicial pode apresentar alguns resultados concretos para o cliente como telas e *layouts*. Possibilita a versão de um protótipo com algumas funcionalidades que será entregue ao cliente. Neste período inicial das atividades de implantação, pelo uso do cliente surgem novas necessidades de melhorias. A equipe frente à novas requisições retorna à análise e implementação das novas regras de negócio, Pressmann (2011).

O Modelo Espiral por sua vez agrega a iteratividade da prototipação e a natureza controladora e sistemática do modelo cascata. Desta forma possibilita um desenvolvimento rápido com versões cada vez mais completas da aplicação, Pressmann (2011).

No Modelo Concorrente, as tarefas trafegam por diversos estados, cada estado representa o estágio em se encontra. As diversas tarefas do projeto são alocadas à estados segundo o estágio de desenvolvimento, ou seja a atividade de Modelagem poderia estar no estado de Revisão enquanto a Construção poderia se encontrar em um estado diferente como Em Exame, segundo Pressmann (2011).

Sob o paradigma de reutilização, do Modelo de Processo Especializado Baseado em Componentes, os módulos ou componentes de software podem reduzir o tempo do ciclo de desenvolvimento das aplicações. Este conjunto de uma ou mais classes orientadas à objeto executam parte da solução de uma requisição do software Pressmann (2011).

Utilizando uma abordagem focada na Sintaxe e Semântica Formal este método busca uma redução de erros de escrita do código. A medida que seguem um padrão, os envolvidos no projeto, reduzem o tempo consumido no desenvolvimento e reduzem o retrabalho que representa custos, Pressmann (2011).

O desenvolvimento de *software*, sob a égide do paradigma Orientado à Aspecto, tem por finalidade atender as demandas refletidas pelas restrições cruzadas. Restrições estas vinculadas a alguns níveis de exigências requeridas pelos clientes, em *softwares* cada vez mais sofisticados. Hoje poucos *softwares* não atendem a regras de negócios que busquem segurança, escalabilidade, transparência, tolerância à falhas e replicação de dados Pressmann (2011).

Na defesa de um processo iterativo, incremental, o Processo Unificado, volta o olhar para a comunicação com o cliente e a visão deste sobre a aplicação. Acolhe o que de melhor se apresenta, em termos de recurso, dos processos tradicionais, Pressmann (2011).

Os profissionais imprimem, no desenvolvimento de uma aplicação, suas características pessoais, ou seja, uma metodologia própria.

O PSP, Processo de *Software* Pessoal, atribui ao profissional a responsabilidade por implementar o controle sobre o projeto que está desenvolvendo. Controle sobre tudo que está produzindo, identificação de falhas e erros, compreensão sobre os mesmos para poder evitar futuras ocorrências, Pressmann (2011).

O objetivo do Processo de Software em Equipe, TSP, segundo Watts Humphrey, (Hum 98) – Pressmann (2011), “é criar uma equipe de projetos autogerida”, ou seja, que se organize para produzir aplicações de alto nível. Esta equipe tem conhecimento sobre suas metas e objetivos, atribui para seus membros papéis e responsabilidades, monitora produtividade e qualidade. Define padrões e identifica processo de equipe que seja compatível com o projeto a ser desenvolvido, acompanha e gerencia.

Entre os principais obstáculos ao desenvolvimento de um sistema estão: buscar uma descrição precisa do problema e descobrir um algoritmo para resolvê-lo, Brooks (7ª Edição).

Estes problemas podem ser resolvidos em ambientes mais ou menos adaptáveis às características humanas ou não. A interface desta adaptação está a cargo das linguagens de programação.

Brooks (7ª Edição), aborda as diferentes linguagens através de uma escala linear, de gerações, conforme um distanciamento do usuário do ambiente de máquina.

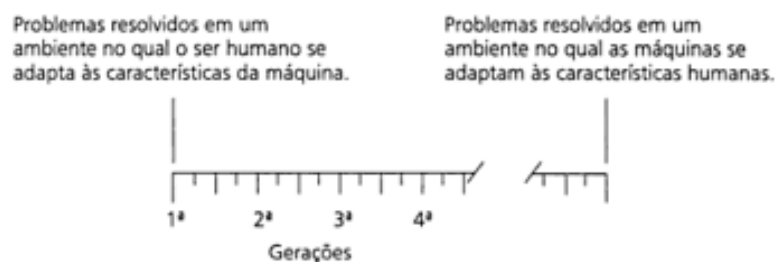


Figura 6 - Brooks (7ª Edição)



As diferentes visões estruturadas, sobre a forma de construir a solução de um problema, os paradigmas, impulsionaram o desenvolvimento de várias linguagens. Estes caminhos surgiram e progrediram de forma independente como mostrado a seguir.

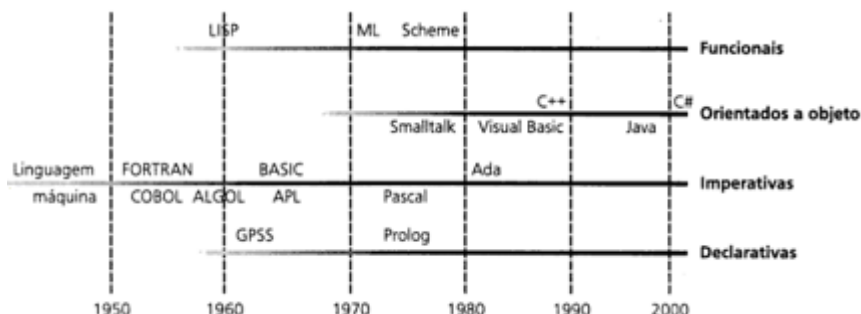


Figura 7 - Brookshear 7ª Edição.

O Paradigma Imperativo ou Procedimental sugere o processo como uma sequência de instruções que quando executadas manipula dados a fim de obter o resultado desejado. Em outras palavras a abordagem do problema sob este paradigma sugere a busca por um algoritmo, para a solução expresso sob a forma de uma sequência de instruções, Brookshear (7ª Edição).

Segundo Brookshear (7ª Edição), o Paradigma Declarativo contextualiza a solução de um problema por uma descrição precisa do mesmo, ao invés de descobrir um algoritmo para resolvê-lo. O esforço está concentrado na implementação de um algoritmo para solucionar problemas em geral. Desta forma para solucionar um problema basta apresentá-lo de forma compatível com o algoritmo desenvolvido.

É um conjunto de “caixas pretas” inter-relacionadas que recebem entradas e produzem saídas que serão fontes de informações para outras até a solução final do problema. Os matemáticos, segundo Brookshear (7ª Edição), nomeiam tais caixas como funções, o que deu origem ao nome. A solução se dá pela construção de funções mais elaboradas a partir de funções primitivas, ou seja, aninhamento de funções simples, que dão a esta forma de codificação do nome de Paradigma Funcional.

Os dados são a essência de um sistema, sob este olhar o Paradigma de Orientação à Objetos atribui a eles a importância da solução de um problema, Filho (2010).

A Orientação a Objetos descreve objetos físicos do mundo real através de entidades denominadas objetos. O objeto é uma unidade autônoma que contém seus próprios dados, a interação entre eles realiza uma tarefa e o conjunto de execuções ordenadas solucionam um problema.

Segundo Bezerra, é uma técnica para modelagem que reduz a diferença semântica entre a realidade modelada e os modelos construídos.

Como o próprio nome diz, o Paradigma Concorrente, faz uso da execução de várias tarefas de forma simultânea e interativa, podendo ser implementadas como programas independentes ou através de um conjunto de *threads* contidas em um único programa. Com isto temos uma forma diferente para o equacionamento de problemas, além de aumentar a *performance* do sistema, pela quantidade de tarefas sendo executadas em um mesmo intervalo e tempo.

#### 4. Arquitetura de Software

Atualmente a grande maioria de sistemas utilizados em grandes computadores são Sistemas Distribuídos. Neste sistema as informações, em processamento, não ficam restritas à uma única máquina, ao contrário, elas são distribuídas e processadas em computadores diferentes.

Neste ambiente podem ser utilizadas diferentes arquiteturas, tais como Arquitetura de Multiprocessadores, Arquitetura Cliente-Servidor, e Arquitetura de Objetos Distribuídos, segundo Sommerville (8ª Edição).

No modelo, **Arquitetura de Multiprocessadores**, de Sistema Distribuído, os processos podem ser executados em processadores distintos, mas não necessariamente. A utilização de vários processadores agiliza o desempenho do sistema bem como sua capacidade de recuperação.

A alocação dos processos para os processadores pode ser predefinida ou ser gerenciado por meio de um *dispatcher* que decidirá a distribuição para os processadores.

Na Arquitetura Cliente-Servidor temos um conjunto de servidores que presta serviço e um conjunto de clientes que usa estes serviços. Para tanto os clientes precisam estar cientes dos servidores disponíveis, na maioria das vezes eles desconhecem quem são os outros clientes, Sommerville (8ª Edição).

Segundo Sommerville, a arquitetura mais simples é chamada de Cliente-Servidor de duas camadas, onde a aplicação é organizada como um ou mais servidores e seus clientes. Esta ainda pode ter duas formas Cliente-magro e Cliente-gordo, no primeiro caso o processamento da aplicação e o gerenciamento de dados são efetuados no servidor, no segundo o servidor é responsável somente pelo gerenciamento de dados.

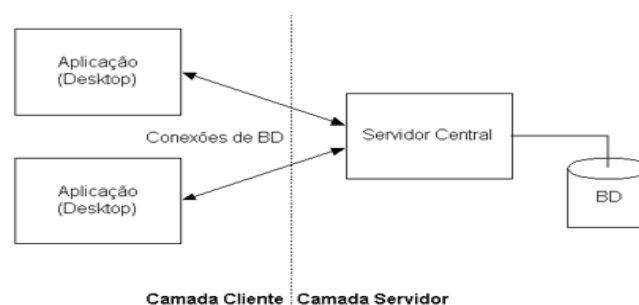


Figura 8 - Arq. Cliente-servidor de duas camadas

Na arquitetura de três camadas processadores diferentes são responsáveis pela apresentação, o processamento da aplicação e o gerenciamento de dados.

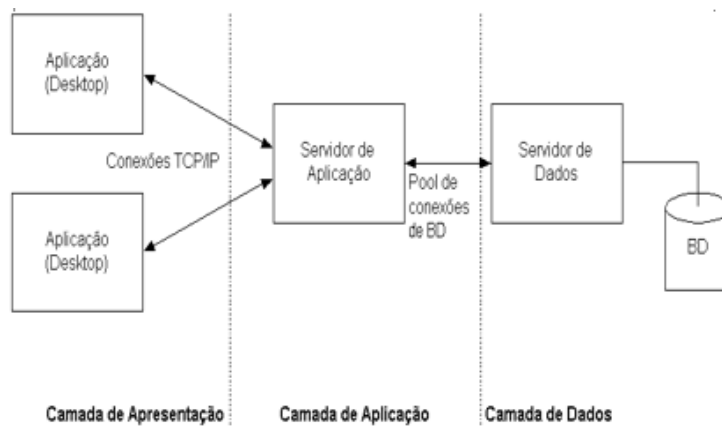


Figura 9- Arq. Cliente-servidor de três camadas

Sommerville (8ª Edição)., na abordagem de Arquitetura de Objetos Distribuídos , demonstra que os servidores podem atuar como clientes de outros servidores, mas não solicitam serviços de clientes, os clientes por sua vez devem saber quais os servidores e os serviços oferecidos por eles

Em uma abordagem mais ampla remove-se a figura de cliente servidor passando para uma arquitetura de objetos distribuídos, desta forma os objetos propiciam uma interface para os serviços fornecidos.

## 5.0 Linguagens

As linguagens são meios de expressão e comunicação, as linguagens naturais facilitam a troca de ideias, mensagens, conhecimento entre pessoas. As de programação são interações entre pessoas/pessoas, pessoas/máquinas e vice-versa. Por sua vez a linguagem obedece a princípios de Sintaxe e Semântica, a primeira especifica a forma correta da escrita a segunda o significado da expressão.

Na linguagem computacional temos um “sotaque” o Paradigma, o primeiro é uma maneira particular de pronuncia, o segundo de forma análoga, identifica um padrão de resolução de problema

A evolução, nas últimas décadas, aponta para alguns tipos distintos:

Programação Imperativa, uma das características deste modelo é o armazenamento de variáveis junto ao código. Entre as que mais se destacam são *Cobol*, *Fortran*, *C*, *Ada* e *Perl*.

Orientada à Objetos, é uma coleção objetos, com seus atributos, que se tornam ativos pela troca de mensagens, entre as Linguagens Orientada à Objetos temos *Smalltalk*, *C++*, *Java*, *C#*, *PHP*.

O entendimento de um problema como uma coleção de funções matemática, é o diferencial em relação às outras linguagens, com exemplos deste paradigma temos *Lisp*, *Scheme*, *Haskell* e *ML*.

A programação Lógica, declarativa ou baseada em regras é modelada pela proposta de um conjunto de regras ou restrições sobre determinado problema. Ou seja, a busca do resultado que um programa deve ter e não como deve ser obtido. A principal linguagem é deste paradigma é *Prolog*, TURCKER (2007).

O PHP, *Hypertext Processor*, é uma linguagem de programação, interpretada, utilizada para desenvolvimento *WEB* podendo ser inserida dentro do Código HTML. É *OpenSource*, tendo um sem número de bibliotecas disponíveis.

É simples, oferece muitos recursos ao programador, seu código é executado dentro do servidor gerando o HTML que é enviado ao cliente, segundo *The PHP Group*, em sua página na Internet.

Na *WhitePaper* de *The Irish PHP UserGroup*, temos alguns dados que demonstram a utilização desta linguagem, dentre a comunidade de desenvolvedores.

A partir de 2007 o PHP foi utilizado em 20.917.850 nomes de domínio e 1.224.183 endereços de IPs, a comunidade de PHP está estimada em 4.500.00 desenvolvedores, estima-se o mercado de desenvolvimento em 4,8 bilhões de *Euros*.

Segundo o mesmo documento, há vários anos a plataforma do PHP não registra alertas de segurança, sendo a mais confiável plataforma para soluções de Internet. Isto explica o desenvolvimento das páginas do *Facebook* e da *Wikipédia*.

Diante do exposto, recaiu a escolha desta linguagem, para o desenvolvimento deste projeto, utilizando o paradigma Procedural, sob o abrigo da teoria dividir para conquistar, em vista da modularização em blocos de instruções para a solução do problema.

Como ferramenta para desenvolvimento optou-se pela utilização do *NetBeans IDE*, versão gratuita de código aberto amplamente utilizada.

A interação do usuário com o Sistema de Gestão de Finanças Pessoais, se dará por uma interface *WEB*, para o projeto e construção da mesma serão utilizados os recursos, sugeridos por DICKETT, JON (2010) em seu livro *Introdução à Programação WEB com XHTML e CSS*.

No desenvolvimento da interface serão explorados os recursos da linguagem, *XHTML* (*Extensible Hypertext Markup Language*) para estruturação da página, a *CSS* (*Cascading Style Sheets*) com o objetivo de controle da aparência e *JavaScript* para adicionar a interatividade desejada à ferramenta de interação.

A *IDE Dreamweaver*, que oferece suporte em PHP, será utilizada para edição do *HTML*, na edição de imagens da página se recorrerá ao software *Fireworks*, e *Flash CS4* como ferramenta para fornecer os recursos de interatividade.

## 6. Sistema de Gerenciamento de Banco de Dados - SGBD

“Banco de dados é uma ferramenta para coletar e organizar informações”, segundo a *Microsoft*, à medida que o número de dados aumenta, crescem as redundâncias e inconsistências dos mesmos.

Os *SGBDs* são constituídos por um conjunto de arquivos e programas inter-relacionados que visam gerir grandes volumes de informações, definindo estruturas de armazenamento, segurança na criação e manipulação de dados.

Segundo o site da empresa de consultoria austríaca *SOLID IT*, no *ranking* dos dez Bancos de Dados, os cinco mais populares são *Oracle*, *MySQL*, *Microsoft SQL Server*, *PostgreSQL* e *DB2*.

Sob a base do Banco de Dados estão os dados descritos em três diferentes grupos: modelos lógicos com base em objetos, modelos lógicos com base em registros e modelos físicos.

204 systems in ranking, November 2013

Rank	Last Month	DBMS	Database Model	Score	Changes
1.	1.	<a href="#">Oracle</a> #	Relational DBMS	1617.19	+33.35
2.	2.	<a href="#">MySQL</a> #	Relational DBMS	1254.27	-77.07
3.	3.	<a href="#">Microsoft SQL Server</a> #	Relational DBMS	1234.46	+27.45
4.	4.	<a href="#">PostgreSQL</a> #	Relational DBMS	190.83	+13.82
5.	5.	<a href="#">DB2</a> #	Relational DBMS	165.90	-9.93
6.	6.	<a href="#">MongoDB</a> #	Document store	161.87	+12.40
7.	7.	<a href="#">Microsoft Access</a> #	Relational DBMS	141.60	-0.89
8.	8.	<a href="#">SQLite</a> #	Relational DBMS	78.78	+0.90
9.	9.	<a href="#">Sybase</a> #	Relational DBMS	77.75	+4.09
10.	10.	<a href="#">Teradata</a> #	Relational DBMS	60.12	+5.70

Figura 10 – Fonte: SOLID IT

No livro, Sistema de Banco de Dados, SILBERSCHATZ E SUDARSHAN (1999), encontra-se referência à diversos modelos tais como Lógico com Base em Objetos, usados na descrição de dados no nível lógico. Modelo Relacional, usando tabelas para representar os dados e seus relacionamentos. Redes, os dados são representados como um conjunto de registros tendo *links* para representar suas relações, Hierárquico, similar ao de Redes com seus registros organizados em árvores.

Segundo Date, no Modelo Relacional, os dados são percebido em forma de tabelas para o usuário, e operadores que geram novas tabelas a partir das primeiras abrindo um leque de ligações, relacionamentos entre as mesmas.

Os relacionamentos, segundo Heuser (2004), podem ser traduzidos por associações entre estes objetos, tabelas.

No Modelo Orientado a Documentos os dados não são armazenados sob estruturas rígidas como tabelas mas sim sob a forma de documentos vagamente definidos. Esta estrutura flexível possibilita incluir novos atributos aos documentos individuais, sem a necessária modificação dos outros. Diferentemente de bancos relacionais que necessitam mudar a definição da própria tabela com consequentes mudanças em todos os registros.

Ao invés de armazenar os dados relacionados em uma área diferente, o banco de dados, integra, a referência, ao próprio documento. Esta diferença possibilita uma maior rapidez de acesso, visto que de outra forma torna-se necessária uma consulta adicional, como afirma a *IBM*.

No Banco de Dados Orientado a Objetos, os objetos, ao contrário dos transientes das linguagens orientadas a objetos, podem ter sua existência estendida passando à objetos armazenados de forma permanente. Ou seja, sua existência é mantida, passando à objetos persistentes, podendo ser compartilhados por outros programas.

Os dados podem ser facilmente identificados e acessados, pois mantem sua integridade e identidade, e não perdem a correspondência direta entre o banco de dados e o mundo real, segundo Navathe e Elmasri.

O Modelo Objeto-Relacional surge com algumas das características dos modelos relacionais, mas voltado a aplicações orientadas a objeto.

Navathe e Elmasri expõe este modelo como um facilitador para desenvolvedores uma vez que o mesmo fornece níveis altos de abstração de modo a seus utilizadores poderem manipular objetos de aplicação em vez de tê-los de construir a partir de dados relacionados.

Os atributos, relacionamentos e métodos do objeto podem ser identificados no modelo de dados, além do que as declarações de tipos de objetos podem ser reutilizadas por herança, o que reduz esforço e tempo de desenvolvimento.

Banco de Dados Oracle com arquitetura para vários usuários em ambiente de *cloud*, possui recursos para gerenciamento de processamento de transações, *big data* e cargas de trabalho de *data warehouse*.

Possui, segundo o fabricante, diversos recursos para proteção de dados contra ameaças, minimiza falhas humanas, de sites, redução de tempo de inatividade e continuidade de aplicativos.

Consolidação de vários bancos de dados, pela ferramenta *Oracle Multitenant*, de forma rápida e segura, compactação e divisão de dados em camadas.

Com estrutura física e lógica separadas permite o gerenciamento físico dos dados não afetando o acesso à estrutura lógica dos mesmos, RAMALHO (2005).

Desde sua criação a mais de trinta anos a Oracle produz produtos para a área de TI, desde servidores, dispositivos de armazenamento, banco de dados, *middleware* e aplicações para *cloud* sistemas operacionais etc Sob este cenário, destaca-se o *MySQL*, banco de dados de código aberto, escalável com base *WEB*, rápido de alto desempenho roda em várias plataformas e ideal para *cloud*, segundo o fabricante. Conhecido por muitos desenvolvedores e de suporte mundial, na versão 5.6 a *Oracle* promete maior escalabilidade, aumenta a agilidade dos desenvolvedores através da otimização de subconsultas, e taxa de transferência mais elevada para consultas.

O *SGBD PostGres*, de código aberto, objeto-relacional roda nos mais usados sistemas operacionais, segundo a Comunidade Brasileira de *PostgreSQL* e possui grande documentação.

Possui suporte para chaves estrangeiras, suporta armazenamento de figuras sons e vídeos, é um banco de dados escalável no gerenciamento de dados e de usuários, para algumas linguagens de programação como *C/C++*, *Java*, *.Net*, *Perl*, *Python*, *Ruby*, possui interfaces nativas.

Tamanho Máx. do Bco de Dados	Ilimitado
Tamanho Máximo de uma Tabela	32 TB
Tamanho Máx. de uma Linha	1.6 TB
Tamanho Máx. de um Campo	1 GB
Máximo de Linhas por Tabela	Ilimitado
Máximo de Colunas por Tabela	250–1600 dependendo do tipo de coluna
Máximo de Índices por Tabela	Ilimitado

Figura 11 – Limites Fonte: Comunidade Brasileira de PostgreSQL

## 7. Processo Escolhido

Para o desenvolvimento deste aplicativo, sob um foco acadêmico, optou-se pelo Modelo Espiral.

Por ser um modelo cíclico, iterativo que mescla características da prototipação e os aspectos sistemáticos e evolutivos do modelo cascata. Esta flexibilidade possibilita, a cada ciclo, um retorno à atividades já executadas, sob um novo contexto mais avançado.

A escolha deste modelo possibilitará ao formando evoluir gradativamente no processo de forma a chegar a um nível que atinja os objetivos propostos neste projeto, Pressmann (2011).

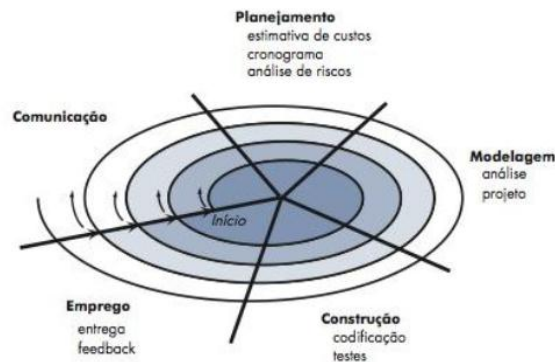


Figura 12 - Fonte: Pressmann (2011) 7ª Edição

## 8. Análise e Projeto

Foram efetuados levantamento dos tipos de movimentação de conta corrente e cartão de crédito, identificando a natureza de cada item para compor a lista de Receitas/Despesas do software de Gestão de Finanças Pessoais.

As publicações do IBGE (2008-2009), da FGV e do livro de MACEDO (2007), também forneceram subsídios a fim de delinear um universo de despesas e a busca de percentuais a serem atribuídos às mesmas.

A organização das atividades será gerenciada pela utilização da Linguagem de Diagramação UML, a fim de simplificar, iterar e unificar as fases deste projeto.

Para o registro das informações levantadas bem criação de diagramas se utilizará de ferramentas de edição de texto, e de uma versão gratuita do Astah (2013), para a criação de diagramas de caso de uso, diagramas de classe e diagramas entidade-relacionamento.

O Administrador parametriza os percentuais de Despesas que serão observados pela aplicação. O Cliente por sua vez poderá alterar estes percentuais customizando-os segundo seu perfil.

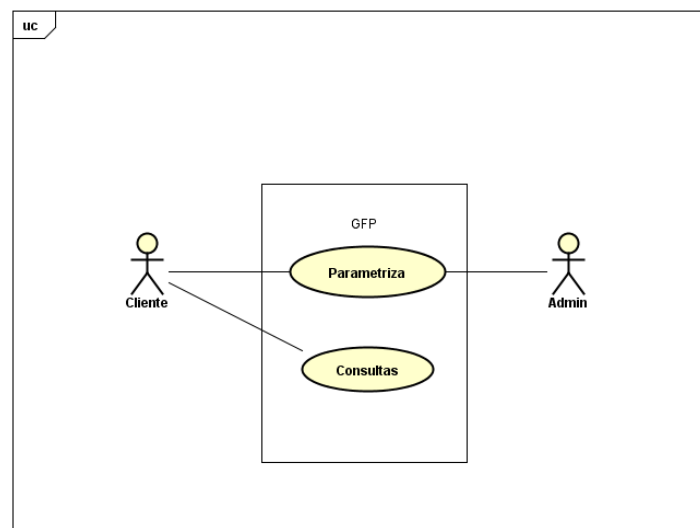


Figura 13 - Caso de Uso - Astah



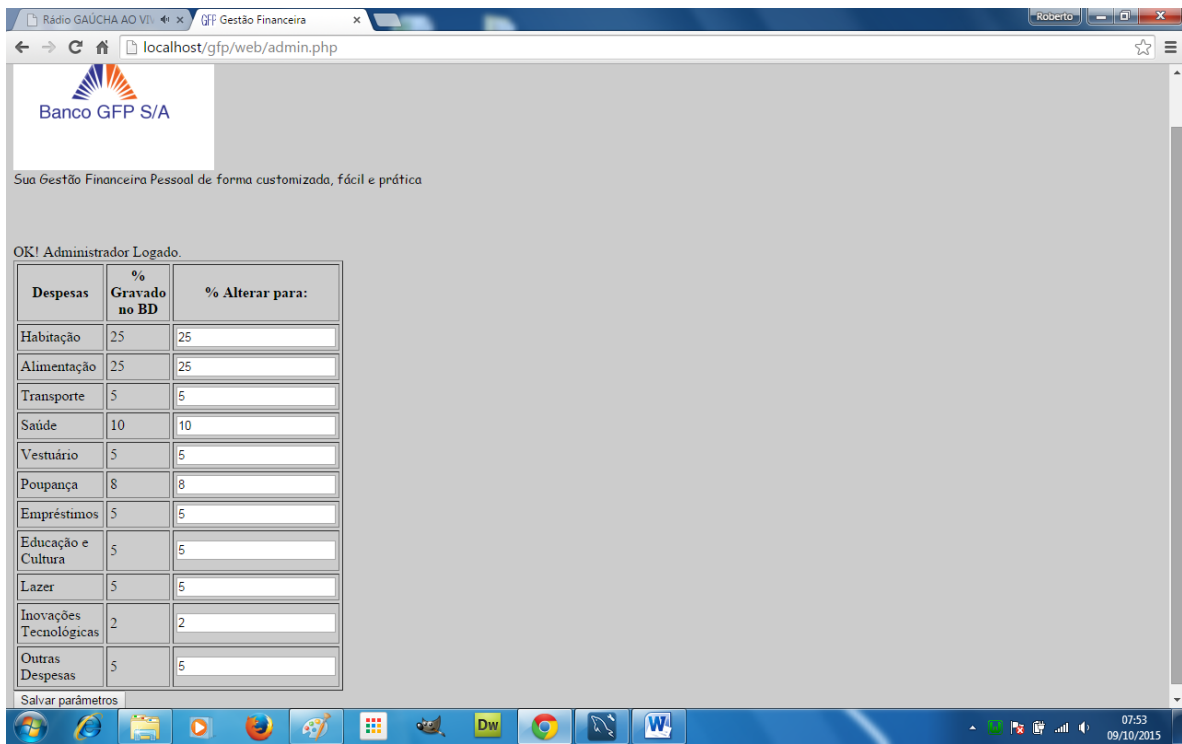


Figura 14 - Manutenção de Parâmetros – Administrador

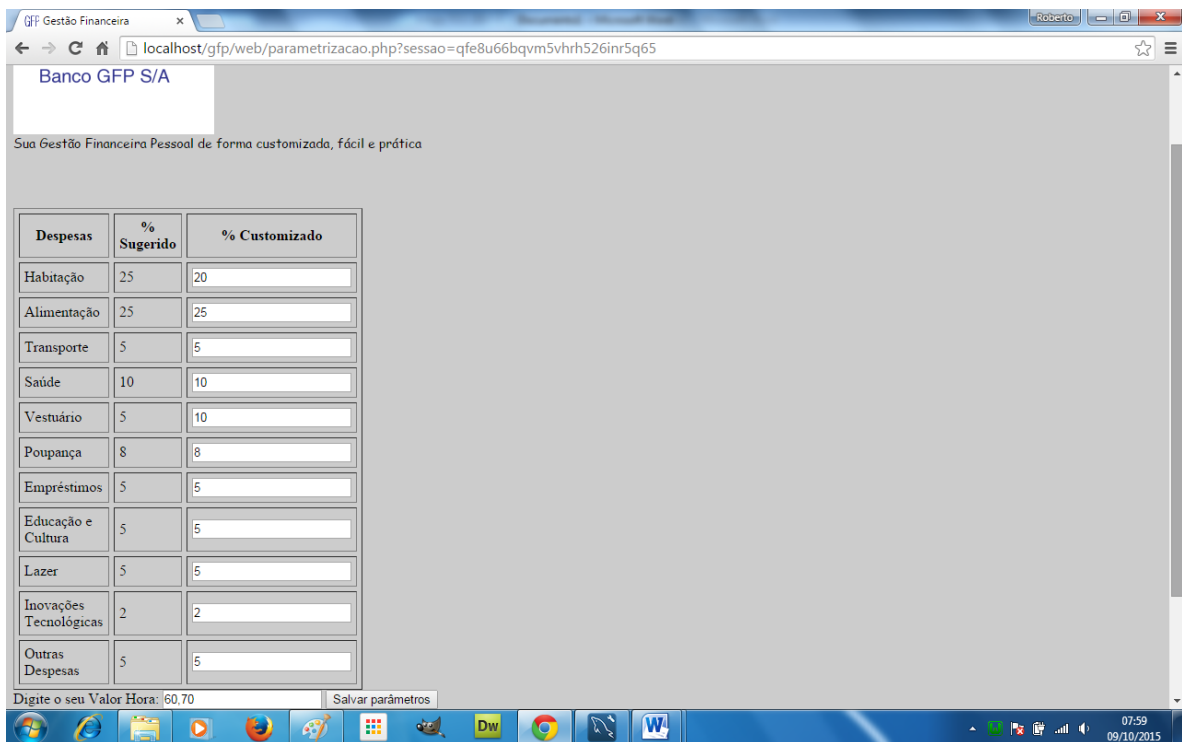


Figura 15 - Manutenção de Parâmetros - Cliente

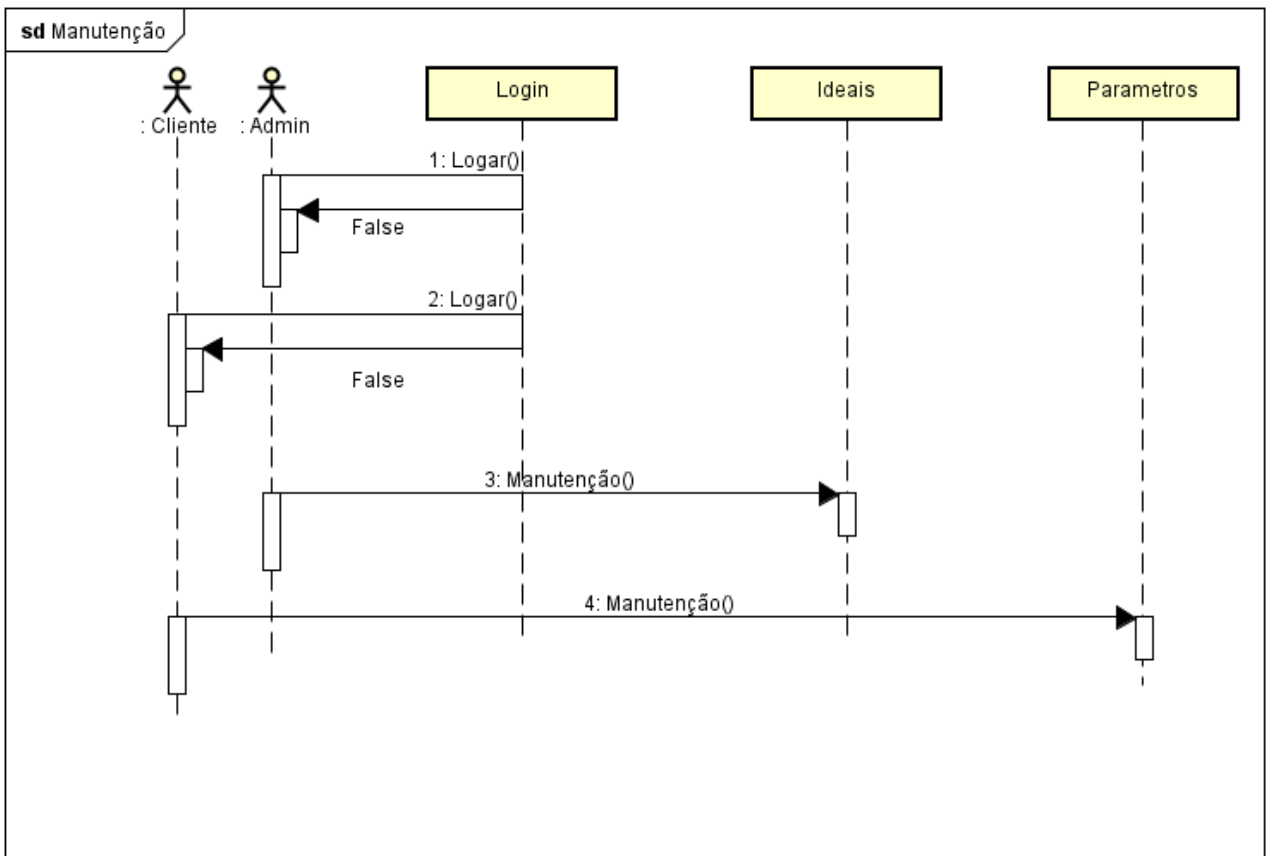


Figura 16 - Diagrama de Manutenção - Astah

A consulta disponibilizará ao cliente informações de um determinado período a cerca das movimentações da conta e os percentuais acumulados.

Home Login Finanças Gráficos > Parametrização

Banco GFP S/A

Sua Gestão Financeira Pessoal de forma customizada, fácil e prática

Data inicial: 01/01/2015 Data final: 31/01/2015 Ok

Despesas	Valor (R\$)	%	H. Trab.	Receitas	Valor (R\$)
Habitação	5.460,00	20	90,0	Salario	15.840,00
Alimentação	4.100,00	25	67,5	Depositos	250,00
Transporte	2.450,00	5	40,4	Resgate de Aplicações	840,00
Saúde	2.540,00	10	41,8	Outras Receitas	28,00
Vestuário	1.650,00	10	27,2		
Poupança	1.500,00	8	24,7		
Empréstimos	0,00	5	0,0		
Educação e Cultura	750,00	5	12,4		
Lazer	950,00	5	15,7		
Inovações Tecnológicas	1.800,00	2	29,7		
Outras Despesas	680,00	5	11,2		
<b>Total Despesas</b>	<b>21.880,00</b>			<b>Total Receitas</b>	<b>16.958,00</b>

Figura 17 - Consulta a movimentos - Finanças

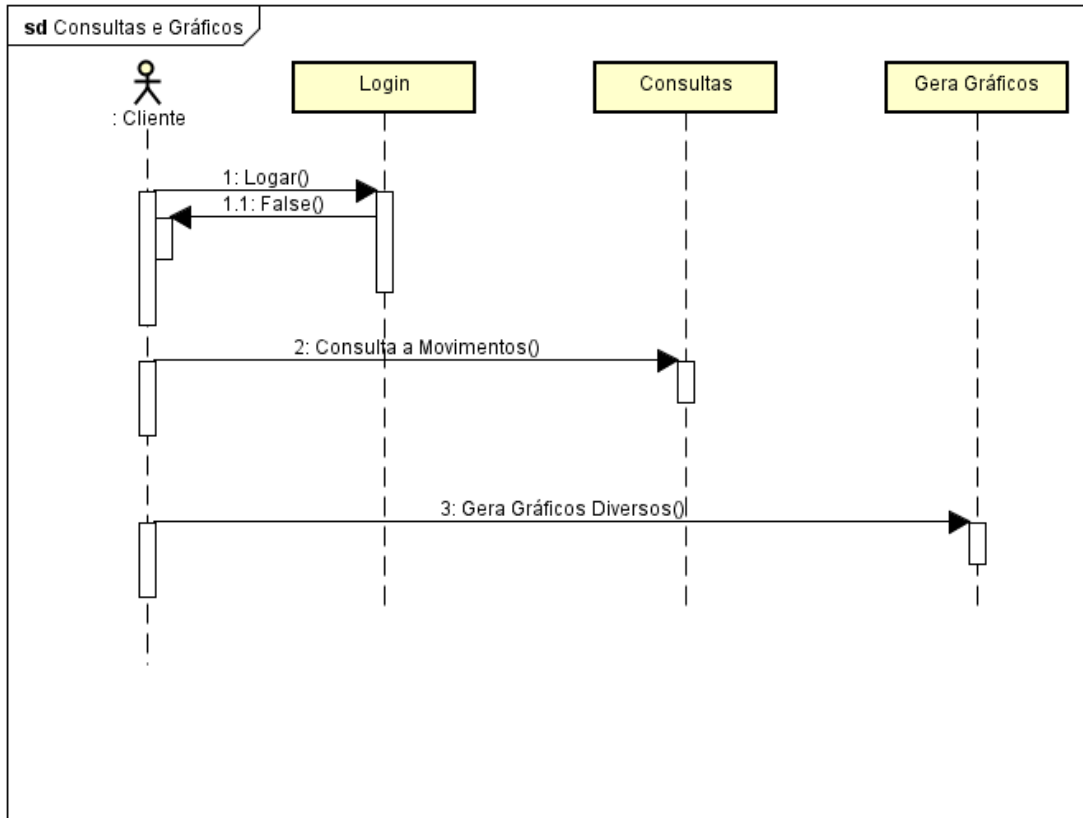


Figura 18 - Diagrama de Consultas e Gráficos – Astah

Como alternativa de visualização de Parâmetros, movimentação e comparação entre períodos o cliente poderá se utilizar da geração de gráficos disponibilizada pelo sistema.

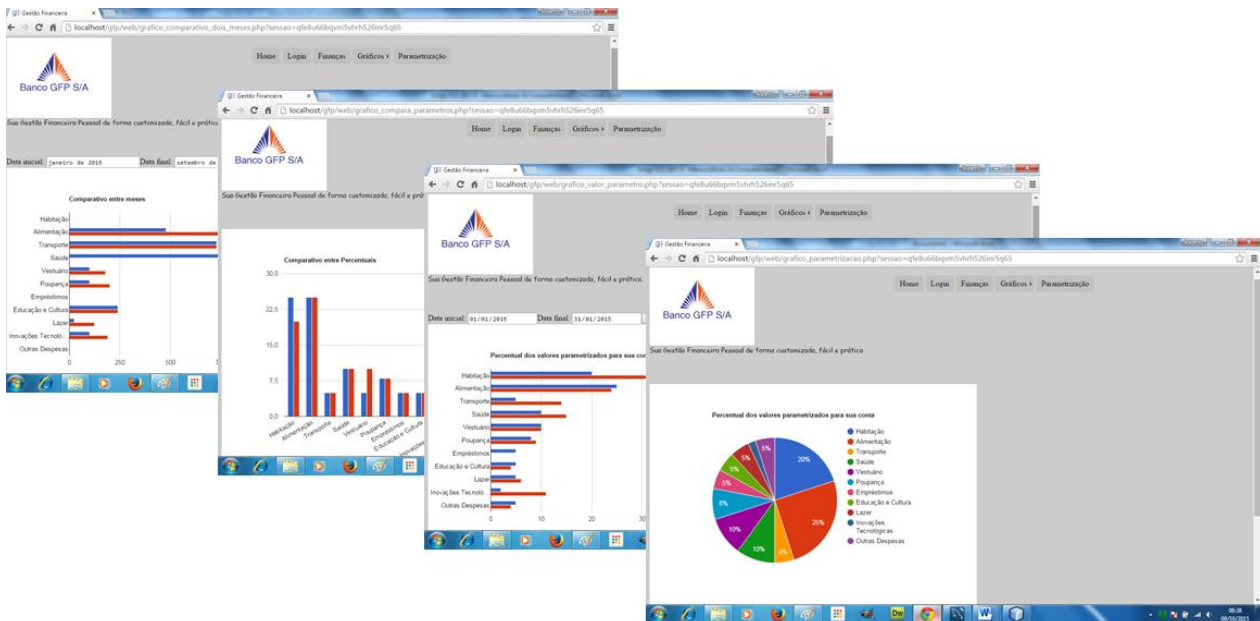


Figura 19 - Gráficos

## 9. Modelo e Banco de Dados escolhido

A utilização do Modelo Relacional de Banco de Dados e a representação de dados utilizando o Modelo Entidade-Relacionamento (MER) serão os escolhidos para desenvolvimento deste projeto.

A forma conceitual amigável e proximidade à uma natural interpretação, de relacionamento humano, entre coisas do mundo real, da sua natural facilidade de trabalho e compreensão entre os “*stakeholders*”, são as características decisivas para tal escolha.

Como Banco de Dados o *MySQL*, por ser um banco de dados de código aberto, confiável fácil de usar e popular, ORACLE (2015). Utilizando o aplicativo *WampServer* e *brModelo* como ferramenta “*freeware*” para criação do modelo relacional.

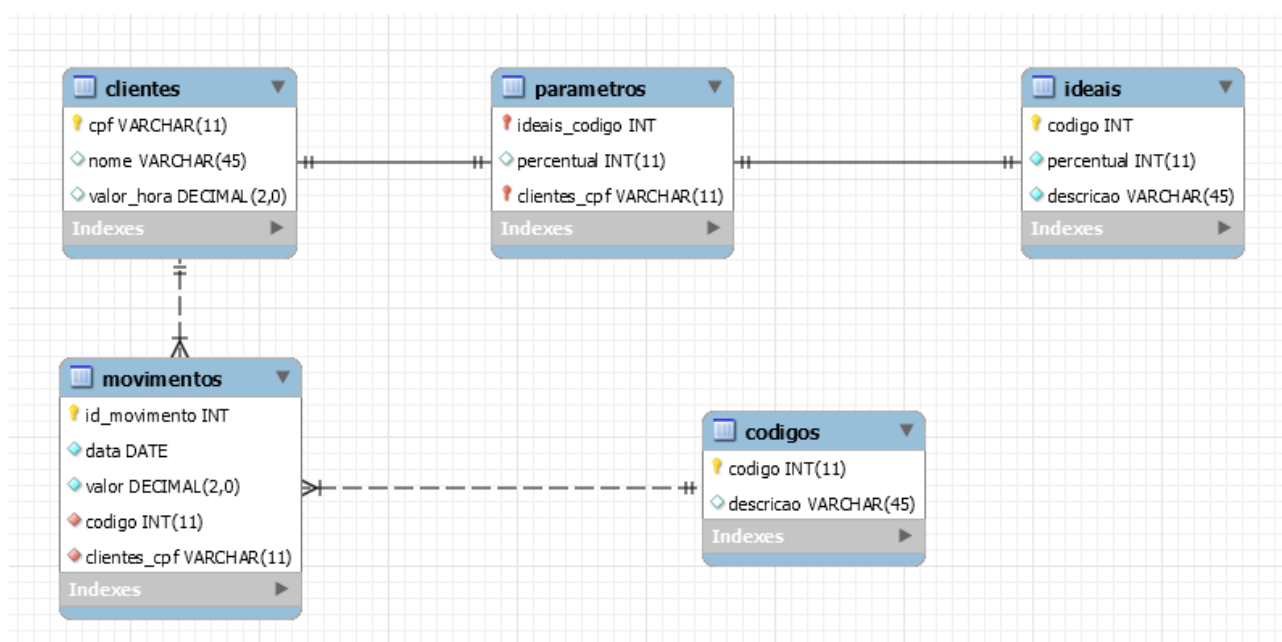


Figura 20 - DER MySQL Workbench

## 10. Linguagem de codificação

Na fase de codificação do sistema desenvolveu-se e implementou-se as funcionalidades necessárias à utilização da aplicação, a partir das anotações efetuadas durante a Fase de Análise buscando atingir o objetivo proposto pelo projeto.

Durante esta fase utilizou-se as linguagens e tecnologias, populares, confiáveis e *freeware*, para documentação das informações do projeto, modelo de Engenharia do Software, escrita do código fonte, banco de dados, e desenvolvimento da interfase WEB.

## 11. Testes

A Fase de Testes teve como Norte os objetivos do projeto, tomando por base as informações coletadas nas diversas fontes citadas e dados fictícios. Serão desenvolvidos testes com estes dados durante o desenvolvimento e implementações, sob o olhar do Modelo Espiral. Este modelo, como visto anteriormente possibilita a reanálise e nova construção aperfeiçoando o *software* a cada ciclo.

Ao final do processo de codificação foram efetuados testes com dados diversos, procurando aproximação à realidade, para todas as funcionalidades do sistema, verificando o correto funcionamento do mesmo.

## 12. Implantação

Concluída a Fase de Testes do sistema, o mesmo será hospedado em um servidor (*MySQL*, *PHP*) que disponibilize acesso a qualquer hora do dia e durante toda a semana.

Neste ambiente serão executados testes de acesso por diversos dispositivos móveis, com browsers diferentes a fim de verificar o desempenho de suas funcionalidades, tempo de resposta e agilidade do mesmo.

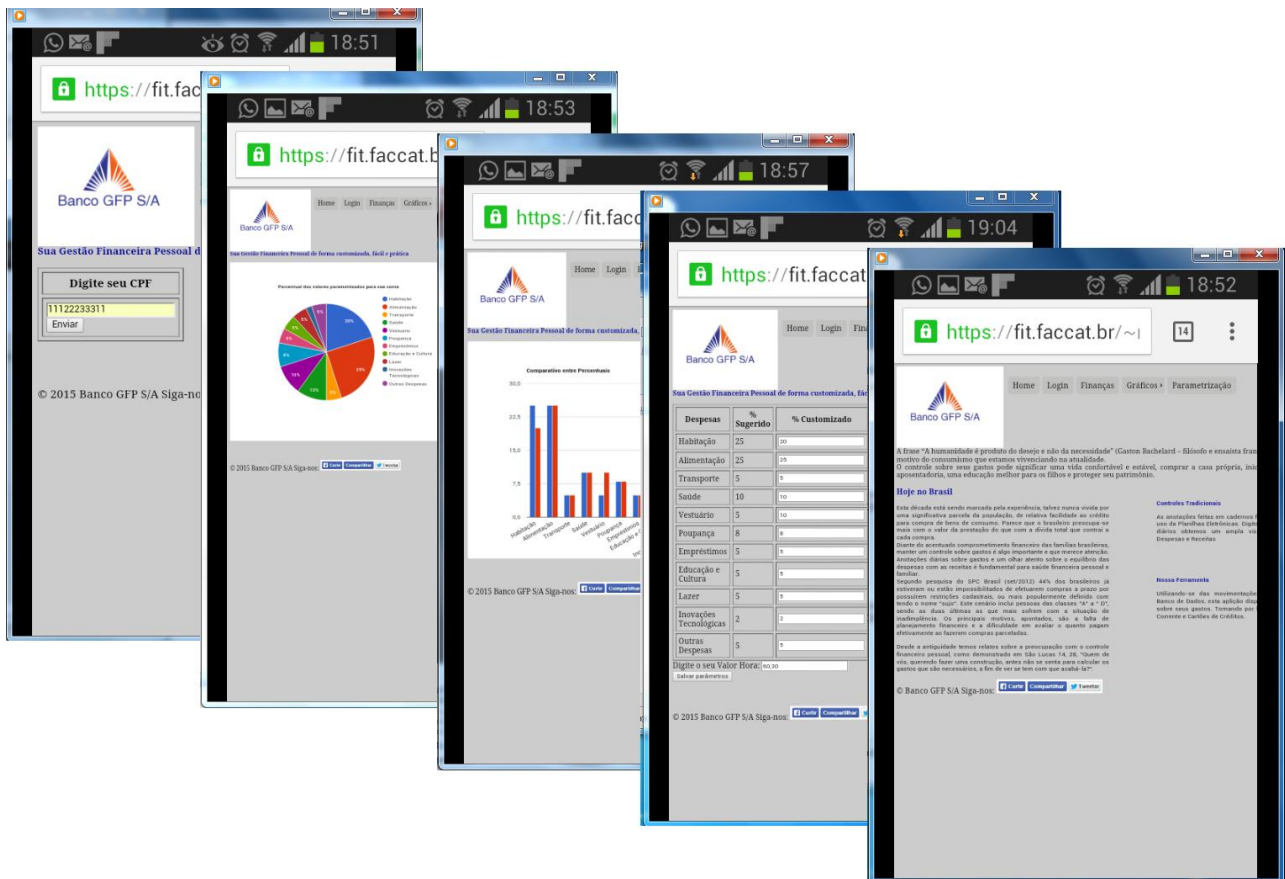


Figura 21 - Acesso via Mobile

### **13. Resultados**

O Software de Gerenciamento de Finanças Pessoais foi resultado da pesquisa e desenvolvimento de uma funcionalidade a ser implementada no Internet Banking para instituições bancárias disponibilizarem, a seus clientes.

Esta ferramenta, hospedada em servidor remoto, com acesso também por dispositivos móveis, disponibiliza aos usuários sua gestão financeira, a partir dos dados gerados pela movimentação financeira de sua conta corrente e cartão de crédito.

Desta forma exime o cliente de qualquer anotação de gastos, na medida em que este faça suas movimentações financeiras via conta corrente e cartão de crédito.

A aplicação gera informações sob a forma de Planilhas e Gráficos segundo parâmetros e critérios sugeridos ou customizados.

Nos testes realizados em servidor remoto, as respostas via Mobile, mostraram-se ágeis, considerando a banda utilizada, notando-se pouca diferença para as requisições solicitadas em localhost.

### **14. Conclusão**

O software foi desenvolvido e testado primeiramente em ambiente local, e finalmente hospedado em servidor remoto, confirmando seu bom desempenho e flexibilidade de acesso por diferentes dispositivos.

Segundo testes realizados pode-se verificar que o mesmo atingiu seus objetivos, quais sejam:

- acesso remoto;
- gestão financeira a partir de movimentos bancários;
- ser uma ferramenta de educação e gestão financeira de despesas e receitas;
- ser uma aplicação diferenciada pelas inexistências das costumeiras anotações e digitações de dados;
- possibilitar análise numérica, didática e ilustrativa dos movimentos financeiros;
- ferramenta inovadora e de fidelização de cliente.

O desenvolvimento desta aplicação de longe esgota as formas e alternativas de disponibilização de informações para análise financeira pessoal.

Como futuro trabalho poderiam ser implementados avisos via SMS quando as despesas estivessem próximas dos percentuais parametrizados. Ou cores diferentes, tipos semáforos, ilustrando nas planilhas estes movimentos, como alerta ao usuário.

## 15. Referências

- ASTAH (2013), <http://astah.net/>
- BEZERRA, Eduardo, Princípios de Análise e Projetos de Sistemas UML 2007 2ª Edição Elsevier Editora Ltda
- BIBLIA CATÓLICA, São Lucas 14, 28 Disponível em: <http://www.bibliacatolica.com.br/biblia-ave-maria/sao-lucas/14/#.UICS3Ki5ddg>
- BROOKSHEAR, J. GLEN, Ciência Da Computação: Uma Visão Abrangente 7ª Edição Editora Bookman
- COMUNIDADE BRASILEIRA DE POSTGRESQL - <http://www.postgresql.org.br/>
- COX, BRAD J. - <http://www.testedemesa.com.br/newsflash/frase-j-cox.html> 2013
- DALL'OGGIO, PABLO Criando Relatórios com PHP Novatec Editora 2013
- DATE, C. J. Introdução a Sistemas de Banco de Dados – 8ª Edição 2004 Elsevier Editora Ltda
- DIA PORTABLE, <http://www.baixaki.com.br/download/dia-portable.htm>
- DIEESE, Departamento Intersindical de Estatística e estudos Socioeconômicos <http://www.dieese.org.br/analisecestabasica/salarioMinimo.html>
- DICKETT, JON (2010), Introdução à Programação WEB com HTML,XHTML e CSS 2ª Edição Editora Ciência Moderna.
- FECOMERCIO-RS (2013), Pesquisa de Endividamento e Inadimplência do Consumidor – PEIC <http://www.cnc.org.br/central-do-conhecimento/pesquisas/pesquisa-nacional-de-endividamento-e-inadimplencia-do-consumido-19>
- FECOMERCIO-SP (2012), Radiografia do Endividamento das Famílias nas Capitais Brasileiras [http://www.fecomercio.com.br/arquivos/arquivo/radiografia\\_endividamento\\_das\\_familias\\_brasileiras\\_aladrmla6.pdf](http://www.fecomercio.com.br/arquivos/arquivo/radiografia_endividamento_das_familias_brasileiras_aladrmla6.pdf)
- FERNANDES, LUIZ GUSTAVO LEÃO [www.inf.pucrs.br/~gustavo](http://www.inf.pucrs.br/~gustavo)
- FGV – Fundação Getúlio Vargas - <http://portal.fgv.br/>
- FILHO, Antonio Mendes da Silva, Introdução a Programação Orientada à Objetos com C++ 2010 Elsevier Editora Ltda
- HEUSER, CARLOS ALBERTO Projetos de Banco de Dados Vol. 4 - Bookman 2004
- IBGE – Instituto Brasileiro de Geografia e Estatística POF 2008 – 2009
- IBM - <http://www.ibm.com/developerworks/br/library/os-mongodb4/>
- MACEDO JUNIOR, Jurandir Sell. A Árvore do dinheiro: Guia para cultivar a sua independência financeira. Rio de Janeiro: Elsevier, 2007.
- MICROSOFT, Noções Básicas de Banco de Dados, <http://office.microsoft.com/pt-br/access-help/nocoes-basicas-do-banco-de-dados-HA010064450.aspx>
- NAVATHE, Shamkant B. e Elmasri, Ramez - Sistemas de Bancos de Dados 2005 Pearson Education do Brasil Ltda.
- NetBeans IDE - [https://netbeans.org/features/index\\_pt\\_BR.html](https://netbeans.org/features/index_pt_BR.html)
- O GLOBO Caderno de Economia (27/05/2013) <http://oglobo.globo.com/economia/endividamento-das-familias-bate-recorde-4399-da-renda-8516655>
- ORACLE (2015), <http://www.oracle.com/br/products/mysql/overview/index.html>
- PRESSMAN, Roger S., Engenharia de Software – Uma abordagem profissional 7ª Edição AMGH Editora Ltda 2011
- RAMALHO, JOSE ANTONIO ALVES – Oracle 10 g 2005 Thomson Learning Ltda
- SILBERSCHATZ, ABRAHAM e SUDARSHAN, HENRY F. KORTH – Sistemas de Banco de Dados. São Paulo Makron Books, 1999
- SOLID IT - IT Consulting & Software Development - <http://db-engines.com/en/ranking>
- SOMMERVILLE, IAN, Engenharia de Software 8ª Edição Editora Pearson Education
- SPC BRASIL, <https://www.spcbrasil.org.br/imprensa/noticia/296-spcbrasildivulgaconcentraodainadimplenciaemsetembroporgeneroidadeevalordadivida>
- TURCKER, ALLAN B. e NOONAM, ROBERT E. Linguagens de Programação – Princípios e Paradigmas 2ª Edição 2007 The M.cGraw-Hill Companies
- THE IRISH PHP USERGROUP, <http://www.php.ie/f/PHPWhitePaper.pdf> 2009-Jan-08
- THE PHP GROUP - <http://www.php.net/>