***INSTITUTO SUPERIOR DE TRANSPORTES E COMUNICAÇÕES***

**Desenvolvimento de um Sistema de Monitoramento de Venda de Recargas na Mcel**

Carlos António Mahumane

Pré-Projecto Final do Curso

Licenciatura em Engenharia Informática e de Telecomunicações

Supervisor:

\_\_\_\_\_\_\_

Departamento de Tecnologias de Informação e Comunicações

Setembro de 2015

***INSTITUTO SUPERIOR DE TRANSPORTES E COMUNICAÇÕES***

**Desenvolvimento de um Sistema de Monitoramento de Venda de Recargas na Mcel**

Carlos António Mahumane

Pré-Projecto Final do Curso

Licenciatura em Engenharia Informática e de Telecomunicações

Supervisor:

\_\_\_\_\_\_\_\_\_\_\_

Departamento de Tecnologias de Informação e Comunicações

Setembro de 2015

**ÍNDICE**

[ÍNDICE DE FIGURAS iii](#_Toc431380906)

[LISTA DAS ABREVIATURAS UTILIZADAS iv](#_Toc431380907)

[PROTOCOLO DA INVESTIGAÇÃO 5](#_Toc431380908)

[1 Justificação do tema 5](#_Toc431380909)

[2 Desenho teórico 5](#_Toc431380910)

[2.1 Problemática 5](#_Toc431380914)

[2.2 Problema 6](#_Toc431380915)

[2.3 Objecto 6](#_Toc431380916)

[2.4 Objectivo Geral 6](#_Toc431380917)

[2.5 Objectivos Específicos 6](#_Toc431380918)

[2.6 Perguntas de Investigação 7](#_Toc431380919)

[3 Metodologia 7](#_Toc431380920)

[3.1 Abordagem de Investigação 7](#_Toc431380921)

[3.2 Estado da tecnologia 7](#_Toc431380922)

[3.3 Hipótese 17](#_Toc431380923)

[3.4 Variáveis de investigação 17](#_Toc431380924)

[3.5 Métodos de Investigação 17](#_Toc431380925)

[3.6 Resultados esperados de investigação 18](#_Toc431380926)

[4 Administração e Controlo de Investigação 19](#_Toc431380927)

[4.2 Cronograma da investigação 19](#_Toc431380928)

[REFERÊNCIAS BIBLIOGRÁFICAS 21](#_Toc431380929)

# ÍNDICE DE FIGURAS

[Figura 1: Diversos tipos de Diagramas UML, (TEOFELO) 11](#_Toc431380493)

[Figura 2: Esquema de uma aplicação usando MVC (PRESSMAN, 1995) 12](#_Toc431380494)

[Figura 3: Esquema de uma aplicação Monótona (PRESSMAN, 1995) 12](#_Toc431380495)

# LISTA DAS ABREVIATURAS UTILIZADAS

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | CD | *Concept Document* | | CMS | *Content Management System* | | DAP | *Directory Access Protocol* | | E/P/S | Entrada/Processo/Saída | | EMAIL | *Electronic Mail* | | LDAP | *Lightweight Directory Access Protocol* | | LGPL | *Lesser General Public License* | | LMS | *Learning Management System* | | MCEL | Moçambique Celular | | MVC | *Model View Control* | | PHP | *Hypertext Preprocessor* | | SMTP | *Simple Mail Transfer Protocol* | | UAT | *User Acceptance Tests* | | UML | *Unified Modeling Language* | | BSCS | Business Support & Control System | | RS | Reengenharia de *Software* | | Mcel | Moçambique Celular, S.A | |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# ProtocOlo da investigação

## Justificação do tema

A venda de recargas é uma das principais fontes de lucro na empresa Mcel. A Mcel possui um sistema denominado **DGRV** *(Sistema de Gestão de TopUps)*,este sistema faz o cálculo total de vendas de Recargas para um determinado intervalo de tempo sem distinguir os dados por antena ou zona.

Cabe ao Departamento de marketing e vendas garantir colocação e a venda das recargas no mercado, mas o sistema actual não distingue as vendas por antenas ou zonas, o que lhes limita o controlo das vendas por zona.

Com base num um sistema de monitoramento de venda de recargas distinguindo as zonas, irá permitir ao departamento de Marketing e Vendas maior controle e possibilitará o desenvolvimento de acções imediatas para melhorar o estado das vendas de recargas.

## Desenho teórico



### Problemática

A Mcel é uma empresa de telecomunicações em Moçambique, ela esta operacional a mais de 17 anos. Possui uma cobertura por quase todo país e até ao momento possui mais de 1700 antenas distribuídas por todo o país.

A Mcel possui uma cadeia de distribuição dos seus serviços desde pacotes iniciais, coletes ate as Recargas. A cadeia de distribuição funciona com base em canais de distribuição, cada canal é responsável por uma determinada região, cada região ou zona pode conter uma ou várias antenas.

O departamento de marketing e vendas tem como principal função garantir a venda e a colocação dos produtos no mercado. A venda de recargas é uma das principais fontes de lucro da Mcel nos clientes do tipo Pré-pago, actualmente ela possui um sistema denominado **DGRV**,este sistema faz o cálculo total de recargas vendidas num determinado intervalo de tempo, sem distinguir:

* O valor da recarga;
* A Antena;
  + Tipo de antena (2G, 3G ou 2G/3G);
  + Localização;
    - Província;
    - Distrito;
    - Localização georreferenciada;

Assim o departamento de Vendas e Marketing não possui dados suficientes e organizados, o que lhes limita:

* Realizar estatísticas e relatórios precisos de TopUps;
* Avaliação da sua performance comercial em termos de TopUps;
* Realizar uma rápida análise da situação actual de TopUps por zona/antena;
* Enviar se necessário, uma equipa de técnicos de distribuição e retalho para apurar a situação da zona;
* Tomar medidas que visam melhorar o estado de venda na zona.
* Encaminhar a uma equipe de técnicos de redes de telecomunicações em caso de avaria.

Com o desenvolvimento de um sistema monitoramento de venda de recargas irá permitir ao departamento de Marketing e Vendas maior controle, assim será possível o desenvolvimento de acções imediatas para melhorar o estado de vendas de recargas pelo país.

### Problema

Como desenvolver um sistema baseado em uma plataforma web para monitorização de vendas de Recargas na Mcel por Antena?

### Objecto

O Objecto de investigação deste projecto é a monitorização de vendas de recargas na Mcel.

### Objectivo Geral

Desenvolver um sistema baseado em uma plataforma web para monitoramento de venda de recargas na Mcel por Antena.

### Objectivos Específicos

* Estudar o sistema actual de registro de venda de recargas na Mcel;
* Analisar e identificar os problemas do sistema actual;
* Fazer o levantamento dos requisitos necessários para melhoria do sistema actual;
* Desenvolver e testar o Sistema.

### Perguntas de Investigação

* Quais são os inconvenientes do sistema actual?
* Quais são as plataformas actuais de registo de TopUps?
* Qual é a relação de um registo de um TopUp e uma antena?

## Metodologia

### Abordagem de Investigação

Este projecto visa desenvolver um sistema que possibilite o monitoramento de vendas de recargas por antena na Mcel, será necessário um contacto directo entre o pesquisador e o ambiente onde a problemática reside. Segundo GODOY (1995), refere que estudos do género são de carácter qualitativo, pois enfatizam das diferentes características as seguintes:

* Obedecem a um plano pré-estabelecido, com o intuito de, enumerar ou medir eventos;
* Utilizam como instrumento para coleta de dados, questionários estruturados, elaborados com questões fechadas, testes e checklists, aplicados a partir de entrevistas individuais, apoiadas por um questionário convencional ou electrônico;
* Utiliza dados que representam uma população específica (amostra), a partir da qual os resultados são generalizados.

### Estado da tecnologia

#### Sistema

Sistema, é definido por (PRESSMAN, 1995), "é um conjunto abstracto, recursivo, sem inicio e sem fim, de objectos inter-relacionados, operando com restrições, que em geral são constituídos por subsistemas", que podem ser classificados em:

* **Sistema com Intenções**: criado com base num conjunto de objectivos.
* Sistema Aberto: Permite a interação com o ambiente, podendo ter diversos graus de aberturas;
* Sistema Dinâmico: Recebe entradas do ambiente, processa-se e produz resultados e pode ser representado pelo ciclo E/P/S (Entrada/Processo/Saida).

#### Sistema de Informação

Sistema de informação, segundo (VARAJAO, 2005), " é um sistema cujo elemento principal é a informação. Seu objectivo é armazenar, tratar e fornecer informações de tal modo a apoiar as funções ou processos de uma organização", e podem ser classificados:

**Sistemas de Informação Rotineiros ou Transacionais**, são os mais simples e os mais comuns nas organizações. Eles apoiam as funções operacionais da organização, aquelas realizadas no dia-a-dia.

**Sistemas de Informações Gerências**, surgiram com o intuito de auxiliar gerentes em suas funções. Com o passar do tempo, este tipo de sistema acabou sendo usado por qualquer funcionário que tome decisões.

**Sistemas de Apoio à Decisão** - recebe como entrada alternativas para solução e um problema e devolve as consequências para cada alternativa. Assim o administrador pode avaliar qual é a melhor alternativa.

#### Informação

Informação é o resultado do processamento, manipulação e organização de dados, de tal forma que represente uma modificação (quantitativa ou qualitativa) no conhecimento do sistema (pessoa, animal ou máquina) que a recebe. (VARAJAO, 2005)

#### Linguagem de programação

Uma linguagem de programação é um vocabulário e um conjunto de regras gramaticais usadas para escrever programas de computador. Esses programas instruem o computador a realizar determinadas tarefas específicas. Cada linguagem possui um conjunto único de palavras-chaves e uma sintaxe específica para organizar as instruções dos programas. (PRESSMAN, 1995). Segundo (Cass, Diakopoulos, & Romero, 2014), as 7 primeiras linguagens de programação mais utilizadas no mundo são: Java, C, C++, Python, PHP, Javascript, Ruby;

#### Java

Java é uma linguagem de programação orientada a objetos desenvolvida pela Sun Microsystems. Modelada depois de C++, a linguagem Java foi projetada para ser pequena, simples e portável a todas as plataformas e sistemas operacionais, tanto o código fonte como os binários. Esta portabilidade é obtida pelo fato da linguagem ser interpretada, ou seja, o compilador gera um código independente de máquina chamado byte-code. No momento da execução este byte-code é interpretado por uma máquina virtual instalado na máquina. Para portar Java para uma arquitetura hadware/s específica, basta instalar a máquina virtual. Além de ser integrada à Internet, Java também é uma excelente linguagem para desenvolvimento de aplicações em geral. Dá suporte ao desenvolvimento de software em larga escala. (Oracle, 2014)

#### Classe

Classe segundo (RICARTE, 2001) " é uma estrutura que abstrai um conjunto de objectos com características similares. Uma classe define o comportamento de seus objectos através de métodos e os estados possíveis destes objectos através de atributos."

#### Pacote

Pacote (do inglês package) segundo (RUBIRA) " é um mecanismo para organizar elementos em grupos. Um pacote pode estar contido em outro pacote. Um sistema pode ser visto como um único pacote de alto nível, que contém todos os elementos do sistema."

#### Correio Electrônico

O correio electrônico, igualmente conhecido pelo nome de *e-mail*, que é a forma abreviada do termo inglês *electronic mail* é um serviço que permite trocar mensagens através de sistemas de comunicação eletrónicos. O conceito é principalmente usado para fazer alusão ao sistema que proporciona este serviço através da Internet mediante o protocolo SMTP (*Simple Mail Transfer Protocol*), mas também permite fazer referência a outros sistemas similares que recorrem a várias tecnologias. (PRESSMAN, 1995).~

#### Base de Dados

Uma base de dados é um simples repositório de informação relacionado com um determinado assunto ou finalidade. Serve para gerir vastos conjuntos de informação de modo a facilitar a organização, manutenção e pesquisa de dados. (PRESSMAN, 1995)

#### UML

UML segundo (TEOFELO) é uma tentativa de padronizar a modelagem orientada a objectos de uma forma que qualquer sistema, seja qual for o tipo, possa ser modelado correctamente, com consistência, fácil de se comunicar com outras aplicações, simples de ser actualizado e compreensível. Existem diversas formas de fazer esta representação em UML:

**Diagramas Estruturais**

Os diagramas estruturais descrevem como as estruturas se relacionam.

* Diagrama de classes;
* Diagrama de pacotes;
* Diagrama de objectos;
* Diagrama de componentes;
* Diagrama de implantação.

**Diagramas comportamentais**

Os diagramas comportamentais descreve como um certo comportamento leva a uma actividade.

* Diagrama de casos de uso;
* Diagrama de sequência;
* Diagrama de colaboração;
* Diagrama de estados;
* Diagrama de actividades.



Figura 1: Diversos tipos de Diagramas UML, (TEOFELO)

#### LDAP

Segundo (PRESSMAN, 1995), é um protocolo para actualizar e pesquisar directórios rodando sobre TCP/IP. Um directório LDAP geralmente segue o modelo X.500, que é uma árvore de nós, cada um consistindo de um conjunto de atributos com seus respectivos valores. O LDAP foi criado como uma alternativa ao muito mais incómodo Directory Access Protocol (DAP).

#### Modelo MVC

È um modelo de três camadas divide um aplicativo de modo que a lógica de negócio resida no meio das três camadas físicas. Isto é chamado de camada física intermediária ou camada física de negócios. A maior parte do código escrito reside na camada de apresentação e de negócio. A arquitetura MVC - (Modelo Visualização Controle) fornece uma maneira de dividir a funcionalidade envolvida na manutenção e apresentação dos dados de uma aplicação. A arquitectura MVC não é nova e foi originalmente desenvolvida para mapear as tarefas tradicionais de entrada, processamento e saída para o modelo de interacção com o usuário. Usando o padrão MVC fica fácil mapear esses conceitos no domínio de aplicações Web multi camadas. (RICARTE, 2001)



Figura 2: Esquema de uma aplicação usando MVC (PRESSMAN, 1995)

#### Aplicações Monótonas

Geralmente este aplicativo continha todas a funcionalidades em um único módulo gerado por uma grande quantidade de linhas de código e de manutenção nada fácil. A entrada do usuário, verificação, lógica de negócio e acesso a banco de dados estava presente em um mesmo lugar. (PRESSMAN, 1995).



Figura 3: Esquema de uma aplicação Monótona (PRESSMAN, 1995)

#### Reengenharia de *software (RS)*

A reengenharia leva tempo, tem um determinado custo financeiro, e absorve recursos que podem ser de outra forma ocupados em problemas imediatos. “A RS é uma actividade de reconstrução. (PRESSMAN, 2010:769)” e para implementa-la é necessário seguir-se um modelo que define seis actividades: análise de inventário, reestruturação da documentação, engenharia reversa, reestruturação do código, reestruturação dos dados, e a engenharia progressiva.

**Análise do inventário**: esta análise consiste na verificação daquela que é a planilha aplicacional da organização, tendo em conta os seguintes detalhes por aplicação: criticidade, capacidade de manutenção actual, suporte, longevidade, referências.

**Reestruturação da documentação**: documentação pobre é a marca registrada de muitos sistemas. Tendo em conta este cenário, este actividade têm como principal foco a criação e/ou actualização da documentação do sistema existe.

**Engenharia reversa**: esta actividade consiste na análise do sistema, com o intuito de criar uma representação do sistema a um nível mais alto de abstração, desvendando deste modo os “segredos” existentes por detrás do sistema.

**Reestruturação do código**: esta actividade consiste na análise do código fonte, de modo a detectar as violações (erros) na contrução do mesmo. Após detectar-se as violações o código é reescrito de outra forma, ou em outra linguagem de programação. À posterior, o código reestruturado é examinado e testado para se assegurar que sobre ele não encorrem anomalias. Concluído este processo é necessário fazer-se a actualização da documentação.

**Reestruturação dos dados**: normalmente esta actividade começa com a engenharia reversa. É redesenhada uma nova arquitetura de dados, construido um novo modelo de dados. Mas isto só acontece quando há alguma anomalia na arquitetura e no modelo de dados previamente existente.

**Engenharia progressiva**: esta actividade é desenvolvida normalmente por “motores de reengenharia”. Nela, recupera-se a informação do projecto existente, utiliza-se a mesma para alterar e reconstruir o sistema em um esforço para melhorar a qualidade do sistema no âmbito geral. Este “motor” reimplementa as funções do sistema existentes e também adiciona novas funções de forma a melhorar o desempenho.

#### OutSystems

“A OutSystems é uma plataforma de desenvolvimento ágil de aplicações web e móveis, geradas através do Java ou .Net nativo da plataforma, que correm na nuvem, localmente (em servidores da organização) ou em ambientes híbridos.” (OUTSYSTEMS, 2015)

É uma plataforma pois é mais do que um RAD – modelo de Desenvolvimento Rápido de aplicações (***R****apid* ***A****pplication* ***D****evelopment* ***m****odel*), pois dentro dela possui-se características como:

***Business Process Managment*** (BPM) – é uma abordagem de modelação de processos de negócio de uma determinada organização. O objectivo desta modelagem é conduzir a uma melhor visão do conjunto de processos da organização e suas interações, criando condições de os otimizar e automatizá-los ao máximo coma ajuda de aplicações.

**Oracle & SQL Server** - ferramentas são utilizadas gestão do modelo de dados que será suportado utilizado pelas aplicações.

**Batch** – (Schedule/Timer/Cron) – ferramenta que é utilizada de forma a ser ter o mecanismo de serviços agendados de forma síncrona. Ex: envio de correio eletrónicos sobre o estado de certos sistemas todos os dias as 7h.

**Controle de versões** - Ferramenta que nos permite ter o controlo das versões das aplicações. Através do *Service Center* pode-se verificar quantas versões foram publicadas (deploy da aplicação), quem foi o responsável pela publicação e quando é que ocorreu.

**1-Click Publish** – Após desenvolver utiliza-se esta ferramenta para efectuar o deploy da aplicação no servidor.

**Gestão End-to-end do Ciclo de vida** – ferramenta que permite ter vários ambientes (Desenvolvimento, Qualidade, Produção) que nos ajuda a desenvolver em módulos diferentes e entregando o produto de forma faseada.

**Multi-Programador** – Permite ter no mesmo projecto ter vários desenvolvedores envolvidos.

**Merge** – É uma tecnologia que permite a fusão do trabalho desenvolvido no projecto. Esta ferramenta é utilizada quando os programadores alteram elementos diferentes de um módulo/projecto, permitindo deste modo a fusão desses elementos dispersos.

**Extensível** – Permite a reutilização do código (**DDL** no caso so .Net e **.JAR** no caso do JAVA) através da ferramenta de integração (Integration Studio).

**Auto-documental** – ferramenta que permite a geração de documentação técnica das aplicações.

**Alta Escalabilidade** – permite o aumento da infraestrutura de forma horizontal. Se a organização possuir um balanceador de carga antes dos servidores, podemos redirecionar pedidos de uma aplicação com muita solicitação para dois servidores ao invés de um.

**Monitoramento de Performance (Service Center)** – ferramenta que nos dá informações generalizadas sobre as aplicações como tempo de resposta da aplicação, erros aplicações, verificação versões aplicacionais, verificação das versões das DDL’s/JAR’s.

**Lock-in** - Se a organização decidir parar de utilizar a plataforma, tem a opção do ter o código das aplicações geradas em nas tecnologias .Net ou Java.

Esta plataforma disponibiliza duas versões: uma organizacional (paga) e uma grátis (com menos recursos que a empresaria) mais para um apredizagem. Existem dois tipos de ferramentas de desenvolvimento (IDE): o Service Studio e o Integration Studio.

A utilização desta plataforma muda o panorama de desenvolvimento transitando de uma análise exaustiva do que o sistema deve fazer, para uma análise simplificada e 100% focada no cliente, transitando do modelo do desenvolvimento de projecto completo (onde cliente avalia o projecto muito tarde) para um modelo de desenvolvimento em blocos menores (antecipação feedback ao cliente). (OUTSYSTEMS, 2015)

#### Testes de Software

São actividades realizadas ao longo do desenvolvimento, verificam se o *software* e seus componentes atendem às suas especificações. Testes de Software garantem que o sistema foi desenvolvido corretamente e o mesmo possui as características de qualidade esperadas. (DA COSTA, 2011)

* Actividades controladas visando identificar problemas;
* Envolve ações que vão do levantamento de requisitos até a execução dos testes propriamente ditos;
* Verifica “resultados encontrados” X “resultados esperados” ;
* Prevenir, localizar e documentar defeitos;
* Minimizar o impacto potencial de defeitos, detectando-os antes da entrega do produto ao cliente.

#### Tipos de Testes de Software

Existem bastantes tipos de testes de *software*, segundo (Da Costa, 2011) alguns dos exemplos:

* Teste de Unidade;
* Teste de Integração;
* Teste de Sistema;
* Teste de Integridade de Dados e de Base de Dados;
* Teste Funcional;
* Teste de Interface com o usuário
* Teste de desempenho;
* Teste de Carga;
* Teste de Segurança;
* Teste de Tolerância a Falhas;
* Teste de Recuperação;
* Teste de Configuração.

#### Testes Unitários

O teste unitário é implementado com base no menor elemento testável (unidades) do *software*, denominado o teste da caixa branca. Implica em testar a estrutura interna (como fluxo lógico e de dados), a função da unidade e os comportamentos observáveis. (Da Costa, 2011). Unidade pode ser uma classe, método, regras. Para realizar estes testes existem duas tecnologias populares e livres:

* **Junit**, realiza os testes unitários baseado em comparação do resultado obtido e esperado. A API - http://www.junit.org/
* **DBUnit**, é uma extensão dos casos de teste do Junit mais complexo que realiza testes unitários voltado para a base de dados.

#### Testes Funcionais

Teste caixa preta que simula a interação do usuário com o sistema, geralmente baseado em cenários de testes extraídos dos casos de uso ou de descrições das funcionalidades. Representam um dos testes de maior nível de granularidade utilizado em fases mais avançadas do ciclo de desenvolvimento. (PRESSMAN, 1995)

### Hipótese

* Com o desenvolvimento de um sistema de monitoramento de venda de recargas por antena na Mcel, com base em metodologias e plataformas ágeis, diminuirá os recursos necessários para o desenvolvimento, aumentará o controlo e melhorará a gestão das vendas de recargas.
* Com o desenvolvimento de um sistema de monitoramento de venda de recargas por antena na Mcel, com base em metodologias tradicionais, permitirá maior estabilidade e administração do sistema;

### Variáveis de investigação

As variáveis de investigação deste projecto são:

* Tolerância a erros;
* Acessibilidade de serviços;
* Planos de redundância;
* Segurança;
* Autorização;
* Autenticação;

### Métodos de Investigação

Como método de investigação usada foi a investigação empírica, pois o processo de pesquisa foi baseado em fenómenos observáveis na realidade. Este trabalho teve as seguintes fases:

* Levantamento dos requisitos com base em entrevistas estruturadas com grupo de Marketing e a recolha do *Concept Document[[1]](#footnote-1)* (CD).
* Analise e observação do estado actual de registo de TopUps e antenas.
* Modelação do projecto usando o UML e a descrição de processos.
* Desenvolvimento do sistema com base em paradigmas de desenvolvimento ágil usando a plataforma OutSystems.
* Testes Unitários ao sistema e testes com base no *User Aceptment Test* (**UAT**).

### Resultados esperados de investigação

Espera-se com a realização deste projecto o alcance dos objectivos pretendidos que são de desenvolver um sistema de Monitoramento de Venda de Recargas na Mcel, respeitando todas as normas e regras usadas no paradigma de desenvolvimento de *software* e bem como as normas e regras da empresa e poder de forma adequada responder a todas variáveis de investigação.

## Administração e Controlo de Investigação

### 4.2 Cronograma da investigação

|  |  |
| --- | --- |
| Actividade | Calendarização |
| Sem 1 | Sem 2 | Sem 3 | Sem 4 | Sem 5 | Sem 6 | Sem 7 | Sem 8 | Sem 9 | Sem 10 | Sem 11 | Sem 12 | Sem 13 | Sem 14 | Sem 15 | Sem 16 |
| Leitura da documentação sobre o registo de venda de recargas |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Análise do sistema actual de registo de Venda de recargas |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Especificação e validação de requisitos. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Definição de Tecnologias a utilizar |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Análise de Riscos |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Modelação de casos de uso |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Desenho de diagramas de sequência, actividade e classe. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Codificação do sistema |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| *Deploy*/Publicação do Sistema |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Testes Unitários, Sistema, Integração. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Elaboração do Relatório de Testes |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Resolução dos erros encontrados |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Testes com base no UAT |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Elaboração do Relatório de Testes |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Elaboração da documentação Técnica e da utilização. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Elaboração do Manual do utilizador |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Formação ao grupo de formadores da empresa. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# REFERÊNCIAS BIBLIOGRÁFICAS

* Cass, S., Diakopoulos, N., & Romero, J. J. (19 de Dezembro de 2014). *http://spectrum.ieee.org/static/interactive-the-top-programming-languages*. Obtido de Interactive: The Top Programming Languages - IEEE Spectrum: <http://spectrum.ieee.org/static/interactive-the-top-programming-languages>
* Correia, E. A. (2009, Julho 10). Concepção de Software para Gestão de Horários Escolares. Coimbra, Portugal.
* Da Costa, A. D. (1 de Agosto de 2011). Testes de Software Testes de Software. Brazil.
* IST, F. (Agosto de 2014). FenixEdu Bible. Lisboa, Portugal.
* LAKATOS, E. M., & MARCONI, M. d. (2000). *Metodologia científica, 3ed.* São Paulo: Atlas.
* LARMAN, C. (2004), *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design ans Iterative Development.* 3th CITY: Addison Wesley Professional
* MARCONI, M. A.; LAKATOS, E. M. (2009), *Metodologia de trabalho cientifico.* 7ª Ed. São Paulo: Editora Atlas S.A. 277p
* MARCONI, M. A.; LAKATOS, E. M. (2012), *Técnicas de pesquisa.* 7ª Ed. São Paulo: Editora Atlas S.A. 277p
* Oracle. (19 de Dezembro de 2014). *Oracle*. Obtido de Java Software: <https://www.oracle.com/java/index.html>
* OUSYSTEMS (10 de Julho de 2015): [http://www.outsystems.com](http://www.outsystems.com/)
* POPPER, K. S. (1975). *A Lógica da pesquisa científica.* São Paulo: Cultrix.
* PRESSMAN, R. S (2001), *Software Engineering: a practitioner’s approach.* 5th New York: McGraw-Hill Companies 860p
* PRESSMAN, R. S (2010), *Software Engineering: a practitioner’s approach.* 7th New York: McGraw-Hill Companies 895p
* PRESSMAN, R. S. (1995). *Engenharia de Software.* Brasil: Makron Books.
* RICARTE, I. L. (2001). *Programação Orientada a Objetos: Uma Abordagem com Java.* Retrieved Julho 10, 2014, from <http://www.dca.fee.unicamp.br/cursos/PooJava/Aulas/poojava.pdf>
* RUBIRA, C. (n.d.). Retrieved Julho 10, 2014, from http://www.ic.unicamp.br/~zanoni/mc336/2012-2s/java/LivroJava/LivroJava-cap\_15.pdf
* Silva, P. N. (n.d.). Retrieved Agosto 20, 2014, from <http://paginas.fe.up.pt/~eol/MIACC/2000/GHES/definicao_de_horario.html>
* TEOFELO, D. (n.d.). Retrieved Julho 5, 2014, from Daniel Teofelo: http://danielteofilo.wordpress.com/2007/10/18/o-que-e-uml/
* VARAJAO, J. (2005). *Arqitectura da Gestao de Sistemas de Informaçao.* Lisboa, Portugal: FCA.

1. *Concept Document* é um documento da Mcel cuja uma das funções é descrever os requisitos do sistema/serviço. [↑](#footnote-ref-1)