****

**Universidade Eduardo Mondlane**

**Faculdade de Engenharia**

**Departamento de Engenharia Electrotécnica**

**Curso de Engenharia Informática**

**Engenharia de Software II**

**Sistema de Gestão de Laboratório Análises Clínicas**

**RELATORIO FINAL**

**GRUPO VI**

|  |  |
| --- | --- |
| **Estudantes** | **Docentes** |
| * Churana, Euclésia Cádia | * dr. Vali Issufo |
| * Dimande, Júlio Roberto | * dr. Bhavika Rugnath |
| * Monjane, Romário Alexandre | * Eng. Edson Michaque |
| * Muchanga, Flórida Germano | * Monitor: Richaldo Elias |
| * Sinalo, Sinalo Américo |  |
| * Viageiro, Victor Felizardo |  |
| * Zunguze, Belarmino Romão |  |

Maputo, Novembro de 2018

****

**Universidade Eduardo Mondlane**

**Faculdade de Engenharia**

**Departamento de Engenharia Electrotécnica**

**Curso de Engenharia Informática**

**Engenharia de Software II**

**Sistema de Gestão de Laboratório Análises Clínicas**

**RELATORIO FINAL**

**GRUPO VI**

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
| **Estudantes** | **Docentes** |
| * Churana, Euclésia Cádia | * dr. Vali Issufo |
| * Dimande, Júlio Roberto | * dr. Bhavika Rugnath |
| * Monjane, Romário Alexandre | * Eng. Edson Michaque |
| * Muchanga, Flórida Germano | * Monitor: Richaldo Elias |
| * Sinalo, Sinalo Américo |  |
| * Viageiro, Victor Felizardo |  |
| * Zunguze, Belarmino Romão |  |

Maputo, Novembro de 2018

**RESUMO**

Vive-se actualmente numa era onde a informática domina grande parte do mercado mundial, sendo os programas de computador (*Software*), ferramentas muito importantes para a maioria das organizações em todo o mundo, possibilitando assim que as mesmas possam realizar as suas actividades com maior eficácia e eficiência. Com as últimas inovações tecnologias e abertura do mercado mundial, o processo de desenvolvimento de *Software* tornou-se mais exigente requerendo integrantes que possuam conhecimentos sólidos nas diferentes áreas de desenvolvimento.

Mas apesar desses avanços há instituições que ainda não estão a par desta situação, adoptando métodos rudimentares de registo e armazenamento de informação, métodos estes que tornam difíceis a procura de informação, possibilitam de perda de informação por deterioração do material usado (papel) e inviabilizam a gestão e outros processos básicos de uma instituição.

Neste contexto, o presente relatório é resultado de um estudo de desenvolvimento de uma aplicação de gestão das Análises Clínicas, caso de estudo Hospital Central de Maputo (HCM), com uso de todas técnicas e recursos aprendidos ao longo das cadeiras de Engenharia de Software e Programação Web.

**Palavras-Chave**: Engenharia de software, software,desenvolvimento.

**ABREVIATURAS E ACRONIMOS**

|  |  |
| --- | --- |
| **CVS** | Concurrent Version System |
| **ESWII** | Engenharia de *Software* II |
| **IDE** | Integrated Development Environment |
| **PHP** | Hypertext Preprocessor |
| **SGLAC** | Sistema de Gestão de Analises Clinicas |

**INDICE DE FIGURAS**

**INDICE DE TABELAS**

INDICE

[1 INTRODUÇÃO 1](#_Toc530543344)

[1.1 Formulação do Problema 1](#_Toc530543345)

[1.2 Metodologia de Trabalho do Grupo 2](#_Toc530543346)

[Classificação da metodologia de Pesquisa 2](#_Toc530543347)

[1.3 Objectivos 3](#_Toc530543348)

[1.3.1 Objectivo Geral 3](#_Toc530543349)

[1.3.2 Objectivos Especificos 3](#_Toc530543350)

[2 REVISÃO TEÓRICA 4](#_Toc530543351)

[2.1 Gerenciamento de Riscos no Desenvolvimento de *Software* 4](#_Toc530543352)

[2.2 Planeamento de Projectos de *Software* 5](#_Toc530543353)

[2.3 Gerenciamento de Qualidade de *Software* 5](#_Toc530543354)

[2.4 Gerenciamento de Configuração de *Software* 6](#_Toc530543355)

[2.5 Uso de Sistemas de Controle de Versões 6](#_Toc530543356)

[2.5.1 Git 7](#_Toc530543357)

[2.6 Laravel 8](#_Toc530543358)

[2.6.1 Surgimento 8](#_Toc530543359)

[2.7 Redmine 9](#_Toc530543360)

[3 CONCLUSÕES 10](#_Toc530543361)

[4 REFERENCIAS BIBLIOGRAFICAS 11](#_Toc530543362)

[5 ANEXOS 12](#_Toc530543363)

# INTRODUÇÃO

O Sistema de Gestão do Laboratório de Análises Clínicas é direccionado a gestão do Laboratório de Analises Clínicas do Hospital Central de Maputo. Este sistema tem como objectivo, em primeiro lugar a informatização do sistema actual de gestão do Laboratório de Analises Clínicas.

O SGLAC tem como objectivo o armazenamento da informação processada no LAC, dinamizar o processamento e troca da informação processada, garantir a segurança e acesso da mesma informação no sentido de que a informação só seja acedida a quem é de direito, fornecer um ambiente de gestão com informação de tempo real.

De referir que o acesso a este sistema é feito pela Internet, tanto pelos usuários internos (recepção, técnicos da instituição e gestores) como usuários externos (pacientes que efectuam requisições de análises).

O sistema esta divido em módulos que representação um grupo de actividades relevantes na gestão do LAC. De forma resumida seguem-se os módulos deste sistema: Autenticação, Atendimento, Gestão de Requisições Clínicas, Estatísticas e Cadastro.

## Formulação do Problema

Actualmente, o Laboratório de Análises Clínicas tem recebido os utentes e regista-os manualmente, preenchendo formulários ou consultando a ficha destes, caso sejam pacientes provenientes das enfermarias do hospital, causando não só demora no atendimento, mas também lentidão nos processos de gestão, busca e organização da informação, perdas de formulários e resultados, demora nas estatísticas realizadas, entre outros.

Visto que a informatização na área de saúde consiste também no estudo da estrutura e do processamento da informação usada para a tomada de decisão, no âmbito da disciplina, a realização deste trabalho permite tornar todos os processos em volta da gestão das análises clínicas automáticos através de uma aplicação web.

## Metodologia de Trabalho do Grupo

Tendo em conta o objectivo geral do trabalho, foi aplicada a pesquisa exploratória de modo a permitir não só a fundamentação teórico-científica do tema, mas também a criação de uma aplicação de gestão para o laboratório de análises. Paralelamente, foi realizada uma pesquisa aplicada, com a finalidade de empregar os conhecimentos adquiridos, durante o curso para a implementação da solução que se propõe neste trabalho.

**Classificação da metodologia de Pesquisa**

A metodologia utilizada no desenvolvimento do presente trabalho pode ser classificada:

**Quanto à técnica aplicada:**

* **Pesquisa documental** – segundo Marconi e Lakatos (2010), a fonte de dados restringe-se a documentos (no caso concreto usou-se o Documento de requisitos);
* **Pesquisa bibliográfica** – consistiu no recurso a fontes secundárias, ou seja, consultou-se a biblliografia pública (livros físicos e electrónicos) que versa assuntos relacionados com o tema em estudo;

**Quanto à natureza:**

* **Pesquisa aplicada** – permitiu gerar conhecimentos sobre a matéria em questão, de modo a garantir que tanto os colegas como os outros pesquisadores possam usar esta fonte para a aplicação teórico-prática, no que tange a trabalhos de culminação de estudos.

**Quanto à abordagem do tema:**

* **Pesquisa qualitativa –** segundo Soares (2003), esta abordagem permite ao autor fazer a pesquisa com maior grau de profundidade e dá-lhe a possibilidade de contribuir com novos conhecimentos.

**Quanto aos objectivos:**

* **Pesquisa exploratória –** para além da fundamentação teórica dos assuntos que foram abordados, no presente trabalho, desenvolveu-se uma aplicação para a gestão do laboratório de análises clínicas

## Objectivos

### Objectivo Geral

* Desenvolver uma aplicação de gestão de Laboratório de Análises Clínicas do Hospital Central de Maputo

### Objectivos Especificos

* Apresentar conceitos relativos ao desenvolvimento de *software*;
* Descrever todos passos necessários para a realização de projectos de *software*;
* Identificar as principais etapas de um projecto de *software*;
* Desenvolver uma aplicação web para o laboratório de análises clínicas utilizando ferramentas colaborativas.

# REVISÃO TEÓRICA

O desenvolvimento de software é um trabalho complexo que envolve criatividade e conhecimento num processo construtivo e, principalmente, colaborativo. Um desenvolvimento eficaz conta com um conjunto de profissionais interdisciplinares e capacitados que utilizam recursos técnicos e ferramentas apropriadas para entregar valor - e não apenas um simples produto.

No decorrer dos projectos de desenvolvimento de software, poderão encontrar-se adversidades, que requerem maior atenção como:

* Falha na comunicação - erro na compreensão das necessidades do projecto;
* Falta de conhecimento - decisão errada sobre a tecnologia que será utilizada;
* Falta de sinergia - equipe de desenvolvedores do projecto não se entende.

Não obstante a essas adversidades, existem conjuntos de boas praticas e ferramentas que podem ser usadas para auxiliar a desenvolver software de qualidade, para o caso em estudo SGLAC fez-se acompanhar de alguns documentos que serão descritos indicando a importância de cada um deles ao longo do trabalho e no final o anexo composto por todos esses documentos.

## Gerenciamento de Riscos no Desenvolvimento de *Software*

O gerenciamento de riscos do projecto é uma da área do conhecimento com os seguintes processos: panejamento do gerenciamento de riscos, identificação de riscos, análise qualitativa de riscos, análise quantitativa de riscos, panejamento de respostas a riscos e monitoramento e controle de riscos.

**Gerenciamento de Riscos do Projecto**

De acordo com MARTINS (2007, p. 66), gerenciamento de riscos é o meio pelo qual estas incertezas são sistematicamente gerenciadas, permitindo avaliar e enfrentar os riscos, evitando o evitável, controlando o controlável e minimizando a imprevisibilidade**.**

Os objectivos do gerenciamento de riscos do projecto são aumentar a probabilidade e o impacto dos eventos positivos e diminuir a probabilidade e o impacto dos eventos adversos ao projecto. Os processos são: planeamento do gerenciamento de riscos, identificação de riscos, análise qualitativa de riscos, análise quantitativa de riscos, planeamento de respostas a riscos e monitoramento e controle de riscos, (PMBOK, 2004, p. 237).

## Planeamento de Projectos de *Software*

O planeamento do projecto é responsável pela quantificação do tempo e orçamento que um projecto custará, reduzindo, assim, as incertezas relativas aos riscos e custos. Nesta etapa o panejamento das actividades é detalhado, os produtos a serem gerados são especificados, os índices de controles são definidos e o cronograma detalhado juntamente da WBS. São estabelecidos os marcos principais do projecto, conforme critérios gerais (entrega de produtos finais) ou específicos (encerramento de uma fase ou um marco financeiro, por exemplo, data de facturamento).

A finalidade do panejamento do projecto é criar um Plano do Projecto que possa ser utilizado para guiar tanto a Execução quanto o Controle do Projecto.

O planeamento de *software* é uma tarefa importante e inclui:

* Identificar tarefas e alocá-las aos membros da equipe;
* Antecipar problemas e preparar soluções.

## Gerenciamento de Qualidade de *Software*

Um dos principais desafios encontrados pelos profissionais é definir o que é qualidade na actualidade. Muitas são as definições de qualidade de *software* propostas hoje em dia. Qualidade de *software* pode ser definida como: “Conformidade a requisitos funcionais e de desempenho explicitamente declarados, a padrões de desenvolvimento claramente documentados e a características implícitas que são esperadas de todos *software* profissionalmente desenvolvido.”(PRESMANN, 2007) Ou seja a qualidade de um *software* se refere não só a padrões de desenvolvimento, mas também garantir que o produto final satisfaça as expectativas do cliente, dentro daquilo que foi requisitado.

Em todas as fases do processo de desenvolvimento de um *software* acontecem defeitos e imprevistos, eles são causados geralmente por erros lógicos, de interpretação, desconhecimento de técnicas, falta de atenção, de motivação, falta de requisitos detalhados e outros. Sabendo disso pode-se afirmar que a garantia de qualidade não é algo que deve ser pensado no fim do projecto, mas sim durante todo o processo de desenvolvimento do projecto ou produto. Essa actividade irá ajudar a remover os defeitos existentes em cada fase. O fato de ser detectado no começo do desenvolvimento traz benefícios para o projecto, pois quanto mais tarde uma falha for corrigida mais cara será sua correcção.

A garantia da qualidade no desenvolvimento de um *software* tem uma série de prevenções que podem ser tomadas para que os problemas que possam por ventura surgir num projecto, sejam mínimos, ou não existam. E existem técnicas/práticas que podem nos auxiliar na missão de manter a qualidade do *software*. Entre elas estão a utilização das práticas, CMM, CMMI, norma ISO /IEC 9126 e também os testes.

## Gerenciamento de Configuração de *Software*

Configuração de um sistema é uma colecção de versões específicas de itens de configuração (hardware, firmware ou software) que são combinados de acordo com procedimentos específicos de construção para servir a uma finalidade particular.

A gerência de configuração vem sendo estudada desde os anos 1960 (Berlack, 1992). Inicialmente, era aplicado da mesma forma para software e hardware, sendo que no final dos anos 60 já haviam padrões de gerência de configuração específicos para software.

**A gerência de configuração de software é um processo abrangente, ao mesmo tempo técnica e gerencial, que se aplica a todas as atividades de engenharia de software, e pode ser visto como um dos principais elementos que compõem o sistema de garantia de qualidade de uma empresa de informática (Leblang, 1988)**. O processo visa identificar e definir os itens considerados relevantes ao projecto, controlar as modificações dos itens, registrar e reportar a situação dos itens e das requisições das alterações, garantir a integridade e consistência dos itens e controlar o armazenamento, manipulação, liberação e entrega dos itens (ISO/IEC 12207, 1995).

## Uso de Sistemas de Controle de Versões

O desenvolvimento de software é quase sempre realizado em equipe e controlar as alterações de cada desenvolvedor de forma que se mantenha a integridade do código fonte do software é uma necessidade recorrente. Por isso, o uso de uma ferramenta que permita o controle de versão de forma que múltiplos usuários possam realizar alterações de forma concorrente pode trazer melhorias no processo de desenvolvimento.

Um sistema de versões concorrente (CVS na sigla em inglês) é uma ferramenta para gerenciar o desenvolvimento de software colaborativo que permite que um grupo de desenvolvedores trabalhe no mesmo conjunto de arquivos fontes de forma que as alterações sejam integradas e persistidas em um repositório que guarde todas essas alterações. O uso de uma ferramenta CVS tona-se imprescindível para que o controle adequado da evolução do software seja atendido e que a produtividade melhore, deixando o desenvolvedor focado em solucionar problemas relativos ao domínio do desenvolvimento ao invés de ter que se preocupar com o estado de cada arquivo que integra a solução do projecto.

O uso de ferramentas que permitem esse controle vem se tornando uma prática comum na indústria e em projectos de software livre, mas ferramentas para uso em ambientes integrados de desenvolvimento (IDE, na sigla em inglês) ainda não funcionam de forma harmónica dependendo da plataforma de desenvolvimento e da IDE utilizada. Neste artigo analisaremos o emprego de um sistema de versionamento integrado a uma IDE de mercado no desenvolvimento de aplicações web. Apresentaremos um estudo de caso demonstrando as ferramentas disponíveis para essa integração, o uso, os benefícios e as eventuais limitações dessa integração e o que pode ser aprimorado na ferramenta utilizada.

### Git

O Git foi criado por desenvolvedores do Kernel do Linux e tem como objetivo **versionar** os arquivos. Por versionar, quer-se dizer que cada arquivo devidamente salvo em um repositório git é registrado como alterações baseadas desde sua primeira versão, ou seja, quando o arquivo em questão é registrado pela primeira vez no repositório. Para o git, cada **commit** (conjunto de alterações) é **visto** como snapshots, ou seja, cópias do estado no qual o repositório estava **no momento em que o commit** foi feito. Ou seja, em um repositório git, consegue-se **voltar ao estado** de um commit anterior a qualquer momento, podendo por exemplo, restaurar o estado após "apagar" aquele arquivo importante ou ver ou entender melhor como o sistema funcionava anteriormente, por exemplo (para o caso de existir alguma tabela mais antiga no base de dados que, por algum motivo, não foi modificada também).

Mas, como que o Git sabe a ordem dos commits? Simples, para o Git, cada commit tem os chamados "parents", que são os commits "anteriores" à ele. Nota-se que uso do plural de "anterior", pois no Git há situações no qual um único commit pode ter mais de um "parent" e há uma situação no qual um determinado commit também pode não ter "parent" (é o caso, por exemplo, do primeiro commit feito em um repositório).

## Laravel

O Laravel é um framework web baseado em PHP para a construção de aplicações web de ponta usando suas sintaxes significativas e graciosas. Ele vem com uma forte coleção de ferramentas e fornece arquitetura de aplicativos. Além disso, inclui várias características de tecnologias como ASP.NET MVC, CodeIgniter, Ruby on Rails e muito mais. Este framework é um framework de código aberto. Ele facilita os desenvolvedores economizando muito tempo e ajuda a reduzir o raciocínio e o planejamento para desenvolver o *website* inteiro a partir do zero. Junto com essa segurança do aplicativo também é o Laravel cuidar. Daí todas as suas características podem impulsionar o ritmo de desenvolvimento web para você. Se alguém estiver familiarizado com o básico do PHP, juntamente com alguns scripts PHP intermediários, então o Laravel pode criar seu trabalho mais facilmente.

### Surgimento

O Laravel foi desenvolvido e criado por Sir Taylor Otwell como uma tentativa de dar um excelente substituto para o framework PHP mais antigo chamado CodeIgniter. E isso foi porque o CodeIgniter não oferecia recursos tão bons como suporte para autenticação embutida do cliente e autorização adequada do usuário. Em 9 de julho de 2011, o Laravel lançou sua primeira versão beta e, mais tarde, no mesmo mês, o Laravel 1 foi lançado. Além da autenticação, o Laravel também possui suporte embutido para localização, visualizações, tratamento de sessões, roteamento da solicitação para o controlador específico e outros recursos surpreendentes.

**Factos Sobre Laravel**

* O Laravel foi desenvolvido por Taylor Otwell em julho de 2011 e foi lançado mais de cinco anos após o lançamento do Codeigniter.
* O Laravel é um framework web baseado em PHP como o Codeigniter.
* O Laravel é um dos frameworks PHP de código aberto.
* O Laravel segue o padrão arquitetural model-view-controller (MVC).
* O Laravel é um dos frameworks PHP mais populares depois do Codeigniter.

**Características de Laravel**

* Controladores de roteamento
* Gerenciamento de configurações
* Testabilidade
* Autenticação e autorização de usuários
* Modularidade
* Recursos do ORM (Object Relational Mapper)
* Fornece mecanismo de modelo
* Esquemas de construção

## Redmine

O Redmine é um software gratuito e mundialmente utilizado de gerenciamento de projetos

que visa manter o registro de todas as tarefas desenvolvidas, contendo Calendário e

Gráfico de Gantt para ajudar na representação visual dos projetos e seus prazos. Ele foi

modelado pela Diretoria de Gestão de Tecnologia da Informação (DGTI) especificamente

para atender às necessidades do IF Sertão-PE.

Sua função principal é facilitar o acesso, registro e acompanhamento das informações

referentes aos projetos desenvolvidos por diversos setores do IF Sertão-PE que tenham

manifestado interesse na utilização da ferramenta.

# CONCLUSÕES

O desenvolvimento do Sistema de Gestão de Analises Clinicas, fruto de um trabalho que teve inicio no primeiro semestre do presente ano (2018) , quando foi produzido sua primeira versão, tem evoluído muito, apesar de manter sempre a sua característica básica de uma ferramenta voltada para o ambiente de gerência hospitalar, abre-se para tantas novas possibilidades sempre criadas com a evolução dos computadores e com a Internet.

Como fruto deste trabalho, surge a base para um trabalho ainda maior, uma vez que todos os esforços têm-se concentrado em tornar concreta uma ferramenta que possa alavancar a criação de outros recursos, potencializando o funcionamento da unidade hospitalar (HCM).

No que diz respeito ao uso do Sistema de Gestão de Analises Clinicas na Internet, além do que já foi explanado, foi a criado um conjunto de Classes que produzem uma interface em HTML. Desta forma poderá haver uma aplicação sendo executada em um servidor enquanto, através de uma interface HTML o usuário envia comandos para a aplicação, que por sua vez produz novas páginas HTML com os resultados. Enfim, a aplicação rodará em um servidor, que não é a máquina do usuário propriamente dita, e através do protocolo HTTP - *Hyper Text Transfer Protocol* - ela interage com o navegador deste usuário.

# REFERENCIAS BIBLIOGRAFICAS

PMBOK, Project Management Institute (PMI). PMBOK - Project Management Body  
of Knowledge. 3ed. 2004. Disponível em:<http://www.pmimg.org.br>, Acessado em: 11 out. 2018.

MARTINS, José Carlos Cordeiro. Gerenciando projetos de desenvolvimento de  
software com PMI, RUP e UML. 4ed. Rio de Janeiro: Brasport, 2007.

PRESSMAN, R. S. “Engenharia de Software”, tradução José Carlos Barbosa dos Santos,  
São Paulo: Pearson Makron Books, 2007.

Soares, E., 2003. Metodologia Científica. São Paulo: Atlas S.A.

Marconi, M. . d. A. & Lakatos, E. M., 2003. Fundamentos de Metodologia Científica. 5 ed. São Paulo (SP): Atlas

# ANEXOS