# Teste de software utilizando inteligência artificial

## Objetivos

* Propor ferramentas comerciais para automatizações de testes de SW
* Desenvolver soluções para testes de SW com uso de IA para melhorar a qualidade dos testes e agilizar o tempo de execução

## Identificação do Problema

Com a inserção no ambiente de desenvolvimento da metodologia Ágil, é necessário também a adequação dos testes utilizando a mesma metodologia.

Os projetos são divididos em sprints, que normalmente duram de 3 a 4 semanas, e por isso viu-se necessidade de aumentar a velocidade de execução dos testes dos sistemas sem diminuir a qualidade. A automação de testes é uma boa ideia, mas, mesmo assim, esbarra na variável tempo, pois criar planos e casos de testes, demanda tempo e pessoal.

O sistema descrito a seguir tenta suprir essa lacuna.

## Proposta

O sistema será desenvolvido com o propósito de:

* Gerar dados para testes
* Gerar planos de testes
* Gerar relatórios de testes
* Identificar anomalias
* Gerar relatórios de anomalias
* Receber resultado de testes de sistema de testes automatizados (Selenium, Jmeter, MFCTools, Cypress, etc.)
* Reportar anomalias nos testes para sistemas externos de gestão/ gestores/DEVs

## Arquitetura

Por ser modular poderá fazer interface com qualquer sistema que deva ser testado.

Diagrama

Descrição gerada automaticamente

### Subsistemas

O sistema Executor de Testes é dividido em módulos e esses módulos possuem subsistemas, descritos a seguir, e que farão parte do desenvolvimento de cada fase mostrada mais adiante.

**Módulo gerador de relatórios**: É formado por um sistema gerador de relatórios e pelo agente de teste automatizado externo. O gerador de relatório pode ser construído com ferramentas visualizadoras de dados específicas para esse fim (ainda não definida).

**Módulo alimentador da base de conhecimento**: É formado por uma Base de Dados e por um módulo interpretador de requisitos. A base de dados pode ser alimentada tanto pelo modulo interpretador de requisitos como por um agente externo (analista de testes, analista de sistemas, etc.). A base de dados deverá ser alimentada com uma linguagem textual lacônica, ou seja, curta e concisa, para cada requisito inserido.

**Módulo Identificador de Anomalias**: É um subsistema que recebe os resultados dos testes executados pelos módulos de testes automatizados externos e pelo módulo executor de testes. De posse desses resultados, é possível através de sistemas de Inteligência Artificial, analisar e identificar dados anômalos ao sistema em teste.

**Módulo Executor de Testes**: É formado pelo Planejador de Testes e Executor de Testes. O módulo planejador de testes é o responsável por interpretar os dados da base de conhecimento e criar casos de testes e planos de testes e enviá-los ao executor de testes. O executor de testes, executa os planos de testes, gera os resultados e envia ao identificador de anomalias para análise.

Diagrama

Descrição gerada automaticamente

## Roadmap

O processo de Desenvolvimento do Sistema Executor de Testes pode ser dividido em fases. Abaixo estão divididas essas fases de desenvolvimento.

**Fase 1**: Integração de softwares de automação de testes com o **Módulo Gerador de Relatórios**. Essa fase foi definida como primeira, pois já existem ferramentas de softwares disponíveis e já configuráveis para uso no sistema em teste. Através do resultado dos testes dessas ferramentas já é possível em um primeiro momento, gerar relatórios para DEVs e Gestores.

**Fase 2**: Essa fase contempla o desenvolvimento de um módulo que contenha a **Base de Conhecimento** para o desenvolvimento das fases posteriores. Essa base de conhecimento será alimentada com os requisitos do sistema.

**Fase 3**: Desenvolvimento do **Identificador de Anomalias**, pois a partir desse ponto já existe a base de conhecimento, essencial para a identificação de anomalias.

**Fase 4**: Desenvolvimento do **Executor de Testes**.

Diagrama

Descrição gerada automaticamente

### Desenvolvimento Modular

O desenvolvimento do sistema pode ser dividido em sprints tendo assim a iteração entre os subsistemas de cada fase.

**Fase 1**:

**Sprint 1**:

Definir e implementar a ferramenta para visualização de dados.

Interface de comunicação com o sistema de automatização de teste.

Analisador de resultados de testes.

**Sprint 2**:

Visualizador de dados.

Interface com sistema de gestão.

Diagrama

Descrição gerada automaticamente

**Fase 2**:

**Sprint 3**:

Implementar a base de conhecimento

**Sprint 4**:

Interpretador de requisitos

Gerador de dados de teste

Diagrama

Descrição gerada automaticamente

Fase 3:

Sprint 4:

Identificador de Anomalias

Interface com o Analisador de resultados

Diagrama

Descrição gerada automaticamente

Fase 4:

Sprint 6:

Interface com a base de conhecimento

Planejador de casos de testes

Sprint 7:

Planejador de planos de testes

Interface com executor de testes

Sprint 8:

Executor de Testes

Interface com sistema em teste

Interface com Identificador de Anomalias

Diagrama

Descrição gerada automaticamente

### Estimativa de Horas

A estimativa de horas para o desenvolvimento do sistema está descrita na tabela abaixo.

|  |  |
| --- | --- |
| **Fases do Projeto** | **Tempo Estimado em horas** |
| **Projeto Executor de Testes** | |
| **Fase 1: Módulo gerador de relatórios** | **320** |
| ***Sprint 1:*** |  |
| Definir e implementar a ferramenta para visualização de dados. | 60 |
| Interface de comunicação com o sistema de automatização de teste. | 40 |
| Analisador de resultados de testes. | 60 |
| ***Sprint 2:*** |  |
| Visualizador de dados. | 80 |
| Interface com sistema de gestão. | 80 |
|  |  |
| **Fase 2: Módulo alimentador da fase de conhecimento** | **480** |
| ***Sprint 3:*** |  |
| Implementar a base de conhecimento | 160 |
| ***Sprint 4:*** |  |
| Interpretador de requisitos | 160 |
| Gerador de dados de teste | 160 |
|  |  |
| **Fase 3: Módulo Identificador de Anomalias** | **160** |
| ***Sprint 5:*** |  |
| Identificador de Anomalias | 120 |
| Interface com o Analisador de resultados | 40 |
|  |  |
| **Fase 4: Módulo Executor de Testes** | **480** |
| ***Sprint 6:*** |  |
| Interface com a base de conhecimento | 60 |
| Planejador de casos de testes | 100 |
| ***Sprint 7:*** |  |
| Planejador de planos de testes | 100 |
| Interface com executor de testes | 60 |
| ***Sprint 8:*** |  |
| Executor de Testes | 80 |
| Interface com sistema em teste | 40 |
| Interface com Identificador de Anomalias | 40 |
| **Quantidade Total de Horas** | **1440** |