Testes automatizados de monitoramento da experiência do usuário (colaborador) no uso de acessos/ferramentas/serviços Web.

Utilização da linguagem de programação Python com Selenium Library e Robot Framework.

1. **MONITORAMENTO**

“O Monitoramento é uma coleta sistemática e uma análise da informação de como um projeto progride” (SHAPIRO, 2002, p. 3).

Significar ter informações que são coletas durante sua execução e que a sua análise contribui para que os responsáveis pela implementação de intervenções acompanhem os resultados alcançados e verifiquem a necessidade de ajustes nas ações para que os objetivos e metas sejam alcançados.

1. **EXPERIÊNCIA DO USUÁRIO**

A experiência do usuário e constituída de seis pilares importantes:

* Estética: Utilizam de cores certas no desenvolvimento do site para proporcionar um ambiente virtual mais agradável aos olhos de quem está utilizando.
* Usabilidade: Interface simples e interativa, mas também funcional e bonita.
* Arquitetura da Informação: Organização de conteúdos.
* Fluxos de Interação: Caminhos práticos que o usuário irá percorrer pela aplicação.
* Conteúdo: Melhorar o conteúdo que é abordado.
* Empatia: Colocar-se no lugar do usuário para compreender o que precisa ser melhorado.

1. **USER EXPERIENCE (UX)**

Termo em inglês para a experiência do usuário, criado para tipificar tudo aquilo que envolve o modo como qualquer usuário interage com o mundo ao seu redor. É tudo o que se refere a experiência do usuário com algum produto ou serviço, desde a pré até o pós-venda. Trata-se de da experiência como um todo: a maneira de experimentar o mundo, a maneira de experimentar a vida, a maneira de experimentar um serviço, aplicativo ou sistema de computador.

1. **DEFINIÇÕES IMPORTANTES**
   1. **Postman**

Postman: O Postman é uma plataforma colaborativa para desenvolvimento de APIs Web por meio do envio de requisições HTTP e análise do retorno obtido.

É utilizado principalmente pela equipe de Tools xCenter para validação de monitoramento web de APIs e WebServices.

* 1. **Applications insights**

Applications Insights: O AppInsights faz parte da suíte do Azure Monitor, utilizado para detectar e monitorar anomalias de desempenho de forma automática.

É utilizado principalmente pela equipe de Tools xCenter para instrumentação de monitoria em recursos alocados na nuvem Azure (mas também é possível capturar dados de ambientes em outras nuvens e *on premisses*).

* 1. **Zabbix**

Zabbix: O Zabbix é uma ferramenta de monitoramento de TI.

Actions: Actions são tarefas pré-programadas que são executadas de acordo com condições específicas (como por exemplo triggers).

1. **MONITORAMENTO DA EXPERIÊNCIA DO USUÁRIO ATRABÉS DE ROBÔS**
   1. **Objetivo**

Agregar valor tecnológico ao ambiente de negócios e operacional (cliente e colaborador).

**5.2 Como funciona**

Através de um editor fácil são criados caminhos por onde o usuário percorre, ou poderá percorrer, e que o robô simulará.

Trata-se de um caminho dividido em passos e são coletadas informações sobre: TEMPO DE RESPOSTA, DISPONIBILIDADE, PERFORMANCE e entre outros.

As informações podem ser enviadas para outras ferramentas de análise de dados ou softwares de gerenciamento como: Zabbix e Nagios.

Instalado em um desktop, a partir daí podendo acessar o sistema a ser monitorado.

* 1. **Teste de software**

Possibilidade de uso para o teste de softwares. Após a alteração, lançamento de atualizações ou correção de bugs podem ocorrer erros nos processos, que podem ser identificados através do monitoramento dos robôs de monitoramento da experiência do usuário.

**5.4** **Automatização de processo**

Geração de relatórios periódicos, podendo ser enviados aos responsáveis por e-mail. Tarefe essa que pode ser realizada também pelo robô.

**5.5** **Principais problemas resolvidos**

Usuários insatisfeitos, descumprimento de SLAs, não atingimento de metas de qualidade.

**5.6 O que a simulação verifica?**

Velocidade, disponibilidade de links de serviços, situações em que erros ocorrem, real dimensão do problema, impacto.

1. **MACHINE LEARNING PARA UX**

Seis algoritmos que os pesquisadores de UX podem aplicar para obter resultados significativos: Regressão, árvores de decisão, agrupamento, regras de associação, mineração de processos, redução de dimensionalidade.

Esses algoritmos possuem três características em comum:

1. Utilizado com sucesso para responder perguntas sobre usuários.
2. Produz resultados compreensíveis ao ser humano.
3. Apropriado para grandes conjuntos de dados.

Os pesquisadores de UX tem perguntas que os dados podem ajudar a responder:

1. Discover: Quando o projeto inicial, você está aberto a qualquer coisa. Técnicas de aprendizado não supervisionado ajudarão a explorar os dados quando você não tem uma resposta certa, quer aprender mais sobre os usuários e depois usá-lo para objetivos futuros.
2. Test hupotheses: Possui ideias específicas e coleta dados específicos. Pode ter várias ideias concorrentes e deseja descobrir qual delas faz mais sentido. Aprendizado supervisionado.
3. Simplify problems: Às vezes os problemas podem ser muito complexos ou grandes demais para serem resolvidos. Abrange as suas primeiras categorias de aprendizado de máquina, mesmo que não resulte diretamente em perguntas respondidas.
4. **PROCESS MINING (MINERAÇÃO DE PROCESSOS)**

Usando quando se tem uma confusão de dados de log mostrando uma sequência de eventos que os usuários executaram e deseja extrair fluxos de trabalho.

A mineração de processos é o único algoritmo de aprendizado de máquina que opera diretamente em logs de eventos não transformados e analisa os fluxos de trabalho diretamente.

A mineração de processos une técnicas de modo a fazer um conjunto de logs de eventos, descobrindo assim o que os usuários do fluxo de trabalho estão realmente fazendo.

Pode ser utilizado para responder muitas perguntas sobre o fluxo de trabalho, como:

O que as pessoas estão realmente fazendo?

Qual é o conjunto mais comum de etapas que os usuários executam?

Quais etapas são mais comuns?

Qual é a sequência mais comum de etapas?

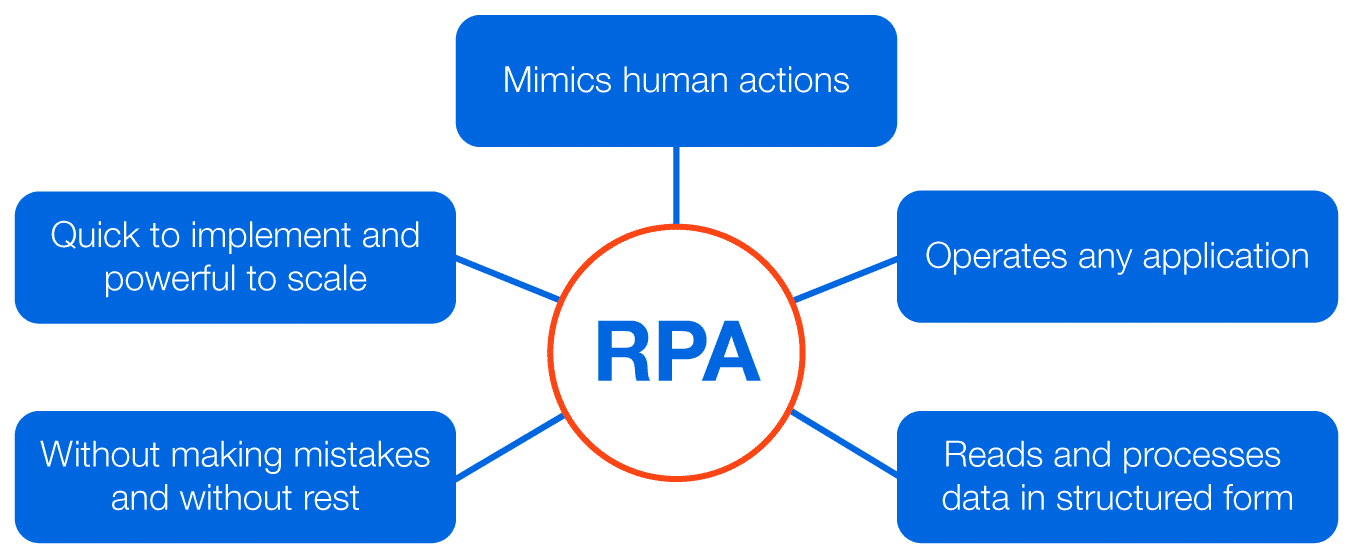
Quais etapas os usuários repetem?

O que os usuários fazem antes da atividade X ou após a atividade X?

Qual é a parte mais lenta do fluxo de trabalho?

1. **ROBOTIC PROCESS AUTOMATION (RPA)**

A Automação de Processo Robótico é a tecnologia que permite a qualquer pessoa hoje em dia configurar um software de computador ou um "robô" para emular e integrar as ações de um ser humano interagindo nos sistemas digitais para executar um processo de negócios. Os robôs RPA utilizam a interface do usuário para capturar dados e manipular aplicativos, assim como os humanos. Eles interpretam, acionam respostas e se comunicam com outros sistemas para executar uma grande variedade de tarefas repetitivas. Apenas substancialmente melhor: um robô de software RPA nunca dorme e não comete nenhum erro.



O sistema RPA observa o usuário enquanto executa uma tarefa na GUI (Graphical User Interface) e a automação entra. As tarefas executadas serão repetidas automaticamente nessa interface gráfica do usuário (GUI).

Os desenvolvedores de RPA são responsáveis ​​pela **criação, design, desenvolvimento e implementação de sistemas RPA**.

Algumas ferramentas RPA mais utilizadas: **Blue Prism, Automantion Anywhere, UI Automation, OpenSpan, UiPath, NICE, Work Fusion.**

**A Automation First Mindset - Primeira mentalidade de automação:**

É preciso observar constantemente seu trabalho e os processos em sua empresa, através das lentes do potencial de automação. Pensar na automação primeiro permite que a empresa se mova com mais rapidez e eficiência, além de ajudar a atender melhor os clientes.

**8.1 Responsabilidade e tarefas de um Desenvolvedor RPA:**

* Fornecer orientação com o design do processo
* Projetar, desenvolver, testar fluxos de trabalho de automação
* Implantação de componentes RPA: bots, robôs, ferramentas de desenvolvimento
* Apoiar a implementação de soluções RPA
* Criar documentação do processo
* Garanta a qualidade da automação (processos de controle de qualidade)

**8.2 Como funciona?**

Os robôs RPA são capazes de imitar muitas ações de usuários humanos. Eles fazem log-on em aplicativos, movem arquivos e pastas, copiam e colam dados, preenchem formulários, extraem dados estruturados e semiestruturados de documentos, raspam dados dos navegadores, fazem cálculos e muito mais.

**8.3 RPA Developer – The role (O papel)**

Design, desenvolvimento e implementação de sistemas RPA.

Criar documentação do processo.

Garantir qualidade da automação.

O desenvolvedor de RPA primeiro trabalham na **análise de processos** **e de requerimentos do cliente** e se eles podem ser estruturados ou automatizados. Com base nesses dados, os desenvolvedores trabalham no **projeto de processos para automatizar**.

1. **Selenium Web Driver**

É uma estrutura portátil para testar aplicativos da web. Fornece ferramenta de reprodução para criar testes funcionais sem a necessidade de aprender uma linguagem de script de teste.

Utilizado para automatização de testes de sistemas que permite ao usuário reproduzi-los rapidamente no ambiente real da aplicação, em função da sua integração direta com o navegador.

Possuem vantagens pois a capacidade de teste do ser humano é limitada, e tarefas repetidas são muito sujeitas a erro e a regressão diminui bastante om o uso de ferramentas automatizadas.

* 1. **Como funciona?**

**Selenium IDE**: é uma ferramenta que permite a rápida prototipagem de scripts de testes. É um *plugin* do Firefox que possibilita gravar o comportamento do usuário: página que ele acessou, textos escritos em formulários, cliques em links e botões etc. Depois de gravadas, estas ações podem ser exportadas para um script de teste em diversas linguagens de programação: Java, Python, Perl, JavaScript etc.

**Selenium WebDriver**: é uma ferramenta que oferece uma API que permite a escrita de forma mais produtiva e organizada de scripts de testes. Essa é a escolha natural quando desejamos escrever testes automatizados para aplicações *web* utilizando o Selenium. Os exemplos desse post utilizam o WebDriver.

1. **Robot Framework**

É um framework para automação de testes de aceitação. Acceptance test-driven development – ATDD). Open source, independente de sistema operacional. Possui sintaxe tabular e abordagem de palavras chave.

* 1. **A estrutura do script pode ser dividida em quatro seções:**

SETINGS: Configuração de bibliotecas, caminhos para arquivos auxiliares (page objects), contextos e hooks.

VARIABLES: Lista das variáreis a serem usadas (com descrição) e definição de algumas dessas variáveis.

TEST CASES: Seção mais importante, sem ela o teste não roda. Aqui ficam os cenários de teste, com ou sem implementação.

KEYWORDS: Definição de palavras chave ou implementação dos cenários de teste descritos acima.

**10.2 Alguns rótulos que ajudam na organização:**

DOCUMENTATION: Feita na seção SETTINGS, não devendo estar entre colchetes.

TAGS: Rótulos mais simples para os cenários. Devido a títulos de cenários geralmente grandes, criar rótulos acaba sendo a melhor opção.

ESPAÇOS E VARIÁVEIS: As variáveis no Robot são representadas por ${variável}. O Robot ignora o espaço entre palavras: ${nome\_da\_variável} = ${nome da variável}.

As divisões dos argumentos são feitas por, no mínimo, dois espaços.

Sinal de igual também pode ser ignorado: ${valor} valor ou ${valor} = valor

INDENTAÇÂO: O Python é uma linguagem [indentada](https://pt.wikibooks.org/wiki/Python/Conceitos_b%C3%A1sicos/Indenta%C3%A7%C3%A3o), ou seja, blocos de comando são separados por espaços ou tabulações, formando uma indentação visual obrigatória. Não existem símbolos de “abre” e “fecha”, o Robot herda essa formatação.

1. **Jenkins C.I**

É um servidor de automação gratuito de código aberto. Ajuda a automatizar a parte não humana do processo de desenvolvimento de software. É um sistema baseado em servidor executado em contêineres de servlet. Como o Apache Tomcat.

É uma aplicação web de Integração Contínua, utilizado para executar testes e criar artefatos de um projeto de software.

A Integração Contínua representa um conjunto de práticas que permitem que um sistema seja compilado, testado e construído em um ambiente separado e independente do de desenvolvimento. Permite que as mudanças que acontecem no código sejam integradas ao sistema mais rapidamente e de forma contínua, pois possui ferramenta (como Jenkins) que automatizam as tarefas de todo esse processo.

1. **Jira Software**

Software comercial desenvolvido pela empresa Atlassian. Ferramenta que permite o monitoramento de tarefas e acompanhamento de projetos garantindo o gerenciamento de todas as suas atividades em um único lugar.

Um projeto JIRA é um agrupamento de tarefas (Issues). Exemplos de projetos: Desenvolvimento de software, campanha de marketing, controle de chamados, controle de processos.

No Jira Software existem dois tipos de modelos de projeto: clássico e de próxima geração.

Clássico: incluem a personalização e flexibilidade avançadas para os quais o Jira foi originalmente desenvolvido,

Próxima Geração: Mostra a experiência reimaginada do Jira Software com opções simplificadas de configuração e customização.

**Jira Software**: Software Development. Plan, track and release software.

**Jira Service Desk**: IT, Customer Service. Create and manage Service Desks.

**Jira Core**: Business. Manage tasks in business projects.

Todo o time de desenvolvimento tem acesso ao projeto no Jira Software, porém somente os Scrum Masters podem criar e gerencias sprints.

Jira Administrator: Controla quais produtos Jira você tem acesso e o que você pode fazer no Jira.

Todos os projetos são feitos de Issues e Workflow.

Uma Issue é uma tarefa (Task) – São únicas e pertencem a apenas um projeto.

Diferentes projetos tem diferentes tipos de Issue.

Issue podem ser de 3 tipos:

**Bug**: Correção de problemas no código.

**Story**: Características. Descrevem funcionalidades pela perspectiva do usuário.

**Task**: Tarefas a serem concluídas.

No cartão de uma Issue estão presentes 5 informações, como: Tipo, prioridade, status, quem a criou e quem está trabalhando nela.

**TAREFAS**

1. Estudar sobre a Experiência do Usuário (UX) e sobre robôs que realizam testes automatizados de monitoramento.
2. Estudar sobre a tecnologia RPA - Robotic Process Automation (Automação de Processo Robótico) e quais são os recursos necessários para a sua criação e implementação.
3. Estudar aprofundado sobre a Selenium Library e o Robot Framework, aprendendo também como efetuar a correta instalação dos mesmos.
4. Aprender ao máximo como utilizar os recursos e comandos da Selenium Library e do Robot Framework através de material de estudo presente no YouTube: *Web Automation using Robot Framework (Selenium with Python) for Beginners*, realizando testes variados em diferentes tipos de sites.
5. Criar um Dashboard interativo com as informações coletadas na etapa de testes de maneira a tornar a leitura e análise mais agradável e rápida
6. Aprender e utilizar o método Web Scraping para absorver informações referente a velocidade de internet em download e upload.
7. Avançar na etapa de testes, com a realização de novas tentativas e experiências agora em um sistema Web utilizado pelos colaborados da área e devidamente especificado pelo Scrum Master.
8. Preparar arquivo de PowerPoint para apresentação do projeto a ser realizada no final do período de estágio.

**REFERÊNCIAS BIBLIOGRÁFICAS**

Don Norman: O termo “UX”.  
<https://www.youtube.com/watch?time_continue=101&v=9BdtGjoIN4E&feature=emb_title>

Experiência do Usuário: como atrais a atenção do público. <https://fluxoconsultoria.poli.ufrj.br/blog/tecnologia-informacao/experiencia-do-usuario/>

O que é Monitoramento?  
<https://www.portaleducacao.com.br/conteudo/artigos/educacao/o-que-e-monitoramento/44205>

OpCEM | Monitoramento da experiência do usuário através de robôs.  
<https://www.opservices.com.br/monitoramento-da-experiencia-do-usuario/>

Aplicando o Machine Learning á pesquisa do usuário: 6 métodos de Machine Learning para gerar insights de experiência do usuário.  
<https://medium.com/athenahealth-design/machine-learning-for-user-experience-research-347e4855d2a8>

Robotic Process Automation (RPA)  
<https://www.uipath.com/rpa/robotic-process-automation>

Stackoverflow  
<https://stackoverflow.com/questions/37181003/is-there-any-open-source-tools-for-rpa-robotic-process-automation>

Book: Automate the Boring Stuff with Python  
<https://automatetheboringstuff.com/>

Robot Framework  
<https://robotframework.org/>

<http://robotframework.org/Selenium2Library/Selenium2Library.html#Choose%20File>

<https://github.com/robotframework/robotframework#installation>

Como iniciar sua carreira como Desenvolvedor RPA (Robotic Process Automation)  
<https://www.profissionaisti.com.br/2019/08/como-iniciar-sua-carreira-como-desenvolvedor-rpa-robotic-process-automation/>

<https://www.freelancermap.com/freelancer-tips/12532-what-does-rpa-developer-do>

<https://blog.aimultiple.com/rpa-developer/>

Python  
<https://www.python.org/downloads/>

Testes Web com Robot Framework para Python: Introdução  
<https://www.concrete.com.br/2018/01/30/testes-web-com-robot-framework-para-python-introducao/>

Web Automation using Robot Framework (Selenium with Python) for Beginners  
<https://www.youtube.com/watch?v=UbYxkUq0Hec&list=PLUDwpEzHYYLsCHiiihnwl3L0xPspL7BPG>

<http://demowebshop.tricentis.com/register>

<http://www.practiceselenium.com/practice-form.html>

<http://testautomationpractice.blogspot.com/>

<https://selenium.dev/selenium/docs/api/java/index>

<https://swisnl.github.io/jQuery-contextMenu/demo.html>

<http://www.dhtmlgoodies.com/scripts/drag-drop-custom/demo-drag-drop-3.html>

https://admin-demo.nopcommerce.com

Web Scraping

<https://www.edureka.co/blog/python-requests-tutorial/>

<https://medium.com/horadecodar/como-fazer-webscraping-com-python-e-beautiful-soup-28a65eee2efd>

Automatizando a geração de pacotes com Jenkins

<https://www.devmedia.com.br/automatizando-a-geracao-de-pacotes-com-o-jenkins/33655>  
  
Curso Python – Biblioteca Tkinter

<https://www.youtube.com/watch?v=vmoH9I9d570&list=PLesCEcYj003ShHnUT83gQEH6KtG8uysUE&index=2>

[Create link to folder with python and tkinter](https://stackoverflow.com/questions/54780086/create-a-link-to-folder-with-python-and-tkinter)

<https://stackoverflow.com/questions/54780086/create-a-link-to-folder-with-python-and-tkinter>

Creating a Stand Alone Executable from a Python Script using PyInstaller

<https://www.youtube.com/watch?v=t51bT7WbeCM>

Jira Software

<https://www.atlassian.com/university>

<https://www.atlassian.com/software/jira/guides/getting-started/basics#step-1-create-a-project>