******

Plano de Atividades para o Trabalho de Conclusão de Curso

Modalidade: Estágio Supervisionado

Curso de Bacharelado em Ciências da Computação – Not

1º sem./2020

1. **Identificação do Aluno**

|  |  |
| --- | --- |
| **Nome:** | Gabriel luiz |
| **Nº UNESP:** | 141153628 |
| **E-mail:** | g[abriel.luiz@unesp.br](mailto:Gabriel.luiz@unesp.br) |
| **Funcionário** | Funcionário desde outubro de 2019 |

1. **Identificação da Empresa**

|  |  |
| --- | --- |
| **Razão Social:** | DAITAN LABS SOLUCOES EM TECNOLOGIA S/A |
| **Endereço:** | Galleria Office Park Ed. 1, Cj. 141, Av. Bailarina Selma Parada, 201, Campinas - SP, 13091-904 |
| **Área(s)/Setor(res) de atuação:** | Computação em Nuvem, Segurança e Comunicação, Ciência de Dados e Análise, Desenvolvimento de *Software*, Manutenção de Produto, Garantia da Qualidade e Teste |
| **Departamento/Setor**  **de realização do estágio:** | Desenvolvimento de *Software* |

1. **Identificação do Responsável na Empresa**

|  |  |
| --- | --- |
| **Nome:** | Flavio Peres |
| **Cargo:** | Gerente de Desenvolvimento |
| **Documento de comprovação de atuação na área de Computação:** | Diploma |
| **Curso Superior/Instituição:** | Bacharelado em Análise de Sistemas/PUC Campinas |

1. ***Datas***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Início das atividades na empresa:** | 01/08/2019 |  | **Início das atividades alvo do estágio:** | 01/03/2020 |
| **Término previsto para encerramento das atividades na empresa:** | Indeterminado |  | **Término das atividades alvo do estágio:** | 30/05/2020 |

1. **Identificação das áreas de aplicação envolvidas no estágio**

Introdução à Computação I e II, Estruturas de Dados I e II, Conceitos de Linguagens de Programação, Análise de Sistemas, Sistemas Operacionais I, Sistemas Operacionais II, Linguagens Comerciais de Programação, Redes de Computadores, Engenharia de Software, Sistemas Orientados a Objetos, Gerenciamento de Projetos: Teoria e Prática - Optativa

1. **Objetivos e Equipes de trabalho**

A empresa Daitan tem como clientes empresas de desenvolvimento de *software* que necessitam de mais desenvolvedores, porém que não possuem espaço ou capacidade de contratação desses funcionários. A Daitan verifica a necessidade de mais desenvolvedores na empresa e forma equipes de tamanho variável para suprir as demandas de desenvolvimento. As equipes são compostas por profissionais com diferentes tipos de perfis, habilidades e níveis técnicos, para atuar desenvolvendo aplicações conforme a necessidade do cliente.

O cliente que estou alocado se chama Symphony. A empresa possui uma aplicação com o mesmo nome (Symphony) para comunicação bancária, um *chat*, que possui ligações de vídeo, áudio, compartilhamento de tela, aplicações extras (como *plugins*) para serem instaladas dentro deste *chat* e expandir suas possibilidades de integrações com outros sistemas.

A Symphony possui em média 100 colaboradores na Daitan, divididos em pequenas equipes, para o desenvolvimento de seus produtos. A equipe que estou alocado possui atualmente 12 colaboradores distribuídos nas seguintes funções: 4 desenvolvedores back-end, 2 desenvolvedores front-end, 1 garantia de qualidade, 1 experiência de usuário, 1 devops, 1 arquiteto, 1 gerente de produto, 1 gerente de desenvolvimento.

Nossa equipe é responsável pelo desenvolvimento de *chatbots* para automações externas com o produto. Um funcionário da empresa bancária que possui o sistema Symphony pode, por exemplo, vender e comprar ações do mercado financeiro diariamente, ligando para compradores que são funcionários de outros bancos que possuem o mesmo sistema. Ao invés de precisar ligar para o outro comprador, ele poderia realizar esta operação via sistema Symphony, visto que possuem esta via de comunicação dos dados. Uma solução para evitar estas ligações entre os funcionários é criar um *chatbot* que realiza a compra e a venda destas ações sem os funcionários precisarem sair do *chat*, ou até mesmo realizar ligações.

Minha função na equipe é desenvolver essas aplicações *back-end* para os *bots* que irão automatizar o processo de compra e venda de ações de segunda instância da bolsa de valores, no qual, atualmente, os usuários deste sistema (funcionários de banco) precisam enviar cotações um a um para todos os seus compradores, informando valores, e tendo que validar também qual é a melhor venda para ele. Neste caso, a automação irá realizar a comunicação com todos os compradores por ele, e toda a negociação é feita através do *chat*, sem a necessidade de e-mails, ligações e tabelas manuais. As tarefas são divididas entre os 4 desenvolvedores back-end, onde após o planejamento do que é necessário para a conclusão do projeto é separado o que cada um deve fazer levando em consideração o prazo de entrega, dependências entre tarefas e tempo estimado para cada tarefa.

1. **Descrição das Atividades do Estágio e Cronograma**

Para este relatório de estágio é proposto o desenvolvimento de dois bots que irão comunicar entre si, no próprio *chat* da Symphony, para realizar a compra e a venda de ações do mercado financeiro de segunda instância, isto é, ações que não são mais gerenciadas pela bolsa de valores, e sim, pelos compradores e vendedores que têm em posse as ações. Um dos *bots* será responsável por anunciar a venda das ações e o outro será responsável por realizar a compra destas ações.

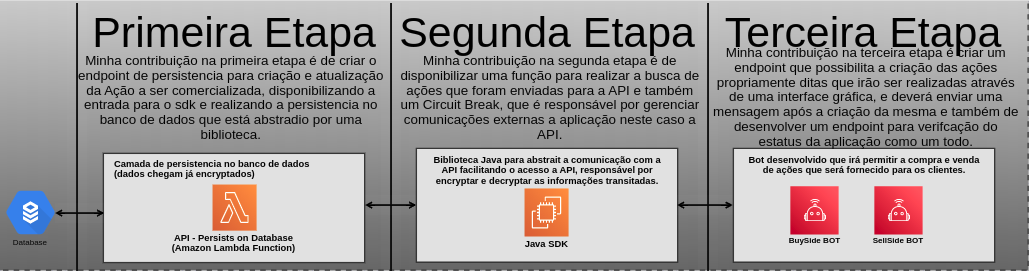
O processo a ser desenvolvido possuirá uma arquitetura dividida em três etapas. Todas as etapas utilizarão a linguagem Java, com o gerenciador de pacotes Maven, para o desenvolvimento da parte do *back-end* com algumas bibliotecas variáveis em cada etapa do processo.

A primeira parte do processo será o desenvolvimento de uma API (interface de programação de aplicações) *web* com arquitetura *serverless*, que utilizando a infraestrutura da Amazon, com os serviços Lambda e API Gateway, será disponibilizado *endpoints* (terminais de comunicação) para persistência dos dados (referentes às ações) encriptados, utilizando o *framework Spring Cloud* para disponibilizar as funções Lambda na Amazon e gerenciar classes.

Existe uma criptografia em cada mensagem enviada neste sistema, no qual essa mensagem só pode ser aberta por uma pessoa que se encontra na sala onde a mensagem foi enviada. Isto é, se possuir uma conversa no chat entre duas pessoas, apenas as duas pessoas que estão nessa sala possuem a chave para conseguir ler a mensagem em questão. A integração utilizará esta segurança para enviar mensagens referente a compra e venda de ações, onde o comprador e o vendedor possuem uma sala em comum com o *chatbot,* que realizará esta operação, possibilitando a comunicação entre eles e a segurança das informações, visto que estes dados são sensíveis.

A segunda parte será o desenvolvimento de um SDK (*Software Development Kit*), uma biblioteca, que irá realizar a abstração destas chamadas na API desenvolvida na primeira etapa, onde sua função é disponibilizar funções e objetos simples para realizar a persistência na API, onde o desenvolvedor que utiliza esta ferramenta deve instanciar essas classes e enviar na chamada das funções. O SDK é responsável por realizar a criptografia dos objetos enviados e a descriptografia quando é realizada alguma comunicação com a API. Nesta etapa não é utilizado nenhum framework.

Com a API de persistência funcionando, e o SDK pronto para ser utilizado, o próximo passo é utilizar este SDK para o desenvolvimento dos bots propriamente ditos, que devem ser capazes de processar mensagens enviadas ao chat para montar os objetos fornecidos pelo SDK e utilizá-lo para realizar a persistência. Quando o objeto é persistido, envia uma mensagem para todos os compradores ou vendedores, comunicando que há uma nova oferta de ação e todos os valores necessários.



**Figura 1:**  Diagrama da arquitetura do chatbot exibindo as etapas a serem implementadas.

A Figura 1 apresenta a arquitetura do projeto que irá ser desenvolvido neste relatório, exibindo quais são as etapas desenvolvidas e as minhas contribuições em cada etapa, exibindo o fluxo de dados entre cada etapa do projeto e o que ela representa. Neste período serão desenvolvidas seis tarefas, duas em cada etapa do processo, de acordo com o seguinte cronograma:

* Primeira Etapa:

1. 01/03/2020 a 15/03/2020: desenvolvimento de um *endpoint* para criação da venda de determinada ação, no qual a entrada será um arquivo contendo informações básicas sobre quem está enviando a solicitação e para quem, com o restante do conteúdo encriptado. Será persistido no banco de dados e depois retornado para o solicitante as informações que foram salvas.
2. 16/03/2020 a 31/03/2020: desenvolvimento do *endpoint* para atualização da ação, no qual o usuário informará o ID desta ação e os novos valores encriptados, que serão persistidos no banco de dados, e o retorno será a ação atualizada.

* Segunda Etapa:

1. 01/04/2020 a 15/04/2020: serei responsável por disponibilizar uma função para consulta de ações criadas. Irei disponibilizar uma classe com métodos públicos onde o usuário utilizará um modelo criado no SDK com filtros, onde é realizado a criptografia desses filtros e enviado para a API, e após o retorno da API os dados são decriptados e enviados como retorno de quem chamou esta função.
2. 16/04/2020 a 30/04/2020: criarei um mecanismo de *circuit break,* responsável por interromper solicitações quando a API não estiver disponível ou quando várias solicitações iguais são realizadas pelo cliente, com possibilidade de configurar o tempo que o serviço fica indisponível.

* Terceira Etapa:

1. 01/05/2020 a 14/05/2020: serei encarregado de criar o *endpoint* para gerar a venda de uma determinada ação, utilizando a biblioteca criada previamente. Irei utilizar a função responsável por persistir o dado e depois, com a confirmação, deve ser enviada uma mensagem de confirmação no chat que a ação foi criada.
2. 15/05/2020 até 29/05/2020: serei responsável pela criação de um *endpoint* para verificar o estado da aplicação como um todo, informando se o *bot* está funcionando, se ele pode se comunicar com o *chat* da Symphony e se seu acesso é disponível externamente.
3. **Bibliografia inicial**

GAMMA, Erich et al. Design Patterns: Elements of Reusable Object-Oriented Software. [S. l.: s. n.], 1994.

DEITEL, H. Java Como Programar, 8a edição, Editora Person, 2010;

SOMMERVILLE, I. Engenharia de Software, 9. ed. São Paulo: Pearson Prentice Hall, 2011.

PROJECT MANAGEMENT INSTITUTE. Um Guia do Conhecimento em Gerenciamento de Projetos (Guia PMBOK. 4. Ed. Pennsylvania: Project Management Institute, 2008.

DEITEL, H. M., DEITEL, P. J., CHOFFNES, D. R., "Sistemas Operacionais: terceira edição", Pearson Prentice Hall, 2005.

**Rio Claro, XX de XXXX de 2020.**

**Assinaturas:**

|  |  |
| --- | --- |
| Flavio Luis Peres | Gabriel Luiz |
| *Orientador do Estágio na Empresa* | *Curso de Bacharelado em Ciências da Computação* |