**Plano de trabalho**

**Desenvolver aplicação para o processo seletivo do Itaú**

**Lucas de Araújo**

**São Paulo, 2019**

**O QUE SERÁ FEITO**

1. Elaborar plano de trabalho
2. Desenvolver aplicação para coletar as últimas postagens do Twitter, dada uma determinada #tag
3. Modelar e implementar uma base de dados para armazenar as informações
4. As informações à serem armazenadas, são os twittes com as seguintes #tags: #openbanking, #apifirst, #devops,cloudfirst, #microservices, #apigateway, #oauth, #swagger, #raml, #openapis
5. Com as informações anteriores armazenadas, será necessário filtrar para exibir as seguintes informações: Quais são os 5 usuários, da amostra coletada, que possuem mais seguidores? Qual o total de postagens, agrupadas por hora do dia (independentemente da #hashtag)? Qual o total de postagens para cada uma das #tag por idioma/país do usuário que postou?
6. Deverá ser criado uma API REST para consumir as informações armazenadas e que foram aplicadas o filtro dos três requisitos anteriores. Além disso, a API deverá expor métricas de execução
7. Deverá ser desenvolvida uma página para fazer a chamada das APIs e mostrar os resultados
8. Deverá ser utilizado uma ferramenta de logging, para criar uma query e mostrar em tempo real os eventos que acontecem na execução da API criada. Importante ter ao menos uma situação de execução com erro
9. Deverá ser utilizado uma ferramenta de monitoração de métricas, criando 3 dashboards para mostrar em tempo real quantidade de execução, a latência e a quantidade de erros da API criada. Importante ter ao menos uma situação de execução com erro
10. Explicar os desvios do planejamento original e a execução, caso houver. Caso contrário, comentar os motivos para esse resultado sem desvio
11. Publicar o projeto no Github e documentar em um README.md os seguintes requisitos: Documentação do projeto; Documentação das APIs; Documentação de arquitetura; Documentação de como podemos subir uma cópia deste ambiente localmente; Manual com prints dos logs e os 3 dashboards, explicando cada um deles

**COMO SERÁ FEITO**

* Será utilizado as ferramentas Microsoft Word e Astah Community / Professional, para a elaboração do projeto
* Será desenvolvida em Python, a aplicação para coletar as métricas do Twitter
* Será utilizado o banco de dados relacional MySQL, para armazenar as informações
* Será utilizado a ferramenta Postman, para desenvolver a API REST
* Será desenvolvida uma página web, com frontend HTML5/CSS3 e backend PHP, para fazer a chamada das APIs e mostrar os resultados
* Será utilizado o Elasticsearch, para elaborar a query que exibirá em tempo real os eventos que acontecem na API
* Será utilizado o Zabbix, para criar os 3 dashboards que mostrarão em tempo real a quantidade de execução, a latência e a quantidade de erros da API
* Será utilizado o GIT, para publicar o projeto no Github.

**QUANDO SERÁ FEITO**

* O projeto iniciará na data 19/12/2019 às 08h, e será concluído na data 02/01/2020 às 22h

**PREVISTO, REALIZADO E DESVIOS**

|  |  |  |
| --- | --- | --- |
| **PREVISTO** | **REALIZADO** | **DESVIOS?** |
| Elaborar plano de trabalho | Sim | Não |
| Desenvolver aplicação para coletar as últimas postagens do Twitter, dada uma determinada #tag | Sim | Sim, de todas as bibliotecas do Python que testei, nenhuma retornou a data do tweet. |
| Modelar e implementar uma base de dados para armazenar as informações | Sim | Não |
| Armazenar os twittes com as seguintes #tags: #openbanking, #apifirst, #devops, #cloudfirst, #microservices, #apigateway, #oauth, #swagger, #raml, #openapis | Sim | Não |
| Filtrar as seguintes informações armazenadas: Quais são os 5 usuários, da amostra coletada, que possuem mais seguidores? Qual o total de postagens, agrupadas por hora do dia (independentemente da #hashtag)? Qual o total de postagens para cada uma das #tag por idioma/país do usuário que postou? | Sim | Sim, consegui filtrar somente quais são os cinco usuários, da amostra coletada e armazenada, com mais seguidores no Twitter |
| Criar uma API REST para consumir as informações armazenadas e que foram aplicadas o filtro dos três requisitos anteriores. Além disso, a API deverá expor métricas de execução | Não | Sim, iniciei o desenvolvimento da API REST, mas não consegui terminar |
| Desenvolver uma página para fazer a chamada das APIs e mostrar os resultados | Não | Sim, como não terminei de fazer a API, consequentemente não fiz a página web proposta (PHP) |
| Utilizar uma ferramenta de logging, para criar uma query e mostrar em tempo real os eventos que acontecem na execução da API criada. Importante ter ao menos uma situação de execução com erro | Não | Sim, como não terminei de fazer a API, consequentemente não fiz a query na ferramenta de logging proposta (Elasticsearch) |
| Utilizar uma ferramenta de monitoração de métricas, criando 3 dashboards para mostrar em tempo real quantidade de execução, a latência e a quantidade de erros da API criada. Importante ter ao menos uma situação de execução com erro | Não | Sim, como não terminei de fazer a API, consequentemente não criei esse dashboard na ferramenta de monitoração proposta (Zabbix) |
| Publicar o projeto no Github e documentar em um README.md os seguintes requisitos: Documentação do projeto; Documentação das APIs; Documentação de arquitetura; Documentação de como podemos subir uma cópia deste ambiente localmente; Manual com prints dos logs e os 3 dashboards, explicando cada um deles | Sim, parcialmente | Sim, como não terminei de desenvolver todo o ambiente do case, a documentação no README não ficou completa |
| O projeto iniciará na data 19/12/2019 às 08h, e será concluído na data 02/01/2020 às 22h | Sim, parcialmente | Sim, conforme alinhado, conseguir desenvolver parte do case no dia 07/01 às 16h |