

EDITAL Nº 046/2020 – LAIS/UFRN

SELEÇÃO DE PESQUISADORES PARA ATUAR NO PROJETO "PESQUISA APLICADA PARA INTEGRAÇÃO INTELIGENTE ORIENTADA AO FORTALECIMENTO DAS REDES DE ATENÇÃO PARA RESPOSTA RÁPIDA À SÍFILIS"

FASE 2 - DESENVOLVIMENTO DO PROJETO

O Projeto a ser desenvolvido deve estar de acordo com a opção de Área de Atuação selecionada na Plataforma de Processos Seletivos, caso contrário o Projeto do candidato não será avaliado.

A entrega do Projeto deverá ser realizada através da Plataforma de Processos Seletivos do LAIS até o final do dia 21/12/2020, sendo necessário informar o link do commit do projeto no GitHub que será considerado para avaliação. Por exemplo: <https://github.com/lais-huol/capacitacao-git/commit/dc125ec4d914788f43d6b98114701f018338bc05>.

Não serão aceitos links fora do padrão exemplificado acima.

Projeto I - Sistemas de informação e Sistemas inteligentes

1. Conteúdo Programático

Conforme o Anexo III do presente edital, o conteúdo programático consiste em: Programação Orientada a Objetos; Programação em Python; Desenvolvimento Web com o framework Django; Consulta a Banco de Dados; Modelagem de Banco de Dados - 1a, 2a e 3a forma normal; HTML 5.

2. Descrição do Projeto

Elaborar uma aplicação que consiste em uma carteira digital de vacinação integrada com as redes de atenção à saúde, onde o usuário do Sistema Único de Saúde

(SUS) poderá realizar o autocadastro e ser vacinado em qualquer estabelecimento de saúde que tenha disponibilidade de vacinas.

3. Requisitos do Projeto

3.1 Obrigatórios

- O projeto deve ser desenvolvido utilizando as seguintes tecnologias:
 - Python 3.6 ou superior;
 - Django 3.1 ou superior;
 - Postgres 10 ou superior.
- Qualquer tipo de biblioteca ou tecnologia adicional pode ser utilizada na construção do projeto, desde que não interfira no deploy;
- Todo o código do projeto deve ser versionado em repositório público no GitHub, com o README.md contendo as instruções para carregar os dados iniciais da aplicação e executá-la localmente, além da URL da aplicação funcionando no Heroku;
- O modelo de dados deve ser elaborado com base nas informações contidas neste documento;
- Não colocar no repositório git dados de autenticação em serviços, como o Gmail, OAuth 2, ou qualquer outro tipo de informação sensível em texto simples;
- Além disso, todos os critérios obrigatórios (apresentados na tabela do item 5) devem ser atendidos para que o projeto seja avaliado;

3.2 Facultativos

- A separação entre front-end (utilizando Vue.JS) e back-end (utilizando Django REST Framework). No mínimo as funcionalidades que envolvem o ator Paciente;

- A autenticação dos usuários, inclusive os pacientes, poderá ser feita utilizando o protocolo OAuth 2, através da integração com o Google ou outro serviço que suporte o protocolo, deixando a aplicação no Heroku funcionando com esta integração;
- Usar Websocket (com a biblioteca Django Channels) para atualizar a fila de espera, tanto no painel de chamada quanto no smartphone do paciente;
- Usar Push Notifications (PWA) para informar ao Paciente quando chegar a sua vez, mesmo quando o sistema não estiver ativo na tela do celular;
- Incluir o arquivo docker-compose.yml na raiz do projeto, de forma que seja possível reproduzir o ambiente de desenvolvimento;
- Envio de e-mail usando os servidores do GMail nas seguintes situações:
 - Quando o usuário se cadastrar
 - Quando o usuário agendar a vacina
 - Quando uma vacina for aplicada/registrada

4. Da aplicação

A aplicação é composta pelos seguintes módulos, que são postos para facilitar o entendimento lógico:

I - Módulo Cartão de Vacina, que tem por objetivo a visualização das vacinas pelo usuário

II - Módulo Agendamento de Vacina, que tem por objetivo atualizar o número de espera da lista de pacientes a serem vacinados.

III - Módulo Administrativo, que tem por objetivo realizar os cadastros de profissionais, estabelecimentos de saúde e os vínculos entre os estabelecimentos e os profissionais, além de permitir a apresentação de relatórios gerenciais.

O Diagrama de Casos de Uso a seguir apresenta as funcionalidades da aplicação distribuídas entre os módulos e atores que serão descritos a seguir.

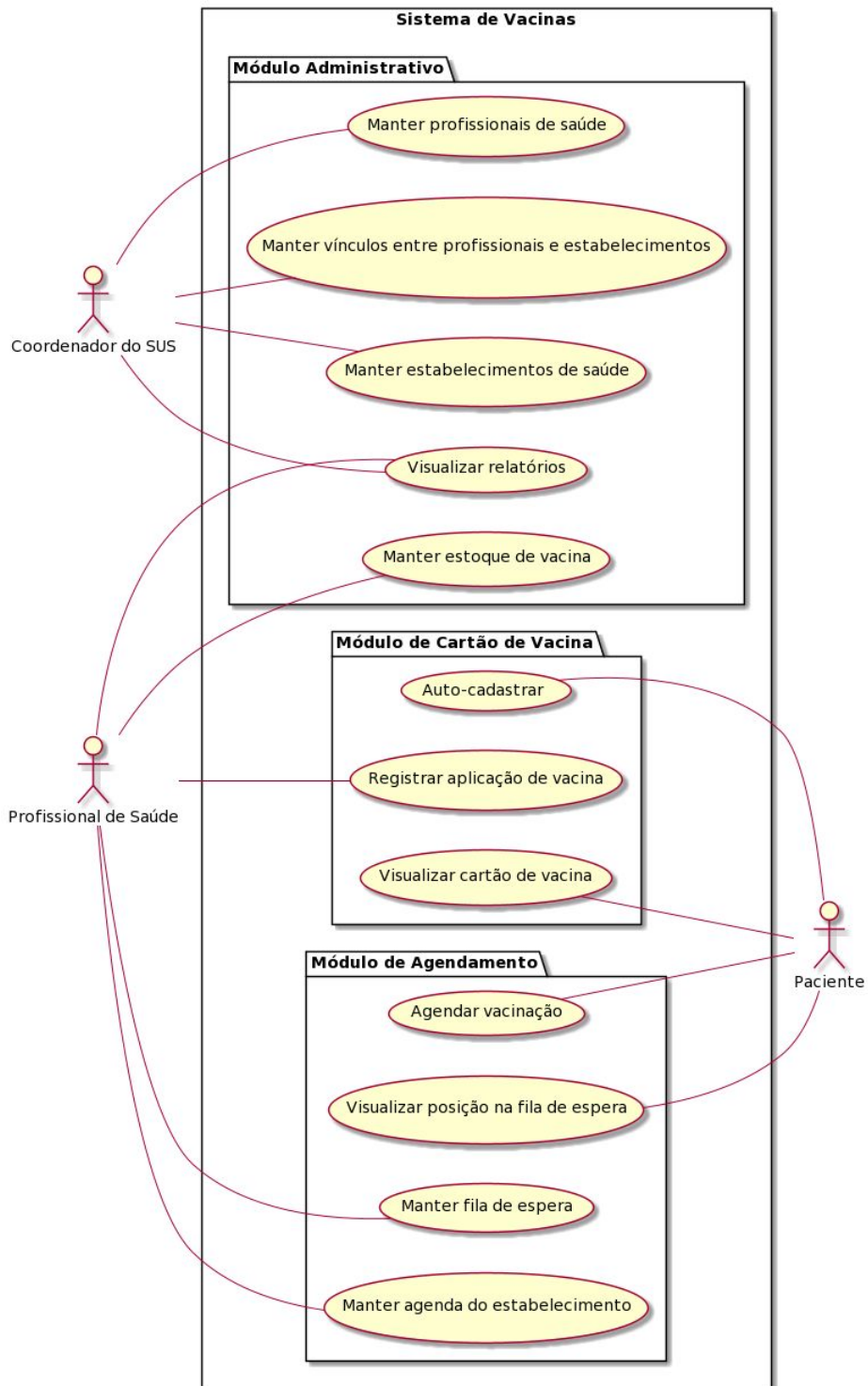


Figura 1. Diagrama de Casos de Uso do Projeto I

Atores:

- Coordenador do SUS: Cadastra as vacinas, profissionais e seus vínculos com os estabelecimentos de saúde;
- Profissional de saúde: Gerencia agenda, gerencia fila de atendimento, gerencia estoque e registra a vacina aplicada dentro do estabelecimento de saúde vinculado;
- Paciente: Autocadastro, selecionar estabelecimento para vacinação, visualizar e gerenciar seu próprio cartão de vacinação.

O Coordenador do SUS importa e cadastra UF, municípios e estabelecimentos (dados disponíveis em <https://drive.google.com/file/d/1gHBf5nWJDP2jluLp1EeOyAMkbZnykbqh/view?usp=sharing>), cadastra as vacinas, os profissionais e os vínculos dos profissionais aos estabelecimentos.

O Profissional de Saúde é o responsável por manter a agenda disponível do estabelecimento para os pacientes, que por sua vez cadastram e aprovam o agendamento de vacinação. Da mesma forma, esse ator também gerencia a chamada de Pacientes a partir da fila de espera.

Os Pacientes solicitam os agendamentos de vacinação, iniciando o processo escolhendo a vacina, UF/município do estabelecimento, e a data/hora disponível. Ao chegar no estabelecimento de saúde, os pacientes aguardam sua vez e são notificados para se dirigirem à sala de vacinação através do sistema, que emite aviso via painel de chamada e também pelo celular do próprio paciente. Após a chamada do paciente pelo Profissional de Saúde, a vacina é aplicada e registrada no cartão de vacina do Paciente. Assim como a vacina é deduzida do estoque.

Caso Paciente realize uma vacina de forma privada, o Paciente pode ele mesmo registrar a vacina, útil para registro de vacinas passadas não anotadas no sistema.

Entretanto, estas vacinas devem ser marcadas de forma a diferenciar as anotadas por Profissionais de Saúde.

5. Critérios de avaliação

Pontos	Descrição	
Obrigatório (total de 5 pontos)		
5	5.1. Interface administrativa	A utilização de filter, search, autocomplete, fieldset, date hierarchy.
	5.2. Normalização do banco de dados	Aplicar 1FN, 2FN e 3FN.
	5.3. Responsividade	Para que o sistema seja acessado através de smartphones.
	5.4. Relatórios	Realizar consultas SQL puras (agregação, projeção e seleção) para geração de gráficos e listagens.
	5.5 Deploy no Heroku	Necessário incluir a URL no README.MD.
	5.6 Instruções	Arquivo README.MD contendo os passos de como rodar o projeto, incluindo fixtures.
Facultativo (total de 30 pontos)		
10	5.7. Separação entre front-end e back-end	Utilizando Vue.JS e Django REST Framework.
5	5.8. Integração com mecanismo de autenticação OAuth 2	Necessário cadastrar a aplicação OAuth2 em algum serviço OAuth 2 suportado, tal como Google.
5	5.9. Websocket	Usado para receber atualizações da fila de espera.
5	5.10. Push notifications	Informa ao usuário quando chegar a sua vez, mesmo quando o sistema não estiver ativo na tela do celular.

3	5.11. Docker + Docker Compose	Incluir o arquivo docker-compose.yml na raiz do projeto.
2	5.12. Envio de e-mail	Envio de email usando os servidores do GMail ou outro.

6. Requisitos para a apresentação de defesa

- O candidato deverá apresentar cada etapa de utilização do sistema para cada ator;
- Apresentar os códigos implementados.

Projeto II - Big Data

1. Conteúdo Programático

Conforme o Anexo III do presente edital, o conteúdo programático consiste em: Apache Hadoop; Apache Hive; ELK Stack; PL/SQL; PostgreSQL ou MySQL; Linguagens de Programação: R ou Python; Bash Script.

2. Descrição

Elaborar um dashboard para consumo de indicadores a partir de uma das fontes de dados disponibilizada abaixo:

- <https://www.kaggle.com/ramirobentes/flights-in-brazil>
- <https://www.kaggle.com/epattaro/brazils-house-of-deputies-reimbursements>
- <https://www.kaggle.com/mateuspgomes/brazil-thrift-stores-data>

3. Requisitos

3.1 Obrigatórios

- Elaborar um dashboard com indicadores que representem o dataset escolhido, bem como permita a obtenção de insights;
- Utilizando uma instância do ELK Stack deverá cumprir as seguintes etapas:
 - Adicionar o dataset escolhido ao Elasticsearch;
 - Preparar um dashboard no Kibana para apresentação dos indicadores.

3.2 Facultativos

- Apresentar de forma documentada (um notebook, por exemplo) da análise dos dados;
- Apresentar código de ingestão dos dados em R ou Python, ou utilizando Logstash, devidamente comentado;
- Customização do dashboard.

4. Requisitos para a apresentação de defesa

- O candidato deverá apresentar cada etapa, explicando como a executou;
- Justificar a escolha do dataset, os indicadores implementados e quais insights é possível obter a partir destes;
- Apresentar os códigos implementados.

5. Critérios de Avaliação

Pontos	Descrição
Obrigatório (total de 5 pontos)	
2	5.1. Adicionar o dataset escolhido ao Elasticsearch.
3	5.2. Preparar um dashboard no Kibana para apresentação dos indicadores.
Facultativo (total de 30 pontos)	
6	5.3. Utilizar o Logstash para adição dos dados no Elasticsearch.
2	5.4. Utilizar o R para adição dos dados no Elasticsearch.
2	5.5. Utilizar o Python para adição dos dados no Elasticsearch.
10	5.6. Customização do Dashboard (logomarca, paleta de cores, tema).
1	5.7. Para cada indicador adicionado ao Dashboard, limitado a 10 pontos , desde que o indicador não seja repetido (mesmo indicador com apresentações diferentes).

Projeto III - Sistemas Embarcados e Biosensores

1. Conteúdo Programático

Conforme o Anexo III do presente edital, o conteúdo programático consiste em: Linguagens de programação: Python, C/C++ ou Arduino; Amplificadores operacionais; Conversores A/D e D/A; Protocolo I2C; Operacionalização de projetos utilizando Arduino ou Raspberry Pi.

2. Descrição

Elaborar um projeto de hardware que seja capaz de detectar a presença de sal ou açúcar na água.

3. Requisitos

3.1 Obrigatórios

- Elaborar uma lista de equipamentos e materiais necessários para a construção do equipamento;
- Em um software para criação de projetos eletrônicos:
 - Elaborar o esquemático do circuito;
 - Realizar a simulação do circuito em funcionamento;
 - Exportar o design da placa de circuito impresso a qual irá reunir todos os componentes eletrônicos utilizados no esquemático.

3.2 Facultativos

- Permitir que o operador do hardware possa utilizar a técnica de voltametria cíclica para analisar a substância aquosa e exportar em forma de gráficos ou CSV o resultados das leituras;
- Possui uma interface gráfica para operação e visualização do resultado da análise.

4. Requisitos para a apresentação de defesa

- O candidato deve explicitar a técnica utilizada e sua teoria;
- Com base na técnica de análise escolhida, apresentar a lista de equipamentos e materiais necessárias para a construção do hardware e explicar como cada item será utilizado para a construção do projeto;
- Apresentar o esquemático do circuito e explicar se dará o seu funcionamento;
- Exibir o código-fonte do sistema de controle do hardware, assim como a simulação do equipamento em funcionamento. Podendo ser aceito a apresentação de pseudo-código, porém será atribuída uma nota inferior.

5. Critérios de avaliação

Pontos	Descrição
Obrigatório (total de 5 pontos)	
1	5.1. Lista de equipamentos e materiais necessários para a construção do equipamento
1	5.2. Esquemático do circuito
2	5.3. Simulação do circuito em funcionamento
1	5.4. Design da placa de circuito impresso
Bônus (total de 30 pontos)	
15	5.5. Permitir que o operador do hardware possa utilizar a técnica de voltametria cíclica para análise da substância aquosa

5	5.6. Exportar em forma de gráficos os dados de leitura da técnica de voltametria cíclica
5	5.7. Exportar em forma de arquivo CSV os dados de leitura da técnica de voltametria cíclica
5	5.8. Interface gráfica para operação e visualização do resultado da análise

Projeto IV - Inteligência Artificial

1. Tecnologias a serem utilizadas

Conforme o Anexo III do presente edital, o conteúdo programático consiste em: Linguagem de programação Python; Aprendizado: Definição e Principais Tipos; Formas de Representação do Conhecimento: Definição, Árvores de Decisão e Redes Neurais; Programação Orientada a Objetos; Bibliotecas Pandas e Scikit-Learn.

2. Detalhes do Projeto

O candidato poderá escolher entre dois problemas distintos: i) um problema de classificação para a detecção de tumor cerebral; e, ii) um problema de saúde fetal.

2.1 Descrição do Problema I - Classificação de Tumor Cerebral

O problema consiste na classificação binária de imagens de ressonância magnética para a detecção de tumor cerebral. Todas as imagens devem ser treinadas com o mesmo shape (tamanho e altura). O candidato pode escolher os melhores parâmetros para a imagem, e quais técnicas de pré-processamento irá utilizar.

Um ponto importante a ser considerado é o balanceamento entre as duas classes. Tendo em vista que, um dos aspectos fundamentais em problemas de classificação é a desigualdade na distribuição das classes. Sendo assim, como parte da solução, deverá analisar se a distribuição das classes irá ou não interferir no resultado do processo de classificação.

É importante que todo o processo seja descrito na documentação, e seja inserida junto com o código no Github.

2.2 Descrição do Problema II - Saúde Fetal

O problema consiste na análise de dados do exame Cardiotocogramas (CTGs) que tem por objetivo medir a frequência cardíaca fetal (FCF), movimentos fetais, contrações uterinas e muito mais fetal para prevenir a mortalidade infantil e materna. O

conjunto de dados contém 2.126 registros, que foram classificados por três obstetricistas especialistas em 3 classes: Normal, Suspeito e Patológico. Sendo assim, o objetivo é criar um modelo multiclasse que classifique os dados nestes três estados de saúde fetal.

A biblioteca Pandas deve ser utilizada para o *load* e manipulação do dataset. Um ponto importante é que deve ser realizada uma varredura nos dados para verificar se há dados faltantes. Se houver, explicar como foi selecionado o problema. É importante que todo o processo seja descrito na documentação, e seja inserida junto com o código no Github.

3. Download do Dataset

- **Problema de Classificação de Imagem:**
<https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection>
- **Problema de Classificação de Saúde Fetal:**
<https://www.kaggle.com/andrewmvd/fetal-health-classification>

4. Requisitos do Projeto

O projeto deve ser desenvolvido utilizando a versão 3.6 ou superior da linguagem de programação Python. Para o problema de classificação de imagem é livre a escolha da melhor técnica, no entanto a solução deve conter os 5 (cinco) passos descritos na Subseção 4.1, e cumprir os critérios obrigatórios descritos na Seção 6, em Critérios Obrigatórios.

Todos os critérios obrigatórios devem ser atendidos para que o projeto seja avaliado.

4.1 Obrigatórios

O projeto tem como requisitos obrigatórios:

- O código do projeto deve ser versionado em repositório público no GitHub;
- O algoritmo de classificação deve conter os seguintes passos:

- Passo 1: Realizar o pré-processamento dos dados;
- Passo 2: Determinar o y_{train} que tem como objetivo ser os rótulos das classes do conjunto de dados;
- Passo 3: Determinar o x_{train} que tem como objetivo ser o conjunto de teste;
- Passo 4: Determinar a dinâmica do treinamento e dizer porque foi utilizada;
- Passo 5: Determinar a métrica de avaliação do modelo proposto: Acurácia.

4.2 Facultativos

- Configurar os critérios de parada no processo de treinamento com o número de vezes em que o conjunto de dados é apresentado ao modelo para a realização do processo de treinamento;
- Transfer learning;
- Técnicas de Validação Cruzada;
- Outras métricas de avaliação do modelo: Matriz de confusão, sensibilidade, especificidade, $f1_score$.

5. Requisitos para a apresentação de defesa

Na apresentação do projeto serão observados os seguintes pontos:

- O candidato deve explicitar a técnica utilizada e sua teoria;
- Explicar o processo de Aquisição do Conhecimento e Representação do Conhecimento utilizados para a resolução do problema;
- Utilizar o repositório Github para versionar o código;
- Apresentar de forma documentada (um notebook, por exemplo) com texto e código do projeto desenvolvido;
- Justificar o problema escolhido.

6. Critérios de avaliação

Pontos	Descrição
Obrigatórios (total de 5 pontos)	
1	Explicar o processo de Aquisição do Conhecimento e Representação do Conhecimento utilizados para a resolução do problema;
1	Utilizar o repositório Github para versionar o código;
1	Listar: o conjunto do domínio; o conjunto de labels; as características dos dados de treinamento; a saída desejada; o modelo para a geração dos dados (ou conhecimento).
1	Apresentar de forma documentada (um notebook, por exemplo) com texto e código do projeto desenvolvido.
1	Apresentar a métrica de avaliação do modelo proposto: Acurácia.
Bônus (total de 30 pontos)	
5	Configurar os critérios de parada no processo de treinamento com o número de vezes em que o conjunto de dados é apresentado ao modelo para a realização do processo de treinamento;
5	<i>Area under the ROC curve</i>
5	<i>Transfer learning</i>
5	Técnicas de Validação Cruzada
5	Outras métricas de avaliação do modelo: Matriz de confusão, sensibilidade, especificidade, f1_score (sem o uso de biblioteca)



5	Deep Learning
---	---------------