**Instruções!**

A organização deste documento segue o que está no formulário de submissão de propostas do <https://sigitec.petrobras.com.br/SIGITEC/>

A parte de ETAPAS e ATIVIDADES foi colocada num segundo documento, que é uma planilha, por ser mais fácil alterá-la.

**Título:**

Aplicação de IA para Previsão de Propriedades de Fluidos Ricos em CO2

**Resumo: 2000 caracteres – Repete objetivo e etapas**

A determinação das propriedades termodinâmicas (número de fases, composição, densidade) e reológicas (viscosidade) de misturas multicomponentes são fundamentais para a previsão de comportamento de sistemas compostos por fluidos de reservatórios e de injeção, tanto em meios porosos como em sistemas de produção. A presença de contaminantes, como o dióxido de carbono, diminui a capacidade de cálculo dessas propriedades através de equações de estado. Por outro lado, o uso de técnicas de Inteligência Artificial tem o potencial de substituir com sucesso essa metodologia.

Nesse projeto será desenvolvido um software com interface gráfica a partir de técnicas de Inteligência Artificial para o cálculo de propriedades termodinâmicas e reológicas de sistemas contendo hidrocarbonetos e dióxido de carbono.

O primeiro passo será a criação de um banco de dados de propriedades termodinâmicas e reológicas de fluidos. Após a coleta dos dados, será realizada a compatibilização, normalização e eliminação de outliers.

Para realizar o treinamento, teste e previsão, o banco de dados será dividido em: i) treino e validação, ii) teste das previsões.

Para o teste, será utilizada uma abordagem de validação cruzada pelo método k-fold, que permitirá não só realizar o teste em si como também permite identificar sobre-ajustes. Além disso, é possível utilizar esta abordagem para determinar os melhores valores para os parâmetros requeridos por certos algoritmos de treino.

As redes serão treinadas para:

1) Determinação do número de fases e composição de cada fase de uma mistura,

2) Estimativa das propriedades termodinâmicas e reológicas de cada fase.

Na etapa seguinte, as propriedades das misturas serão estimadas e os resultados comparados com os valores remanescentes, para validação do software. Para garantir as capacidades preditivas das redes a serem desenvolvidas neste projeto, o número de dados usados para treiná-las será pelo menos 10 vezes maior que o número de parâmetros requeridos.

# Objetivo Geral : 500 caracteres

O objetivo geral reside no desenvolvimento de aplicativo aprimorado por técnicas de IA (aprendizado de máquina, redes neurais, etc.) para previsão de propriedades termodinâmicas e reológicas de fluidos contendo teores variáveis de CO2, similares aos encontrados ou usados no pré-sal brasileiro.

# Duração

24 meses.

# Resultados esperados:

* PRODUTO

# Descrição: 300 caracteres

Desenvolvimento de software/aplicativo (multiplataforma com interface gráfica amigável), aprimorado por técnicas de IA para previsão de propriedades termodinâmicas e reológicas de fluidos contendo teores variáveis de CO2, similares aos encontrados ou usados no pré-sal brasileiro.

# Metodologia: 34976 caracteres

As propriedades termodinâmicas e reológicas de misturas multicomponentes são fundamentais para a previsão de comportamento de sistemas compostos pelos fluidos de reservatórios e de injeção, tanto nos meios porosos como nos sistemas de produção. As principais propriedades físicas (densidade, compressibilidade e viscosidade) são determinadas pela sua composição química, pressão e temperatura do sistema. Além disso, é importante prever e calcular o equilíbrio de fases, ou seja, determinar o número de fases, suas composições e quantidades relativas. Tradicionalmente, essas propriedades são obtidas ajustando-se os parâmetros de uma equação de estado a dados experimentais.

Apesar do grande esforço dedicado ao desenvolvimento de equações de estado, ainda não existe um modelo universal. Além disso, conduzir experimentos de laboratórios em condições reais de um sistema de produção de petróleo para todos os cenários possíveis é uma tarefa impossível de ser executada em termos práticos, devido à grande quantidade e diversidade de sistemas a analisar. Finalmente, a presença de contaminantes, como por exemplo o dióxido de carbono, aliada à impossibilidade de caracterização completa do fluido contido num reservatório de petróleo, torna ainda mais desafiadora a utilização de equações de estado que sejam capazes de calcular as propriedades termodinâmicas e reológicas de fluidos e de sistemas multifásicos.

Algumas técnicas de Inteligência Artificial, baseadas em dados, têm o potencial de substituir as abordagens baseadas em modelos físicos. Isto é, podem extrair a informação física subjacente num conjunto de dados, com os quais foram “treinadas”, e interpolar propriedades físicas dos fluidos neste domínio, sem fazer uso de equações de estado.

O primeiro passo na implementação do aprendizado de máquina será a criação de um banco de dados experimentais de propriedades termodinâmicas e reológicas de fluidos. Esse banco de dados será composto por propriedades de substâncias puras e misturas. Os dados experimentais de misturas, numa primeira etapa, serão restritos a sistemas binários, para limitar a quantidade de sistemas. Também serão incluídos os dados dos fluidos de reservatórios existentes no pré-sal. Nos sistemas binários um dos componentes será o dióxido de carbono. Os sistemas serão divididos em monofásicos e bifásicos, em função da variação da composição de cada fase.

Após a coleta dos dados experimentais, será realizada a compatibilização a um mesmo sistema de unidades, bem como a sua normalização. Também serão categorizados de acordo com a faixa de temperatura, pressão e composição (com ênfase na quantidade de dióxido de carbono nas misturas). Também será necessário verificar a consistência dos dados, com o intuito de eliminar “outliers”.

Uma vez construída uma base de dados consistente e abrangente, é possível realizar a escolha e treinamento do algoritmo mais indicado para o problema. Algumas publicações indicam que uma rede neural totalmente conectada produz melhores resultados, em problemas similares (determinação de propriedades de materiais), em comparação com outros algoritmos de IA.

Devido ao crescente interesse e atividade nas técnicas de inteligência artificial, existe uma variedade de bibliotecas e pacotes disponíveis que implementam algoritmos de “Machine Learning“ e redes neurais em uma variedade de linguagens de programação, tais como Python e C++. Estas bibliotecas serão empregadas na seleção do algoritmo de solução.

Para realizar o treinamento, teste e previsão, o banco de dados compilado nas etapas anteriores será dividido em duas partes: uma para treino e validação, e outra para teste das previsões. Essas partes não terão o mesmo tamanho, e a maior parte dos dados deverá ser utilizada para treino e teste das redes, embora a proporção exata terá de ser determinada no decorrer da execução do projeto, e será função do tamanho do banco de dados e número de dados experimentais.

A etapa de treino (aprendizagem) da rede neural é dependente do tipo de rede a ser treinada, que por sua vez depende tanto dos dados de entrada quanto do objetivo (saída, “output”) da rede. Para o teste da rede, será utilizada uma abordagem de validação cruzada pelo método “k-fold”, que permitirá não só realizar o teste da rede em si como também permite identificar sobre-ajustes (“overfitting”). Além disso, com pequenas modificações, é possível utilizar esta abordagem também para determinar os melhores valores para os parâmetros requeridos por certos algoritmos de treino.

Serão treinadas redes para os seguintes propósitos:

1) Determinação do número de fases de uma mistura: determinar se um sistema é monofásico ou bifásico numa dada temperatura, pressão e composição inicial (verificação de estabilidade). Caso seja um sistema monofásico, determinar se esta fase é líquida ou gasosa.

2) Determinação da composição de cada fase de uma mistura bifásica: dada uma certa condição de temperatura, pressão e composição global, determinar a composição de cada fase.

3) Estimativa das propriedades de uma fase: numa dada temperatura, pressão, composição e fase, determinar as propriedades desta fase, como a massa específica, a viscosidade, etc.

Na etapa da previsão, serão realizadas estimativas das propriedades das misturas, conforme os objetivos das redes treinadas. Estas estimativas serão comparadas com os valores disponíveis na base de dados remanescente, para validação do “software” como um todo. Somente após a validação das redes treinadas é possível realizar previsões com elas. Para garantir as capacidades preditivas das redes a serem desenvolvidas neste projeto, o número de dados usados para treiná-las será pelo menos 10 vezes maior que o número de parâmetros requeridos por elas (pesos e “bias”).

## Metodologia Desenvolvimento do Software:

Uso de metodologia de desenvolvimento de software híbrida, desenvolvida no LDSC/LENEP/CCT/UENF.

Na primeira edição do livro "Programação Orientada a Objeto com C++: Aprenda a Programar em Ambiente Multiplataforma com Software Livre" (Bueno, 2003) desenvolvemos uma metodologia de software relativamente tradicional, cíclica, que mistura conceitos dos modelos TMO, UP e espiral, com as etapas de:

- Concepção (especificação, requisitos);

- Elaboração;

- Análise Orientada a Objeto - Modelagem UML;

- Projeto do Sistema - infraestrutura, recursos;

- Projeto Orientado a Objeto;

- Desenvolvimento, Implementação;

- Testes, depuração;

- Documentação (em todas as fases);

- Ciclos de manutenção;

Esta metodologia foi utilizada no desenvolvimento de alguns softwares no LDSC, incluindo SAIL - Software de Análise de Imagens Livre (códigos com autor), LIB\_LDSC (https://github.com/ldsc/lib\_ldsc), LVP - Laboratório Virtual de Petrofísica (https://github.com/ldsc/lvp). O LVP, por exemplo, teve uma primeira versão usando QT4 e depois foi portado para Qt5. A LIB\_LDSC também teve diversas atualizações para atender as demandas do doutorado do aluno Leandro Puerari.

Posteriormente foram incorporadas à metodologia utilizada, conceitos de Engenharia de Software Ágil, com uso de SCRUM, FDD e ciclos de desenvolvimento.

Antes de descrever a metodologia em si, cabe aqui destacar que a mesma tenta considerar uma realidade híbrida para softwares desenvolvidos no meio acadêmico, pois os modelos ainda estão sendo desenvolvidos e testados. Diferente de uma realizada mais comercial, em que se aplicam modelos e normas pré-estabelecidos. Outro aspecto é que os alunos não estão acostumados com o uso de ferramentas como:

- Engenharia de Software, UML.

- Sistemas de controle de versões - git/github.

- Montadores - cmake.

- Debuggers.

- Documentação profissional (Javadoc/doxygen, uso de TeX/LaTeX/Lyx).

- Otimização serial e paralelização.

o que leva um certo tempo para aprender e colocar em prática.

Apresenta-se a seguir a metodologia de desenvolvimento de software sugerido. É uma metodologia mista, com características determinísticas, definidas pelos requisitos e análise criteriosa. É uma metodologia em espiral, cíclica e iterativa (com características dos modelos espiral, TMO, UP, SCRUM e FDD).

Entre suas características pode-se citar:

- Reduz problemas de custos com alterações ao adotar o uso de ciclos para elaboração e análise criteriosa (usa característica de UP e modelo espiral); e, ao mesmo tempo, possibilitar a refatoração dos sistemas a cada ciclo subsequente (característica de FDD).

- Cada módulo ou parte do software é desenvolvida em ciclos (com formatação variável a depender da complexidade dos sistemas) que podem ser desmembrados em atividades especializadas cuja execução é definida pela equipe (característica de RAD e SCRUM, a equipe tem mais autonomia).

- O cliente tem acesso ao software através de repositórios como exemplo o github.

- No início do projeto é feita uma reunião que define o nível e forma de interação da equipe do cliente com a equipe de desenvolvimento. Em alguns casos os clientes podem atuar diretamente no desenvolvimento do sistema.

- O cliente participa das reuniões. Isto faz com que tenha maior conhecimento do software e possa contribuir com o desenvolvimento.

- A metodologia incentiva o uso de bibliotecas conhecidas, o que permite reaproveitamento de código.

- Incentiva a adoção de padrões de projeto (padrões de desenvolvimento).

- Usa métodos de refatoração, o que permite corrigir falhas e adicionar modificações e novos requisitos mais facilmente. Na refatoração eliminam-se códigos redundantes, códigos e funcionalidades que posteriormente foram identificadas como sendo inúteis (característica de SCRUM-FDD).

- Trabalha-se com reuniões periódicas (SCRUM).

As Figuras no site abaixo ilustram a metodologia

[Site: https://sites.google.com/view/professorandreduartebueno/ensino/introdu%C3%A7%C3%A3o-ao-projeto-de-engenharia#h.x930x7ct2wsx ]

Ciclo Concepção e Análise:

Tem como objetivo desenvolver um modelo abrangente e ao mesmo tempo identificar/listar funcionalidades/características/módulos/assuntos a serem desenvolvidos. Esta etapa deve ser desenvolvida de forma cíclica e iterativa. A cada iteração o modelo deve ser mais abrangente e abstrato (a visão dos micro detalhes será feita posteriormente nos Ciclos de Planejamento e Detalhamento).

- Especificação, Requisitos, Casos de Uso.

- Elaboração - diagramas de assuntos, pacotes, protótipos.

- Análise orientada a objeto - diagramas classe, sequência, comunicação, atividades, máquina de estado.

- Projeto do sistema - protocolos, recursos, controle, arquitetura.

- Projeto orientado a objeto - diagramas componentes e implantação; revisão.

- Documentação.

Esta etapa deve considerar a Norma Qualidade de Software (ISO/IEC 9126: NBR 13596) que inclui definições de:

- Funcionalidade – satisfaz as necessidades.

- Confiabilidade – é imune a falhas.

- Usabilidade – é fácil de usar.

- Eficiência – é rápido.

Ciclos Planejamento e Detalhamento:

– Nesta etapa cada funcionalidade é melhor estudada e detalhada, identificando-se complexidades e demandas adicionais.

– Inclui detalhamento do modelo estrutural de UML (como o diagrama de classes).

– Inclui detalhamento dos diagramas dinâmicos de UML (como os diagramas de caso de uso, de sequência, comunicação e máquina de estado).

– Detalhamento do uso de bibliotecas.

– Detalhamento dos testes.

– O resultado é um plano detalhado para desenvolvimento das funcionalidades do ciclo.

Ciclos Construção

- Codificação.

- Documentação.

- Testes unidade.

- Teste integração.

A documentação dos códigos é feita ao longo de todo processo, usando-se sistemas como JAVADOC/Doxygen;

Já o manual do usuário e o manual técnico-científico costumam ser desenvolvidos no final do processo.

Entrega do ciclo

- Ao final do ciclo um novo conjunto de funcionalidades (classes/módulos) foi adicionado ao software que deve estar funcional.

Produto final

- No último ciclo novidades são proibidas (congeladas).

– Foca-se na eliminação de bugs.

– Na realização dos testes finais.

– Na confecção do manual do usuário.

– Na confecção do manual técnico-científico.

– Na revisão e finalização da documentação.

– Na geração e disponibilização da distribuição (ex: github ou arquivo dist.tar.gz, eventualmente um instalador).

### Etapas gerais do projeto:

• Definição de objetivos específicos.

• Revisão de literatura.

• Construção da base de dados.

• Estudo dos algoritmos de IA - protótipos.

• Desenvolvimento/Implementação do Software.

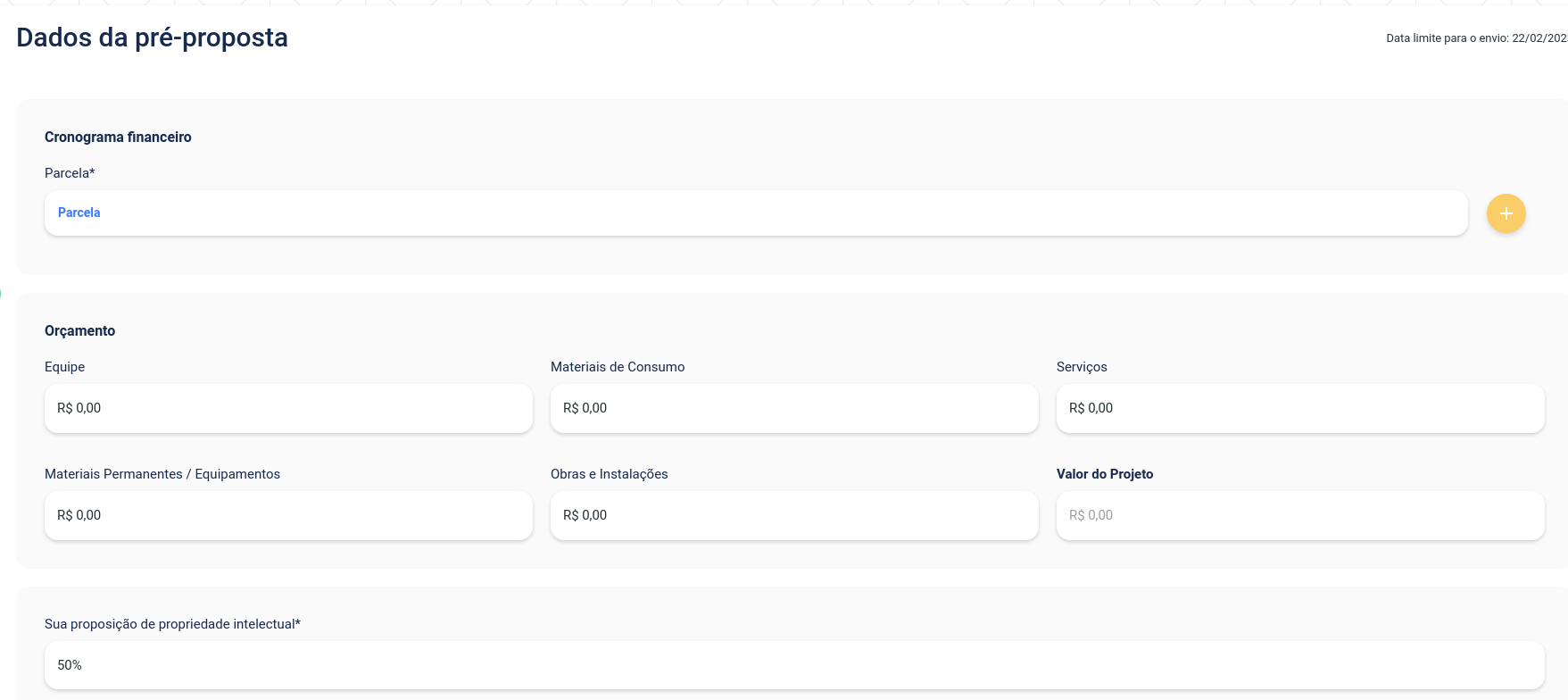
NOTA: Algumas informações sobre softwares desenvolvidos estão disponibilizadas na página de internet: https://sites.google.com/view/professorandreduartebueno/softwares-desenvolvidos

NOTA: Havendo necessidade de compras de bibliotecas ou softwares proprietários para compilar, debugar ou rodar os sistemas de forma não acadêmica a responsabilidade pela compra e licenciamento é da Petrobras.

**Etapas/Atividades**

CONSULTE O SEGUNDO DOCUMENTO A PLANILHA

**Orçamento:**



**PIT: Bruno**

256 caracteres (não coube o ponto final!)

O grupo possui experiência no tema através de publicação de artigos, orientações de mestrado/doutorado e projetos de pesquisa na área de propriedades de fluidos e Inteligência Artificial. Também atuamos no desenvolvimento de software com interface gráfica.