

# Sumário

1. CANVAS do Projeto Aplicado	4
1.1 Desafio	5
1.1.1 Análise de Contexto	5
1.1.2 Personas	6
1.1.3 Benefícios e Justificativas	7
1.1.4 Hipóteses	8
1.2 Solução	9
1.2.1 Objetivo SMART	9
1.2.2 Premissas e Restrições	11
1.2.3 Backlog de Produto	13
2. Área de Experimentação	14
2.1 Sprint 1	16
2.1.1 Solução	16
• Evidência do planejamento:	16
• Evidência da execução de cada requisito:	16
• Evidência dos resultados:	16
2.1.2 Experiências vivenciadas	16
2.2 Sprint 2	17
2.2.1 Solução	17
• Evidência do planejamento:	17
• Evidência da execução de cada requisito:	17
• Evidência dos resultados:	17
2.2.2 Experiências vivenciadas	17
2.3 Sprint 3	18
2.3.1 Solução	18
• Evidência do planejamento:	18
• Evidência da execução de cada requisito:	18
• Evidência dos resultados:	18
2.3.2 Experiências vivenciadas	18
3. Considerações Finais	19
3.1 Resultados	19
3.2 Contribuições	19
3.3 Próximos passos	19



## 1. CANVAS do Projeto Aplicado

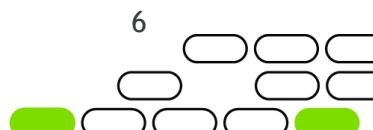
Figura 1. CANVAS



### 1.1 Desafio

#### 1.1.1 Análise de Contexto

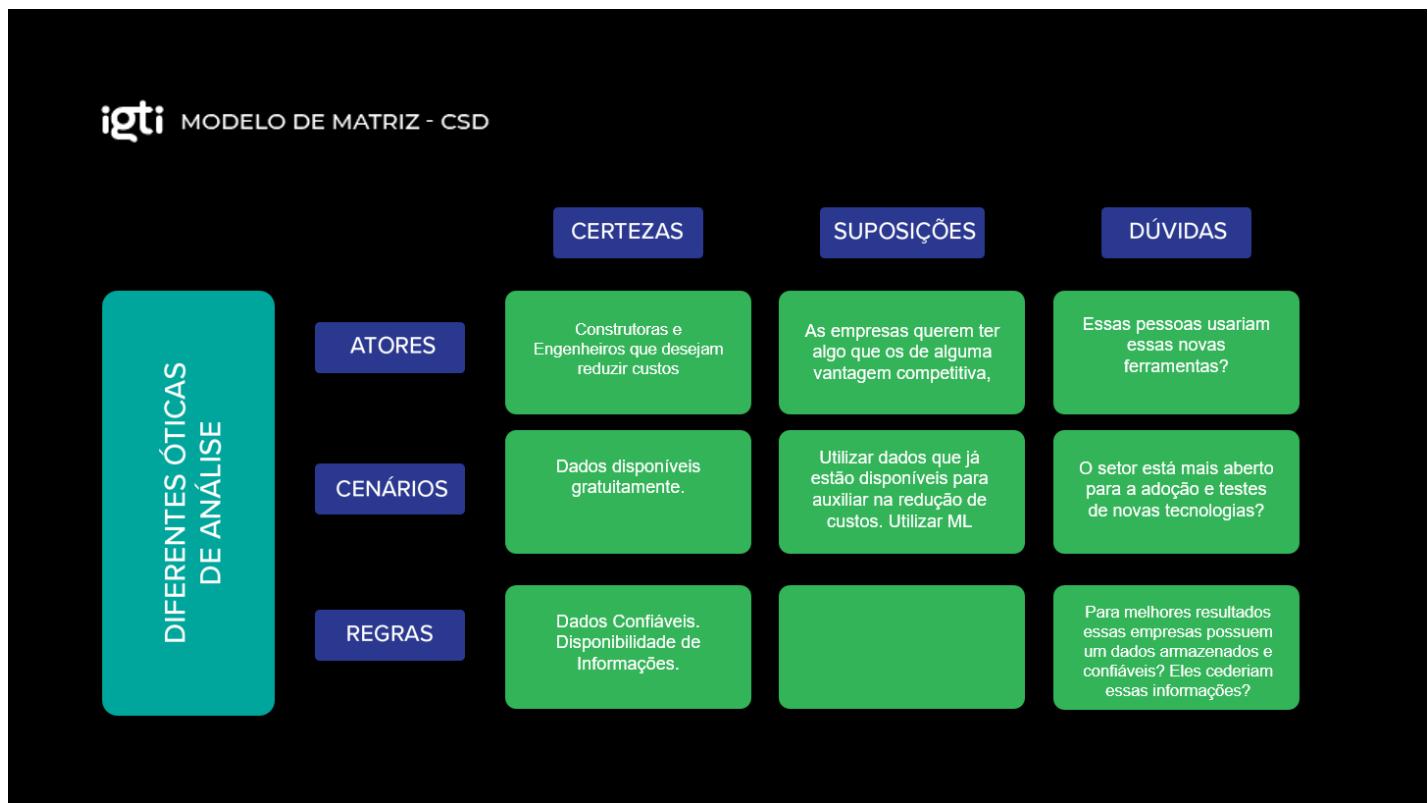
A indústria da construção civil enfrenta muitos desafios que impedem o seu crescimento, e um desses problemas é a dificuldade de orçar uma obra. Uma das principais dificuldades reside na diferença entre o valor orçado inicialmente e o valor real da construção. Isso pode acontecer quando o orçamento não contempla todo o escopo real da obra ou quando os preços ficam defasados devido ao espaço de tempo entre o orçamento e o início da obra. Isso resulta na necessidade de pedidos de pleitos, redução da qualidade do material ou até mesmo contratação de mão de obra desqualificada, entre outras situações. Por isso, a tarefa de orçar uma obra de forma que consiga enfrentar o tempo e a volatilidade dos preços é algo que traria melhores oportunidades para uma empresa. Isso faria com que o empreendimento não tenha



que arcar com prejuízos devido à incerteza do mercado e possa investir em tecnologia, qualidade e mão de obra, podendo se destacar no mercado e mantendo a sua competitividade.

Com os problemas expostos utilizaremos a matriz CSD (certezas, suposições e dúvidas) para um melhor conhecimento do contexto e na busca de uma solução mais assertiva, expondo o que já se sabe, as hipóteses e as perguntas que precisam ser feitas ficam expostas de forma visual e clara.

*Figura 2. Matriz CSD*



Para complementar a análise, utilizaremos a ferramenta de análise do contexto do problema POEMS.



*Figura 3. POEMS (Fonte: Autora)*

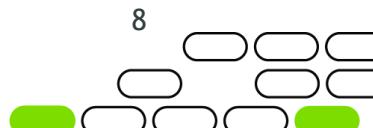


Para se entender melhor como funciona um processo de orçamentação, uma breve explicação:

Tudo se inicia analisando as propostas, convites e licitações. Realizar a análise e orçamentação de obras é um processo que ajuda a determinar a viabilidade e rentabilidade de um projeto de construção. Primeiro, é preciso compreender os detalhes do projeto para avaliar se é adequado para a empresa. Em seguida, a equipe responsável se reúne para planejar a análise da proposta e dividir as tarefas. Isso se tiver uma equipe é claro.

Então os principais insumos são levantados, como materiais, mão de obra e equipamentos, usando informações de projetos anteriores ou referências de mercado. Com base nesses dados, os custos dos insumos e os custos indiretos, como impostos e administração, são estimados, considerando também uma margem de lucro.

Durante a análise, pontos fortes da empresa que possam trazer vantagens competitivas são identificados, além de avaliar os possíveis riscos e desafios. Por fim, a proposta é elaborada com os preços cotados e informações relevantes sobre a empresa, garantindo que esteja de acordo com os requisitos da licitação ou do cliente.



### 1.1.2 Personas

Para que possamos analisar mais a fundo o problema, precisamos entender as necessidades do cliente e, para isso, é necessário definir Personas que representam o perfil do usuário ideal do produto. Essas Personas devem ter as características e necessidades que serão atendidas pela solução, lembrando que essas informações são fictícias.

#### Persona 1:

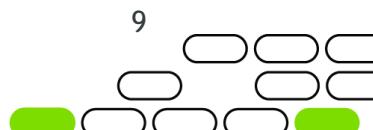
João Silva é um engenheiro de custos e orçamentos de 32 anos que trabalha em uma empresa de construção em Vitória, ES. Sua responsabilidade é realizar orçamentos e prever com precisão e rapidez os custos e insumos necessários para cada obra. Ele é um profissional dedicado e comprometido, que busca constantemente aprimorar seus conhecimentos e habilidades na área de orçamentos de obras. Ele é detalhista e exigente, buscando sempre a precisão e a confiabilidade nos dados de orçamentação das obras.

João está atento às mudanças tecnológicas e acredita que a adoção de novas tecnologias pode tornar sua rotina de trabalho mais eficiente e permitir uma maior precisão nos orçamentos. Ele está aberto a novas tecnologias, desde que sejam acessíveis e fáceis de implementar na empresa. No entanto, ele se preocupa com o custo dessas tecnologias e acredita que precisaria de mais conhecimento e capacitação para implementá-las.

João enfrenta desafios na previsão de custos das obras, pois muitas vezes não consegue obter informações precisas e em tempo hábil sobre os preços dos insumos, o que gera insegurança quanto aos orçamentos das obras. Ele não gosta de depender dos outros para coletar os preços, pois muitas vezes não recebe resposta a tempo e precisa ir atrás das cotações uma a uma.

Além disso, João se preocupa com a variação dos preços dos materiais de construção, o que impacta diretamente no orçamento dos projetos. Ele está em busca de soluções que possam melhorar a eficiência e a precisão de seu trabalho, permitindo que ele preveja com mais precisão os custos e insumos das obras.

Apesar dos desafios, João é um profissional motivador e inspirador, que sabe reconhecer e valorizar o trabalho de sua equipe. Ele está sempre aberto ao diálogo e à colaboração, buscando envolver toda a equipe nos processos de tomada de decisão.



Além disso, João é um profissional comprometido com a qualidade e a segurança das obras, garantindo que todos os projetos atendam aos mais altos padrões técnicos e regulatórios.

**Persona 2:**

Rafael Alves é um engenheiro civil recém-formado de 28 anos que está abrindo sua própria empresa de construção civil. Sua paixão pela área de construção civil e seu espírito empreendedor o levaram a buscar constantemente novas oportunidades de negócio, mas ele tem enfrentado dificuldades para ganhar obras devido à alta concorrência no mercado.

Rafael é um empreendedor motivador e otimista, que busca sempre inovação e tecnologia para melhorar seus processos e entregas.

Para superar a alta concorrência no mercado, Rafael busca criar um diferencial competitivo para sua empresa, oferecendo serviços de alta qualidade e diferenciais competitivos. Ele está sempre aberto ao diálogo e à colaboração, buscando envolver todos os colaboradores em suas estratégias de negócio.

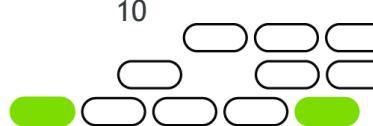
Rafael sabe que o mercado de construção civil é altamente competitivo e, portanto, busca constantemente novas estratégias para se destacar da concorrência. Ele acredita que a tecnologia pode ser um grande diferencial para sua empresa, mas enfrenta desafios em implementá-la.

Além disso, Rafael sabe que a previsão de custos é fundamental para garantir a rentabilidade dos projetos, mas tem enfrentado dificuldades em estimar os custos dos materiais e mão de obra, o que pode impactar negativamente sua rentabilidade. Ele está em busca de soluções que possam melhorar a eficiência e precisão de seu trabalho de orçamentação.

**Persona 3:**

Joana Almeida é uma engenheira civil, com MBA em Gestão de Projetos, de 42 anos de idade e gênero feminino. Com 15 anos de experiência no setor de construção civil, incluindo trabalhos em obras públicas e privadas, com ênfase em projetos residenciais e comerciais de pequeno e médio porte. Joana é a fundadora da empresa Almeida Construções e Reformas, que conta com uma equipe de 50 colaboradores, incluindo engenheiros, arquitetos, mestres de obras e operários.

Joana é uma líder nata, com habilidades de liderança que permitem gerenciar sua equipe com eficiência. Sua facilidade de comunicação com a equipe, clientes e



fornecedores é uma de suas maiores forças. Além disso, Joana é proativa e busca soluções rápidas e eficientes para os problemas enfrentados pela empresa.

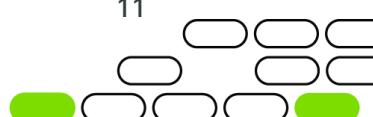
No entanto, Joana enfrenta dois grandes problemas em sua empresa de construção civil:

- Dificuldade em prever custos de materiais e mão de obra: Joana tem dificuldades em estimar os custos reais dos materiais e mão de obra, o que frequentemente resulta em estouro do orçamento. Além disso, a variação dos preços dos materiais de construção impacta diretamente no orçamento dos projetos.
- Problemas com fornecedores, gerando atrasos e aumentos no orçamento: Joana enfrenta desafios na relação com fornecedores, como atrasos nas entregas das cotações e valores dos insumos incompatíveis com a realidade.

Joana está buscando soluções para esses problemas, com o objetivo de melhorar a rentabilidade e a eficiência da empresa.

Para entender melhor as necessidades e expectativas dos clientes, foram utilizados mapas de empatia para considerar os pensamentos e sentimentos do orçamentista e do empreendedor em relação ao processo de orçamento. O objetivo era identificar aspectos mais específicos desse processo, apontar problemas e, por fim, identificar dores como perdas financeiras, queda na qualidade do trabalho, falta de investimentos e tensão nos relacionamentos com os clientes.

*Figura 4. Exemplo de Mapa de empatia. Persona 1 - João Silva (Fonte: Autora)*



## igtí MAPA DE EMPATIA



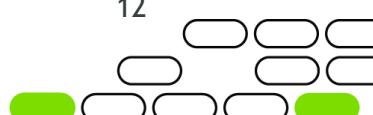
### 1.1.3 Benefícios e Justificativas

Para explorar quais são os benefícios e as justificativas para propor uma solução que resolva ou, pelo menos, ajude as empresas em seu enfrentamento, utilizaremos a metodologia Business Design Blueprint, que é uma abordagem que visa compreender profundamente o cotidiano do cliente, mapeando todas as ações necessárias para solucionar os desafios propostos. Através dessa metodologia, torna-se mais fácil e intuitivo explorar e desenvolver propostas de solução que atendam às necessidades específicas do cliente.

Foi realizado o preenchimento de uma tabela utilizando a metodologia e na Tabela 1 consta um pedaço do que foi feito para realizar essa análise mais aprofundada.

Tabela 1 - Business Desing Blueprint (Fonte: Autora)

Ações do Cliente	Estimar custos
------------------	----------------



<b>Objetivos</b>	Estimativas precisa e pontual de custos de construção.
<b>Atividades</b>	Coletando dados sobre preços de entrada, custos de mão de obra e escopo do projeto.
<b>Questões</b>	Como contabilizar flutuações de preços e incertezas?
<b>Barreiras</b>	Incertezas nas estimativas, flutuações de preços e falta de base de dados própria.
<b>Funcionalidades</b>	Modelos de orçamento personalizáveis, análise preditiva.
<b>Interação</b>	Geração de previsões de custos de insumos.
<b>Mensagem</b>	Compreensão das tendências e maior precisão no processo de orçamento.
<b>Onde ocorre</b>	Aplicativo de desktop ou móvel, plataforma baseada na web.
<b>Tarefas aparentes</b>	Gerar previsões
<b>Tarefas escondidas</b>	Cálculos complexos, análise de dados.
<b>Processos de Suporte</b>	Melhorias no algoritmo, atualizações do modelo.
<b>Saída desejada</b>	Previsões de preços e análise de tendências

Com as análises realizadas para o entendimento das tarefas necessárias para se realizar um orçamento de obras, percebe-se que a tarefa é algo complexo que envolve muitas fases diferentes, como por exemplo analisar o projeto corretamente, saber quais materiais empregar, entender todos os aspectos na execução de um serviço, solicitar alterações de projeto para o arquiteto e entre outros.

E dentro desses processos macros na realização de orçamentos da empresa sempre se busca relatórios de preços de insumos fornecidos por instituições governamentais o DER-ES, SINAPI entre outros, porém esses preços são os atuais do mercado da região e que estão sujeitos a alterações até o início da obra.

E para isso o Cientista de Dados pode ajudar criando uma solução que ajude a realizar essas previsões mais elaboradas e em vários níveis.

Pensando em como explicar essa proposta, preenchemos uma ferramenta muito útil chamada “Explicação da Proposta de Valor”, contendo assim as dores, remédios, o criador de ganho dessa proposta e qual será o produto e as tarefas que ele realizará.

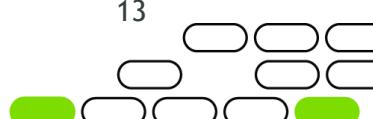
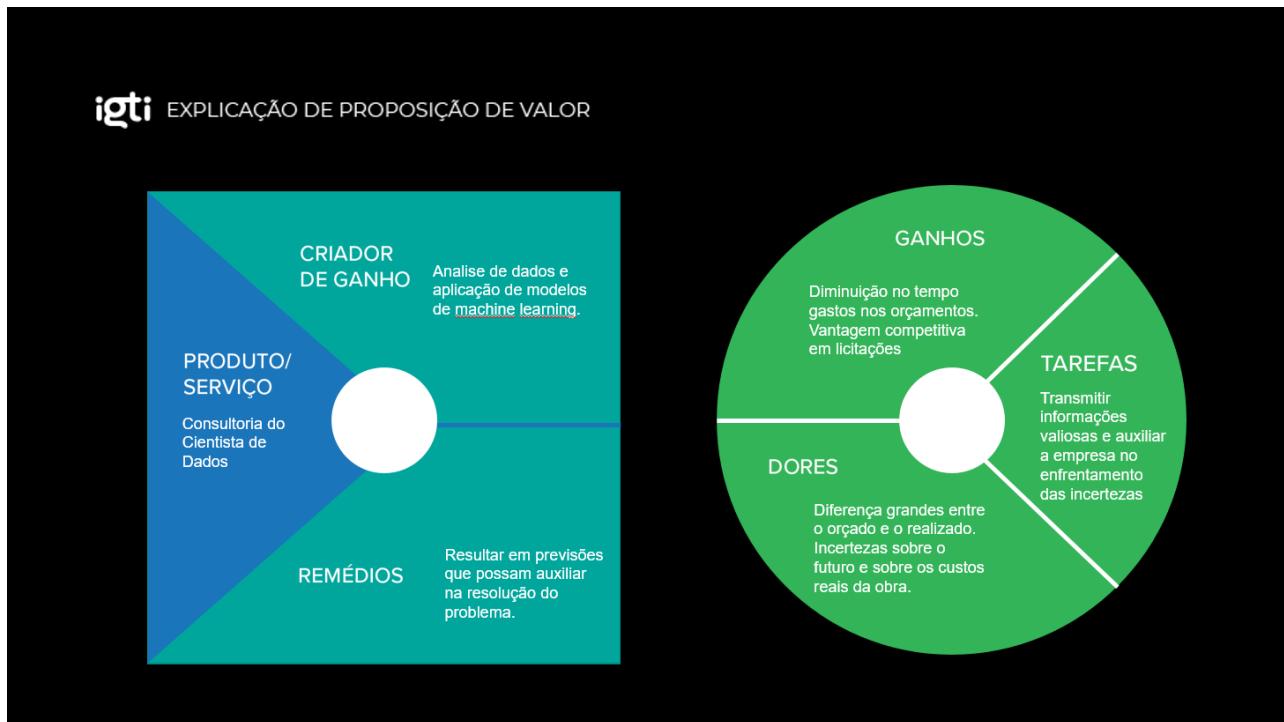


Figura 5. Explicação da proposta de Valor (Fonte: Autora)

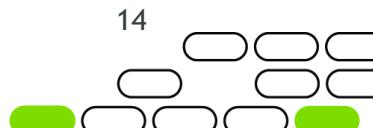


#### 1.1.4 Hipóteses

Estimativas de custo não confiáveis do projeto podem ter enormes implicações econômicas e financeiras. Portanto, é um aspecto ao qual as empresas dão muita atenção. No entanto, o grau de atenção e investimento não é dado na mesma proporção entre empresas pequenas e grandes. Fornecer serviços que ataquem alguns desses problemas que as empresas de pequeno a médio porte enfrentam de forma mais assertiva e objetiva daria uma visão de longo prazo e as auxiliaria nos problemas diários. Para expor melhor as hipóteses, a tabela 1.3 contém as observações e as hipóteses a serem testadas.

Tabela 3. Tabela de Observações e Hipóteses (Fonte: Autora)

Observação	Hipótese
Estimativas de custo imprecisas podem causar grandes prejuízos financeiros.	Obra de construção civil tem custos influenciados por fatores externos.
Pequenas e médias empresas resistem a mudanças devido a incertezas e custos de novas tecnologias.	Buscam serviços baratos com alto valor agregado sem alterar a estrutura atual.
Empresas usam bases públicas de preços quando não têm dados próprios para orçamentos.	Consulta e extração de informações brutas, sem análises complexas pelos orçamentistas.
Índices financeiros pouco usados para entender o mercado.	Índices são subutilizados por falta de compreensão de seu potencial.
Informações imprecisas sobre custos de materiais.	Empresas falham em atualizar preços de insumos por ser um processo lento e demorado.



Obtenção de cotações confiáveis e rápidas é difícil.

Empresas fazem parcerias com fornecedores, tornando cotações pouco confiáveis devido à alta concorrência.

Tecnologia insuficiente para orçamentação.

Falta de investimento e conhecimento limitam uso de ferramentas para tarefas.

Pequenas empresas competem por preços baixos em mercado acirrado.

Competitividade pode levar a prejuízos e falência por falta de consideração ao futuro.

Com as hipóteses expostas e todo o contexto do problema analisado, vamos listar abaixo as ideias para a sua resolução e priorizar aquelas mais possíveis.

Com o método de Matriz de Priorização BASICO (Benefícios, Abrangência, Satisfação, Investimento, Cliente, Operações) podemos priorizar quais soluções devem ser adotadas e, assim, criar um ranking. As notas que julgarão a prioridade das soluções devem seguir o seguinte critério:

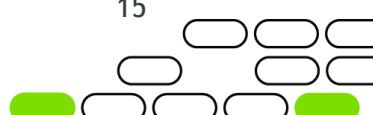
*Figura 5. Dimensões matriz BASICO*

Escala	B - Benefícios	A - Abrangência	S - Satisfação	I - Investimentos	C - Cliente	O - Operacionalidade
5	De vital importância	Total (de 70 a 100%)	Muito grande	Pouquíssimo investimento	Nenhum impacto	Muito fácil
4	Significativo	Muito grande (de 40 a 70%)	Grande	Algum investimento	Impacto pequeno	Fácil
3	Razoável	Razoável (de 20 a 40%)	Média	Médio investimento	Médio impacto	Média facilidade
2	Poucos benefícios	Pequena (de 5 a 20%)	Pequena	Alto investimento	Impacto grande	Difícil
1	Algum benefício	Muito pequena	Quase não é notada	Altíssimo investimento	Impacto muito grande no cliente	Muito difícil

(Fonte: <https://engenhariaexercicios.com.br/gestao-de-qualidade/matriz-gutbasico-conceito-aplicacao-das-matrices-priorizacao/>)

*Tabela 3: Matriz BASICO - priorização de ideias (Fonte: Autora)*

SOLUÇÕES	B	A	S	I	C	O	TOTAL	PRIORIZAÇÃO
Predição dos preços dos insumos relevantes da construção.	5	5	4	5	3	4	26	1
Projeção dos índices	5	5	3	5	3	5	26	1
Fornecimento de projeção por categoria de insumo	5	5	4	4	3	4	25	2
Extração, transformação e carregamento dos dados das obras	5	5	2	3	3	2	20	3
Atualização dos preços dos insumos, após selecionar o período.	4	3	2	2	3	4	18	4
Fornecimento de sugestões de preços de insumos	4	3	2	3	3	4	19	5
Recomendação de outros métodos construtivos	4	4	5	2	2	1	18	4



A partir da análise das ideias, priorizamos as aquelas com as maiores pontuações, e com isso “Predição dos preços dos insumos relevantes da construção.” e “Projeção de índices” serão as ideias selecionadas.

## 1.2 Solução

### 1.2.1 Objetivo SMART

O objetivo SMART (específico, mensurável, atingível, relevante e temporal) do projeto é prever as tendências de preços dos insumos mais representativos da construção civil com base em índices econômicos relevantes. Este objetivo é específico, pois aborda o desafio direto de proporcionar previsões mais precisas e confiáveis que os engenheiros de orçamento podem utilizar para melhorar a eficiência do planejamento de custos. É mensurável, pois o sucesso do projeto pode ser quantificado pela precisão e confiabilidade das previsões de tendências de preços produzidas pelo modelo. É atingível, dado que as técnicas e tecnologias necessárias para realizar previsões baseadas em índices econômicos são acessíveis e utilizáveis. A relevância do objetivo reside na solução que oferece a um problema prático e significativo enfrentado por engenheiros de orçamentos na indústria da construção civil. Por fim, é temporal, pois o projeto tem um prazo claramente definido para a conclusão e as previsões produzidas pelo modelo devem ser atualizadas regularmente para manter sua eficácia e relevância.

### 1.2.2 Premissas e Restrições

Premissas do projeto:

- Duas a quatro horas de disponibilidade por dia;
- Utilização da linguagem Python.



- Dados utilizados serão públicos e suas fontes são o SINAPI, CBIC e outros.

Restrições do projeto:

- Prazo até o fim de maio/2023;
- Orçamento somente para a assinatura mensal do Colaboratory do google.

Identificar e gerenciar os riscos são uma etapa crucial no desenvolvimento de um projeto. Além de definir as premissas e restrições, é importante avaliar possíveis obstáculos e saber como mitigá-los caso ocorram. Dessa forma, a equipe pode agir rapidamente e tomar decisões informadas para garantir o sucesso do projeto.

Abaixo está a Tabela 4 que é a matriz de riscos do projeto, nela estão mapeados os riscos potenciais, como por exemplo, a possibilidade de nossa solução não ter dados suficientes, ou que não conseguimos os resultados esperados nos modelos de aprendizado de máquina e neles nos identificamos os impactos potenciais assim como as ações a serem tomadas para cada caso.

*Tabela 4: Matriz de Riscos (Fonte: Autora)*

Riscos identificados	Impacto potencial	Ações Preventivas	Ações Corretivas
Atraso no cronograma do projeto	Tarefas acumuladas, entrega de projeto incompleto. Reprovação	Planejamento eficiente	Revisar e ajustar as Sprints, simplificar o projeto
Dificuldades na integração de sistemas e tecnologias	Funcionalidade comprometida, incompatibilidade	Avaliação prévia de compatibilidade, uso de padrões abertos	Solução de problemas, buscar alternativas compatíveis
Falhas na comunicação com o orientador	Atrasos, erros no desenvolvimento	Estabelecer canais de comunicação eficazes	Melhorar a comunicação, revisar e corrigir erros
Não conseguir os resultados esperados propostos	Reprovação, atraso no projeto	Começar pelo mais simples, ter um plano B	Pivatar de solução.
Baixa qualidade do código ou arquitetura	Dificuldade de manutenção, problemas de escalabilidade	Estabelecer padrões de qualidade, revisões de código	Refatorar o código, melhorar a arquitetura
Erros e bugs na solução final	Insatisfação do cliente, retrabalho	Testes rigorosos, envolvimento do cliente no processo de validação	Corrigir erros, melhorar a qualidade, aprender com os erros
Não conseguir bons resultados na avaliação do modelo	Não obter sucesso do objetivo	Verificar a existência de soluções parecidas para outras áreas	Utilizar outras técnicas ou outros indicadores
Dados insuficientes	Falha do projeto	Antes de iniciar verificar a disponibilidade de dados	Buscar mais fontes de dados



Armazenamento insuficiente	Gastos com cluster	Análise criteriosa dos dados	públicos	Utilizar técnicas de redução de dimensionalidade.
----------------------------	--------------------	------------------------------	----------	---

### 1.2.3 Backlog de Produto

A determinação do backlog do produto é um passo crucial no processo de desenvolvimento de software e projetos em geral. O backlog do produto compreende uma lista ordenada de funcionalidades, melhorias e correções de erros a serem desenvolvidos ao longo do projeto. Essa lista auxilia a compreender o escopo do trabalho, as prioridades e os objetivos.

O Trello é uma ferramenta de gerenciamento de projetos baseada em quadros Kanban que pode ser utilizada para organizar e acompanhar as sprints em um projeto ágil. Com o Trello, se pode criar listas para diferentes estágios do desenvolvimento, como "A Fazer", "Em Andamento" e "Concluído". São adicionados cartões às listas para representar tarefas específicas, estabelecer prazos e adicionar descrições e comentários detalhados. Esta abordagem visual e colaborativa torna o processo de desenvolvimento mais transparente e eficiente, permitindo que a equipe se concentre em entregar valor de forma rápida e iterativa.

Em suma, a definição do backlog do produto, a utilização de metodologias ágeis e o uso de ferramentas como o Trello são elementos fundamentais para o sucesso de um projeto. Essas práticas garantem que a equipe esteja alinhada, que o trabalho seja priorizado e que os objetivos do projeto sejam alcançados de forma eficiente e eficaz.

A construção do backlog proposto neste projeto envolve a organização e priorização das tarefas a serem executadas em três sprints de duas semanas cada. As atividades foram planejadas de forma a abordar, de maneira eficiente, a preparação e análise dos dados, a previsão de índices econômicos e a análise de viabilidade, bem como o desenvolvimento e validação do modelo de previsão de preços dos insumos. O backlog foi criado com o objetivo de fornecer um roteiro claro e estruturado para a equipe, garantindo o alinhamento das prioridades e o progresso efetivo em direção aos objetivos do projeto.



## Preparação e análise dos dados

- Obter e organizar os dados dos índices econômicos e insumos.
- Combinar e limpar os dados conforme necessário.
- Verificar e tratar valores ausentes ou inconsistentes nos dados.
- Realizar análises exploratórias dos dados para identificar tendências e padrões.

## Previsão de índices econômicos e análise de viabilidade

- Selecionar um modelo de previsão de séries temporais adequado para os índices econômicos.
- Preparar os dados para treinamento e teste do modelo.
- Treinar e testar o modelo usando os dados disponíveis.
- Avaliar o desempenho do modelo e decidir se a previsão de índices é viável para o projeto.
- Se a previsão for viável, prosseguir; caso contrário, ajustar o escopo do projeto conforme necessário.

## Desenvolvimento e validação do modelo de previsão de preços dos insumos

- Selecionar um modelo de previsão de séries temporais adequado para os preços dos insumos.
- Preparar os dados para treinamento e teste do modelo, incluindo as informações dos índices econômicos.
- Treinar e testar o modelo usando os dados disponíveis.
- Avaliar o desempenho do modelo e ajustar os hiperparâmetros, se necessário.
- Documentar os resultados e preparar a entrega final do projeto.



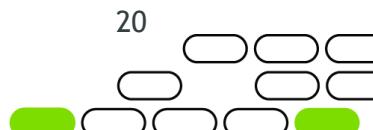
*Figura 6. Backlog do projeto no Trello. (Fonte: Autora)*

The screenshot shows a Trello board with the following structure:

- Projeto Aplicado - 2023** (Board title)
- Backlog do Produto** (List)
  - Obter e organizar os dados dos índices econômicos e insumos.
  - Combinar e limpar os dados conforme necessário.
  - Verificar e tratar valores ausentes ou inconsistentes nos dados.
  - Realizar análises exploratórias dos dados para identificar tendências e padrões.
  - Previsão de índices econômicos e análise de viabilidade
  - Selecionar um modelo de previsão de séries temporais adequado para os índices econômicos.
  - Preparar os dados para treinamento e teste do modelo.
  - Treinar e testar o modelo usando os dados disponíveis.
  - Avaliar o desempenho do modelo e decidir se a previsão de índices é viável para o projeto.
  - Desenvolvimento e validação do modelo de previsão de preços dos insumos
  - Agrupar os insumos em categorias relevantes.
  - Selecionar um modelo de previsão de séries temporais adequado para os preços dos insumos.
  - + Adicionar um cartão
- Visível à Área de trabalho** (Section)
  - Backlog do Produto
  - Sprint 1
  - Sprint 2
  - Sprint 3
- Quadro** (Section)
  - Sprint 1
  - Sprint 2
  - Sprint 3
- Power-Ups**
- Automação**
- Filtro**
- Compartilhar**

## 2. Área de Experimentação.

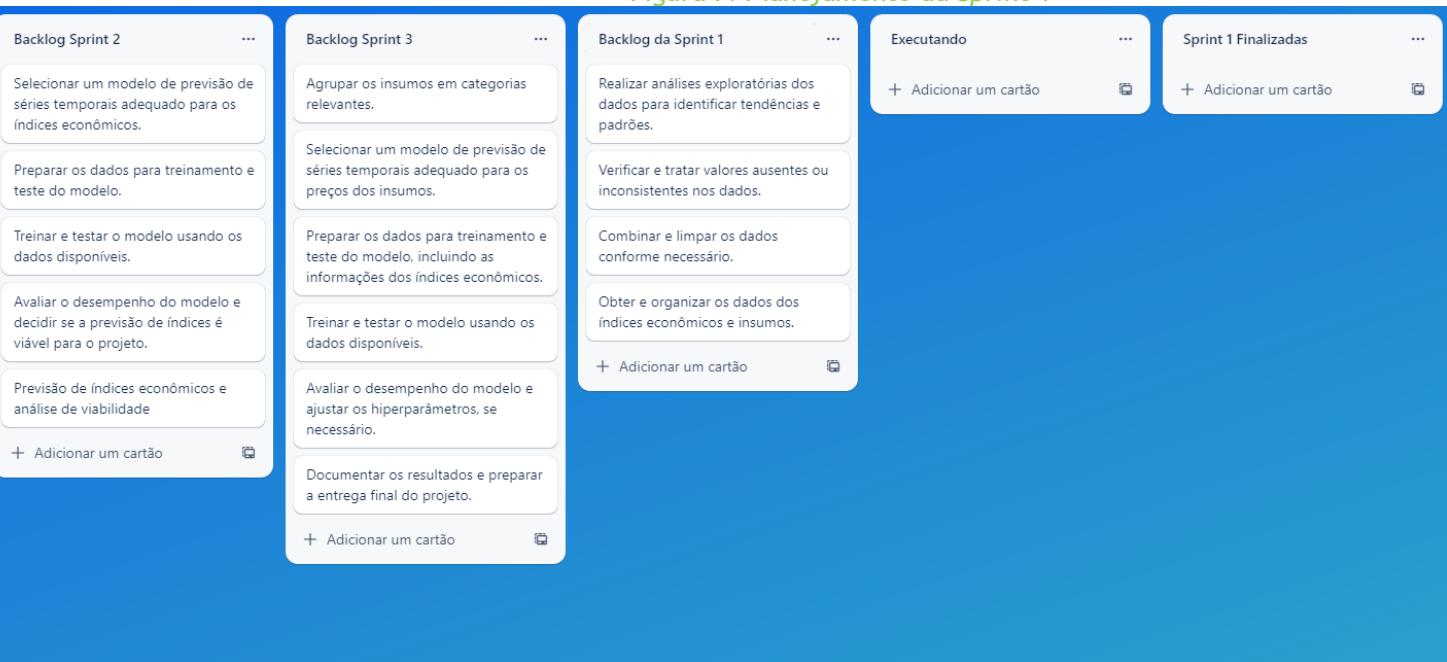
### 2.1 Sprint 1



## 2.1.1 Solução

- Evidência do planejamento:

*Figura 7. Planejamento da Sprint 1*



- Evidência da execução de cada requisito:

Github do projeto: [https://github.com/deboraxhx/projeto\\_aplicado](https://github.com/deboraxhx/projeto_aplicado)

ITEM 1: Obter e organizar os dados dos índices econômicos e insumos.

*Figura 8: Dados baixados do SINAPI*



- |  |  |
|--|--|
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_032015_N...</a> | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_201704_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_032016_N...</a> | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_201705_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_032017_N...</a> | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_201801_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_042010</a>      | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_201806_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_042011</a>      | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_201807_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_042013_N...</a> | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_201808_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_042014_N...</a> | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_201809_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_042015_N...</a> | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_201810_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_052010</a>      | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_201811_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_052011</a>      | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_201812_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_052013_N...</a> | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_201901_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_052014_N...</a> | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_201902_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_052015_N...</a> | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_201903_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_052016_N...</a> | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_201904_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_062010</a>      | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_201905_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_062011</a>      | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_201906_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_062013_N...</a> | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_201907_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_062014_N...</a> | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_201908_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_062015_N...</a> | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_201909_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_062016_N...</a> | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_201910_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_072009</a>      | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_201911_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_072010</a>      | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_201912_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_072011</a>      | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_202001_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_072013_N...</a> | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_202002_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_072014_N...</a> | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_202003_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_072015_N...</a> | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_202101_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_082009</a>      | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_202102_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_082010</a>      | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_202103_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_082011</a>      | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_202104_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_082013_N...</a> | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_202105_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_082014_N...</a> | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_202106_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_082015_N...</a> | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_202107_N...</a> |
| <a href="#"> SINAPI_Preco_Ref_Insumos_ES_082016_N...</a> | <a href="#"> SINAPI_Preco_Ref_Insumos_ES_202108_N...</a> |

Figura 9: Dados dos insumos e relatórios

- [MATERIAIS](#)
- [cub\\_es](#)
- [selic](#)
- [inpc](#)
- [relatorio\\_insumos](#)



Figura 10. Código da extração dos dados do .pdf

```

# Definir parâmetros da extração de dados
parametros = {
    'pages': 'all',
    'guess': False,
    'area': [188, 0, 810, 700],
    'columns': [32, 74, 430, 500],
    'stream': True,
}

# Caminho para a pasta com os arquivos PDF
caminho_pasta_pdf = 'dados/source/PDF3'
caminho_pasta_csv = 'dados/raw/raw3'

# Iterar sobre todos os arquivos na pasta de PDFs e converter para CSV
for nome_arquivo in os.listdir(caminho_pasta_pdf):
    if nome_arquivo.lower().endswith('.pdf'):
        # Extrair dados da tabela do arquivo PDF
        caminho_arquivo_pdf = os.path.join(caminho_pasta_pdf, nome_arquivo)
        dados_tabela = tabula.read_pdf(caminho_arquivo_pdf, **parametros)

        # Concatenar os DataFrames em uma única tabela
        tabela_completa = pd.concat(dados_tabela)

        # Adicionar uma coluna com o nome do arquivo e excluir a terceira coluna
        tabela_completa['data'] = os.path.splitext(nome_arquivo)[0]
        tabela_completa.drop(
            tabela_completa.columns[[2, 3]], axis=1, inplace=True)

        # Salvar a tabela completa como um arquivo CSV com o nome do arquivo de origem,
        nome_csv = os.path.join(
            caminho_pasta_csv, os.path.splitext(nome_arquivo)[0] + '.csv')
        tabela_completa.to_csv(nome_csv, index=False,
                              sep=';', quoting=csv.QUOTE_ALL)

# Lista para armazenar os DataFrames de cada arquivo CSV
dfs = []

# Iterar sobre todos os arquivos na pasta de CSVs e adicionar à lista de DataFrames
for caminho_arquivo in os.listdir(caminho_pasta_csv):
    if caminho_arquivo.lower().endswith('.csv'):
        # Leitura do arquivo CSV
        df = pd.read_csv(
            os.path.join(caminho_pasta_csv, caminho_arquivo),
            header=0,
            names=['codigo', 'descricao', 'valor', 'data'],
            skipfooter=1,
            engine='python',
            quoting=3,
            sep=";",
            error_bad_lines=False,
            na_values=['', 'NA'])
    )

    # Remover as aspas duplas de todas as células do DataFrame
    df.replace('"', '', regex=True, inplace=True)

    # Adicionar o DataFrame à lista
    dfs.append(df)

# Concatenar os DataFrames em um único DataFrame
dados_completos = pd.concat(dfs, ignore_index=True)

# Remover linhas com valores nulos
dados_completos.dropna(inplace=True)

# Salvar o arquivo completo
dados_completos.to_csv(
    'dados/merged/dados_completos_3.csv', index=False, sep=';')

```



Figura 11. Código da extração dos dados .xls

```
import os
import pandas as pd

# Caminho para a pasta com os arquivos
caminho_pasta = r'dados/source/XLS DATA 3'

# Lista para armazenar os dataframes de cada arquivo
dfs = []

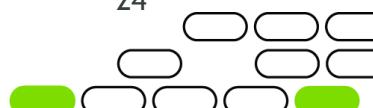
# Lista com os nomes de cada coluna

colunas = ['codigo', 'descricao', 'unidade', 'origem', 'valor']

# Iterar sobre todos os arquivos na pasta
for nome_arquivo in os.listdir(caminho_pasta):
    if nome_arquivo.lower().endswith('.xls'):
        # Ler o arquivo e pular as 6 primeiras linhas
        df = pd.read_excel(os.path.join(caminho_pasta, nome_arquivo),
                           skiprows=6, names=colunas)
        # Deletar as colunas 3 e 4
        df.drop(df.columns[[2, 3]], axis=1, inplace=True)
        # Adicionar uma coluna com o nome do arquivo
        df['data'] = nome_arquivo.split('.')[0][-6:]
        # Adicionar o dataframe à lista
        dfs.append(df)

# Concatenar todos os dataframes em um só
df_concat = pd.concat(dfs)

# Salvar o dataframe em um arquivo .csv
df_concat.to_csv('dados/merged/dados_completos_5.csv', index=False, sep=';')
```



- ITEM 2: Combinar e limpar os dados conforme necessário.

Como os dados foram extraídos em formatos diferentes, foram necessários serem realizados alguns tratamentos para padronizar e deletar os dados nulos, principalmente para os arquivos em pdf. E por isso foram criados arquivos ETL\_pt para cada extração diferente.

*Figura 12. Importação e padronização dos dados*

#### LEITURA DOS DADOS

```
[ ] # Ler o arquivo CSV e tratar os dados
dados = pd.read_csv('/content/dados_completos_5.csv',
                     engine='python',
                     sep=";")

```

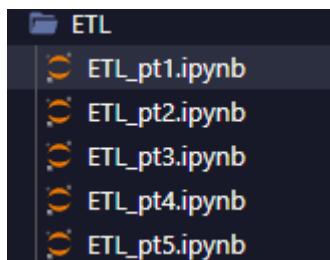
#### CONVERSÃO DOS DADOS

```
[ ] # Converter a coluna de data para o formato mm/yyyy
dados['data'] = pd.to_datetime(dados['data'].astype(str).str.zfill(6), format='%m%Y').dt.strftime('%m/%Y')

# Aplicar a expressão regular na coluna e converter para float
dados['codigo'] = dados['codigo'].str.replace(r'^[0-9,\.]', '', regex=True)

[ ] # Conversão para float
dados['codigo'] = pd.to_numeric(dados['codigo'].str.replace(',', '.'), errors='coerce')
```

#### DADOS FALTANTES



```
[ ] # Verificando os dados faltantes

dados_faltantes = dados[dados.isna().any(axis=1)]
print(dados_faltantes['valor'].unique())
print(dados_faltantes['codigo'].unique())
```

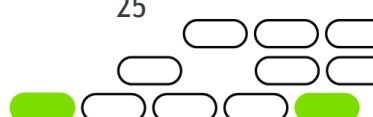
```
[nan]
[ nan 5279. 5280. 5303. 5299.]
```

dados.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21169 entries, 0 to 21168
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype  
---  --          -----          ----- 
 0   codigo       21165 non-null   float64 
 1   descricao    21161 non-null   object  
 2   valor        21161 non-null   object  
 3   data         21169 non-null   object  
dtypes: float64(1), object(3)
memory usage: 661.7+ KB
```

```
[ ] # Deletar as linhas que contêm dados faltantes

dados.dropna(inplace=True)
```



Após a limpeza preliminar foram combinados os dados em um só arquivo chamado “`insumos.csv`”.

Nas figuras abaixo mostra as combinações feitas para análise dos dados e operações realizadas para retirar informações que se julgou irrelevantes.

*Figura 13. Combinação dos dados de MATERIAIS com INSUMOS*

#### 0. PREPARAÇÃO DOS DADOS

##### 0.1 CARREGANDO OS DADOS

```
[3] # Schema do dados
schema = {
    'codigo': 'int64',
    'valor': 'float64',
}

insumos = pd.read_csv('/content/drive/MyDrive/PROJETO APLICADO/dados_insumos.csv',
                     encoding='ISO-8859-1', sep = ";", dtype=schema)

classif = pd.read_csv('/content/drive/MyDrive/PROJETO APLICADO/materiais categorizados/MATERIAIS.csv',
                     encoding='ISO-8859-1', sep = ";")
```

##### 0.2 CHECAGEM DAS INFORMAÇÕES DOS DADOS PARA COMBINAÇÃO

42 células ocultas

##### 0.3 SELEÇÃO DOS DADOS

```
[6] # Deletar as colunas sem utilizade
classif.drop(columns = ['PRECO MEDIANO R$', 'ENCARGOS', 'mês', 'ano', 'localidade'], inplace=True)

# Renomeação dos dados classif
classif.columns =[ 'codigo', 'descricao', 'unidade', 'coeficiente', 'cat_1',
                   'commodities', 'classe', 'cat_2', 'cat_3', 'cat_4', 'outros']
```

##### 0.4 COMBINAÇÃO DOS DADOS

```
[7] # Realiza o merge usando a coluna "codigo" como chave de junção
dados = pd.merge(insumos, classif, on='codigo')
```



*Figura 14. Combinação dos dados INSUMO e RELATORIO DE INSUMOS*

▼ 1. PREPARAÇÃO DOS DADOS

▼ 1.1 CARREGANDO OS DADOS

```
✓ [65] # Schema do dados
      schema = {
          'codigo': 'int64',
          'valor': 'float64',
      }

      path = '/content/drive/MyDrive/PROJETO APLICADO/'

      relatorio = pd.read_csv(path+'materiais categorizados/relatorio_insumentos.csv',
                               encoding='ISO-8859-1', sep = ";")
```

► 1.2 CHECAGEM DAS INFORMAÇÕES DOS DADOS PARA COMBINAÇÃO

[ ] 45 células ocultas

▼ 1.3 SELEÇÃO DOS DADOS

```
✓ [72] # Deletar as colunas sem utilizade
      relatorio.drop(columns = ['vinculo'], inplace=True)
```

▼ 1.4 COMBINAÇÃO DOS DADOS

```
✓ [73] # Realiza o merge usando a coluna "codigo" como chave de junção
      dados_1 = pd.merge(insumentos, relatorio, on='codigo')
```

*Figura 15. Combinação dos dados INSUMO e RELATORIO DE INSUMOS*

▼ 2.1 CARREGANDO DOS DADOS

```
✓ [104] # Caminho dos dados
      caminho_pasta = '/content/drive/MyDrive/PROJETO APLICADO/indices'

      cub = pd.read_csv(caminho_pasta+'/cub_es.csv', sep=';')
      indice = pd.read_csv(caminho_pasta+'/inpc.csv', sep=';')
      juros = pd.read_csv(caminho_pasta+'/selic.csv', sep=';')
```

► 2.2 CHECAGEM DAS INFORMAÇÕES DOS DADOS PARA COMBINAÇÃO

[ ] 48 células ocultas

▼ 2.3 TRATAMENTO DOS DADOS

```
✓ [109] # Transformando os dados para combinação
      cub['data'] = pd.to_datetime(cub['data'], format='%d/%m/%Y')
      cub['data'] = cub['data'].dt.strftime('%m/%Y')

      indice['data'] = pd.to_datetime(indice['data'], format='%d/%m/%Y')
      indice['data'] = indice['data'].dt.strftime('%m/%Y')

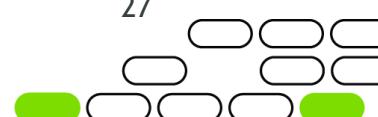
      juros['data'] = pd.to_datetime(juros['data'], format='%d/%m/%Y')
      juros['data'] = juros['data'].dt.strftime('%m/%Y')
```

▼ 2.4 COMBINAÇÃO DOS DADOS

```
✓ [ ] # Realiza o merge usando a coluna "data" como chave de junção
      indice1 = pd.merge(indice, juros, on='data')

      indices = pd.merge(indice1, cub, on='data')

      dados_2 = pd.merge(insumentos, indices, on='data')
```



- ITEM 3: Verificar e tratar valores ausentes ou inconsistentes nos dados.

*Figura 16. Verificação de dados inconsistentes*

```
# Calculo para conferencia dos dados:
a = len(dados['codigo'].unique())
b = len(dados['data'].unique())

qtd_esperada = a * b
qtd_real = dados.shape[0]

diff = qtd_esperada - qtd_real

print('Quantidade esperada de instâncias:',qtd_esperada)
print('Quantidade real de instâncias:',qtd_real)
print('Diferença:', diff )
```

Quantidade esperada de instâncias: 534612  
 Quantidade real de instâncias: 413401  
 Diferença: 121211

*Figura 17. Tratamento de dados inconsistentes*

Filtrar os dados por somente os insumos que estiverem presentes em todos os meses

```
# Verificação dos dados
contagem_check = dados.groupby('codigo', as_index=False).count()
contagem_check['count'] = contagem_check['descricao_X']
contagem_check = contagem_check[['codigo', 'count']]

# Realiza o merge usando a coluna "codigo" como chave de junção
dados = pd.merge(dados, contagem_check, on='codigo')

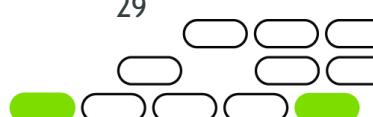
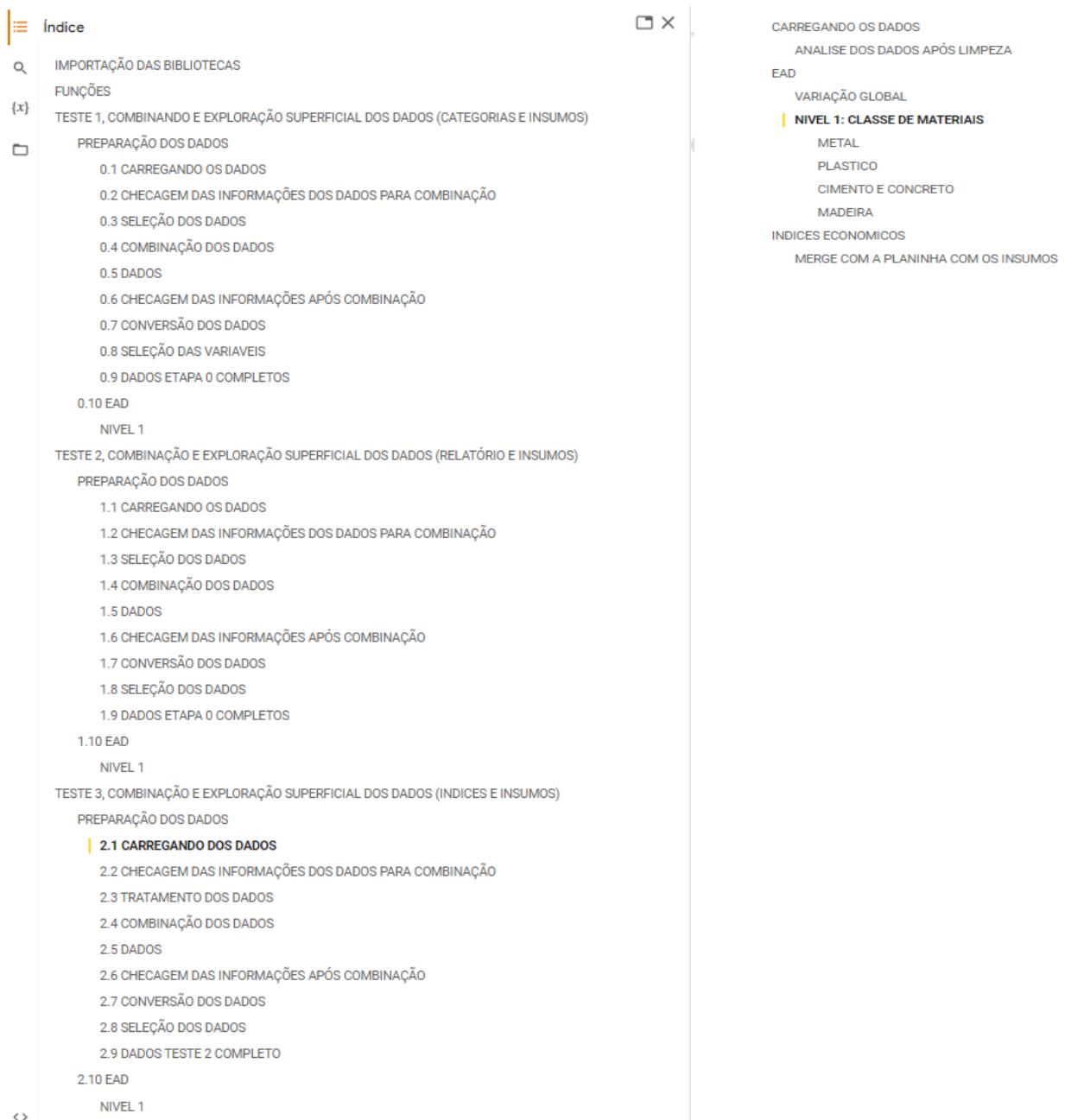
# Filtro
dados_filtrado = dados[dados['count'] > 155]
```

- ITEM 4: Realizar análises exploratórias dos dados para identificar tendências e padrões.

Foram realizados testes de combinações de dados para definir qual será a melhor abordagem seguir. Na figura 18, no lado esquerdo da imagem mostra a estrutura do notebook em que foram realizados os testes de combinações. E no lado direito mostra o EAD mais profundo realizado para o teste 1.



Figura 18. Estruturas das analyses



Abaixo estão algumas evidencias das analise exploratória realizada:

*Figura 19. Visualização dos dados teste 1, combinação materiais classificados e insumos.*

valor x data

```
[64] # Converte a coluna 'data' para o formato de data com o formato atual
dados['data'] = pd.to_datetime(dados['data'], format='%m/%Y')

# Agrupa os valores por mês e calcula a soma
agrupoado_por_mes = dados.groupby(pd.Grouper(key='data', freq='M')).mean()

# Plota um gráfico de linhas com os valores
plt.subplots(figsize=(20, 4))
plt.plot(agrupoado_por_mes.index, agrupoado_por_mes['valor'])
plt.title('Variação mensal do preços ao longo do tempo(GLOBAL)')
plt.xlabel('data')
plt.ylabel('Valor médio')
plt.show()
```



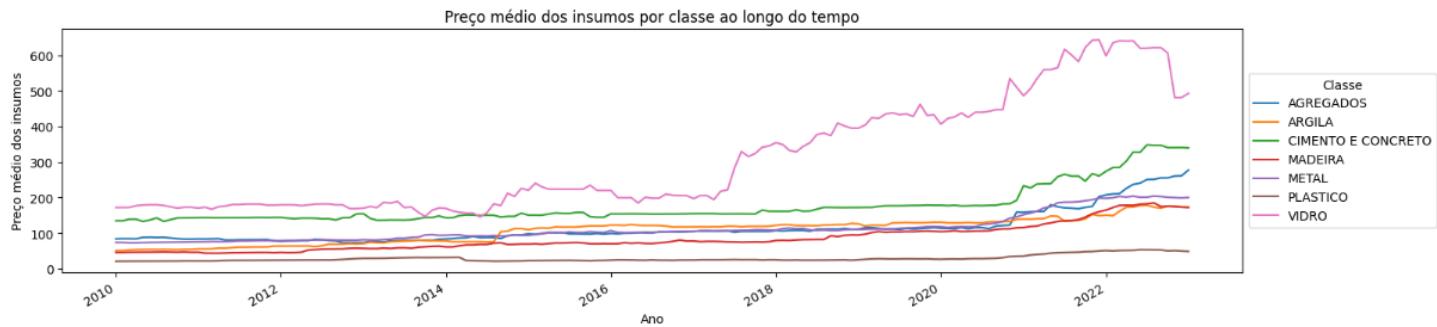
*Figura 20. Visualização dos dados teste 1, analisando as classes dos insumos.*

```
# Agrupa os dados por classe e data e calcula a média dos valores
agrupoado = agrupo_classe.groupby(['classe', 'data']).mean().reset_index()

# Cria um dataframe com as colunas "data" e uma coluna para cada classe
pivotado = agrupoado.pivot(index='data', columns='classe', values='valor')

# Plota um gráfico de linha para cada classe
ax = pivotado.plot(figsize=(17,4))

# Adiciona um título e uma legenda
ax.set_title('Preço médio dos insumos por classe ao longo do tempo')
ax.legend(title='Classe', loc='center left', bbox_to_anchor=(1, 0.5))
plt.xlabel('Ano')
plt.ylabel('Preço médio dos insumos')
plt.show()
```



*Figura 21. Visualização dos dados teste 2, análise da macro classe MATERIAL, combinação relatório de insumos x insumos.*

valor x data para a categoria MATERIAL

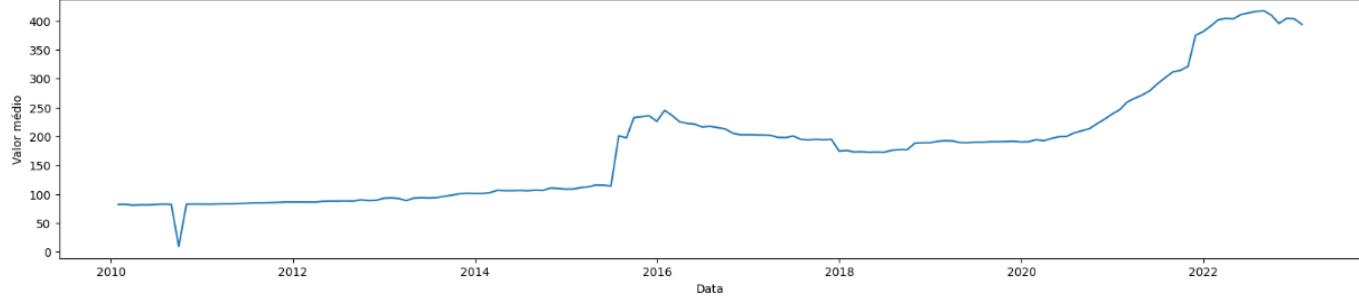
```
# Converte a coluna 'data' para o formato de data com o formato atual
dados_1['data'] = pd.to_datetime(dados_1['data'], format='%m/%y')

dados_1_material = dados_1[dados_1['macro_classe'] == 'MATERIAL']

# Agrupa os valores por mês e calcula a soma
agrupado_por_mes = dados_1_material.groupby(pd.Grouper(key='data', freq='M')).mean()

# Plota um gráfico de linha com os valores
plt.subplots(figsize=(20, 4))
plt.plot(agrupado_por_mes.index, agrupado_por_mes['valor'])
plt.title('Variação mensal do preços ao longo do tempo(GLOBAL)')
plt.xlabel('Data')
plt.ylabel('Valor médio')
plt.show()
```

Variação mensal do preços ao longo do tempo(GLOBAL)



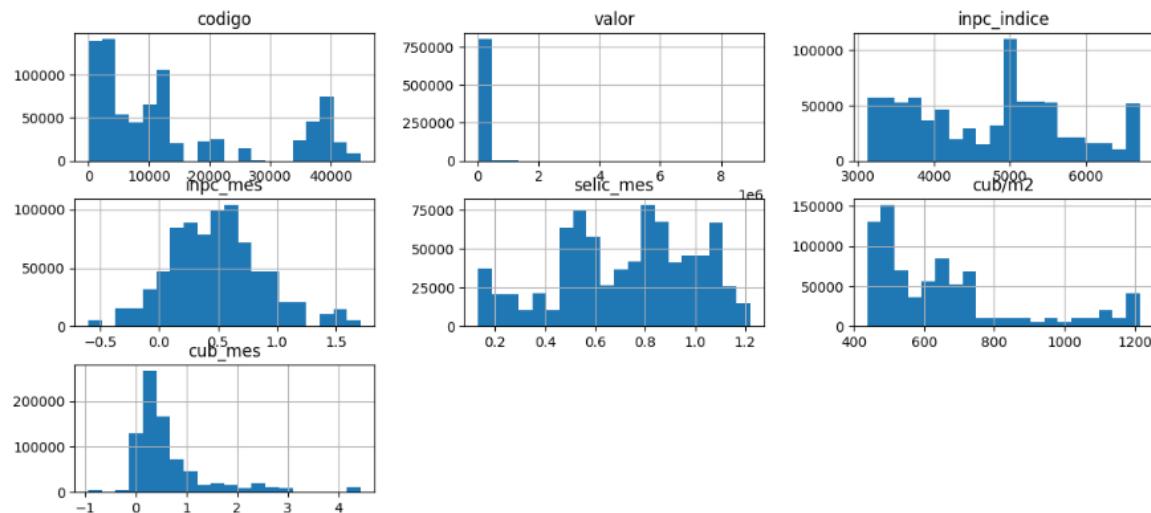
*Figura 22. Análise descritiva e visual da distribuição dos dados do teste 3, insumos x índices.*

# Quantis

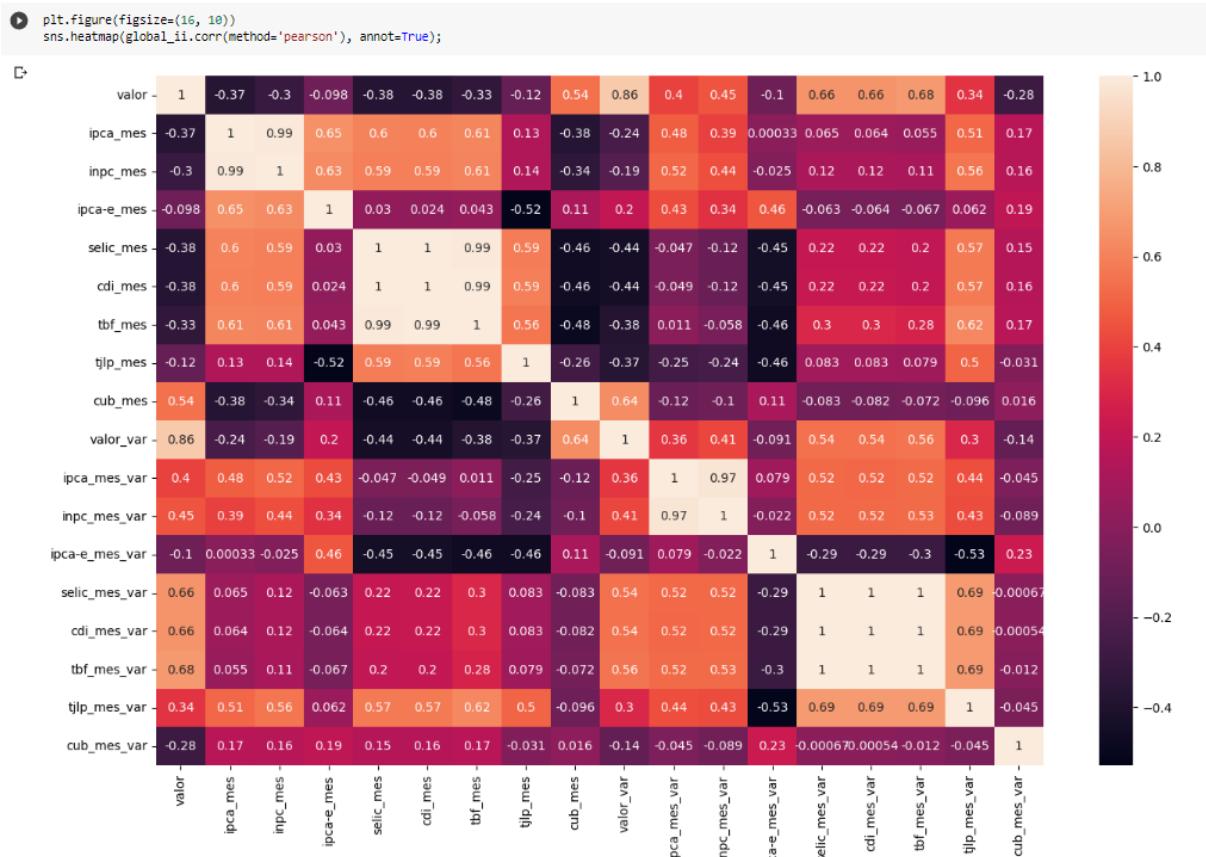
```
round(num_attributes.describe([0.05, 0.25, 0.50, 0.75, 0.90, 0.95, 0.99]).T, 2)
```

	count	mean	std	min	5%	25%	50%	75%	90%	95%	99%	max
codigo	807700.0	14437.66	13989.10	1.00	581.00	3461.00	10519.00	21040.00	39144.00	39879.00	43053.00	44945.00
valor	807700.0	18153.99	173205.31	0.00	1.24	8.20	28.81	148.30	1153.68	8477.33	451508.82	8958602.97
inpc_indice	807700.0	4709.61	1002.55	3124.76	3205.83	3813.73	4918.12	5355.77	6160.89	6614.89	6685.86	6706.15
inpc_mes	807700.0	0.49	0.40	-0.60	-0.13	0.21	0.49	0.73	0.99	1.20	1.51	1.71
selic_mes	807700.0	0.72	0.28	0.13	0.19	0.52	0.76	0.94	1.07	1.12	1.18	1.22
cub/m2	807700.0	657.54	212.57	437.51	456.24	490.32	607.76	720.29	1075.63	1175.75	1209.77	1214.56
cub_mes	807700.0	0.65	0.79	-0.94	0.07	0.18	0.40	0.81	1.74	2.39	4.27	4.44

```
[123] # Visualização da distribuição dos dados
dados_2.hist(figsize=(14, 6), bins=20);
```



*Figura 23. Analise de correlação com o valor médio e variação mensal dos preços dos insumos, insumos x índices.*



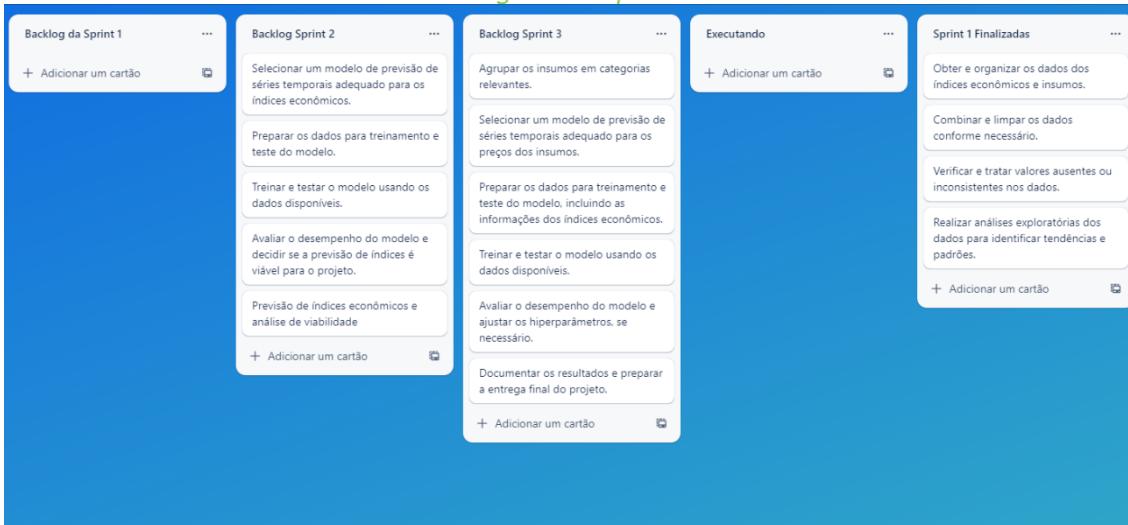
*Figura 24. Conclusões da análise de correlação da figura 24.*

- A variação mensal do preço médio dos insumos em termos gerais tem uma boa correlação com o cub mensal.
- Existem correlações boas entre valor(preços médio) as variações mensais da selic, cdi e tbf, mas esses índices são altamente correlacionados e por isso só ficará um.
- Assim como o IPCA e o INPC. Só ficará um índice de cada. Um da taxa de juros, um de índices de preço.
- O índice cub também se manterá

### ● Evidência dos resultados:

Com essa sprint foram extraídos, carregados e transformados os dados necessários para o projeto. Após, realizou-se as análises exploratórias dos dados, e com isso podemos tirar conclusões e insights para a continuação do projeto.

*Figura 25. Sprint Concluída*



Abaixo estão as evidencias dos resultados obtidos ao final da Sprint:

*Figura 25. Dados da combinação do teste 1: insumos x material classificado.*

codigo	valor	data	descricao_y	coeficiente	cat_1	commodities	classe	cat_2	cat_3	cat_4	outros	x
135341	11316	264.80	2017-02-01	TAMPÃO FOFO SIMPLES COM BASE, CLASSE B125 CARG...	0.868217	MATERIAL	ferro fundido	METAL	Instalações hidrossanitárias e hidráulicas	Ferragens de outros construtores	NaN	NaN
23079	5078	7.87	2013-01-01	PREGO DE ACO POLIDO COM CABECA 16 X 27 (2 1/2 ...	1.072017	MATERIAL	aco polido	METAL	Materiais de Fixação	Parafusos, porcas, chumbadores	NaN	NaN
330273	34734	1508.31	2016-05-01	DISJUNTOR TERMICO E MAGNETICO AJUSTAVEIS, TRIP...	155.656458	MATERIAL	Metal e plástico.	OUTROS MATERIAIS DE CONSTRUÇÃO	Instalações Elétricas e Comunicação	NaN	NaN	NaN
360430	40839	52.38	2017-02-01	PARAFUSO ASTM A307 - GRAU A, SEXTAVADO, ZINCA...	52.308802	MATERIAL	aço carbono	METAL	Materiais de Fixação	Parafusos, porcas, chumbadores	NaN	NaN
225421	113	8.70	2012-12-01	ADAPTADOR PVC SOLDAVENT CURTO COM BOLSA E ROSCA...	2.690598	MATERIAL	pvc	PLÁSTICO	Instalações hidrossanitárias e hidráulicas	Tubos e conexões (rígidos)	NaN	Água fria
43576	4998	315.46	2016-06-01	PORTA DE MADEIRA-DE-LEI TIPO MEXICANA SEM EMEN...	1.436869	MATERIAL	madeira angelim	MADEIRA	Esquadrias	Portas e Janelas	NaN	NaN
58141	3679	410.30	2022-11-01	JUNTA DILATACAO ELASTICA PARA CONCRETO (FUGENB...	310.836735	MATERIAL	plástico	PLÁSTICO	Materiais estruturais	Plásticos de outros construtores	NaN	NaN
336265	38600	5.40	2022-11-01	CANAleta DE CONCRETO ESTRUTURAL 14 X 19 X 39 C...	2.040816	MATERIAL	concreto	CIMENTO E CONCRETO	Blocos Estruturais e de Vedaçao	Produtos de concreto pré-moldado	Blocos, tijolos, telhas e lajes	Não armado
109226	11749	36.91	2016-08-01	VALVULA DE ESFERA BRUTA EM BRONZE. BITOLA 3/4 ...	0.672437	MATERIAL	bronze	METAL	Instalações hidrossanitárias e hidráulicas	Torreiras e Valvulas para louças sanitárias	NaN	NaN
59651	3746	40.55	2011-08-01	LAJE PRE-MOLDADA TREPLICADA (LAJOTAS + VIGOTAS)...	1.760465	MATERIAL	concreto armado	CIMENTO E CONCRETO	Materiais estruturais	Produtos de concreto pré-moldado	Blocos, tijolos, telhas e lajes	Armado

*Figura 26. Conclusões teste 1.*

#### CONCLUSÕES TESTE 1:

- Algumas variáveis têm valores faltantes, mas na classificação mais macro dos insumos os dados estão completos.
- Na variável CLASSE, temos uma grande diferença nas quantidades de insumo de cada categoria. 'METAL', 'PLÁSTICO', 'MADEIRA' e 'CIMENTO E CONCRETO' são os mais plausíveis para prosseguir com as análises.
- No gráfico da média dos valores dos insumos por mês, nota-se aumentos consideráveis em determinados pontos.
- A quantidade esperada para a série histórica está completa não bate com a realidade, o que pode explicar isso é a descontinuação de itens ao longo da série histórica. Portanto, explica o gráfico com aumentos inesperados.



*Figura 27. Dados da combinação do teste 2, relatório de insumos x insumos.*

	codigo	valor	data	cod_rep	Descrição do Insumo	unidade	coeficiente	categoria	macro_classe	🔗
0	414	0.07	2010-01-01	411	ABRACADEIRA DE NYLON PARA AMARRACAO DE CABOS, ...	UN	0,325428	REPRESENTADO	MATERIAL	🔗
1	414	0.06	2011-01-01	411	ABRACADEIRA DE NYLON PARA AMARRACAO DE CABOS, ...	UN	0,325428	REPRESENTADO	MATERIAL	🔗
2	414	0.06	2012-01-01	411	ABRACADEIRA DE NYLON PARA AMARRACAO DE CABOS, ...	UN	0,325428	REPRESENTADO	MATERIAL	🔗
3	414	0.05	2013-01-01	411	ABRACADEIRA DE NYLON PARA AMARRACAO DE CABOS, ...	UN	0,325428	REPRESENTADO	MATERIAL	🔗
4	414	0.06	2014-01-01	411	ABRACADEIRA DE NYLON PARA AMARRACAO DE CABOS, ...	UN	0,325428	REPRESENTADO	MATERIAL	🔗
5	414	0.04	2015-01-01	411	ABRACADEIRA DE NYLON PARA AMARRACAO DE CABOS, ...	UN	0,325428	REPRESENTADO	MATERIAL	🔗
6	414	0.04	2016-01-01	411	ABRACADEIRA DE NYLON PARA AMARRACAO DE CABOS, ...	UN	0,325428	REPRESENTADO	MATERIAL	🔗
7	414	0.05	2017-01-01	411	ABRACADEIRA DE NYLON PARA AMARRACAO DE CABOS, ...	UN	0,325428	REPRESENTADO	MATERIAL	🔗
8	414	0.04	2018-01-01	411	ABRACADEIRA DE NYLON PARA AMARRACAO DE CABOS, ...	UN	0,325428	REPRESENTADO	MATERIAL	🔗
9	414	0.04	2019-01-01	411	ABRACADEIRA DE NYLON PARA AMARRACAO DE CABOS, ...	UN	0,325428	REPRESENTADO	MATERIAL	🔗

*Figura 28. Conclusões teste 2.*

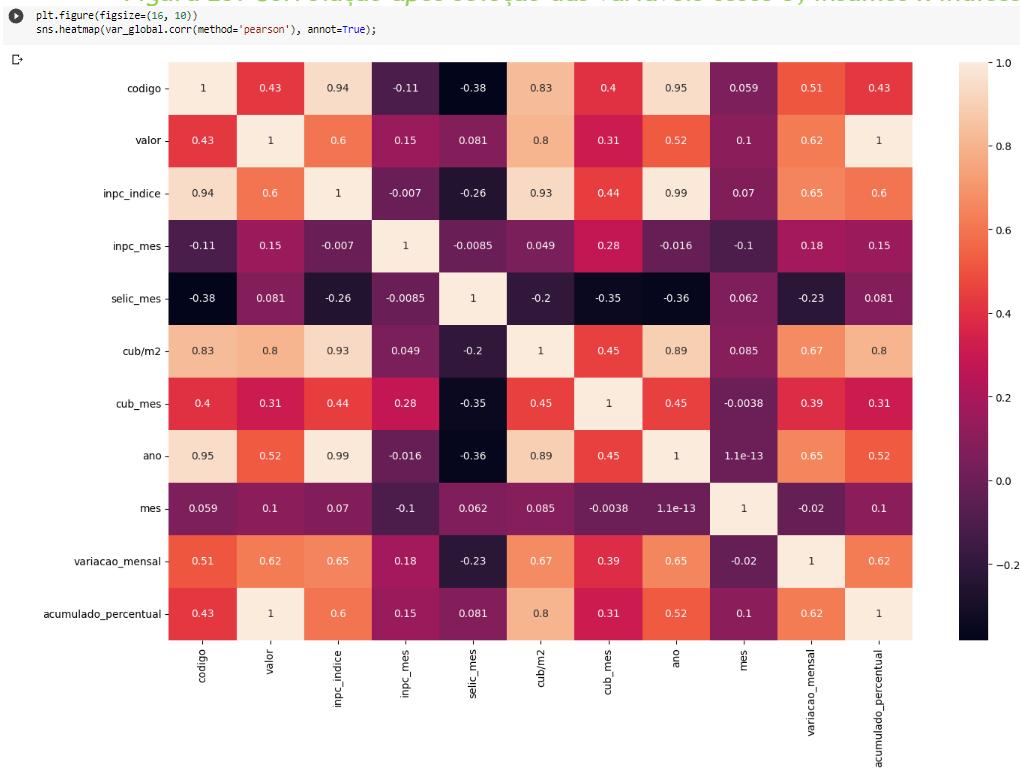
#### CONCLUSÕES TESTE 2:

- Essa junção tem como objetivo pegar todos os dados de insumos que estão sendo atualizados atualmente pelo SINAPI para realizar uma análise em insumos que não foram incluídos nas categorias para predição.
- Algumas variáveis têm valores faltantes, mas são solucionados quando filtrados, por isso manterei.
- No gráfico da média dos valores dos insumos por mês, nota-se aumentos consideráveis em determinados pontos.
- A quantidade esperada para a série histórica está completa não bate com a realidade, o que pode explicar isso é a descontinuação de itens ao longo da série histórica. Portanto, explica o gráfico com aumentos inesperados.
- A diferença de insumos entre a quantidade de insumos pode ser analisada de duas formas: primeiro, a descontinuação de itens e algum problema na extração dos dados.
- Mesmo com essas diferenças, podemos filtrar os dados e deixar somente os que têm as séries históricas completas. Além disso, podemos utilizar o cod\_rep e o coeficiente para nos ajudar a prever os valores dos insumos, visto que eles categorizam e calculam seu preço de sua forma. Com isso em mente, podemos então seguir por um caminho mais robusto.

Figura 24. Dados da combinação do teste 3, insumos x índices.

	codigo	descricao	valor	data	inpc_index	inpc_mes	selic_mes	cub/m2	cub_mes
0	414	ABRACADEIRA DE NYLON PARA AMARRACAO DE CABOS, ...	0.07	2010-01-01	3124.76	0.88	0.66	441.67	0.01
1	4715	PEDRA ARDOSIA CINZA IRREGULAR	7.28	2010-01-01	3124.76	0.88	0.66	441.67	0.01
2	4491	PECA DE MADEIRA 3A/4A QUALIDADE 7,5 X 7,5CM (3...	3.67	2010-01-01	3124.76	0.88	0.66	441.67	0.01
3	4500	PECA DE MADEIRA 3A/4A QUALIDADE 7,5 X 10CM NAO...	5.68	2010-01-01	3124.76	0.88	0.66	441.67	0.01
4	4512	PECA DE MADEIRA 3A/4A QUALIDADE 2,5 X 5CM NAO ...	1.16	2010-01-01	3124.76	0.88	0.66	441.67	0.01
5	4505	PECA DE MADEIRA 3A/4A QUALIDADE 1 X 7CM NAO AP...	1.45	2010-01-01	3124.76	0.88	0.66	441.67	0.01
6	4509	PECA DE MADEIRA 3A QUALIDADE 2,5 X 10CM NAO AP...	1.88	2010-01-01	3124.76	0.88	0.66	441.67	0.01
7	4497	PECA DE MADEIRA 3A QUALIDADE 10 X 10CM NAO APA...	5.09	2010-01-01	3124.76	0.88	0.66	441.67	0.01
8	4704	PEDRA ARDOSIA CINZA 20 X 40CM E = 1CM	9.31	2010-01-01	3124.76	0.88	0.66	441.67	0.01
9	4510	PECA DE MADEIRA 3A QUALIDADE 1,5 X 4CM NAO APA...	1.01	2010-01-01	3124.76	0.88	0.66	441.67	0.01

Figura 25. Correlação após seleção das variáveis teste 3, insumos x índices



### • 2.1.2 Experiências vivenciadas

Após a realização da Sprint, muitas ideias, descobertas e desafios foram encontrados.

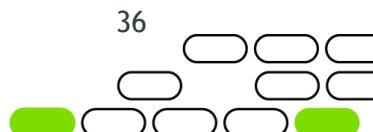
Abaixo estão as experiências vivenciadas e as lições aprendidas na Sprint 1:

- Devido à pouca experiência em manipulação de dados em outros formatos, aprendemos que extraír dados de PDF é mais sujeito a falhas e perda de

dados. Por esse motivo, a execução tornou-se mais demorada do que o previsto.

- Criar o conjunto de dados chamado MATERIAIS foi bem complicado, pois teve que ser feita boa parte manualmente. Após a importação, percebeu-se algumas falhas. Por esse motivo, será interessante avaliar na próxima etapa sua verdadeira utilidade.
- Foi perdido muito tempo para classificar os dados para o conjunto de dados chamado MATERIAIS. Portanto, para futuros projetos que necessitem classificar manualmente, é importante começar com uma abordagem mais macro e já analisar, para então prosseguir adicionando mais variáveis.
- Com o nível de dificuldade maior do que o esperado, o MVP pode ser realizado em um nível mais macro. A quantidade de insumos caiu moderadamente, devido à nem todos estarem presentes em toda a série histórica.
- Após realizar testes de combinações de dados, percebemos que dados que inicialmente não estavam nos planos de serem usados tornaram-se úteis.
- Organizar melhor a Sprint é necessário. Poderíamos adicionar mais detalhes e dividir alguns itens do backlog em duas partes, como por exemplo, ter criado um item para a organização dos arquivos do projeto.
- Analisar os dados mais profundamente em algumas questões será necessário. Por isso, seria interessante incluir um item no backlog da Sprint 2 nesse sentido para o desenvolvimento do projeto.

## 2.2 Sprint 2



## 2.2.1 Solução

- Evidência do planejamento:

*Figura 25. Planejamento da Sprint 2*

**Projeto Aplicado - 2023** Visível à Área de trabalho Quadro

**Backlog do Produto**

+ Adicionar um cartão

**Backlog da Sprint 1**

+ Adicionar um cartão

**Backlog Sprint 2**

Avaliar o desempenho do modelo e decidir se a previsão de índices é viável para o projeto.  
🕒 3

Previsão de índices econômicos e análise de viabilidade

Selecionar um modelo de previsão de séries temporais adequado para os índices econômicos.  
🕒 1

Realizar análises exploratórias dos dados para identificar tendências e padrões.[2]  
🕒 2

Obter e organizar os dados dos índices econômicos e insumos.[2]  
🕒 1

Preparar os dados para treinamento e teste do modelo.  
🕒 2

Treinar e testar o modelo usando os dados disponíveis.  
🕒 1

+ Adicionar um cartão

**Backlog Sprint 3**

Selecionar um modelo de previsão de séries temporais adequado para os preços dos insumos.  
🕒 1

Agrupar os insumos em categorias relevantes.

Preparar os dados para treinamento e teste do modelo, incluindo as informações dos índices econômicos.

Treinar e testar o modelo usando os dados disponíveis.

Avaliar o desempenho do modelo e ajustar os hiperparâmetros, se necessário.

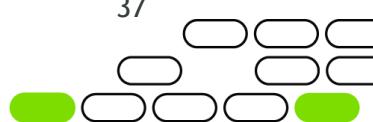
Documentar os resultados e preparar a entrega final do projeto.

+ Adicionar um cartão

**Executando**

+ Adicionar um cartão

- Evidência da execução de cada requisito:



Link para os notebooks:

- Notebook com algumas análises dos dados:

[https://colab.research.google.com/drive/1Y65\\_t2uGJ4UybCRmElsMJ9YFcD9na6nj?usp=share\\_link](https://colab.research.google.com/drive/1Y65_t2uGJ4UybCRmElsMJ9YFcD9na6nj?usp=share_link)

- Notebook para a previsão dos índices econômicos:

[https://colab.research.google.com/drive/1josBOTwEdpy9qFTX2YQG9HIFmVd1mjEJ?usp=share\\_link](https://colab.research.google.com/drive/1josBOTwEdpy9qFTX2YQG9HIFmVd1mjEJ?usp=share_link)

ITEM 1: Realizar análises exploratórias dos dados para identificar tendências e padrões. [2]

Figura 27. Código do gráfico para análise da variação dos insumos e índices

```
1 insumo_rate = pd.concat(insumo_ratio, axis =1)
2
3 insumo_rate.fillna(0, inplace=True)
4
5 px.line(insumo_rate)
```

Figura 28. Código da decomposição das series temporais.

```
[7] 1 decomposition = seasonal_decompose(df_inpc.values,
2                                model='additive',
3                                period=6)
4 plot_decomposition(decomposition)
```

ITEM 2: Obter e organizar os dados dos índices econômicos e insumos. [2]

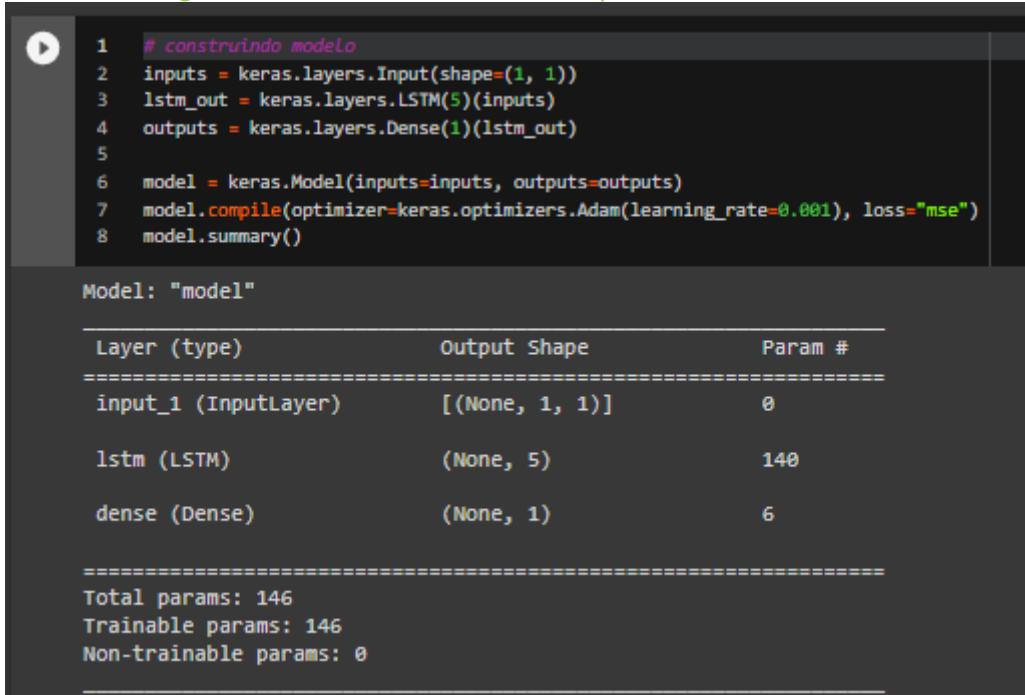
Figura 29. Código para capturar dados da API do IBGE.

```
[ ] 1 url = 'https://servicodados.ibge.gov.br/api/v3/agregados/7060/periodos/202001|202002|202003|202004|202005|202006|202007|202008|202009|202010|202011|202012'
2
3
4 response = requests.get(url)
5 data = response.json()
6
7 # Acessando a série dentro dos resultados
8 results = []
9
10 for i in range(11):
11     serie = data[0]['resultados'][i]['series'][0]['serie']
12     nome = data[0]['resultados'][i]['classificacoes'][0]['categoria']
13     # Selecionando as chaves do dicionário correspondentes aos meses a partir de janeiro de 2021
14     serie = {k:v for k,v in serie.items() if int(k[:4]) >= 2009 and int(k[4:]) >= 1}
15     result = pd.DataFrame.from_dict(serie, orient='index', columns=[list(nome.values())[0]])
16     results.append(result)
17
18
```

ITEM 3: Selecionar um modelo de previsão de séries temporais adequado para os índices econômicos.

Para a previsão de índices econômicos foram escolhidos dois modelos básico, Auto Arima e LSTM.

*Figura 30. Modelo LSTM e o total de parâmetros do modelo.*



```
1 # construindo modelo
2 inputs = keras.layers.Input(shape=(1, 1))
3 lstm_out = keras.layers.LSTM(5)(inputs)
4 outputs = keras.layers.Dense(1)(lstm_out)
5
6 model = keras.Model(inputs=inputs, outputs=outputs)
7 model.compile(optimizer=keras.optimizers.Adam(learning_rate=0.001), loss="mse")
8 model.summary()
```

Model: "model"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 1, 1)]	0
lstm (LSTM)	(None, 5)	140
dense (Dense)	(None, 1)	6

Total params: 146  
Trainable params: 146  
Non-trainable params: 0



*Figura 31. Modelo Auto Arima*

```
1 import pmdarima as pm
2
3 # treinando o modelo
4 model = pm.auto_arima(train,
5                         seasonal=True,
6                         m=6,
7                         trace=True,
8                         error_action='ignore',
9                         suppress_warnings=True)
10
11 # relatório
12 print(model.summary())
```

ITEM 4: Preparar os dados para treinamento e teste do modelo.

*Figura 31. Transformações logarítmica e suavização exponencial*

#### ▼ Transformação logaritmica

```
[ ] 1 # Aplicando a transformação Logaritmica
2 inpc_log = np.log(df_inpc)
3
4 # Suavização exponencial tripla (Holt-Winters)
5 model = ExponentialSmoothing(inpc_log, trend='add', seasonal='add', seasonal_periods=6)
6 fit = model.fit()
7 data_log_smoothed = fit.fittedvalues
8
9 # Decomposição da série temporal transformada
10 result = seasonal_decompose(data_log_smoothed, model='additive', period=6)
11 plot_decomposition(result)
12
```



Figura 32. Pré-processamento dos dados para o modelo LSTM

▼ PREVISÃO COM A REDE LSTM (DADOS TRANSFORMADOS)

```
[ ] 1 # utilizando o keras: https://keras.io/
2 import keras
3 from sklearn.preprocessing import MinMaxScaler

[ ] 1 # extraindo a série (np.array)
2 series = data_log_smoothed.values.reshape(-1, 1)

[ ] 1 # divisão em treino e teste
2 n, p = len(data_log_smoothed), 0.9
3 train = series[:int(n*p)]
4 test = series[int(n*p):]

[ ] 1 # normalização (max 1 e min 0)
2 scaler_train = MinMaxScaler()
3 train = scaler_train.fit_transform(train)

[ ] 1 # normalização (max 1 e min 0)
2 scaler_test = MinMaxScaler()
3 test = scaler_test.fit_transform(test)

[ ] 1 # convertendo a série para uma estrutura x_train, y_train
2 x_train, y_train = train[:-1], train[1:]
3 x_test, y_test = test[:-1], test[1:]

[ ] 1 # x_train, y_train: primeiras 5 Linhas
2 print(np.c_[x_train, y_train][:5, :])

[[0.          0.01035866]
 [0.01035866  0.02197114]
 [0.02197114  0.03311952]
 [0.03311952  0.04465277]
 [0.04465277  0.0468567 ]]
```

ITEM 5. Treinar e testar o modelo usando os dados disponíveis.

Figura 32. Treino para teste do modelo LSTM

▼ TREINAMENTO

```
▶ 1 # fit no modelo
2 model.fit(x=x_train, y=y_train, batch_size=1, epochs=50)

Epoch 1/50
139/139 [=====] - 2s 3ms/step - loss: 0.2416
Epoch 2/50
139/139 [=====] - 0s 3ms/step - loss: 0.0848
Epoch 3/50
139/139 [=====] - 0s 3ms/step - loss: 0.0445
Epoch 4/50
139/139 [=====] - 0s 3ms/step - loss: 0.0362
Epoch 5/50
139/139 [=====] - 0s 3ms/step - loss: 0.0311
Epoch 6/50
139/139 [=====] - 0s 3ms/step - loss: 0.0268
Epoch 7/50
139/139 [=====] - 0s 3ms/step - loss: 0.0228
Epoch 8/50
139/139 [=====] - 0s 3ms/step - loss: 0.0187
Epoch 9/50
139/139 [=====] - 1s 5ms/step - loss: 0.0154
Epoch 10/50
```



ITEM 6: Avaliar o desempenho do modelo e decidir se a previsão de índices é viável para o projeto.

*Figura 33. Códigos para análise do modelo*

```
[ ] 1 # plot da predição
2 n_train, n_test = len(y_train), len(y_test)
3 plt.figure(figsize=(10, 5))
4 plt.plot(np.arange(n_train), y_train_orig,'.-', label='Treino')
5 plt.plot(np.arange(n_train, n_train+n_test), y_test_orig,'.-', label='Teste')
6 plt.plot(np.arange(n_train, n_train+n_test), forecast_orig,'.-', label='Predição')
7 plt.grid()
8 plt.legend()
9 plt.show()

[ ] 1 forecast_metrics(y_test_orig, forecast_orig)
```

ITEM 7: Selecionar um modelo de previsão de séries temporais adequado para os preços dos insumos.

*Figura 34. Pesquisa sobre modelos deep learning para previsão de times series.*

Isaac Godfried  
Jan 9 · 16 min read · Member-only · Listen

## Advances in Deep Learning for Time Series Forecasting and Classification: Winter 2023 Edition

The downfall of transformers for time series forecasting and the rise of time series embedding methods. Plus advances in anomaly detection, classification, and optimal (t) interventions.



Photo from myself (Great Sand Dunes NP at sunset)

It has been quite sometime since I've written an update on the state of deep learning for time series. Several conferences have come and gone and the field as a whole has advanced in several different ways. Here I will attempt to cover some of the more promising as well as critical papers that have come out in the last year or so as well as updates to the [Flow Forecast](#) framework [FF].

Fonte: <https://towardsdatascience.com/deepar-mastering-time-series-forecasting-with-deep-learning-bc717771ce85>



### Links das pesquisas

<https://pytorch-forecasting.readthedocs.io/en/stable/models.html>

<https://towardsdatascience.com/temporal-fusion-transformer-time-series-forecasting-with-deep-learning-complete-tutorial-d32c1e51cd91>

- Evidência dos resultados:

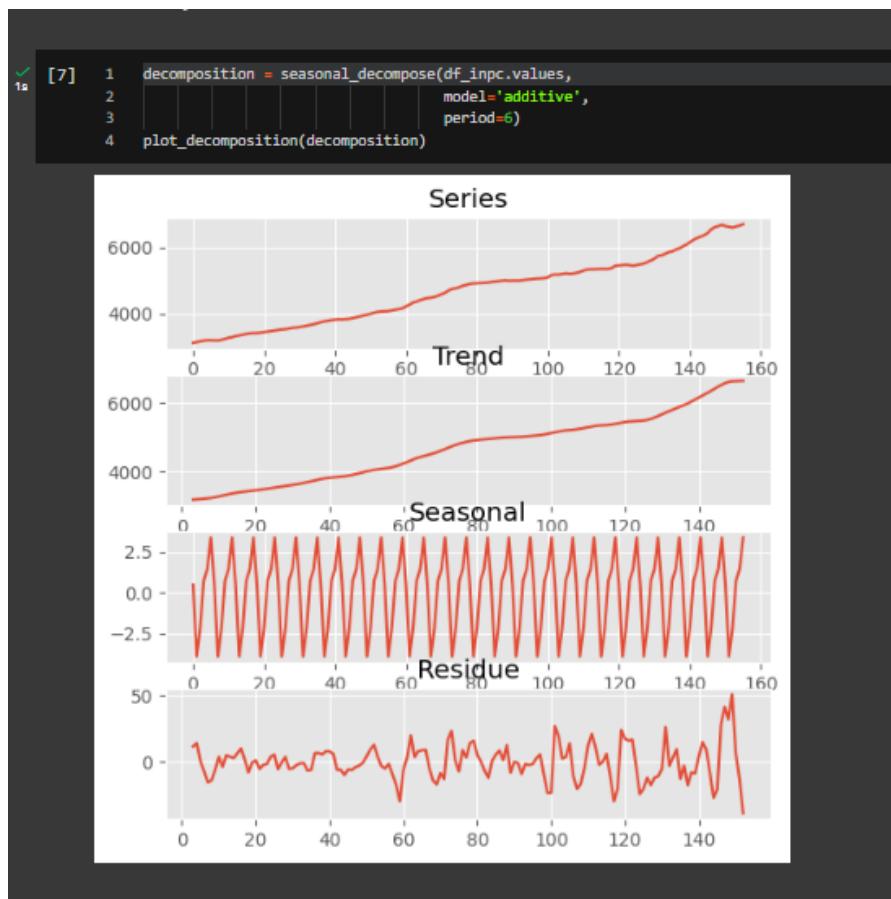
*Figura 35. Resultado da coleta dos IBGE*

	2103005.Ferragens	2103008.Material de eletricidade	2103012.Vidro	2103014.Tinta	2103032.Revestimento de piso e parede	2103039.Cimento	2103040.Tijolo
202001	-0.33		-0.52	3.39	0.44		1.13
202002	-0.24		1.69	0.13	0.32		0.01
202003	0.42		0.71	-0.07	0.47		-1.42
202004	0.59		-1.30	-0.16	-0.03		-0.14
202005	-0.63		-2.69	-6.97	1.44		1.29

*Figura 36. Analise da Variação dos dados dos insumos e do IBGE*



Figura 37. Analise da Séries Temporais



Após a conclusão de algumas etapas podemos então treinar o modelo, e tivemos alguns bons resultados para a previsão dos índices econômicos e com isso podemos selecionar qual modelo utilizar para prever essas variáveis.

Figura 38. Treinamento do modelo LSTM

```
[1] #_fit no modelo
2 model.fit(x=x_train, y=y_train, batch_size=1, epochs=50)
39/139 [=====] - 1s 5ms/step - loss: 1.0495e-04
Epoch 23/50
39/139 [=====] - 1s 5ms/step - loss: 1.0518e-04
Epoch 24/50
39/139 [=====] - 1s 4ms/step - loss: 1.0232e-04
Epoch 25/50
39/139 [=====] - 1s 5ms/step - loss: 1.0665e-04
Epoch 26/50
39/139 [=====] - 1s 5ms/step - loss: 1.0690e-04
Epoch 27/50
39/139 [=====] - 1s 5ms/step - loss: 1.0174e-04
Epoch 28/50
39/139 [=====] - 1s 5ms/step - loss: 1.0720e-04
Epoch 29/50
39/139 [=====] - 1s 5ms/step - loss: 1.0119e-04
Epoch 30/50
39/139 [=====] - 0s 3ms/step - loss: 1.0533e-04
Epoch 31/50
39/139 [=====] - 0s 3ms/step - loss: 1.0196e-04
Epoch 32/50
39/139 [=====] - 0s 3ms/step - loss: 1.0400e-04
Epoch 33/50
39/139 [=====] - 0s 3ms/step - loss: 1.0575e-04
Epoch 34/50
39/139 [=====] - 0s 3ms/step - loss: 1.0479e-04
Epoch 35/50
39/139 [=====] - 0s 3ms/step - loss: 1.0439e-04
Epoch 36/50
39/139 [=====] - 0s 3ms/step - loss: 9.3296e-05
Epoch 37/50
39/139 [=====] - 1s 4ms/step - loss: 1.0704e-04
Epoch 38/50
39/139 [=====] - 1s 4ms/step - loss: 1.1676e-04
Epoch 39/50
39/139 [=====] - 1s 4ms/step - loss: 1.0412e-04
Epoch 40/50
39/139 [=====] - 1s 4ms/step - loss: 1.1181e-04
Epoch 41/50
39/139 [=====] - 1s 4ms/step - loss: 1.0124e-04
Epoch 42/50
39/139 [=====] - 1s 4ms/step - loss: 1.0797e-04
Epoch 43/50
39/139 [=====] - 0s 3ms/step - loss: 1.0611e-04
Epoch 44/50
39/139 [=====] - 0s 3ms/step - loss: 1.0588e-04
Epoch 45/50
39/139 [=====] - 0s 3ms/step - loss: 1.2608e-04
Epoch 46/50
```

Figura 39. Resultado do teste do modelo LSTM para previsão do índice CUB



Figura 40. Resultado do teste do modelo LSTM para previsão da SELIC

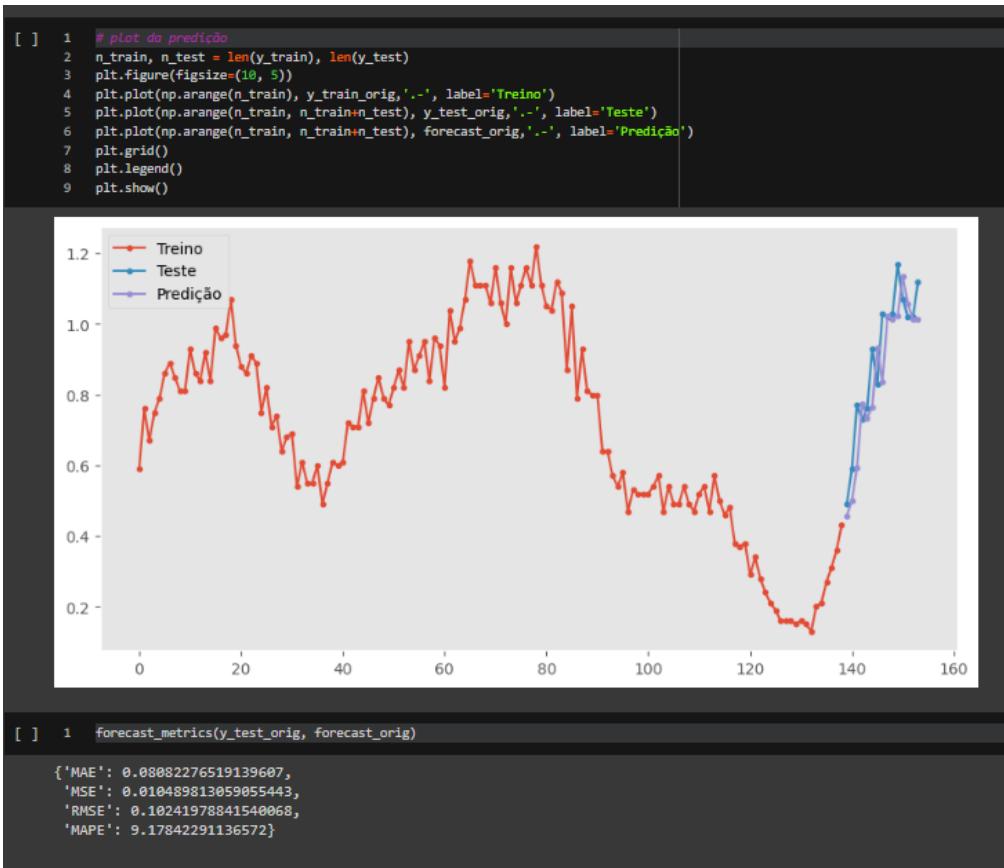
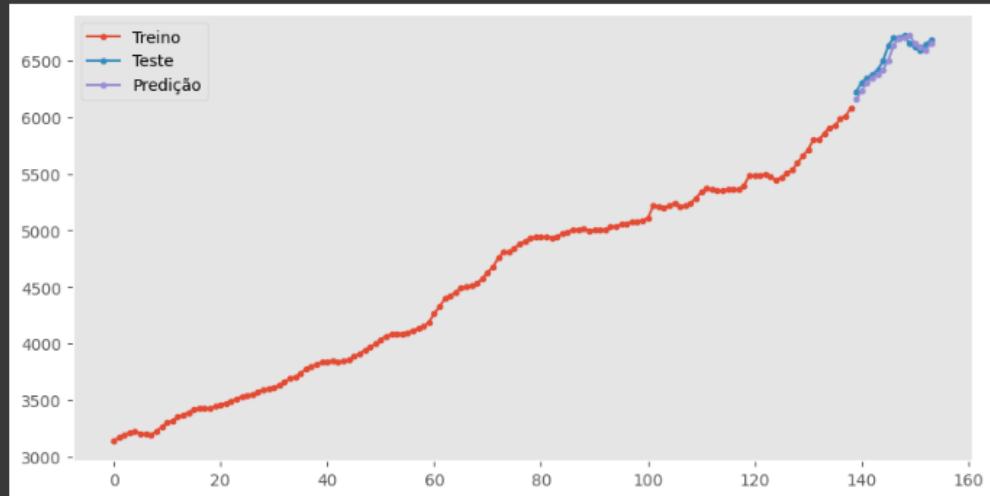


Figura 41. Resultado do teste do modelo LSTM para previsão da SELIC

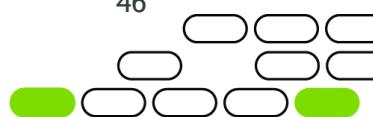
▼ RESULTADO

```
[ ] 1 # plot da predição
2 n_train, n_test = len(y_train), len(y_test)
3 plt.figure(figsize=(10, 5))
4 plt.plot(np.arange(n_train), y_train_orig,'.-', label='Treino')
5 plt.plot(np.arange(n_train, n_train+n_test), y_test_orig,'.-', label='Teste')
6 plt.plot(np.arange(n_train, n_train+n_test), forecast_orig,'.-', label='Predição')
7 plt.grid()
8 plt.legend()
9 plt.show()
```



```
[ ] 1 forecast_metrics(y_test_orig,forecast_orig)
```

```
{'MAE': 52.03065485542454,
'MSE': 3574.817128995051,
'RMSE': 59.78977445178273,
'MAPE': 0.7976754945648481}
```



Como para se definir qual é a melhorar arquitetura, cheguei ao resultado de que algumas arquiteturas do Pytorch Forecasting podem atender as necessidades do projeto. Na figura 42 mostra um resumo do escopo de cada arquitetura.

*Figura 42. Arquiteturas da biblioteca Pytorch Forecasting.*

### Selecting an architecture

Criteria for selecting an architecture depend heavily on the use-case. There are multiple selection criteria and you should take into account. Here is an overview over the pros and cons of the implemented models:

Name	Multiple						Interac between series
	Covariates	targets	Regression	Classification	Probabilistic	Uncertainty	
RecurrentNetwork	x	x	x				
DecoderMLP	x	x	x	x			x
NBeats				x			
NHiTS	x	x	x				
DeepAR	x	x	x		x	x	x [1]
TemporalFusionTransformer	x	x	x	x			x

[1] Accounting for correlations using a multivariate loss function which converts the network into a DeepVAR model.

*Figura 42. Sprint 2*

Backlog Sprint 1	Backlog Sprint 2	Backlog Sprint 3	Sprint 2 Finalizadas	Sprint 3 Finalizadas
+ Adicionar um cartão	+ Adicionar um cartão	+ Adicionar um cartão	Realizar análises exploratórias dos dados para identificar tendências e padrões.[2]	+ Adicionar um cartão
			Avaliar e organizar os dados dos índices econômicos e insumos. [2]	
			Selecionar um modelo de previsão de séries temporais adequado para os índices econômicos.	
			Preparar os dados para treinamento e teste do modelo.	
			Treinar e testar o modelo	

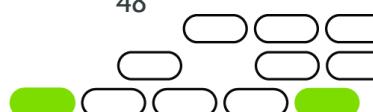
## 2.2.2 Experiências vivenciadas

Conforme mencionado na Sprint 1, foi imprescindível realizar análises adicionais nos dados antes do treinamento dos modelos. O processo de treinar e testar os modelos com o objetivo de prever índices econômicos mostrou-se mais simples do que a coleta, extração e processamento dos dados em si.

É importante salientar que, dependendo do modelo de previsão escolhido, pode ser necessário um trabalho adicional de modelagem e pré-processamento dos dados. Isso ocorre porque, embora existam diversos modelos de previsão de séries temporais, a formatação dos dados pode variar significativamente entre eles.

A atividade "Previsão de índices econômicos e análise de viabilidade" foi postergada para a próxima Sprint, sendo substituída pelo item "Selecionar um modelo de previsão de séries temporais adequado para os preços dos insumos". A decisão foi tomada uma vez que os modelos de previsão de índices apresentaram resultados satisfatórios e a previsão para os meses subsequentes será realizada na etapa final do projeto. Além disso, a alteração objetiva priorizar o progresso do projeto, focando na abordagem dos próximos itens e na estruturação do projeto de modo a convergir para a solução final proposta.

Ao longo das experiências em cada Sprint, identificou-se a necessidade de aprimorar o detalhamento dos backlogs, uma vez que estes não representam de maneira precisa todo o processo executado no projeto. Alguns itens deveriam ter sido incluídos de forma recorrente em todas as Sprints, enquanto outros poderiam ser desmembrados em partes menores, permitindo uma compreensão mais clara e gerenciamento eficiente das tarefas. Essa percepção destaca a importância de refinar continuamente a organização dos backlogs, visando otimizar o planejamento e a execução das atividades ao longo das Sprints futuras no âmbito do projeto.

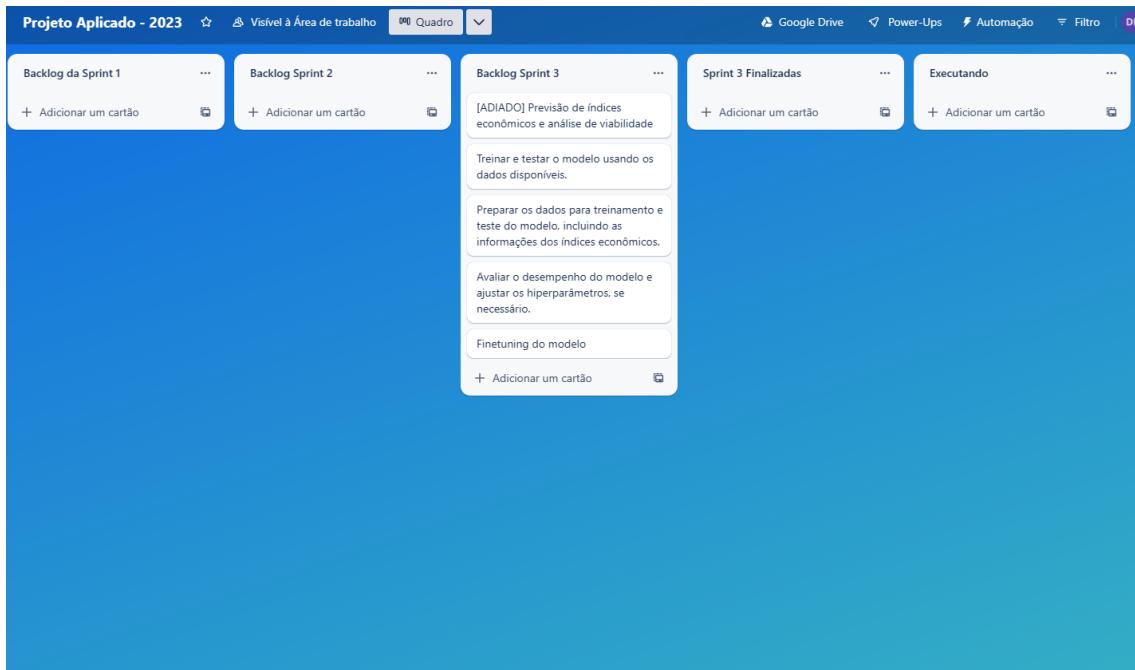


## 2.3 Sprint 3

### 2.3.1 Solução

- Evidência do planejamento:

*Figura 43. Sprint 3*



- Evidência da execução de cada requisito:

ITEM 1: Previsão de índices econômicos e análise de viabilidade.

*Figura 44. Código para previsão de 12 meses do índice.*

```
# prever 12 períodos futuros além dos dados de teste
n_future_preds = 12
forecast_future = model.predict(n_periods=len(test) + n_future_preds)

# extrair apenas os 12 últimos valores, que são os períodos futuros
future_preds = forecast_future[-n_future_preds:]
```



ITEM 2: Preparar os dados para treinamento e teste do modelo, incluindo as informações dos índices econômicos.

*Figura 45. Código do pré-processamento dos dados para o formato aceito pelo modelo.*

```

1  indices = pd.read_csv('/content/drive/MyDrive/PROJETO APLICADO/0105_dados/indices_merge.csv', index_col=0)
2  indices.data = pd.to_datetime(indices['data'])
3  #
4  insumos = pd.read_csv('/content/drive/MyDrive/PROJETO APLICADO/0105_dados/insumos_merge.csv', index_col=0)
5  insumos.data = pd.to_datetime(insumos['data'])

[ ] 1  # Copia dos dados importados
2  insumos1 = insumos.copy()
3  indices1 = indices.copy()

[ ] 1  # Transforma os dados em objeto
2  insumos1['codigo'] = insumos1['codigo'].astype(str)
3  insumos1['cod_rep'] = insumos1['cod_rep'].astype(str)

Mesclando os dados para o treinamento do modelo

[ ] 1  # Merclar os dados
2  df = pd.merge(indices1, insumos1, on='data', how='outer')

[ ] 1  # Retirar excedente dos dados
2  df_cortados = df.loc[df['data'] >= '2023-02-01']
3  df1 = df.loc[df['data'] < '2023-02-01']
4  #
5  #copiando os dados
6  df2 = df1.copy()

Selecionando os dados para o treinamento

[ ] 1  df3 = df2[['data', 'time_idx', 'codigo', 'cod_rep','commodities','valor','valor_var_mes', 'valor_var_cumsum', 'valor_cumsum', 'inpc_mes']

[ ] 1  # Preenchendo os NAN
2  df3['valor_cumsum_fill'] = df3['valor_cumsum'].fillna(0)
3  # criando uma feature nova para dizer se os dados são reais ou não, por causa do fillna
4  df3['Valor_cumsum_is_real'] = df3['valor_cumsum'].notna().astype(int)
5  # Reordenando os dados
6  df3 = df3.sort_values(by=['codigo', 'data'])

[ ] 1  df4 = df3.copy()
2  #
3  # Função para selecionar aleatoriamente uma amostra de cada grupo 'cod_rep'
4  def sample_from_group(group):
5      return group.sample(n=3)
6  #
7  # Agrupar os dados por 'cod_rep' e selecionar uma amostra de cada grupo
8  sampled_data = df4.groupby('cod_rep').apply(sample_from_group).reset_index(drop=True)
9

```



ITEM 3: Treinar e testar o modelo usando os dados disponíveis.

Figura 46. Fornecendo as informações para o treinamento do modelo.

```
1 train_data_feat['time_idx'] = train_data_feat['time_idx'].astype(np.int64)
2 train_data_feat['valor_var_cumsum'] = train_data_feat['valor_var_cumsum'].astype(np.float32)
3
4 training = TimeSeriesDataSet(
5     train_data_feat[lambda x: x.time_idx <= training_cutoff],
6     time_idx="time_idx",
7     target="valor_cumsum_fill",
8     group_ids=["codigo"],
9     min_encoder_length=max_encoder_length // 2,
10    max_encoder_length=max_encoder_length,
11    min_prediction_length=1,
12    max_prediction_length=max_prediction_length,
13    static_categoricals=["codigo",'cod_rep'],
14    time_varying_known_reals= ["time_idx", "mes", "ano_numerico", "trimestre", 'inpc_mes_value'],
15    time_varying_unknown_reals=[
16        "valor_cumsum_fill",
17        "valor_var_cumsum",
18        'valor_media_movel_3m',
19        'valor_tendencia',
20        'valor_var_mes',
21        'ind_transf_value',
22        'valor_cumsum_is_real',
23        'selic_media_anual',
24
25    ],
26    #target_normalizer=GroupNormalizer(
27    #    groups=["codigo"], transformation="softplus"
28    # ), # we normalize by group
29    add_relative_time_idx=True,
30    add_target_scales=True,
31    add_encoder_length=True,
32    allow_missing_timesteps=True,
33
34 )
35
36
37
38 validation = TimeSeriesDataSet.from_dataset(training, train_data_feat, predict=True, stop_randomization=True)
39
40 # create dataloaders for our model
41 batch_size = 64
42 # if you have a strong GPU, feel free to increase the number of workers
43 train_dataloader = training.to_dataloader(train=True, batch_size=batch_size, num_workers=30)
44 val_dataloader = validation.to_dataloader(train=False, batch_size=batch_size * 10, num_workers=30)
```



Figura 47. Código para a criação do modelo e treinamento..

```

pl.seed_everything(42)

early_stop_callback = EarlyStopping(monitor="val_loss", min_delta=1e-4, patience=5, verbose=True, mode="min")
lr_logger = LearningRateMonitor()
logger = TensorBoardLogger("lightning_logs")

trainer = pl.Trainer(
    max_epochs=45,
    accelerator='gpu',
    devices=1,
    enable_model_summary=True,
    gradient_clip_val=0.022,
    callbacks=[lr_logger, early_stop_callback],
    logger=logger)

tft = TemporalFusionTransformer.from_dataset(
    training,
    learning_rate= 0.047,
    hidden_size=85,
    attention_head_size=2,
    dropout=0.206,
    hidden_continuous_size=29,
    output_size=7, # there are 7 quantiles by default: [0.02, 0.1, 0.25, 0.5, 0.75, 0.9, 0.98]
    loss=QuantileLoss(),
    log_interval=10,
    reduce_on_plateau_patience=10)

trainer.fit(
    tft,
    train_dataloaders=train_dataloader,
    val_dataloaders=val_dataloader)

```

ITEM 4: Avaliar o desempenho do modelo e ajustar os hiperparâmetros, se necessário.

Figura 48. Código para a análise do modelo.

```

import torch

actuals = torch.cat([y[0] for x, y in iter(val_dataloader)]).to('cuda')
predictions = best_tft.predict(val_dataloader).to('cuda')

#average p50 loss overall
print('MAE:', (actuals - predictions).abs().mean().item())

#average p50 loss per time series
mape = ((actuals - predictions).abs() / actuals).mean()
print('MAPE:', (mape * 100).item(), '%')

smape = (2.0 * (actuals - predictions).abs() / (actuals.abs() + predictions.abs())).mean()
print('SMAPE:', (smape * 100).item(), '%')

```



Figura 49. Código para avaliar o modelo e visualizar as previsões

```

1 raw_predictions = best_tft.predict(val_dataloader, mode="raw", return_x=True)
2 x = raw_predictions[1]
3 print(raw_predictions._fields) # ('output', 'x', 'index', 'decoder_lengths', 'y')
4 print('\n')
5 print(len(raw_predictions.output))
6

INFO:lightning.pytorch.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES: [0]
('output', 'x', 'index', 'decoder_lengths', 'y')

8

1 for idx in range(50): # plot 10 examples
2     fig, ax = plt.subplots(figsize=(10, 4))
3     best_tft.plot_prediction(raw_predictions.x, raw_predictions.output, idx=idx, add_loss_to_title=True, ax=ax)

interpretation = best_tft.interpret_output(raw_predictions.output, reduction="sum")
best_tft.plot_interpretation(interpretation)

predictions = best_tft.predict(val_dataloader, return_x=True)
predictions_vs_actuals = best_tft.calculate_prediction_actual_by_variable(predictions.x, predictions.output)
best_tft.plot_prediction_actual_by_variable(predictions_vs_actuals)

```

## ITEM 5: Finetuning do modelo.

Figura 50. Código para a otimização dos hyperparametros.

```

1 import pickle
2
3 from pytorch_forecasting.models.temporal_fusion_transformer.tuning import optimize_hyperparameters
4
5
6 # create study
7 study = optimize_hyperparameters(
8     train_dataloader,
9     val_dataloader,
10    model_path="optuna_test",
11    n_trials=40,
12    max_epochs=15,
13    gradient_clip_val_range=(0.01, 1.0),
14    hidden_size_range=(8, 128),
15    hidden_continuous_size_range=(8, 128),
16    attention_head_size_range=(1, 4),
17    learning_rate_range=(0.001, 0.1),
18    dropout_range=(0.1, 0.3),
19    trainer_kwarg=dict(limit_train_batches=30),
20    reduce_on_plateau_patience=4,
21    use_learning_rate_finder=False, # use Optuna to find ideal learning rate or use in-built learning rate finder
22 )
23
24 # save study results - also we can resume tuning at a later point in time
25 with open("test_study.pkl", "wb") as fout:
26     pickle.dump(study, fout)
27
28 # show best hyperparameters
29 print(study.best_trial.params)

```



- Evidência dos resultados:

*Figura 51. Previsão do índice econômico.*

```
In [19]: np.exp(future_preds)
```

```
Out[19]: 2023-04-01    655.753047
          2023-05-01    658.941539
          2023-06-01    663.298955
          2023-07-01    668.466345
          2023-08-01    671.177975
          2023-09-01    673.853193
          2023-10-01    676.817070
          2023-11-01    680.080696
          2023-12-01    684.541089
          2024-01-01    689.855118
          2024-02-01    692.647822
          2024-03-01    695.417768
Freq: MS, dtype: float64
```

*Figura 52. Treinamento do modelo.*

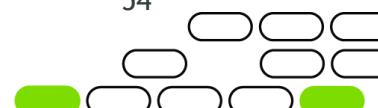
+ Código	+ Texto	
1   logging_metrics	ModuleList	0
2   input_embeddings	MultiEmbedding	21.6 K
3   prescalers	ModuleDict	986
4   static_variable_selection	VariableSelectionNetwork	23.4 K
5   encoder_variable_selection	VariableSelectionNetwork	110 K
6   decoder_variable_selection	VariableSelectionNetwork	45.7 K
7   static_context_variable_selection	GatedResidualNetwork	29.4 K
8   static_context_initial_hidden_lstm	GatedResidualNetwork	29.4 K
9   static_context_initial_cell_lstm	GatedResidualNetwork	29.4 K
10   static_context_enrichment	GatedResidualNetwork	29.4 K
11   lstm_encoder	LSTM	58.5 K
12   lstm_decoder	LSTM	58.5 K
13   post_lstm_gate_encoder	GatedLinearUnit	14.6 K
14   post_lstm_add_norm_encoder	AddNorm	170
15   static_enrichment	GatedResidualNetwork	36.6 K
16   multihead_attn	InterpretableMultiHeadAttention	21.6 K
17   post_attn_gate_norm	GateAddNorm	14.8 K
18   pos_wise_ff	GatedResidualNetwork	29.4 K
19   pre_output_gate_norm	GateAddNorm	14.8 K
20   output_layer	Linear	602
-----		
567 K	Trainable params	
0	Non-trainable params	
567 K	Total params	
2.271	Total estimated model params size (MB)	

Epoch 6: 100%

```
INFO:lightning.pytorch.callbacks.early_stopping:Metric val_loss improved. New best score: 5.886
INFO:lightning.pytorch.callbacks.early_stopping:Metric val_loss improved by 1.606 >= min_delta = 0.0001. New best score: 4.279
INFO:lightning.pytorch.callbacks.early_stopping:Monitored metric val_loss did not improve in the last 5 records. Best score: 4.279. Signaling Trainer to stop.
```

*Figura 53. Base do modelo.*

```
INFO: LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES: [0]
INFO:lightning.pytorch.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES: [0]
tensor(6.9738, device='cuda:0')
```



### *Figura 54. Resultado do Finetuning*

```
[1 2023-05-02 08:49:56.228] Trial 36 finished with value: 6.460273651885986 and parameters: {'gradient_clip_val': 0.01025493845654496, 'hidden_size': 100, 'dropout': 0.2219506668195716, 'hidden_...  
INFO:pytorch_lightning.utilities.rank_zero:GPU available: True (cuda), used: True  
INFO:pytorch_lightning.utilities.rank_zero:TPU available: False, using: 0 TPU cores  
INFO:pytorch_lightning.utilities.rank_zero:IPU available: False, using: 0 IPUs  
INFO:pytorch_lightning.utilities.rank_zero:HPU available: False, using: 0 HPUs  
INFO:pytorch_lightning.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES: [0]  
INFO:pytorch_lightning.utilities.rank_zero:'Trainer.fit' stopped: 'max_epochs=15' reached.  
[1 2023-05-02 08:52:34.393] Trial 37 finished with value: 7.39112863201984 and parameters: {'gradient_clip_val': 0.040215707077119134, 'hidden_size': 125, 'dropout': 0.20878217574234175, 'hidden_...  
INFO:pytorch_lightning.utilities.rank_zero:GPU available: True (cuda), used: True  
INFO:pytorch_lightning.utilities.rank_zero:TPU available: False, using: 0 TPU cores  
INFO:pytorch_lightning.utilities.rank_zero:IPU available: False, using: 0 IPUs  
INFO:pytorch_lightning.utilities.rank_zero:HPU available: False, using: 0 HPUs  
INFO:pytorch_lightning.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES: [0]  
INFO:pytorch_lightning.utilities.rank_zero:'Trainer.fit' stopped: 'max_epochs=15' reached.  
[1 2023-05-02 08:55:08.784] Trial 38 finished with value: 7.251089096609336 and parameters: {'gradient_clip_val': 0.01400989181009384, 'hidden_size': 83, 'dropout': 0.1886940631781429, 'hidden_...  
INFO:pytorch_lightning.utilities.rank_zero:GPU available: True (cuda), used: True  
INFO:pytorch_lightning.utilities.rank_zero:TPU available: False, using: 0 TPU cores  
INFO:pytorch_lightning.utilities.rank_zero:IPU available: False, using: 0 IPUs  
INFO:pytorch_lightning.utilities.rank_zero:HPU available: False, using: 0 HPUs  
INFO:pytorch_lightning.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES: [0]  
INFO:pytorch_lightning.utilities.rank_zero:'Trainer.fit' stopped: 'max_epochs=15' reached.  
[1 2023-05-02 08:57:42.886] Trial 39 finished with value: 5.617138862609863 and parameters: {'gradient_clip_val': 0.019593210819943658, 'hidden_size': 72, 'dropout': 0.15545164530319736, 'hidden_...  
INFO:pytorch_lightning.utilities.rank_zero:GPU available: True (cuda), used: True  
INFO:pytorch_lightning.utilities.rank_zero:TPU available: False, using: 0 TPU cores  
INFO:pytorch_lightning.utilities.rank_zero:IPU available: False, using: 0 IPUs  
INFO:pytorch_lightning.utilities.rank_zero:HPU available: False, using: 0 HPUs  
INFO:pytorch_lightning.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES: [0]  
INFO:pytorch_lightning.utilities.rank_zero:'Trainer.fit' stopped: 'max_epochs=15' reached.
```

*Figura 53. Resultados do modelo após o Finetuning*

```
INFO:lightning.pytorch.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES: [0]
MAE: 7.299492359161377
MAPE: 3.6184682846069336 %
SMAPE: 3.557260036468506 %
```

*Figura 53. Gráficos para análise das previsões*

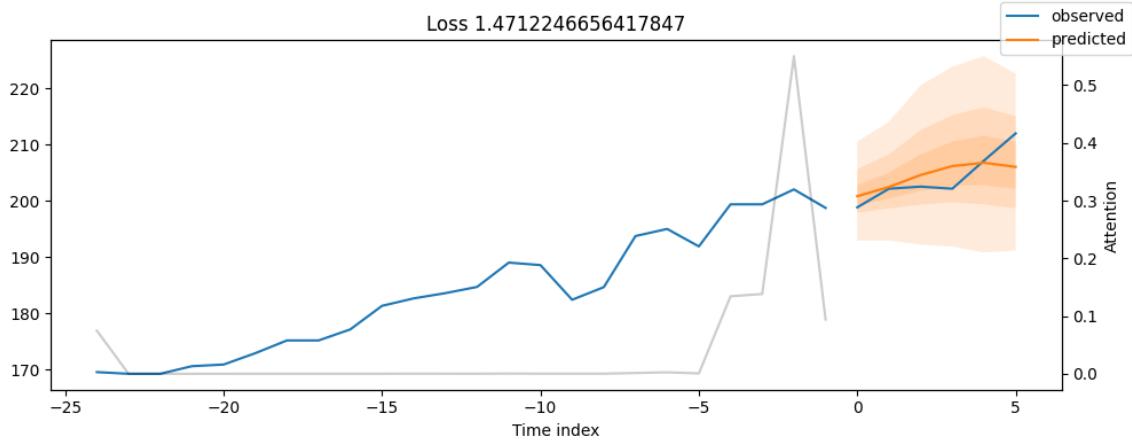
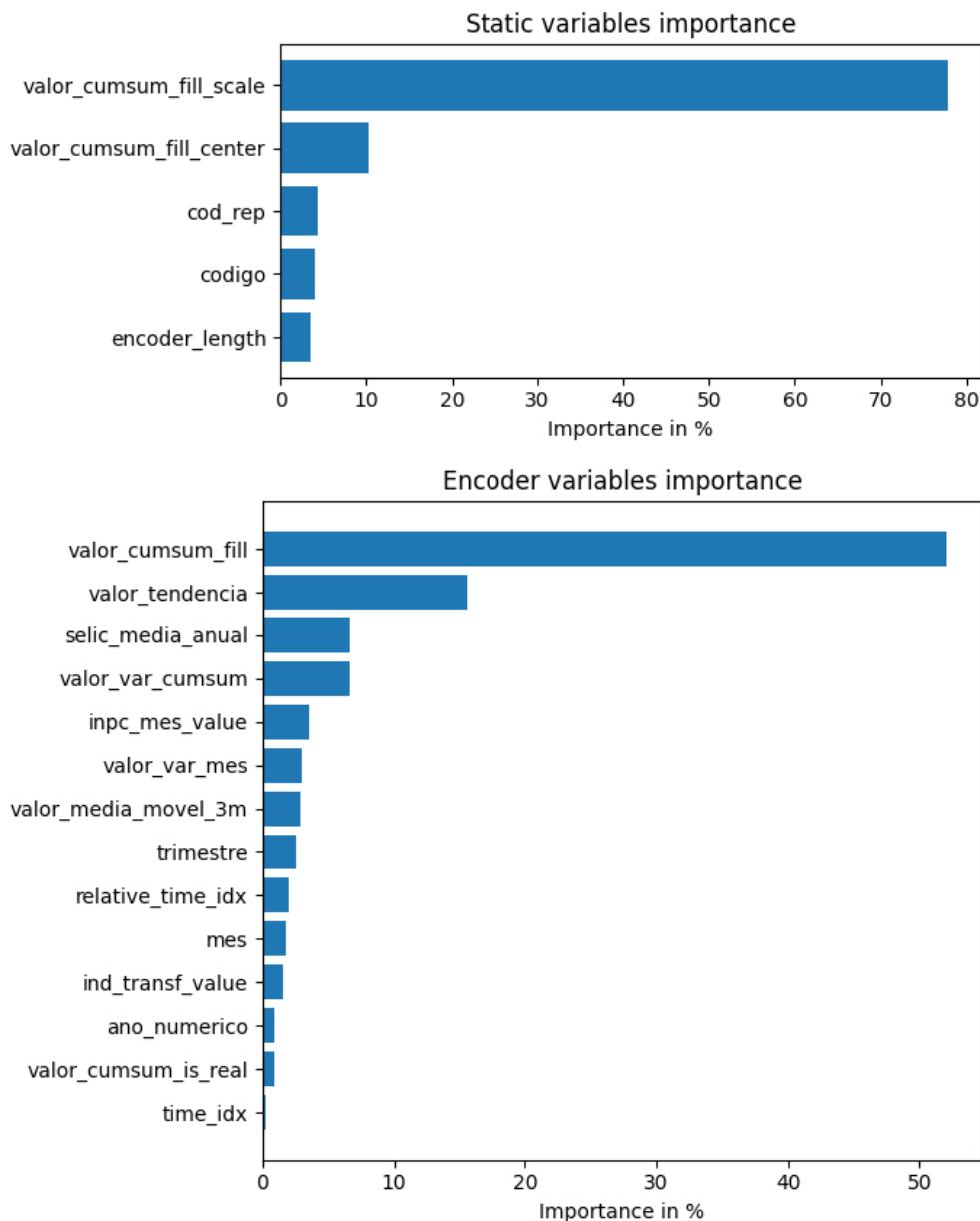
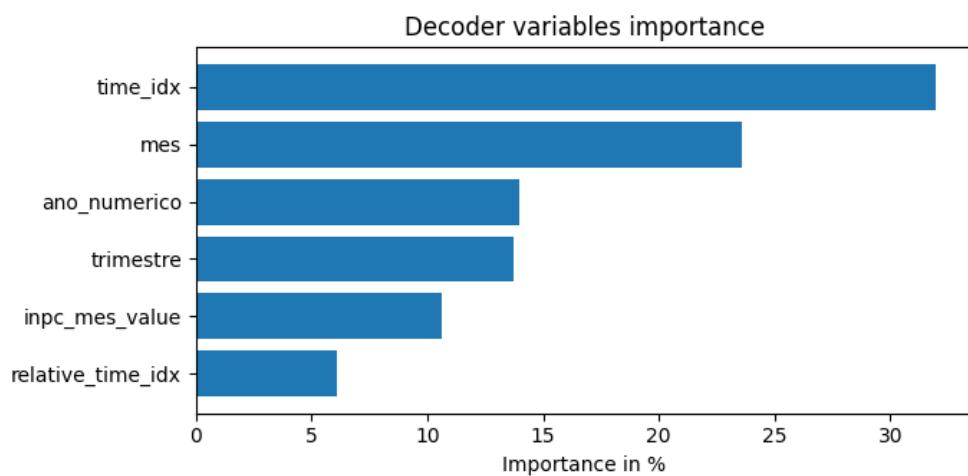


Figura 54. Gráfico para a visualização da importância das variáveis para o modelo.





*Figura 55. Evidencias dos experimentos dos modelos.*

 model\_30/04.v0.ipynb

model\_30/04.v1.ipynb

 model\_30/04.v4.ipynb

 model\_30/04.v5.ipynb

 model\_30/04.v6.ipynb 

 model\_30/04.v7.ipynb

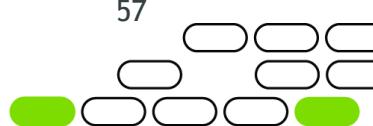
 model\_30/04.v8.ipynb

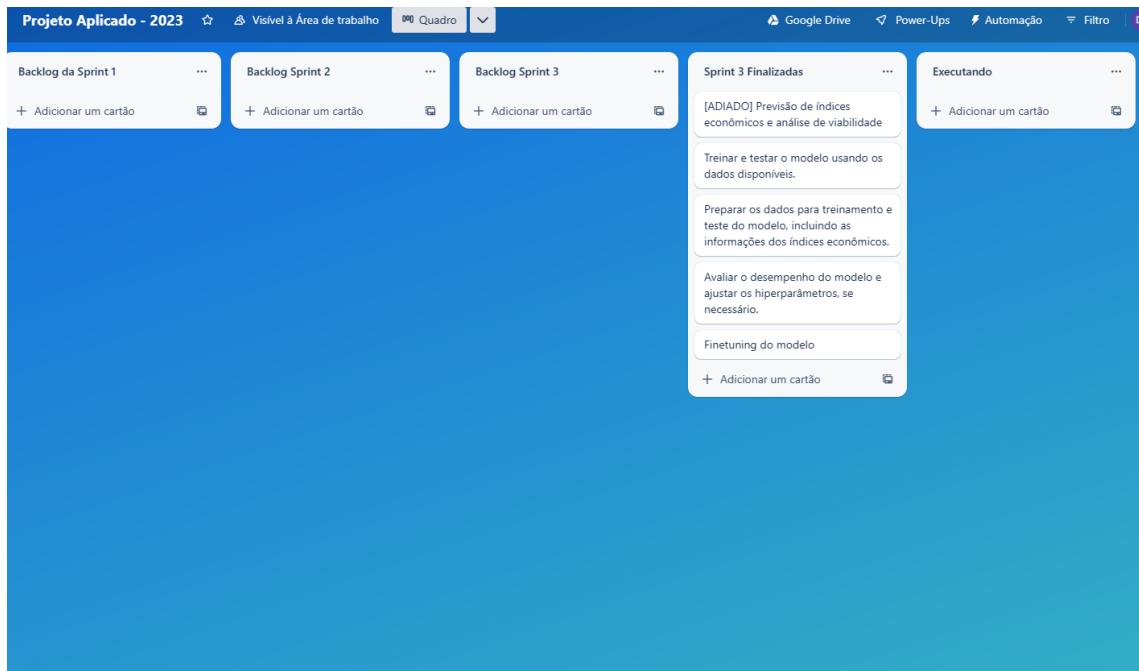
 model\_30/04.v9.ipynb

 model\_30/04.v10.ipynb

 model\_30/04.v12.ipynb

*Figura 55. Sprint 3.*





### 2.3.2 Experiências vivenciadas

Para realizar essa Sprint foram gastos muitas horas em testes de modelos, com diversas combinações de indicadores e insumos. E para isso foi necessário realizar muitos pré-processamento e adaptação dos dados.

Treinar o modelo escolhido foi bem demorado, além de que nos últimos dias da Sprint o ambiente virtual do Colaboratory do Google não estava funcionando da mesma forma que anteriormente, minando a possibilidade de realizar outros experimentos.

Para realizar o teste de modelos foi necessário dividir em vários notebooks e versionar, visto que fica muito confuso e trabalhoso treinar em um mesmo notebook.



### 3. Considerações Finais

#### 3.1 Resultados

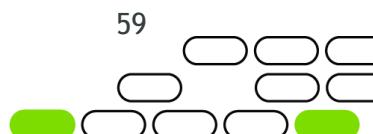
Neste projeto, trilhamos uma jornada cheia de obstáculos, impulsionados pela meta de aplicar previsões de índices ajudarem na previsão das tendências de preços dos insumos mais representativos da construção civil. O primeiro desafio enfrentado envolveu a coleta, extração e transformação de dados de tabelas em formatos PDF e XLS, uma tarefa que se mostrou mais complexa e demorada do que inicialmente prevíamos. Uma análise rigorosa dos dados coletados foi realizada, observando-os sob diversas perspectivas e combinações para definir a melhor rota a seguir.

Em seguida, nos concentramos na previsão de índices econômicos - especificamente, INPC, CUB e Selic. O melhor desempenho foi obtido com a rede neural LSTM. Entre os índices, o INPC apresentou resultados mais consistentes, tornando-se a variável conhecida que optamos por usar no nosso modelo.

Após pesquisa e análise de diversos modelos de deep learning que poderiam atender aos nossos objetivos, decidimos aplicar o Temporal Fusion Transformers (TFT). O TFT é um modelo de aprendizado profundo projetado especificamente para realizar previsões precisas em séries temporais complexas e multivariadas. Ele consegue fazer isso combinando os benefícios de um modelo de atenção global (que permite que ele aprenda quais entradas são importantes) com a eficiência de um modelo local (que permite que ele aprenda diferentes padrões em diferentes horizontes de tempo).

Essa escolha foi motivada por vários fatores. Primeiro, nossa série temporal era relativamente curta, o que dificulta o aprendizado de padrões individuais. Além disso, havia um grande número de insumos a serem analisados, tornando o treinamento individual impraticável. Queríamos um modelo que pudesse aprender e aproveitar as correlações entre os diferentes insumos. Além disso, o TFT permitiu a inclusão de variáveis externas, como previsões de índices econômicos, para melhorar a precisão das previsões dos preços dos insumos.

Com o TFT, alcançamos resultados satisfatórios na previsão das tendências de preços dos insumos representativos do SINAPI para um período de seis meses. Entretanto, é importante notar que nosso modelo teve dificuldades em captar os padrões de queda



dos insumos e em lidar com séries históricas mais curtas. Quanto à mensurabilidade do projeto, o erro percentual absoluto médio (MAPE) foi de 3,61% na previsão. Este é um resultado médio entre as previsões, com alguns insumos tendo um desempenho extremamente bom e outros nem tanto. Importante salientar que o foco do modelo está na estimativa de aumentos percentuais, não em preços absolutos.

Mesmo com a necessidade de melhorias, os resultados do nosso modelo demonstra que conseguimos oferecer uma resposta direta à problemática enfrentada pelos engenheiros de orçamentos, ao disponibilizar previsões que auxiliam no planejamento e controle de custos, principalmente em projetos que não se iniciam imediatamente.

### 3.2 Contribuições

A relevância do nosso projeto é evidenciada pela significativa contribuição que oferece para o desafio proposto, implementando um modelo preditivo de tendências de preços dos insumos. Essa inovação distingue-se das abordagens existentes por sua capacidade de adaptação a diferentes conjuntos de dados e por sua aplicabilidade a uma variedade de contextos. O modelo, baseado na arquitetura TFT, proporciona uma solução viável e valiosa, ao responder a uma dificuldade real vivenciada pelos engenheiros de orçamentos, aprimorando a precisão das previsões de custos e, consequentemente, a eficiência dos projetos de construção.

### 3.3 Próximos passos

Considerando o objetivo do projeto, estamos planejando os próximos passos para aprimorar o modelo e torná-lo uma solução robusta e confiável para a indústria da construção civil.

Em primeiro lugar, os resultados obtidos destacaram a necessidade de melhorar a capacidade do modelo em prever padrões de queda dos insumos. Duas alternativas estão sendo consideradas para essa melhoria. A primeira é a utilização de técnicas de reamostragem, como SMOTE, para criar um equilíbrio entre os padrões de aumento e queda nos dados de treinamento.



Em segundo lugar, estamos planejando expandir a base de dados do modelo para incluir informações de outras regiões, ampliando assim sua aplicabilidade a diferentes mercados. Isso também pode contribuir para o aprimoramento do modelo ao proporcionar uma maior diversidade de dados para o treinamento.

Terceiro, é fundamental estabelecer uma infraestrutura robusta para coletar, processar e atualizar continuamente os dados. Isso garantirá que o modelo sempre tenha as informações mais recentes e possa fornecer previsões precisas.

Outro aspecto crítico será a implementação de uma interface de usuário intuitiva que permita aos engenheiros de orçamento interagir facilmente com o modelo e interpretar suas previsões. Isso pode incluir recursos como gráficos interativos, alertas de alterações significativas nas tendências de preços e relatórios personalizáveis.

Finalmente, será importante estabelecer parcerias com empresas de construção civil para testar o serviço em um ambiente real, coletar feedback e fazer ajustes conforme necessário. A longo prazo, a visão é transformar este projeto em uma solução de previsão de preços de insumos autônoma e confiável, que possa ser oferecida como um serviço para empresas de construção civil ou como um recurso interno para uma equipe de cientistas de dados.

