



Ricardo Vianna da Paixão de Barros

**Modelo estocástico preditivo de eventos de sobrepreço,
aplicado a obras de infraestrutura.**

Trabalho de Conclusão de Curso

Trabalho apresentado como requisito parcial à conclusão do curso de Engenharia Civil na Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil.

Orientador: Cristiano Saad.

Rio de Janeiro
Julho de 2023



Agradecimentos

À minha mãe, Heidi Vianna, e ao meu pai, Ricardo Barros, que não mediram esforços para me proporcionar educação e estiveram sempre presentes com inesgotável amor, exemplo, incentivo e acolhimento. À minha tia e madrinha, Leila, pelo apoio, imprescindível à realização desta conquista.

A todos os meus amigos da época de colégio, que me acompanham na vida, sempre disponíveis quando necessário, cujos laços de amizade permanecerão para sempre.

Aos meus amigos e companheiros de curso de Engenharia Civil, em especial Bruna Vallejos, Guilherme Piacsek e Marcos Vinicius Alves, com quem dividi importantes memórias dessa trajetória, que contribuiu para a formação de quem eu sou hoje.

E à Gabriela Veloso, presença constante e parceira, pelo carinho, palavras de sabedoria e incentivo genuíno, sempre fonte de estímulo e motivação.

Agradeço também a todos os que estiveram direta ou indiretamente comigo nesse processo de construção do conhecimento e, em especial, ao meu orientador, Prof. Cristiano Saad, por me conduzir à esta conquista com tanta dedicação. Levarei comigo este aprendizado e gratidão sincera por sua liderança.

Ricardo Vianna da Paixão de Barros



Resumo

No Brasil, um dos pilares do desenvolvimento econômico e da qualidade de vida da população é a realização de obras públicas, especialmente as de infraestrutura, que sustentam toda a atividade produtiva do país. Em abril de 2023, o Governo Federal anunciou o lançamento do futuro plano nacional de investimentos em infraestrutura, confirmado em junho do mesmo ano. O novo plano seria uma terceira edição do Programa de Aceleração do Crescimento (PAC), que reunirá investimentos federais diretos, concessões e Parcerias Público-Privadas e deve englobar mais de mil projetos. Com o novo PAC, as obras de infraestrutura ganham protagonismo na economia nacional, tornando oportuno os esforços para melhorar a eficiência do controle orçamentário de cada empreendimento. Nesse contexto, aborda-se, no presente trabalho, o desenvolvimento de um sistema que propicie mais eficiência à análise e controle de custos. A confecção de um modelo estocástico, desenvolvido por meio da linguagem de programação Python, utilizou como base de dados os gastos reais e previstos contratualmente de empreendimentos reais, com o objetivo de apontar os instantes, ao longo de uma construção, em que há maior probabilidade de ocorrências que levem a sobrepreço. Como resultado, foram criados modelos agrupados em três cenários: otimista, neutro e pessimista. Dentre as aplicações deste conjunto de modelos, destaca-se o auxílio ao planejamento e ao controle de orçamento, contribuindo para uma gestão eficiente do empreendimento.

Palavras-chave: Obras de infraestrutura; planejamento de obras; controle orçamentário; modelo estocástico; Python.



Abstract

In Brazil, one of the pillars of economic development and quality of life for the population is the implementation of public works, especially infrastructure projects, which support the entire productive activity of the country. In April 2023, the Federal Government announced the launch of the future national infrastructure investment plan, confirmed in June of the same year. The new plan would be a third edition of the Growth Acceleration Program (PAC), which will bring together direct federal investments, concessions, and Public-Private Partnerships, and is expected to encompass over a thousand projects. With the new PAC, infrastructure projects gain prominence in the national economy, making efforts to improve the efficiency of budgetary control for each project opportune. In this context, this study addresses the development of a system that provides greater efficiency in the analysis and control of costs. The creation of a stochastic model, developed using the Python programming language, utilized actual and contractually forecasted expenses from real projects as the database, with the objective of identifying the stages during construction with a higher probability of events leading to cost overruns. As a result, models were created clustered into three scenarios: optimistic, neutral, and pessimistic. Among the applications of this set of models, particular emphasis is placed on aiding in planning and budget control, contributing to efficient project management.

Keywords: Infrastructure projects; construction planning; budget control; stochastic model; Python.



SUMÁRIO

1	INTRODUÇÃO	1
1.1	Contextualização e Motivo	3
1.2	Objetivos	5
1.3	Organização do Trabalho	6
2	REVISÃO TEÓRICA.....	7
2.1	Obras de infraestrutura	8
2.2	Fluxo de caixa de uma obra	10
2.3	Python	11
2.4	Processos estocásticos.....	14
3	METODOLOGIA.....	18
3.1	Excel	18
3.2	Python	18
4	ESTUDO DE CASO	23
4.1	Obtenção dos dados e Pré-processamento.....	23
4.2	Python	24
5	CONCLUSÃO E DISCUSSÕES.....	31
	REFERÊNCIAS BIBLIOGRÁFICAS	33
	ANEXO A	37
	ANEXO B	53
	ANEXO C	60
	ANEXO D.....	73



ÍNDICE DE FIGURAS

Figura 1 – Panorama da Construção Civil em 2020.....	4
Figura 2 – Exemplo de série temporal em seu formato em números discretos e acumulado ..	19
Figura 3 – Exemplo de divisão das séries validadas em 3 clusters	22
Figura 4 – Gráfico da obra PARTAGE2.....	23
Figura 5 – Informações e plotagem das séries validadas, geradas pelo parâmetro [0.11881362929570444, 0.12282164753014266].....	26
Figura 6 – Plotagem das séries validadas.....	27
Figura 7 – Plotagem dos clusters formados	28

ÍNDICE DE TABELAS

Tabela 1 – Série em valores discretos, obtida para a obra PARTAGE2.....	24
Tabela 2 – Tabela Resumo.....	25
Tabela 3 – Tabela com os parâmetros de 1 a 6 e suas respectivas séries médias	29
Tabela 4 – Tabela com os parâmetros de 7 a 12 e suas respectivas séries médias	29
Tabela 5 – Tabela com os parâmetros de 13 a 18 e suas respectivas séries médias	30
Tabela 6 – Tabela com os parâmetros de 19 a 24 e suas respectivas séries médias	30



1 INTRODUÇÃO

Gastos com obras de infraestrutura (estradas, ferrovias, portos, hidrovias, energia entre outras) representam importantes investimentos para o país, com benefícios indiretos na economia, como a criação e a manutenção de empregos e a melhoria da produtividade e da competitividade (PEREIRA, 2018).

Com relevante papel para o desenvolvimento nacional, segundo o SindusCon-SP (SINDUSCON, 2023), a atividade da construção brasileira equivale a 4% do PIB brasileiro. No entanto, para 2023, de acordo com a Câmara Brasileira da Indústria da Construção (CBIC), com a redução das expectativas de desempenho da economia brasileira e a falta da previsibilidade da queda da taxa de juros, a expectativa de crescimento do setor caiu de 2,5% para 2,0%.

A indústria de construção civil, da mesma forma em que é responsável por uma grande fatia da economia, ela também enfrenta conflitos associados a gastos e perdas. Mesmo com a previsão de queda de crescimento, como bem ressalta Flyvbjerg (2014), o setor de infraestrutura envolve valores grandiosos. Por esta razão, são necessários altos investimentos de capital, submetidos a diversas incertezas e de difícil controle de custos. Segundo Pio et al (2018), o Brasil precisará investir 8,7 trilhões de reais em infraestrutura até 2038 para eliminar o “hiato de investimento” (a diferença entre o investimento acumulado e o necessário). Com isso, a efetividade do planejamento se torna essencial para o alcance dos resultados objetivados.

Nesse quesito, Goldman (2004) observa que o planejamento se constitui em um dos principais fatores para o sucesso de qualquer empreendimento. Na mesma linha de raciocínio, Chiavenato (1983) também destaca o planejamento como sendo a primeira função administrativa, por ser aquela que atua como base para as demais, estabelecendo, de forma antecipada, os objetivos que devem ser atingidos e como fazer para alcançá-los. E Assed (1986) contribui ao afirmar que o planejamento é a função administrativa que abrange a seleção de objetivos, diretrizes, planos, processos e programas. Para que esses objetivos sejam atingidos de forma eficiente, a empresa precisa alcançar harmonia entre os recursos físicos e os recursos financeiros disponíveis. Para o autor, essa harmonia é obtida por meio de um planejamento racional. Já Formoso (2001) conceitua planejamento como um processo gerencial, que abrange a definição de objetivos e a determinação dos



procedimentos necessários para alcançá-los, sendo eficaz quando realizado em conjunto com o controle.

Complementando, Mattos (2010) pondera que, na construção civil, o gerenciamento de uma obra é um trabalho complexo, por ser uma atividade que envolve uma grande quantidade de variáveis, além de ser desenvolvida em um ambiente particularmente dinâmico e mutável. Entre essas variáveis, uma das mais importantes, segundo Ribeiro (2016), é um orçamento feito corretamente. Ele pode ajudar na previsibilidade financeira em obras e evitar desperdícios. Ainda citando Mattos (2006), o autor reforça que a estimativa de custo é uma avaliação elaborada com base em custos históricos e, também, feito com a comparação de projetos anteriores semelhantes. No seu entender, é comum ver distorções em orçamentos, pois precisam de muitas informações e conhecimento detalhado dos serviços. Nesse quesito, tem destaque o papel do fluxo de caixa (MENON, 2020), no qual despesas e receitas são segmentadas por mês de ocorrência, trazendo uma visão muito mais clara da situação do projeto e do potencial de retorno do investimento.

Diante dessas constatações, a análise das demonstrações financeiras e fluxos de caixa das obras de infraestrutura é uma tarefa fundamental, que requer planejamento face ao grande volume de dados e agilidade nesta verificação, de forma a trazer economia e eficiência ao empreendimento, evitando sobrepreços e evasão de recursos.

Silva (2007) reforça que o fluxo de caixa é considerado por muitos analistas um dos principais instrumentos de análise, proporcionando-lhes identificar o processo de circulação do dinheiro através da variação de caixa. Já para Zdanowicz (2002), o fluxo de caixa é um instrumento de gestão financeira que possibilita ao administrador planejar, organizar, coordenar, gerir e controlar os recursos monetários, elencando suas entradas e saídas. O autor reforça, também, que controlar o fluxo de caixa é tão importante quanto o processo de planejamento, pois um depende do outro para que ambos possam ser úteis e práticos.

Tais afirmações vêm ao encontro do que afirma Menon (2020), ao dizer que o fluxo de caixa traz uma visão direta de quanto o capital investido está trazendo de retorno. O autor destaca ainda que a quebra de custos por período de ocorrência traz ainda uma maior transparência sobre as receitas e despesas e como elas podem ser modeladas para melhorar o desempenho do investimento.



De acordo com Mattos (2010), estudos realizados no Brasil e no exterior indicam que deficiências no planejamento e no controle estão entre as principais causas da baixa produtividade do setor, de suas elevadas perdas e da baixa qualidade dos seus produtos. Logo, esta análise, respaldada pela observação de Mattos, tem justificativa pois, como observado por Nascimento (2021), nos últimos anos, o setor da construção civil tem investido nas técnicas de gerenciamento que possibilitem aos empreendedores um controle mais eficaz de todos os processos elencados, desde a fase preliminar de concepção do projeto até a fase final da construção da obra.

Nascimento (2021), ressalta ainda que, tradicionalmente, na construção civil, as áreas técnica e estrutural sempre foram consideradas partes mais importantes durante a fase de execução. Porém, com as grandes demandas de mercado e, atualmente, com recursos financeiros cada vez mais reduzidos, as construtoras, sejam públicas ou privadas, precisam de maior investimento em planejamento e controle, de forma a identificar e corrigir pontos críticos no gerenciamento dos seus processos.

O planejamento, ou a concepção, e o projeto de engenharia, em geral, empregam ou desenvolvem modelos determinísticos para análise técnica e mercadológica, não levando em consideração as incertezas envolvidas. O desenvolvimento de procedimentos que incorporem a análise de incertezas nos modelos, visando reduzir riscos, torna-se um diferencial.

Este trabalho visa trazer uma contribuição às empresas responsáveis por obras públicas ou privadas, que objetivem alcançar mais previsibilidade, controle e redução de incertezas de custos na elaboração de seus projetos.

1.1 Contextualização e Motivo

Entre 2019 e 2020, a Pesquisa Anual da Indústria da Construção (PAIC) mostrou estabilidade na participação de obras de infraestrutura no valor gerado pelo setor. O segmento caiu para a segunda colocação, entre 2011 e 2020. Dos R\$ 325,1 bilhões gerados em valor de incorporações, obras e/ou serviços em 2020, R\$ 106,3 bilhões são provenientes do segmento de obras de infraestrutura.

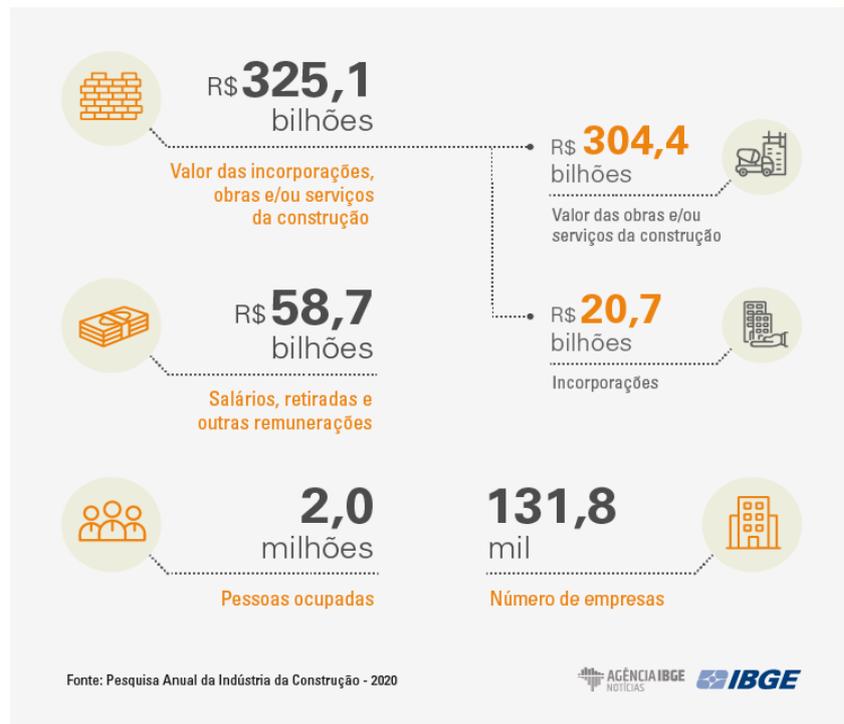


Figura 1 – Panorama da Construção Civil em 2020

Fonte: IBGE

Vale destacar que a indústria de construção civil enfrenta conflitos associados a gastos e perdas. Por esta razão, nos últimos anos, tem experimentado adaptações e melhorias nos seus processos de gerenciamento para atender às demandas de mercado, enquanto cumprem metas de redução de custos.

As empresas perceberam que investir em gestão e controle de processos é inevitável, pois, sem essa sistemática gerencial, os empreendimentos perdem de vista seus principais indicadores: prazo, custo, lucro, retorno sobre o investimento e fluxo de caixa. Informação rápida é um insumo que vale ouro. (MATTOS, 2010).

De acordo com Leite e Possamai (2001 apud SZAJUBOK; ALENCAR; ALMEIDA, 2006), o planejamento da programação de obra em engenharia requer, inicialmente, previsão das atividades a serem realizadas, dos recursos necessários, dos custos estimados, dos prazos e tantos outros elementos importantes para a execução e para o acompanhamento da obra. De acordo com Nascimento (2021), os principais pontos que devem compor esse planejamento são:

- Especificações de todos os materiais que serão usados em cada etapa da obra;
- Seleção dos fornecedores;
- Dimensionamento dos locais de armazenamento;



- Forma de controle desses materiais;
- Definição do consumo desses materiais;
- Produtividade das equipes responsáveis por cada atividade;
- Forma de distribuição do material;
- Controle da aplicação no serviço.

Tal análise impacta diretamente na previsão dos custos (especialmente os de valores mais significativos e que merecem um controle maior), desde a aquisição até a aplicação na obra.

A motivação para a realização deste trabalho surgiu da constatação da importância da utilização de tecnologia para auxiliar a gestão eficiente dos recursos públicos e privados, visando o controle do orçamento das obras de infraestrutura e a otimização dos investimentos.

1.2 Objetivos

Os ganhos de produtividade na execução das obras de infraestrutura estão diretamente ligados à gestão dos custos e à manutenção do cronograma. Os estudos técnico-econômicos que comprovam a exequibilidade do projeto dependem essencialmente da previsibilidade financeira.

O intuito desse estudo foi propor uma ferramenta que contribua para identificar os instantes no fluxo de caixa de uma obra de infraestrutura em que possa haver ocorrências que gerem sobrepreço, fornecendo insumos para maior controle orçamentário, rigor na gestão e identificação de evasão de recursos. O método utilizado foi o desenvolvimento de modelo estocástico, realizado a partir de base de dados de obras de infraestrutura no país.



1.3 Organização do Trabalho

A estrutura do presente trabalho está dividida em cinco capítulos: 1. **Introdução**; 2. **Revisão Teórica**; 3. **Metodologia**; 4. **Estudo de Caso**; 5. **Discussões e Conclusões**. O primeiro capítulo introduz e contextualiza o tema, apresentando também os objetivos do estudo, as justificativas e estrutura do trabalho. O segundo capítulo fundamenta o estudo com autores que abordam temas relacionados aos assuntos tratados e dão base para o estudo de caso. No terceiro capítulo está descrito o passo a passo aplicado para a realização do estudo proposto. Já o quarto capítulo é composto pela aplicação, na prática, da metodologia proposta. Nele, são detalhados os valores trabalhados no estudo e os resultados obtidos ao longo do desenvolvimento deste trabalho. No quinto capítulo são analisados os resultados obtidos no estudo de caso e feitas conclusões a partir das informações reunidas. Tais dados serão tratados a partir das informações levantadas na Revisão Teórica, a fim de mostrar se os objetivos do trabalho foram atingidos.



2 REVISÃO TEÓRICA

Esse estudo se baseou em levantamento bibliográfico coletado de banco de artigos científicos, livros, revistas e textos acadêmicos, especificamente o CAPES periódicos. Inicialmente, com o objetivo de contextualizar o objetivo da pesquisa e aprofundar o conhecimento sobre o tema proposto, foram consultados textos que exemplificam a origem, a natureza e a utilização de modelos estocásticos e de que forma poderiam ser aplicados em Engenharia Civil.

Aprofundando a discussão sobre análises preditivas de custos em diferentes etapas de uma obra, Nakamura (2014), afirma que, saber de antemão e com o máximo de exatidão possível o quanto vai custar uma obra, incluindo todos os seus diversos itens, representa um diferencial competitivo que permite subsidiar a tomada de decisões e antecipar-se à concorrência, além de ajudar a garantir a viabilidade do negócio e a maximizar o resultado financeiro do empreendedor. Em linha com esse pensamento, Udacity (2017) destaca que a análise preditiva auxilia na inteligência competitiva para uma melhor tomada de decisões. Já para Efraim Turban (2017), é a prática de determinação de padrões de dados para prever resultados e as tendências futuras, que permite às empresas reagir conforme as coisas acontecem, de forma a serem proativas.

Procurou-se, também, traçar um panorama da importância para a economia brasileira das obras públicas de infraestrutura e as previsões do setor, com o objetivo de destacar a validade do estudo proposto como uma ferramenta a mais à disposição das empresas na busca da gestão eficiente dos custos de obra. Foi feita ainda uma breve revisão sobre a linguagem de programação Python, utilizada no estudo, que é versátil e tem sido empregada para propósitos diversos. Além de rodar em basicamente qualquer arquitetura de sistema, ela pode ser usada para um leque enorme de aplicações em diferentes áreas, como o da Engenharia Civil.

Esta linha de pesquisa comprova o interesse do mundo acadêmico em aprofundar o conhecimento sobre como incorporar as incertezas nos projetos de engenharia e construções, e o objetivo de desenvolver métodos de aplicação de probabilidade e estatística para alcançar resultados que permitam maior previsibilidade e segurança na tomada de decisão.



2.1 Obras de infraestrutura

De acordo com a Classificação Nacional de Atividades Econômicas (CNAE), do Instituto Brasileiro de Geografia e Estatística (IBGE), a Divisão 42 (Obras de infraestrutura) da Seção F (Construção), detalha obras de infraestrutura como sendo:

- 42.1 – Construção de rodovias, ferrovias, obras urbanas e obras-de-arte especiais;
- 42.2 – Obras de infraestrutura para energia elétrica, telecomunicações, água, esgoto e transporte por dutos;
- 42.9 – Construção de outras obras de infraestrutura.

Já o Departamento Intersindical de Estatísticas e Estudos Socioeconômicos (DIEESE, 2013) classifica a atividade construtiva em três segmentos:

- Construção de edifícios – formado pelas obras de edificações residenciais e de incorporação de empreendimentos imobiliários;
- Obras da construção pesada ou obras de infraestrutura;
- Serviços especializados.

Segundo Mello & Amorim (2009), a indústria da construção civil é composta por uma complexa cadeia produtiva que abrange setores industriais diversos. E o desenvolvimento do país está diretamente relacionado aos investimentos em obras públicas e a eficiente gestão de planejamento que antecede esse empreendimento pela administração pública (SALOMÃO; PÊGO; RHIS & COELHO, 2019).

Para Ferreira e Neto (2010), as obras de infraestrutura demandam altos investimentos iniciais com longos prazos de maturação. Tais fatores reduzem a atratividade destes investimentos fazendo com que sua provisão seja mais adequada ao poder público. Razão pela qual o planejamento orçamentário adequado à realização das obras é fundamental para o sucesso do investimento e devem levar em consideração os gastos futuros com manutenção e operação.

Com isso, os recursos disponíveis para novos investimentos ficam ainda mais reduzidos, o que certamente explica o relevante número de obras públicas paradas no Brasil (SALOMÃO; PÊGO; RHIS & COELHO, 2019). De acordo com dados do relatório do Tribunal de Contas da União (TCU), o Brasil tinha, em novembro de 2022, 8,6 mil obras paralisadas ou em atraso, sendo em grande parte no campo de infraestrutura, como sistemas de saneamento básico e pavimentação de rodovias. As obras receberam



investimento do governo federal de R\$ 27,2 bilhões. (O TEMPO BRASÍLIA, 2022). Segundo a CBIC - Confederação Brasileira da Indústria e Comércio (2018), falhas no planejamento e execução pela administração pública contribuíram para este cenário, gerando aumento dos custos previstos e do tempo de execução do serviço.

Como bem ressaltaram Salomão; Pêgo; Rhis & Coelho (2019), um sistema de infraestrutura pouco desenvolvido dificulta a expansão da economia, limitando a operação de serviços essenciais para toda cadeia produtiva do país. Em linha com essa reflexão, para Gomide e Pereira (2018), o aumento de investimentos em infraestrutura é uma forma de se favorecer o crescimento econômico e o bem-estar social. Além disso, investimentos dessa natureza são geradores de ganhos e de produtividade, e estão relacionados à demanda da sociedade por bens e serviços públicos de qualidade.

Devido à maior fiscalização e o auxílio da mídia e dos mecanismos de transparência recentemente adotados no país, para Medeiros Lima (2016), a ineficiência estatal em aplicar seus recursos tornou-se mais evidente, fazendo com que a população tivesse um comportamento mais consciente sobre o processo de gestão dos recursos públicos.

Hoje, com o avanço da tecnologia da informação, desenvolveram-se muitas ferramentas específicas para o setor da construção civil e, por conseguinte, para obras de infraestrutura, capazes de facilitar e automatizar os processos de gestão, o que, na visão de Sallaberry (2009) tornou as empresas do setor mais competitivas. Para Nascimento (2021), as empresas construtoras precisam se conscientizar de que tais ferramentas representam um grande passo evolutivo na gestão do empreendimento: elas são de extrema importância para o ganho da produtividade de uma empresa e, principalmente, para o acesso a informações mais seguras e em tempo real.

A tecnologia da informação também tem reflexos sobre a previsibilidade financeira, que está relacionada ao controle das receitas e despesas das obras. Um modelo preditivo de custos tem alto potencial em tornar-se uma ferramenta útil para o segmento de obras de infraestrutura. Com o anúncio do novo plano nacional de investimentos em infraestrutura do Governo Federal, o setor público, historicamente o principal contratante, será um dos principais beneficiários. Uma melhor efetividade na destinação dos recursos públicos promoveria a continuidade das obras dentro do orçamento e prazo previstos (SALOMÃO; PÊGO; RHIS E COELHO, 2019).



2.2 Fluxo de caixa de uma obra

Quando se faz referência a controle das receitas e despesas de um projeto de construção civil, a ferramenta de gestão orçamentária é o fluxo de caixa, uma representação gráfica ou analítica das entradas e saídas de dinheiro ao longo do tempo, construído para dar apoio a decisões empresariais, estudar aplicações de resíduos de caixa de permanência temporária e servir de base para a obtenção dos indicadores necessários para a análise financeira (SCHLINDWEIN, 2009 e COSTA NETO et al, 2003).

O fluxo de caixa de uma obra é importante para garantir a viabilidade econômica do projeto, evitar a falta ou o excesso de recursos, otimizar o uso do capital de giro e facilitar a tomada de decisão. Para Ribeiro (2020), o fluxo de caixa permite aos gestores da obra terem uma visão holística ou completa em relação ao negócio, permitindo traçar metas e planejar os gastos dos próximos meses, de maneira a fazer com que a obra seja realizada com sucesso. Ele pode ser elaborado de diferentes formas, mas geralmente segue os seguintes passos:

- Definir o escopo, o cronograma e o orçamento do projeto, considerando os custos diretos e indiretos, os impostos, as margens de lucro e as contingências.
- Estimar as receitas do projeto, considerando as formas e os prazos de pagamento dos clientes, os reajustes contratuais e as possíveis multas ou penalidades.
- Projetar as despesas do projeto, considerando os custos fixos e variáveis, os pagamentos aos fornecedores e prestadores de serviços, os impostos, as taxas e os encargos financeiros.
- Elaborar o fluxo de caixa projetado, comparando as receitas e as despesas em cada período (mensal, trimestral, anual etc.) e calculando o saldo acumulado.
- Analisar o fluxo de caixa projetado, identificando os períodos de déficit ou superávit de caixa, a necessidade de captação ou aplicação de recursos financeiros, a taxa interna de retorno (TIR) e o valor presente líquido (VPL) do projeto.
- Acompanhar o fluxo de caixa realizado, registrando as entradas e saídas efetivas de caixa em cada período e comparando-as com as projeções.
- Revisar o fluxo de caixa projetado, ajustando as premissas e as estimativas conforme a evolução do projeto e as mudanças no cenário.



2.3 Python

2.3.1 Sobre a Linguagem de Programação

Para o avanço da tecnologia, a programação é fundamental. Python é uma linguagem de programação de alto nível, interpretada e de propósito geral. Ela foi criada por Guido van Rossum e lançada em 1991, em Amsterdã, na Holanda. Desde então, a linguagem Python ganhou popularidade rapidamente devido à sua sintaxe limpa e legível, facilidade de uso e grande comunidade de desenvolvedores.

Segundo Silva e Silva (2019), a linguagem Python é gerida pela PSF (Python Software Foundation), fundada em 2001. Trata-se de uma organização sem fins lucrativos que mantém e coordena o uso da linguagem, sendo apoiada por grandes empresas como o Google, Microsoft e, no Brasil, a Globo.com que também utiliza o Python nos seus sistemas.

De acordo com o criador da linguagem, com o passar do tempo, o Python foi evoluindo e, aos poucos sua estrutura foi ganhando mais componentes (POSSAMAI, 2022), como por exemplo, as compreensões de lista (*list comprehension*), as funções *filter*, *map*, e diversas outras características. Hoje, a linguagem é uma das mais usadas no mercado, e já vem instalada de fábrica em vários sistemas operacionais, como por exemplo, o AmigaOS, netBSD, MacOS e todas as distribuições Linux.

Por ser uma linguagem extremamente simples e robusta, além de contar com uma arquitetura bem projetada, ela fornece um bom desempenho e legibilidade do código, o que explica sua presença em várias aplicações do cotidiano, dando suporte às demais tarefas. Além disso, a sintaxe do Python utiliza indentação significativa, o que torna o código mais claro e facilita a leitura. Para Petters (2004), a filosofia do Python enfatiza o princípio do "zen do Python", que valoriza a simplicidade, clareza e facilidade de compreensão do código. “Se a implementação é difícil de explicar, é uma má ideia. Se a implementação for fácil de explicar, pode ser uma boa ideia.” (Tim Petters, 2004).

Outra característica que marca esta linguagem é sua extensa biblioteca padrão, que fornece um conjunto abrangente de módulos e pacotes para realizar várias tarefas, proporcionando economia de tempo e esforço. Os módulos são arquivos individuais que permitem aos desenvolvedores organizar e compartilhar suas funcionalidades de maneira modular, o que facilita a manutenção e o reuso de código. Os módulos podem ser



importados em outros programas Python para estender suas funcionalidades. Ao importar um módulo, as funções, classes e variáveis definidas nele ficam disponíveis para uso no programa atual.

Além da biblioteca padrão, a comunidade Python é conhecida por sua vasta gama de bibliotecas de terceiros. Essas bibliotecas são desenvolvidas por membros da comunidade e são bem abrangentes, como *machine learning*, visualização de dados, desenvolvimento web, entre outras. Através do gerenciador de pacotes do Python, chamado pip, os desenvolvedores podem facilmente instalar e utilizar essas bibliotecas de terceiros em seus projetos.

Em vez de escrever códigos do zero para realizar uma tarefa específica, os desenvolvedores podem aproveitar as funcionalidades pré-existentes nas bibliotecas. Isso acelera o desenvolvimento de aplicativos e permite que os programadores se concentrem na lógica específica do seu projeto, em vez de se preocuparem com detalhes de implementação.

Na mesma linha de pensamento, para Fontana e Voll (2021), uma das principais vantagens em utilizar Python é o grande número de bibliotecas disponíveis para os mais diversos fins. Já para Vicentiner, Mattedi e Mello (2020), Python tem se notabilizado por sua variedade de bibliotecas direcionadas para a área de ciência de dados, aumentando sua eficiência enquanto ferramenta de tratamento e análise de dados e estatísticas.

Em resumo, Python é uma linguagem de programação poderosa e fácil de aprender, que oferece uma ampla variedade de bibliotecas e módulos para facilitar o desenvolvimento de software. As bibliotecas fornecem funcionalidades pré-definidas, enquanto os módulos organizam e compartilham código reutilizável. Com sua ampla adoção e comunidade ativa, Python continua a ser uma escolha popular para uma variedade de aplicações, desde desenvolvimento web até ciência de dados e inteligência artificial.

2.3.2 Biblioteca BirDePy

Para o estudo em questão, foi utilizada a biblioteca BirDePy, um pacote Python desenvolvido para analisar processos gerais de nascimento e morte em tempo contínuo, que inclui funções que podem ser usadas para:

- Aproximar probabilidades de transição (`birdepy.probability()`);



- Estimar valores de parâmetros a partir de dados (`birdepy.estimate()`);
- Simular caminhos de amostra (`birdepy.simulate.discrete()` e `birdepy.simulate.continuous()`);
- Gerar previsões (`birdepy.forecast()`).

Para Hautphenne e Patch (2021), o principal objetivo do novo pacote Python BirDePy é fornecer funções fáceis de usar, que permitem que os parâmetros de PSDBDPs (*Population-Size-Dependent-Birth-and-Death-Processes*) discretamente observados sejam estimados. O uso e eficácia do pacote é demonstrado através de uma variedade de exemplos de PSDBDPs, bem como casos de estudos envolvendo dados anuais de contagem populacional de duas espécies de aves ameaçadas de extinção.

A função *birdypy.estimate()* (de estimativa de parâmetros) é composta por quatro métodos de cálculo de estimativa. Dois deles são baseados em calcular o parâmetro que maximiza a verossimilhança do dado observado, sendo um diretamente (DNM) e outro utilizando um algoritmo de maximização de expectativa (EM). Os dois restantes contam com estimativa por quadrados mínimos (LSE) e computação aproximada Bayesiana (ABC). Para fins deste trabalho, optou-se pelo uso da estimativa por DNM.

- **Direct Numerical Maximization (DNM):**

Segundo Hautphenne e Patch (2021), a maximização numérica direta (DNM) é um *framework* popular de estimação de parâmetros. Essa abordagem, conhecida como estimativa por máxima verossimilhança (MLE), busca encontrar os parâmetros que maximizam a função de verossimilhança descrita abaixo. O método de maximização numérica direta consiste em encontrar esse parâmetro que maximiza a probabilidade do dado Z sob o modelo assumido.

$$\sum_{j=1}^m \sum_{i=1}^{n_j} \log (p_{z_{i-1},j,z_{i,j}}(\Delta_{i,j}; \theta))$$

2.3.3 Biblioteca **tslearn**

De acordo com Tavenard et al (2020), o **tslearn** é uma biblioteca de aprendizado Python que fornece ferramentas para ajustar modelos relevantes para dados de séries temporais. Os dados de séries temporais são entendidos como séries de recursos coletados



ao longo do tempo, incluindo rotinas de pré-processamento, extração de recursos e modelos de *machine learning* para classificação (BAGNALL et al., 2017; FAWAZ et al., 2019), regressão e agrupamento (AGHABOZORGI et al., 2015).

Séries temporais univariadas e multivariadas podem ser tratadas em *tslearn*. Além disso, os conjuntos de dados podem conter séries temporais de comprimento variável, conforme discutido abaixo.

Dentro dessa vasta biblioteca, há funções dedicadas para *clustering*, sendo essas:

- `tslearn.clustering.KernelKMeans` (realiza o agrupamento (*clustering*) de séries temporais por meio do algoritmo Kernel K-Means);
- `tslearn.clustering.KShape` (realiza o agrupamento de séries temporais por meio do algoritmo KShape);
- `tslearn.clustering.TimeSeriesKMeans` (realiza o agrupamento de séries temporais por meio do algoritmo Time Series K-Means); e
- `tslearn.clustering.silhouette_score` (Calcula o coeficiente médio de silhouette das amostras fornecidas).

Para fins do projeto, será utilizada a classe `TimeSeriesKMeans`, visto que essa se caracteriza por trabalhar com séries temporais, sendo a mais adequada para realizar os agrupamentos desejados.

2.4 Processos estocásticos

2.4.1 Processo determinístico e estocástico

Os processos estocásticos proporcionam um tipo de modelo matemático que permite a inclusão de incerteza em suas estimativas, simulando o comportamento de variáveis aleatórias. Segundo Taylor e Karlin (1998), os processos estocásticos são formas de quantificar as relações dinâmicas de sequências de eventos aleatórios. Para os autores, os modelos estocásticos desempenham um papel importante na elucidação de muitas áreas das ciências naturais e da engenharia. Eles também ressaltam que tais modelos podem ser usados para analisar a variabilidade inerente aos processos biológicos e médicos; para lidar com as incertezas que afetam as decisões gerenciais e com as complexidades das interações psicológicas e sociais; ou ainda para fornecer novas perspectivas em outros estudos matemáticos e estatísticos.



Para Hellemo, Barton e Tomaszgard (2018), a maioria dos problemas práticos de decisão envolve incerteza em algum nível, e a programação estocástica foi introduzida por Dantzig e Beale (1955) para lidar com parâmetros incertos em modelos de otimização. A abordagem utilizada foi modelar um processo de decisão de tempo discreto, no qual parâmetros incertos são representados por cenários e suas respectivas probabilidades. Em um programa estocástico baseado em cenários, decisões são tomadas e valores incertos são revelados em pontos discretos no tempo. Algumas decisões são tomadas antes que os valores reais dos parâmetros incertos sejam conhecidos.

De acordo com Lars et al (2018), a descrição de um problema pode ser classificada como determinística, que são problemas em que não há incerteza substancial; ou problemas com incerteza substancial nos quais a distribuição dos parâmetros estocásticos é conhecida, por exemplo, com base em dados históricos ou opinião de especialistas. Um modelo determinístico é um modelo que não possui elementos aleatórios e que a relação entre as entradas e as saídas do modelo são totalmente determinadas.

Já um modelo estocástico possui um ou mais elementos aleatórios e a relação entre as entradas e as saídas do modelo é incerta. Um modelo estocástico tem a capacidade de lidar com incertezas nas entradas aplicadas. Modelos estocásticos possuem alguma aleatoriedade inerente - o mesmo conjunto de valores de parâmetros e condições iniciais levará a um conjunto de saídas diferentes.

Um exemplo de processo estocástico é uma coleção de variáveis aleatórias indexadas por um parâmetro de tempo. Os métodos estocásticos correspondem aos processos em que, para cada valor das variáveis de entrada de um modelo, vai existir uma distribuição de probabilidades dos valores a serem assumidos pela variável dependente, a qual varia com o tempo, tornando as variáveis, em parte, aleatórias (ROGERS; SANTOS; LEMES, 2008).

Para C. Bisset & S.E. Terblanche (2021), a programação estocástica é um recurso para os futuros pesquisadores devido à previsão de incerteza, que foi ignorada em experimentos determinísticos globalmente. É difícil encontrar exemplos de sistemas para analisar que não incluam aleatoriedade ou incerteza.

Uma das principais vantagens dos modelos estocásticos é que eles são capazes de capturar a incerteza de forma mais realista e precisa do que modelos determinísticos. De acordo com Taylor e Karlin (2015), os modelos determinísticos assumem que as variáveis



do sistema possuem valores fixos e previsíveis, o que nem sempre é verdadeiro na prática. Já os modelos estocásticos levam em conta as incertezas e variações que podem ocorrer em uma dada situação, permitindo uma simulação mais realista do comportamento do sistema. Os autores afirmam que os modelos estocásticos são úteis para a identificação de riscos e para a tomada de decisões informadas em ambientes de incerteza. Para eles, os modelos estocásticos têm evoluído rapidamente ao longo das últimas décadas, graças ao desenvolvimento de técnicas avançadas de simulação e análise de dados. Atualmente, existem diversas ferramentas de software que permitem a construção e análise de modelos estocásticos com facilidade e precisão.

Ainda segundo Taylor e Karlin (2015), um modelo determinístico prevê um único resultado de um determinado conjunto de circunstâncias. Um modelo estocástico prevê um conjunto de possíveis resultados ponderados por suas verossimilhanças ou probabilidades. No entanto, para os autores, os fenômenos não são em si mesmos inerentemente estocásticos ou determinísticos. Em vez disso, modelar um fenômeno como estocástico ou determinístico é uma escolha do observador. A escolha do modelo depende do propósito do observador, sendo a utilidade deste modelo o melhor critério para a escolha. Para ser útil, o modelo deve refletir todos os aspectos do fenômeno em estudo e ser passível de cálculo, permitindo a dedução de previsões ou implicações importantes sobre o fenômeno.

Em resumo, os modelos estocásticos são uma ferramenta poderosa para a análise e previsão de eventos aleatórios em diversas áreas. Eles são capazes de capturar a incerteza de forma realista, permitindo a avaliação de diferentes cenários e a identificação de riscos em ambientes de incerteza. No entanto, é importante ter em mente suas limitações e a necessidade de uma análise cuidadosa dos dados para garantir a precisão dos resultados obtidos.

2.4.2 Processos de Poisson e Markov

Dentre os modelos gerados por um processo estocástico, há dois principais tipos: um modelo baseado no processo de Poisson e um modelo baseado em processo Markoviano.

Um processo de Poisson é o tipo de processo estocástico que conta o número de eventos que ocorrem em um intervalo de tempo contínuo. Por exemplo, o número de



ligações telefônicas que chegam em uma central em uma hora. Um processo de Poisson tem duas propriedades principais: a probabilidade de ocorrer um evento em um pequeno intervalo de tempo é proporcional ao comprimento do intervalo e independente do tempo e dos eventos anteriores. E a probabilidade de ocorrerem mais de um evento em um pequeno intervalo de tempo é desprezível (DIAS, Edgard, 2011).

Para Taylor e Karlin (2015), o processo de Markov é outro tipo de processo estocástico que descreve a evolução aleatória de um sistema que pode estar em diferentes estados. Um processo de Markov tem uma propriedade fundamental: a probabilidade de transitar para um estado futuro depende apenas do estado atual e não dos estados passados. Dessa forma, um processo Markoviano se caracteriza por não possuir memória.



3 METODOLOGIA

O presente capítulo tem por objetivo, descrever a metodologia aplicada para a criação do modelo proposto. Essa é dividida em duas etapas, a primeira consistindo em tratamento de dados com a ferramenta Excel, e a segunda em cálculos e estimativas com a linguagem de programação Python.

3.1 Excel

A criação do modelo estocástico, proposto neste trabalho, inicia-se pela coleta de dados de obras passadas, sendo essas de mesma natureza daquela para a qual o modelo será aplicado. Com um pré-processamento desses dados, as informações desejadas restantes serão os fluxos de caixa previstos e reais, para cada obra observada. A partir disso, os dados serão trabalhados de maneira a se obter, para cada obra, uma série temporal de valores discretos composta apenas por -1, 0 e 1. Primeiramente, é obtido uma série temporal a partir da diferença entre os fluxos de caixa (para cada dia, semana ou mês, conforme os dados permitirem), avaliando-se se foi gasto o mesmo que previsto ou se houve um sobrecusto ou economia no período observado. Essa série então é convertida em um modelo de nascimento e morte, composto pelos valores de -1, 0 e 1, sendo -1 atribuído aos momentos em que houve economia, 0 para quando foi gasto a quantia prevista e 1 para os instantes em que houve sobrepreço.

Com o tratamento dos dados concluído, as amostras de séries, em valores discretos de -1, 0 e 1, são divididas em dois conjuntos. O primeiro é dedicado às amostras para *cross-validation*, utilizadas para a obtenção dos parâmetros que geram o modelo desejado, representando 90% da base utilizada. Os 10% restantes compõem o conjunto de amostras para validação do modelo criado.

3.2 Python

Tendo ambas as bases definidas, passa-se a utilizar a linguagem de programação Python, considerando a sua eficiência para os cálculos necessários e para a plotagem dos gráficos desejados.



3.2.1 Validação cruzada

Partindo da base composta por 90% do *dataset* original, utiliza-se um processo de *cross-validation* para a obtenção dos parâmetros que irão gerar o modelo desejado. Para cada amostra presente na base, será percorrido o passo a passo descrito a seguir:

- i. A amostra selecionada será separada das restantes, sendo chamada de **amostra teste**. As demais amostragens irão compor a **base de treino**.
- ii. É obtida uma série mais provável (**SMP**) a partir da base de treino, utilizando-se a função *fluxo_mais_provavel()*, que servirá para ampliar a quantidade de simulações geradas. A função consiste em, para cada instante de tempo, verificar se o valor mais recorrente, entre as amostras, equivale a -1 (economia), 0 (gasto esperado) ou 1 (sobrecusto).
- iii. Todas as amostras (as séries da base de treino, a série SMP e a amostra teste) são transformadas em séries acumuladas, por meio da função *converte_para_serie_acumulada()*.

Exemplo de série em valores discretos:
 [0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, -1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, -1, 0, 0, 1, 0, 1, -1, 1, 1, 0]

Exemplo de série acumulada:
 [0, 0, 1, 2, 2, 3, 3, 3, 4, 5, 5, 5, 6, 7, 7, 8, 7, 7, 7, 7, 7, 7, 7, 7, 8, 9, 9, 10, 10, 9, 9, 9, 10, 10, 11, 10, 11, 12, 12]

Figura 2 – Exemplo de série temporal em seu formato em números discretos e acumulado
 Fonte: Autor

- iv. É utilizada a função *parametros_estimados()* para se estimar os parâmetros para cada série de treino, incluindo a série SMP. A função consiste em utilizar o método de estimativa DNM, já definido no tópico **2.3.2**, fornecido pela função *estimate()* da biblioteca ByrDePy, também já apresentada. Dessa forma, é obtido uma lista de parâmetros, sendo um para cada série da base de treino, e um para a série SMP (parâmetro **SMP**).



- v. É calculado um parâmetro médio (**PM**) a partir da lista de parâmetros anterior, que também será utilizado para a simulação de modelos que virá em seguida. Para isso, é utilizada a função *parametro_medio()*, que calcula a média aritmética dos parâmetros fornecidos.
- vi. Com os parâmetros estimados (parâmetro PM, SMP e lista de parâmetros de treino), inicia-se a simulação de modelos, que ocorrerá com o auxílio de 3 principais funções.
- ❖ A primeira função, *simulate_series()* é responsável diretamente por simular uma série temporal a partir de um parâmetro fornecido. Então, essa série gerada é interpolada para garantir que tenha o mesmo comprimento da amostra teste. A simulação da série propriamente dita é realizada com o auxílio da função *simulate()*, da biblioteca BirDePy, já apresentada anteriormente no tópico **2.3.2**.
 - ❖ Já a função *calibração_do_modelo()* utiliza a função *simulate_series()* e confere se a série retornada atende a certas métricas definidas, quando comparada à amostra de teste. O número de simulações é determinado pelo usuário e, ao seu término, todas as séries que atenderem às restrições são retornadas. As métricas utilizadas e seus respectivos valores de restrições são:
 - Coeficiente de correlação de Pearson, calculado pela função *np.corrcoef()*. Deve ser superior a 95%;
 - MAE - erro absoluto médio. Deve ser inferior a 10;
 - MSE - erro quadrático médio. Deve ser inferior a 100;
 - SMAPE - erro percentual absoluto médio simétrico. Deve ser inferior a 120.
 - ❖ Por fim, a função *simulação_de_modelos_calibrados()* é responsável por repetir as funções anteriores até que um número mínimo, determinado pelo usuário, de modelos calibrados seja retornado, assim garantindo amostras para a validação final do modelo estocástico proposto.



- vii. Tendo o conjunto de modelos calibrados pela amostra teste definida, os passos anteriores são repetidos para cada opção de amostra teste dentro do *dataset* fornecido.

3.2.2 Validação do modelo

Com todos os modelos retornados pelas simulações com a lista de parâmetros e os parâmetros SMP e PM, inicia-se a etapa final dessa metodologia. Utilizando a função *validação_do_modelo()*, todas as séries salvas são avaliadas pelas mesmas métricas da etapa anterior, tendo como diferença a série de comparação. Para a etapa de validação, a série a ser comparada é aquela gerada pela função *fluxo_mais_provavel()*, dessa vez aplicada à base de amostras para validação, composta por 10% do *dataset* original. Com esse novo filtro, são obtidos os modelos finais, calibrados e validados.

3.2.3 Pós-processamento – Clusterização

Para analisar os resultados finais obtidos pelas simulações e estimativas anteriores, utiliza-se uma técnica de clusterização (agrupamento) para dividir os parâmetros validados em subgrupos. A partir da biblioteca *tslearn* do Python, a função *TimeSeriesKMeans()* é aplicada para realizar os agrupamentos desejados. Para tal, é preciso gerar uma única série para cada um dos modelos finais validados. Dessa forma, para as séries temporais validadas anteriormente, será obtida uma série média dentre aquelas que forem geradas por um mesmo parâmetro, calculando a média aritmética para cada instante.

Por fim, essas séries médias são, então, clusterizadas em um número de grupos definido pelo usuário, possibilitando assim uma plotagem desses agrupamentos e a interpretação visual dos resultados finais.

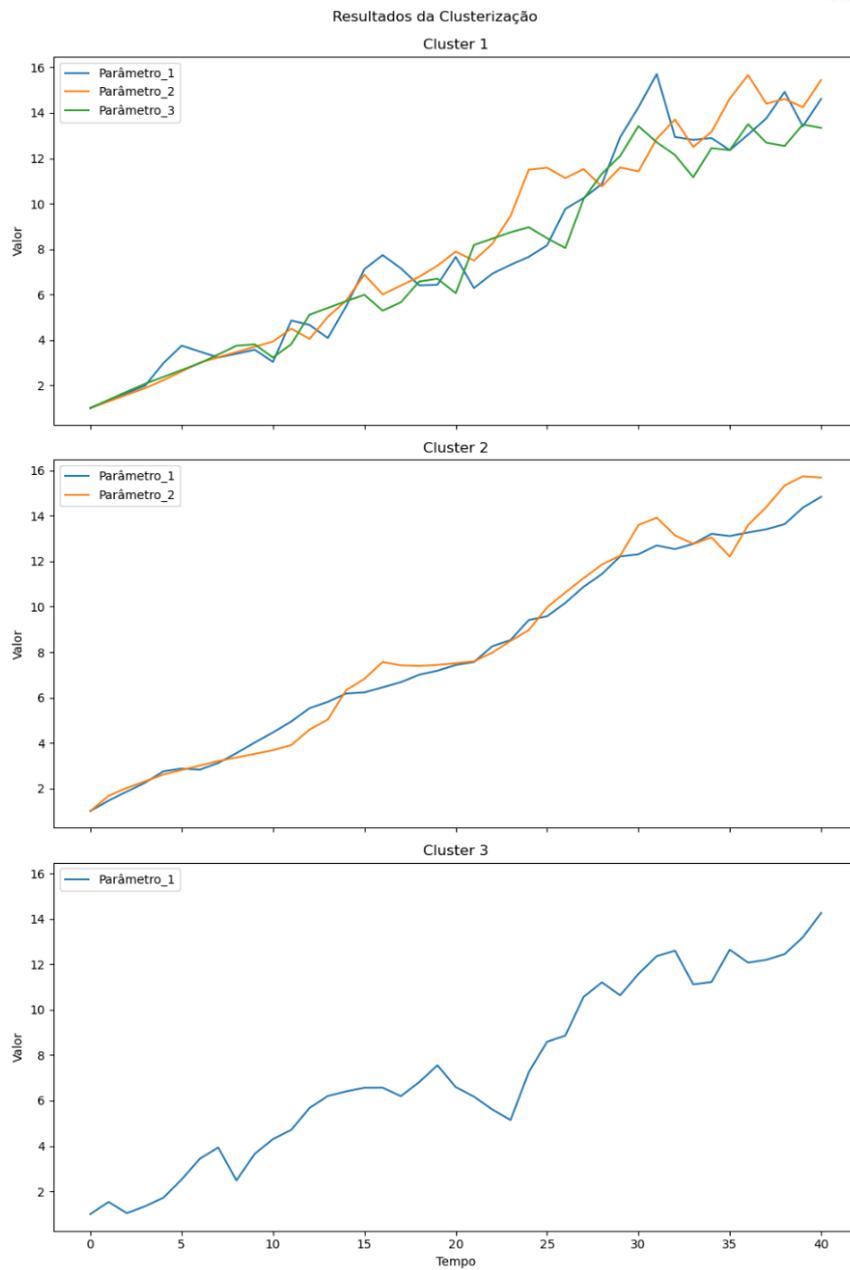


Figura 3 – Exemplo de divisão das séries validadas em 3 clusters.

Fonte: Autor



4 ESTUDO DE CASO

Neste capítulo, é realizado o estudo para obtenção do modelo estocástico desejado, a partir da aplicação da metodologia descrita anteriormente.

4.1 Obtenção dos dados e Pré-processamento

Os dados utilizados para o respectivo trabalho foram informações obtidas a partir de obras de infraestrutura. Essas informações foram fornecidas no formato de gráficos que apontavam o avanço dos gastos da obra ao longo do tempo, comparando entre o orçamento disponibilizado por contrato e o gasto real. Os gráficos analisados estão presentes no Anexo A.

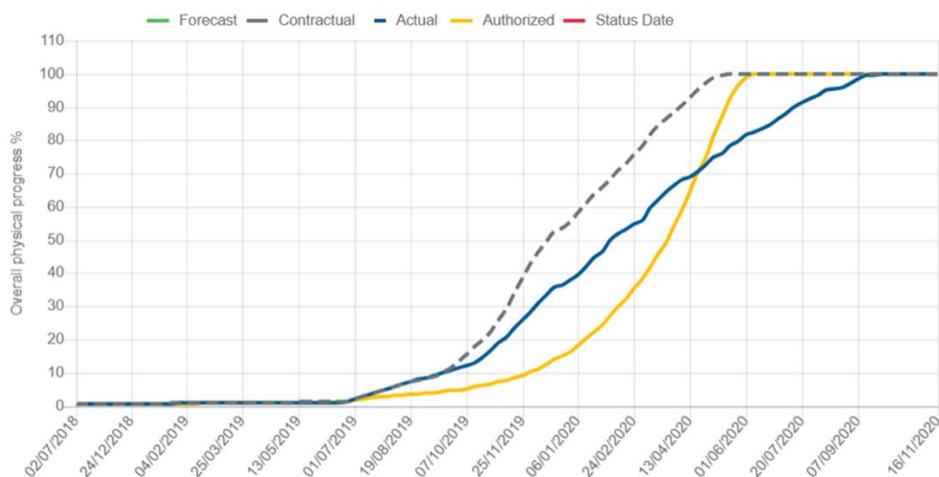


Figura 4 – Gráfico da obra PARTAGE2

Fonte: Autor

Ao todo, foram obtidos gráficos para 48 obras. Assim, para cada um, a duração da obra foi dividida em 40 intervalos e, para cada intervalo, foi definido um valor de 0, 1 ou -1 considerando as curvas azuis (gasto real) e amarelas (gasto previsto por contrato). Para aqueles instantes em que as curvas coincidem, foi adotado o valor 0. Para quando a curva azul estiver sobre a amarela, foi adotado o valor de 1. Por fim, para a curva azul estando abaixo da amarela, aplicou-se o valor de -1. A partir desse tratamento dos dados, foi montado o arquivo em Excel que está ilustrado no Anexo B.



DATA	PARTAGE2
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	1
14	1
15	1
16	1
17	1
18	1
19	1
20	1
21	1
22	1
23	1
24	1
25	1
26	1
27	1
28	1
29	0
30	-1
31	-1
32	-1
33	-1
34	-1
35	-1
36	-1
37	0
38	0
39	0
40	0

Tabela 1 – Série em valores discretos, obtida para a obra PARTAGE2
 Fonte: Autor

4.2 Python

Tendo as séries em valores discretos para cada uma das 48 obras, esses dados foram passados ao Python, dividindo as amostras em dois conjuntos. O primeiro, utilizado para a etapa de *cross-validation*, é composto por 90% das amostras, equivalente a 44 unidades. Já o segundo conjunto foi formado pelas 4 obras restantes, compondo a base de validação.

4.2.1 Validação cruzada

Prosseguindo com a metodologia utilizada, foi dado início à etapa de *cross-validation*. Nela, para cada uma das 44 séries da sua respectiva base, uma é selecionada como **série para teste** e as 43 restantes compõem a base de treino. Seguindo o passo a passo descrito no Capítulo 3, são estimados parâmetros de três maneiras diferentes. A primeira, gerando



o **Parâmetro SMP**, é por meio da série mais provável, calculada com a base de treino aplicada à função *fluxo_mais_provavel()*. A segunda maneira, é estimando um parâmetro para cada uma das 43 séries, individualmente, obtendo uma **Lista de Parâmetros**. Já a terceira, obtendo o **Parâmetro PM**, consiste em calcular o parâmetro médio entre aqueles presentes na Lista de Parâmetros.

Com isso, em cada repetição, foram simulados 45 parâmetros, sendo 43 para a Lista de Parâmetros, 1 para o Parâmetro SMP e 1 para o Parâmetro PM e, para cada um desses, foram simuladas séries temporais, que representam um cenário possível para uma obra. Comparando com a série teste, são calculadas as métricas (MAE, MSE e SMAPE apresentadas na Metodologia) e definido se a série simulada é válida ou não. Para cada parâmetro, foram realizadas baterias de 50 simulações, até que fossem encontradas 5 séries válidas ou atingisse o limite de 50 repetições (equivalente a 2500 simulações). Para cada parâmetro individualmente, se houver uma série validada, este passa a ser considerado um parâmetro calibrado, que será utilizado para a etapa seguinte de validação final.

	Parâmetros estimados	Séries simuladas	Séries validadas	Parâmetros calibrados	Tempo
Parâmetros SMP	44	77.850	99	2	0:03:23
Parâmetros PM	44	89.650	72	16	0:03:53
Listas de Parâmetros	1.892	4.259.200	562	34	3:04:51
Total	1.980	4.426.700	733	52	3:12:07

Tabela 2 – Tabela Resumo

Fonte: Autor

4.2.2 Validação dos modelos

A partir das séries validadas e seus respectivos parâmetros calibrados, realizou-se a validação final desses modelos. Para cada série já simulada, foram aplicadas as mesmas métricas utilizadas na etapa de cross-validation, exceto que, agora, cada série temporal foi comparada à base de validação, composta pelas 4 séries restantes do *dataset* original. Para essa comparação, foi gerada uma série mais provável a partir dessa base e, então, cada série foi novamente comparada.

Para aquelas que atenderam às métricas, foram traçados gráficos para uma melhor visualização do seu comportamento. Esses estão apresentados no Anexo C. Na sequência,



vê-se exemplo de um dos parâmetros validados, com cada uma das suas séries simuladas, validadas, e as respectivas métricas.

Modelo: [0.11881362929570444, 0.12282164753014266]

Simulada_1:

MSE: 5.527822742681606

MAE: 1.890131281953733

SMAPE: 78.23210194342931

Correlation: 0.9535948283243955

Simulada_2:

MSE: 7.492285812962924

MAE: 2.44753682935083

SMAPE: 81.53249360075506

Correlation: 0.9563744702291896

Simulada_3:

MSE: 6.009913576645781

MAE: 2.036661101266076

SMAPE: 79.1778803114932

Correlation: 0.9505734904858473

Simulada_4:

MSE: 17.32820303527061

MAE: 3.860249767099414

SMAPE: 90.91770553502727

Correlation: 0.9564667540621916

Simulada_5:

MSE: 5.419857408367271

MAE: 2.0587204588636587

SMAPE: 77.5007165017328

Correlation: 0.9553625736099749

Simulada_6:

MSE: 3.3828661114734486

MAE: 1.5574997606550025

SMAPE: 75.87956256544578

Correlation: 0.9710912641480236

Simulada_7:

MSE: 3.4462737110297907

MAE: 1.5868342095367611

SMAPE: 73.65290775713792

Correlation: 0.9625109506104625

Simulada_8:

MSE: 12.868067037408736

MAE: 3.4102725565492973

SMAPE: 88.37249168982868

Correlation: 0.9813091211015792

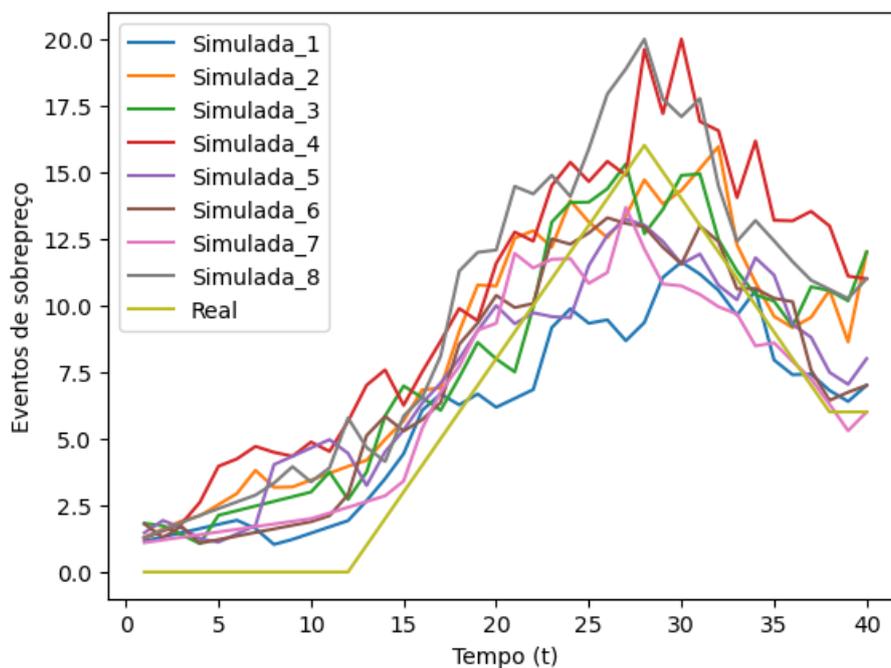


Figura 5 – Informações e plotagem das séries validadas, geradas pelo parâmetro [0.11881362929570444, 0.12282164753014266]

Fonte: Autor



Ao fim da etapa de validação, foram obtidas 88 séries validadas, equivalentes a 24 parâmetros validados. Abaixo, está o comportamento de todas essas séries, sobrepostas.

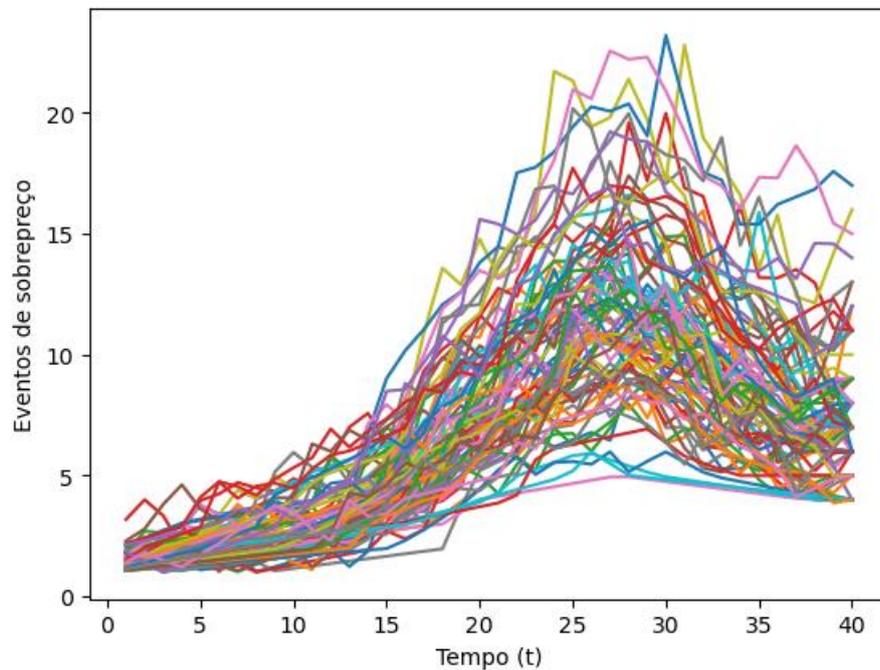


Figura 6 – Plotagem das séries validadas

Fonte: Autor

4.2.3 Pós-processamento

Com os parâmetros finais validados e suas respectivas séries, foi realizada a última etapa da metodologia adotada. Para cada um dos 24 parâmetros, foi calculada uma série média e, com a biblioteca *tslearn*, foi então realizada a clusterização desses modelos. Abaixo, estão representados os agrupamentos construídos, assim como os parâmetros analisados e suas respectivas séries.

De posse desses resultados, parte-se para a análise visual dos gráficos gerados, o que permitirá interpretações e conclusões da aplicação dos modelos.



Resultados da Clusterização

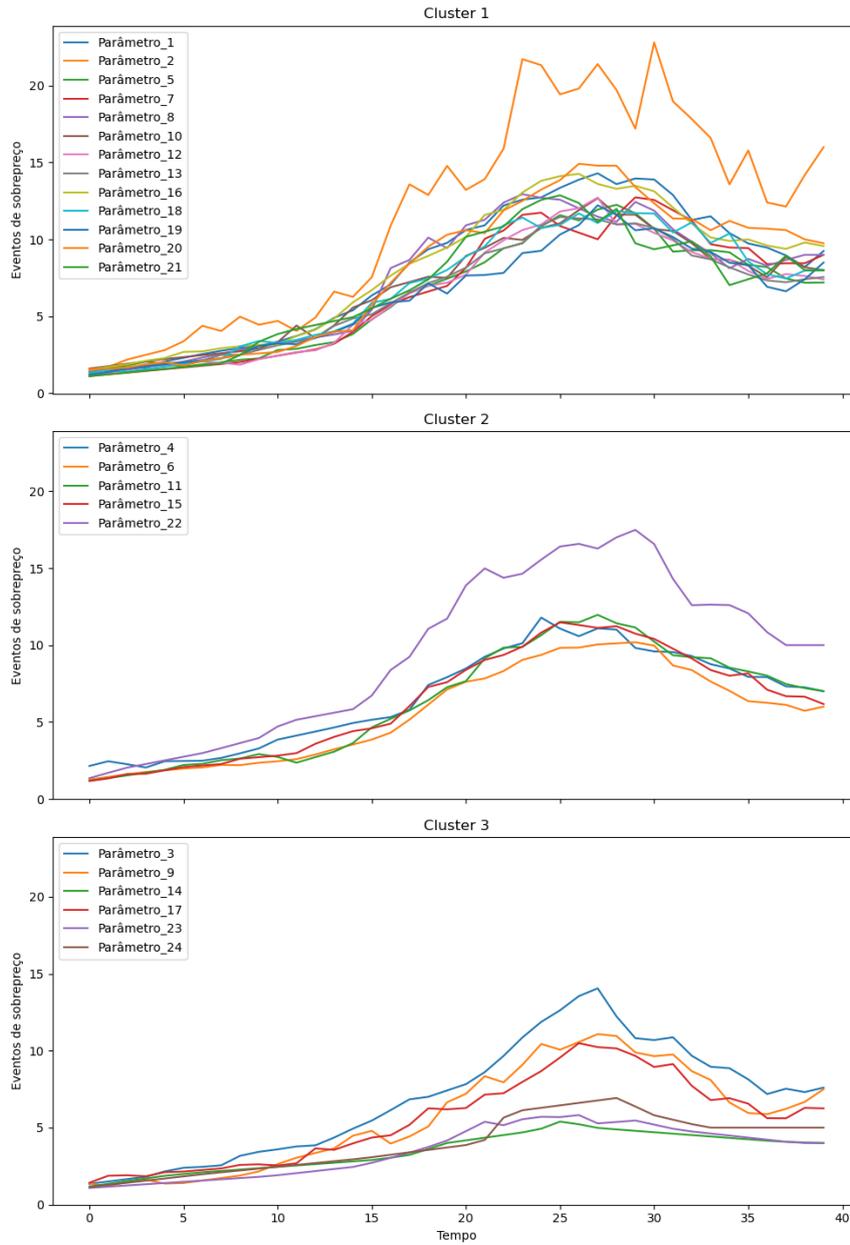


Figura 7 – Plotagem dos clusters formados

Fonte: Autor



TEMPO	Parâmetro 1: [0.118813629570444, 0.12282164753014266]	Parâmetro 2: [0.31437127602021914, 0.27318183003278473]	Parâmetro 3: [0.1050502084165468, 0.07076715620958508]	Parâmetro 4: [0.0896317243767274, 0.06976351811352355]	Parâmetro 5: [0.10039500615769674, 0.10333319421298007]	Parâmetro 6: [0.05727385475501244, 0.0659949712713608]
1	1,400539306	1,171061169	1,345933324	2,143500229	1,45846477	1,257383018
2	1,522795449	1,706671973	1,49910452	2,448344853	1,649370381	1,428910482
3	1,637380986	2,192030403	1,652275716	2,256923253	1,797408177	1,600437947
4	1,657666522	2,500127091	1,805446912	2,040581919	2,055294344	1,753617292
5	2,075250169	2,808223779	2,154683145	2,449615206	1,894206225	1,863188951
6	2,303276602	3,380295178	2,390515953	2,46598001	1,732427326	1,972760609
7	2,552763185	4,389443448	2,450700422	2,482344814	1,876356634	2,043847408
8	2,763633966	4,047922668	2,550647359	2,664575509	2,020285943	2,210517953
9	2,938877491	4,981726497	3,167068791	2,963117152	2,176604087	2,194094094
10	3,085343593	4,456520917	3,427982468	3,282138816	2,264517299	2,350949901
11	3,355741243	4,706785398	3,588102198	3,857033433	2,811185641	2,446510681
12	3,72652327	4,05720744	3,776808552	4,117668973	2,88873405	2,585284201
13	4,159492326	4,920839693	3,848139162	4,378304514	3,135121967	2,884782981
14	4,905576673	6,610965368	4,355589334	4,644637448	3,321553811	3,227404917
15	5,400771458	6,266189178	4,934952262	4,934036853	3,855203347	3,538425888
16	6,361802361	7,546895357	5,457100559	5,147087525	4,801234768	3,85676335
17	7,073720966	10,8926927	6,135824877	5,318759132	5,598566072	4,305931308
18	8,497364124	13,58113744	6,839296078	5,793395371	6,571079268	5,154102124
19	9,362188463	12,88921095	7,003374935	7,405305836	6,983974324	6,130580123
20	9,77455347	14,78193476	7,415499496	7,91380845	7,44579791	7,112989519
21	10,61276544	13,21728898	7,824684361	8,475750146	7,921083809	7,615387507
22	10,9182777	13,91948543	8,608556829	9,239696657	8,519011626	7,830842981
23	12,2036794	15,89877449	9,662416229	9,77804677	9,424752373	8,313667403
24	12,58523273	21,71748366	10,85743943	10,11838834	9,747314982	9,037793085
25	12,73242526	21,32719064	11,8719933	11,78450871	10,84651524	9,361371243
26	13,35417495	19,42968645	12,62576169	11,07203779	11,57861227	9,819760037
27	13,88205863	19,8114476	13,54778689	10,5765168	11,23630828	9,83890811
28	14,29787254	21,39850282	14,05181014	11,08524954	11,93946796	10,04053491
29	13,593001	19,71880672	12,22684907	11,00585789	12,25211886	10,11145586
30	13,96215082	17,19672136	10,81528105	9,811804689	11,69590484	10,1796721
31	13,89639678	22,80558624	10,69245277	9,58884178	10,69981067	9,967333189
32	12,8868499	18,9726175	10,87242968	9,546065681	9,208685636	8,674327105
33	11,26575216	17,82429825	9,670777613	9,285026247	9,29467666	8,377473556
34	11,5031428	16,59505807	8,956842627	8,758194186	9,304410067	7,631897124
35	10,41279307	13,57613238	8,861417209	8,480052748	9,163717773	7,025915865
36	9,737159813	15,78358329	8,135200059	7,931402387	8,347969059	6,350557567
37	9,467754544	12,40255814	7,181707136	7,920215426	8,201292573	6,247089277
38	8,963249564	12,13684164	7,528138852	7,313785487	7,466174811	6,113716472
39	8,201201022	14,18507631	7,311002568	7,254636032	7,184060887	5,727573225
40	9,25	16	7,6	7	7,2	6

Tabela 3 – Tabela com os parâmetros de 1 a 6 e suas respectivas séries médias

Fonte: Autor

TEMPO	Parâmetro 7: [0.0688395576723993, 0.025218936132029596]	Parâmetro 8: [0.06400264536200635, 0.03647968857365036]	Parâmetro 9: [0.06604965459144374, 0.04978964757176377]	Parâmetro 10: [0.0953110862577362, 0.042044077601490044]	Parâmetro 11: [0.09957151766252241, 0.04287615772858795]	Parâmetro 12: [0.0928326597057208, 0.07533463846694832]
1	1,112893056	1,174679626	1,381304036	1,60936866	1,176010151	1,413771506
2	1,225786112	1,349359253	1,172986756	1,763792866	1,352020302	1,586071012
3	1,338679168	1,524038879	1,484327064	1,918217073	1,528030453	1,627068381
4	1,451572224	1,698718506	1,618722255	2,072641279	1,704040605	1,81754704
5	1,56446528	1,873398132	1,361084437	2,227065486	1,892715682	1,813725394
6	1,677358336	2,068910793	1,399421865	2,353501382	2,20626586	2,010360889
7	1,790251392	2,319282522	1,560979641	2,478103288	2,278256062	2,057685296
8	1,903144448	2,569654251	1,722537416	2,602705194	2,51704012	1,978454925
9	2,029221707	2,82002598	1,884095191	2,7273071	2,627486415	1,861696974
10	2,234922532	3,053833149	2,154504958	2,909661874	2,914250791	2,215636384
11	2,440623358	3,245292483	2,630359654	3,171712365	2,729435557	2,453755568
12	2,646324184	3,436751816	3,042124107	4,405399368	2,353396662	2,677286182
13	2,852025009	3,62821115	3,34765639	3,594406669	2,716525362	2,794540283
14	3,21884962	3,819670484	3,653188673	4,416488312	3,069595459	3,26719464
15	3,99886186	4,067804516	4,474403389	5,57430173	3,63400438	4,467536027
16	5,017259593	5,439465217	4,79177308	6,046937153	4,648645026	4,77372655
17	5,768395733	8,138827502	3,964838094	6,891436059	5,208259095	5,670207875
18	6,251868608	8,669532716	4,426097935	7,230244464	5,75003267	6,49275848
19	6,616018767	10,12112699	5,074060569	7,576665765	6,419508304	6,981161657
20	6,980168925	9,434931084	6,647669552	7,507545003	7,260522578	7,189833774
21	8,154682318	10,9181484	7,197822779	8,9306505	7,652315677	7,710200757
22	10,03846961	11,27851345	8,341339774	9,45469257	9,135697478	9,147521197
23	10,58331017	12,39383466	7,94476585	10,10093915	9,834258206	9,942327465
24	11,59818467	12,9411926	9,081984588	9,968697611	9,879556513	10,61011549
25	11,73990219	12,7088009	10,43751913	10,74435476	10,64336444	10,97084673
26	10,87203184	12,5818556	10,06619526	11,05169972	11,5138309	11,61672976
27	10,44093693	12,01931395	10,56083371	11,90242151	11,47848669	12,03000961
28	10,00984201	11,47753909	11,07907369	12,67899032	11,96443684	12,72116406
29	11,56326779	11,14980459	10,95456658	11,62152706	11,41796882	10,96283086
30	12,73515537	12,44201601	9,890606011	11,60710604	11,1430273	11,00891371
31	12,56750039	11,86924442	9,642691199	10,6767217	10,22309894	10,42365007
32	11,89668029	10,66244868	9,754589284	10,55044553	9,339694212	9,939817908
33	11,22489347	9,854080794	8,670687838	9,756082214	9,211124545	9,18176683
34	9,690564943	9,112260422	8,097541473	8,726048977	9,143534606	8,745384671
35	9,476742496	8,12450162	6,63463085	8,667334149	8,532045037	8,689387345
36	9,435215006	8,737535447	5,948950059	8,296381899	8,284924731	7,951494807
37	8,394843251	8,288853997	5,872378405	7,541226129	8,013102335	7,434879603
38	8,452476167	8,669744139	6,220034116	8,864821841	7,468262465	7,748898228
39	8,451983643	9	6,673809612	8,191642554	7,202856089	7,626257924
40	9	9	7,5	8	7	7,4

Tabela 4 – Tabela com os parâmetros de 7 a 12 e suas respectivas séries médias

Fonte: Autor



TEMPO	Parâmetro 13: [0.1087103089156497, 0.0825991661898965]	Parâmetro 14: [0.017399115799055242, 0.0341051385569888]	Parâmetro 15: [0.0830877016997886, 0.08304582639195181]	Parâmetro 16: [0.07546314441346764, 0.06963222359510693]	Parâmetro 17: [0.08785105179872042, 0.0985921451800363]	Parâmetro 18: [0.08611783212949338, 0.09355277736970888]
1	1,372787971	1,173737418	1,167464288	1,531596209	1,418264469	1,276568016
2	1,540978857	1,347474837	1,334928577	1,724702515	1,87480705	1,553136032
3	1,599392984	1,521212255	1,611466292	1,875467422	1,903543601	1,398613908
4	1,842707934	1,694949673	1,63605321	2,11476433	1,839287766	1,563235015
5	1,983403121	1,867505282	1,859909758	2,290853845	2,103465825	1,731514073
6	1,965792176	1,984877617	2,067976081	2,692137242	2,146881841	1,900412358
7	2,137278233	2,097048168	2,171237609	2,726688419	2,245556283	2,069310644
8	2,492482847	2,185559413	2,265294314	2,929380955	2,348025847	2,237503568
9	2,505491116	2,274070658	2,610944875	3,054554794	2,578713232	3,044655611
10	2,81975782	2,362581903	2,712360826	3,149524693	2,614613369	3,397345239
11	3,11624129	2,451093148	2,813776777	3,173881748	2,536269657	3,275140376
12	3,356399064	2,539604393	2,976480508	3,731365522	2,683536239	3,468893149
13	3,600147509	2,628115638	3,580684586	4,12538278	3,648656418	3,798002064
14	4,380845194	2,716628883	4,032761459	4,858070445	3,555700013	4,026176405
15	4,866035284	2,805138128	4,405044389	5,914025662	3,976284385	4,537121382
16	5,13627417	2,893649372	4,589615579	6,704044143	4,353111782	5,883500676
17	5,858829399	3,043845596	4,891359442	7,644895849	4,503207281	6,131751976
18	6,541349451	3,224179852	6,03802907	8,418915101	5,180301816	7,162951787
19	7,161827476	3,608226646	7,277362993	8,933050761	6,252576704	7,457523335
20	7,580930841	4,000946939	7,587911445	9,477227792	6,189185816	8,01339962
21	8,169015252	4,172730489	8,389732458	10,1647216	6,273360659	8,881505794
22	9,100672096	4,344266811	9,035118875	11,59520674	7,144425563	9,548966473
23	9,415755863	4,515803134	9,357897407	11,91684292	7,286350716	10,91356054
24	9,768187788	4,687339457	9,89411193	13,05022216	7,971145504	11,42297184
25	10,90473779	4,930082083	10,80947663	13,81294016	8,674460067	10,73909396
26	11,46197574	5,391240289	11,47524189	14,11119771	9,55608291	10,94761414
27	11,36804579	5,218147803	11,3045138	14,26378038	10,48947559	11,69129617
28	11,3021177	4,982883565	11,11719353	13,60589397	10,23297052	11,06793372
29	10,9806377	4,883284682	11,22737923	13,28852801	10,14902555	11,84497198
30	11,05137833	4,791400172	10,74306068	13,48705795	9,662737235	11,71030376
31	10,7451087	4,699515662	10,40131601	13,13257482	9,940116286	11,68595903
32	10,03096007	4,607631153	9,764054604	12,07486741	9,131339691	10,45080208
33	8,956084825	4,515746643	9,127424979	11,03587055	7,716354328	11,11010743
34	8,705576597	4,423862133	8,37855088	10,12878709	6,790101444	9,76930775
35	8,186890849	4,331977623	8,008348011	9,904696273	6,914821	10,3839859
36	7,722587265	4,240093113	8,159763947	10,00227473	6,555350041	8,565018376
37	7,33394039	4,148208603	7,100700026	9,600566209	5,614640092	7,742259594
38	7,217269722	4,077927575	6,677197862	9,385910687	5,607913812	7,457719788
39	7,415559786	4,035554763	6,642222046	9,799878942	6,285501288	7,976969605
40	7,555555556	4	6,166666667	9,571428571	6,25	8

Tabela 5 – Tabela com os parâmetros de 13 a 18 e suas respectivas séries médias

Fonte: Autor

TEMPO	Parâmetro 19: [0.09269246127912027, 0.095311084956461614]	Parâmetro 20: [0.1062603737644523, 0.10410598591914483]	Parâmetro 21: [0.07987343161815483, 0.04990865437932415]	Parâmetro 22: [0.05034814480209794, 0.0161149616087683]	Parâmetro 23: [0.057946978501754044, 0.017456502406577404]	Parâmetro 24: [0.04928101978936014, 0.01464552459389336]
1	1,202323697	1,560997009	1,118077261	1,348933083	1,080117831	1,138468348
2	1,404647393	1,615661193	1,236154722	1,697866165	1,160235661	1,276936696
3	1,60697109	1,670325376	1,354232083	2,031763616	1,240353492	1,415405044
4	1,749362581	1,888390009	1,472309444	2,268591705	1,320471323	1,553873392
5	1,881685101	2,026713293	1,590388804	2,505419794	1,400589153	1,692341741
6	2,013953622	1,799715383	1,708464165	2,742247883	1,480706984	1,830810089
7	2,146249142	2,10129046	1,826541526	2,979075971	1,560824815	1,969278437
8	2,278544663	2,2807523	1,944618887	3,300360091	1,640942645	2,093944721
9	2,504941007	2,501921583	2,482811498	3,62971591	1,721060476	2,214675659
10	3,080561914	2,590720369	3,3494063	3,95912573	1,801178306	2,335406596
11	3,237057215	2,689292268	3,849501657	4,70615419	1,906146667	2,456137534
12	3,18818263	3,075779701	4,179675675	5,145786765	2,040968876	2,576688471
13	3,605743048	3,663133714	4,436700098	5,375187109	2,175791086	2,697599408
14	3,984204795	3,99555987	4,693724521	5,604587453	2,310613295	2,818330346
15	4,47593925	4,100000279	4,950748944	5,833987797	2,445435505	2,939061283
16	5,544281704	5,761513433	5,521150328	6,721376932	2,713060189	3,078104951
17	5,911172046	7,183708033	6,142182833	8,37956791	3,033464222	3,235812488
18	6,020499606	8,42694266	6,69491858	9,232820249	3,37466828	3,393520025
19	7,146634503	9,530576562	7,40144482	11,0574077	3,739644764	3,551227562
20	6,480392405	10,3061163	8,554326811	11,71772471	4,16307235	3,708935099
21	7,653023102	10,6061114	10,16899258	13,87704926	4,768622141	3,866642637
22	7,689960995	10,41138415	10,50648441	14,98199378	5,378962825	4,197135319
23	7,82516364	11,88364701	10,84397623	14,37332308	5,157102145	5,648049331
24	9,114387422	12,50400608	11,96804605	14,6376533	5,544094035	6,127540245
25	9,264189727	13,24201492	12,57157132	15,5526655	5,704183754	6,287281861
26	10,32027676	13,85478032	12,86781313	16,40545232	5,684596516	6,447023478
27	10,93232587	14,91685043	12,36661322	16,57690193	5,815615889	6,606765094
28	12,21842371	14,79791967	11,13648518	16,27160609	5,275220771	6,766506711
29	11,59818686	14,78923373	11,99109185	16,99950833	5,36493026	6,926248327
30	10,59455771	13,4048619	9,748208214	17,48649906	5,46353175	6,372310992
31	10,7327602	12,34826006	9,360726425	16,56561342	5,195601821	5,803904087
32	10,15606175	11,36260199	9,613917108	14,30107943	4,927671893	5,515826785
33	9,40567115	11,33381525	9,873436498	12,59341814	4,750096644	5,227749484
34	9,19583977	10,59824323	8,908647818	12,62684681	4,617243884	5
35	8,48218312	11,20617132	7,031036792	12,60049362	4,484391124	5
36	8,331743904	10,75490445	7,414616339	12,06149801	4,351538363	5
37	6,92347587	10,69984615	7,798195886	10,83754339	4,218685603	5
38	6,630617856	10,60562079	8,945662625	10	4,085832843	5
39	7,407455267	10,00455617	8	10	4,001050139	5
40	8,5	9,75	8	10	4	5

Tabela 6 – Tabela com os parâmetros de 19 a 24 e suas respectivas séries médias

Fonte: Autor



5 CONCLUSÃO E DISCUSSÕES

Este capítulo apresenta as principais conclusões e discussões sobre o modelo estocástico desenvolvido para prever eventos de sobrecusto em obras de infraestrutura. Desenvolvido com o objetivo de contribuir para o planejamento de novas obras, este estudo fornece informações que podem auxiliar no gerenciamento e antecipação de eventos de sobrepreço. A seguir, são discutidos os resultados obtidos a partir do modelo gerado, suas aplicabilidades e limitações.

Ao fornecer previsões de possíveis cenários de crescimento ou diminuição dos eventos de gastos e sobrepreço, o modelo estocástico proposto pode ser aplicado na fase de planejamento de novas obras de infraestrutura, possibilitando identificar os instantes em que há maior probabilidade de ocorrência desses eventos. Isso permite aos engenheiros se antecipar ou se prevenir contra tais sobrecustos.

Essa antecipação aos eventos de sobrecustos torna-se possível uma vez que, a partir do modelo calculado, é possível realizar a análise dos diferentes cenários no decorrer da obra. Como resultado do trabalho, foram obtidos 24 modelos utilizáveis. Dentre esses, 13 apresentam uma perspectiva mais pessimista, com aumento dos eventos de sobrepreço de maneira abrupta e alcançando um valor absoluto maior (**Cluster 1**). Outros 6 modelos gerados são considerados otimistas, com uma distribuição mais gradual desses eventos e valores absolutos inferiores (**Cluster 3**). Também, foram gerados 5 modelos considerados neutros (**Cluster 2**), com desempenho que se encontra entre os outros dois clusters apresentados.

Com isso, pode-se adotar modelos mais pessimistas no início de uma obra, podendo ser substituído por modelos mais otimistas conforme o seu avanço, caso este seja o cenário real. Tal fato possibilita, aos profissionais envolvidos no empreendimento, estarem preparados e adotarem medidas adequadas para mitigar os impactos financeiros causados por esses sobrecustos.

Além disso, analisando os principais comportamentos dos modelos, foi observado que o valor máximo dos eventos de sobrecusto se concentra entre os períodos equivalentes a 50% e 75% da duração total da obra. Os cenários mais otimistas indicam um crescimento mais bem distribuído ao longo dessa duração, enquanto que os cenários mais pessimistas apontam um crescimento acentuado, a partir da fase equivalente a 15% do tempo de obra. Quanto a esses valores absolutos, os cenários pessimistas indicam que, de 40 fases de



uma obra, entre 10 e 15 delas haverá sobrecustos. Os cenários neutros indicam por volta de um máximo de 10 eventos e, os cenários otimistas, indicam valores máximos entre 5 e 10 eventos. Essas informações são essenciais para um bom planejamento e alocação de recursos adequados em cada fase da obra.

Se comparado a métodos tradicionais de estimativa de sobrepreço, a principal diferença reside no fato de que alguns desses métodos, como comparação com projetos similares e análise paramétrica, podem apresentar informações enviesadas ou dificuldades na determinação dos parâmetros. O modelo estocástico proposto, por sua vez, utiliza informações de obras semelhantes, mas com a adição de variabilidades. A adição dessas imprevisibilidades ao planejamento é o que permite, ao gerente da obra, se precaver perante cenários pessimistas que não tenham um histórico de ocorrência.

Quanto à praticidade de utilização do método por esses mesmos gerentes de obra e planejamento, é importante destacar que, com um conhecimento básico na linguagem Python e uma base de dados adequada à obra em questão, é possível utilizar o código desenvolvido neste trabalho. Isso torna o uso do modelo relativamente fácil, desde que haja profissionais com as habilidades necessárias para trabalhar com a linguagem de programação.

No entanto, algumas limitações foram identificadas. A principal delas consiste na necessidade de possuir uma base de dados, completa o suficiente para uma análise de maior qualidade. Além disso, é importante ter alguém com conhecimento em Python para utilizar o código criado, desde a leitura da respectiva base, assim como para adequá-lo às necessidades intrínsecas ao projeto estudado, seja exigindo por mais simulações de séries ou necessitando de mais clusters para o agrupamento final. Essas limitações podem ser superadas com treinamentos e aprimoramentos técnicos, bem como com a disponibilidade de dados adequados.

Trabalhos futuros podem explorar novas aplicações do modelo, aprimorando sua integração com ferramentas de gerenciamento de projetos, de forma a melhorar a eficiência do planejamento e o controle de custos em obras de infraestrutura.



REFERÊNCIAS BIBLIOGRÁFICAS

- AGÊNCIA CBIC. **CBIC reduz expectativas de crescimento do setor**. Disponível em: <<https://cbic.org.br/cbic-reduz-expectativas-de-crescimento-do-setor-esclarece-economista/>>. Acesso em: 21 abr. 2023.
- AGHABOZORGI, S.; SHIRKHORSHIDI, A. S.; WAH, T. Y. **Time-series clustering—a decade review**. *Information systems*, v. 53, p. 16-38, 2015.
- ASSED, J. A. **Construção civil: viabilidade, planejamento, controle**. Rio de Janeiro: Livros Técnicos e Científicos, 1986.
- BAGNALL, A. et al. **The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances**. *Data mining and knowledge discovery*, v. 31, p. 606-660, 2017.
- BARBARÁ, D. **PIB da construção cresceu 6,9% em 2022**. SINDUSCON-SP, 02 mar. 2023. Disponível em: <<https://sindusconsp.com.br/pib-da-construcao-cresceu-69-em-2022-2/>>. Acesso em: 10 maio 2023.
- BEALE, E.M.L... **On Minimizing a Convex Function Subject to Linear Inequalities**. Julho 1955. Disponível em: <<https://doi.org/10.1111/j.2517-6161.1955.tb00191.x>> Acesso em: 18 maio 2023.
- BISSET, C.; TERBLANCHE, S.E... **A Systematic Literature Review on Stochastic Programming Models in the Retail Industry**. Publicado On-line em: 12 jan 2022. Disponível em: <https://hdl.handle.net/10520/ejc-indeng_v32_n3_a10> Acesso em: 30 de maio, 2023.
- CARDOSO, G.F.; SOUZA G.S.; CARVALHO L.F.; RIBEIRO K.C.S... **Análise estocástica de uma empresa do setor siderúrgico**. *Revista Ciências Administrativas*, v. 24, n. 1, p.1-12, 2018. Universidade de Fortaleza. Disponível em: <<https://ojs.unifor.br/rca/article/view/6855/pdf>>. Acesso em: 12 de abril, 2023.
- CHIAVENATO, I. **Introdução à Teoria Geral da Administração**. 3a. ed. São Paulo: McGraw-Hill Brasil, 1983.
- COSTA NETO, A. L.; BRIM, J. V.; AMORIM, P. H. M. **Estudo de um modelo para análise prévia de viabilidade econômico-financeira de empreendimentos imobiliários em Salvador**. Trabalho de Conclusão de Curso (Especialização) - Universidade Federal da Bahia, Salvador, 2003.
- DEPARTAMENTO INTERSINDICAL DE ESTATÍSTICA E ESTUDOS SOCIOECONÔMICOS - DIEESE. **Estudo Setorial da Construção 2012 - Nº 65 - Maio de 2013**. São Paulo, 2013. Disponível em: <<https://www.dieese.org.br/estudossetorial/2012/estPesq65setorialConstrucaoCivil2012.html>>. Acesso em: 13 abr. 2023.
- FERREIRA, I.M.N.; CAMPOS NETO, C.A.S... **Infraestrutura Econômica no Brasil: diagnósticos e perspectivas para 2025**. Livro 6 – Volume1 IPEA – Instituto de Pesquisa



Econômica Aplicada, 2010. <Disponível em: <<https://repositorio.ipea.gov.br/handle/11058/3210>>. Acesso em 23 jun. 2023.

FERREIRA, P.C... **Investimento em infraestrutura no Brasil: fatos estilizados e relações de longo prazo** – Pesquisa e Planejamento Econômico (PPE), Rio de Janeiro, v 26, n 2, p 231-252 ago. 96. Instituto de Pesquisa Econômica Aplicada (Ipea), 1996.

FLYVBJERG, B. **Policy and planning for large-infrastructure projects: problems, causes and cures**. Environment and Planning B: planning and design, v. 34, p. 578-597, 2007.

FONTANA, É.; PEDERSEN VOLL, F. A. **Conceitos Básicos de Programação em Python**. Universidade Federal do Paraná, Departamento de Engenharia Química, 2020/1.

FORMOSO, T. C. **Planejamento e controle da produção em empresas de construção**. Porto Alegre: Universidade Federal do Rio Grande do Sul, 2001.

GOLDMAN, P. **Introdução ao planejamento e controle de custos na construção civil brasileira**. 4. ed. atual. São Paulo: Pini, 2004.

GOMIDE, A.A.; PEREIRA, A.K... **Governança da política de infraestrutura: condicionantes institucionais ao investimento** – Instituto de Pesquisa Econômica Aplicada (Ipea) –2018. Disponível em: <<http://repositorio.ipea.gov.br/handle/11058/8482>>. Acesso em 2, jun 2023

GOMIDE, A. de Á.; PEREIRA, A. K. **Capacidades estatais para políticas de infraestrutura no Brasil contemporâneo**. Revista de Administração Pública, v. 52, n. 5, set./out. 2018. Disponível em: <<https://doi.org/10.1590/0034-761220170006>>. Acesso em: 20, mai de 2023.

HAUTPHENNE, S.; PATCH, B... **Birth-and-death Processes in Python: The BirDePy Package**. The University of Melbourne, 2021

HELLEMO, L. et al. **Decision-Dependent Probabilities in Stochastic Programs with Recourse**. Computational Management Science, v.15, n.3-4, p.369-395, ago. 2018.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA - IBGE. **Pesquisa anual da indústria da construção**. Disponível em: <<https://www.ibge.gov.br/estatisticas/economicas/industria/9018-pesquisa-anual-da-industria-da-construcao.html?t=noticias-e-releases>>. Acesso em: 21 abr. 2023.

ISMAIL FAWAZ, H. et al. **Deep learning for time series classification: a review**. Data mining and knowledge discovery, v. 33, n. 4, p. 917-963, 2019.

LEITE, M. O.; POSSAMAI, O. **A utilização das curvas de aprendizagem no planejamento da construção civil**. In: ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, 21., 2001, Santa Catarina. Anais [...] Santa Catarina, 2001.

LIMA, V.G.M... **Sobrepço e Superfaturamento de Obras Públicas e Indicadores Educacionais: uma análise de suas relações nos Estados brasileiros**. Universidade de Brasília, Programa de Pós-Graduação em Ciências Contábeis, 2016. Disponível em:



<https://repositorio.unb.br/bitstream/10482/20903/1/2016_VictorGodeirodeMedeirosLima.pdf> Acesso em 25 maio 2023.

MATTOS, A. D. **Como preparar orçamentos de obras: Dicas para orçamentistas, estudo de caso, exemplos.** São Paulo: Editora Pini, 2014.

MATTOS, A. D. **Planejamento e controle de obras.** 1. ed. São Paulo: PINI, 2010.

MELLO, L.C.B.B.; AMORIM S.R. **O subsetor de edificações da construção civil no Brasil: uma análise comparativa em relação à União Europeia e aos Estados Unidos. Produção,** v.19, nº2, p 388-399 maio/agosto2009.

MENON, D. **Análise da viabilidade econômica por meio do fluxo de caixa.** In: Novos Negócios da Tecverde Engenharia – Site Smartus – out. 2020. Disponível em: <<https://smartus.com.br/viabilidade-economica-fluxo-de-caixa-empreendimento/>>. Acesso em: 01 julho 2023.

NAKAMURA, J. **Como fazer o gerenciamento de obras.** Revista Online AU PINE, Edição nº 245, agosto, 2014.

NASCIMENTO, L. M. **Análise do processo de gerenciamento de estoque de materiais na construção civil.** Revista Organização Sistêmica, Curitiba, v. 10, n. 19, p. 3-16, 2021.

O TEMPO Brasília. **Obras paradas no Brasil somam R\$ 27,2 bilhões, informa TCU** Publicado em 26 de novembro de 2022. Disponível em: <<https://www.otempo.com.br/politica/governo/obras-paradas-no-brasil-somam-r-27-2-bilhoes-informa-tcu-1.2772885>>. Acesso em 15 abril, 2023.

PEREIRA, A. K. **Governança da política de infraestrutura: condicionantes institucionais ao investimento.** IPEA. 2018. cap. 8, p. 191.

PETTERS, T... **Zen do Python.** Disponível em <<https://peps.python.org/pep-0020/#the-zen-of-python>>. Acesso em 3 mai. 2023.

PIO, C.; BARBOZA, M. de M.; COSTA, M. E. L.; ANDRADE, R. B.; SILVA, W. B.; CORRÊA, P. M. **Desafios ao aumento do investimento privado em infraestrutura no Brasil** - Relatório de Conjuntura Nº 5 - Presidência da República – Secretaria Geral da Presidência - Secretaria Especial de Assuntos Estratégicos – jun. 2018.

POSSAMAI, M... **Python,** Publicado em Plain English. Disponível em <<https://python.plainenglish.io/guido-van-rossum-the-mind-behind-python-d6ba487e3cf0>>. Acesso em 13 jun.2023

RIBEIRO, M. **Mais Controle** Blog – dezembro, 2016. Disponível em: <<https://maiscontroleerp.com.br/previsibilidade-financeira-em-obras/>>. Acesso em: 02 jun. 2023.

RIBEIRO, L. **Fluxo de caixa de obras: como fazer e por que é importante?** Tecplaner Blog – 2020. <Disponível em: <https://tecplaner.com.br/tecplaner.com.br/fluxo-de-caixa-de-obras/>>. Acesso em: 02 jun. 2023.



RITTER, D. **Lançamento do futuro plano nacional de investimentos em infraestrutura.** CNN Brasil, Brasília, 05 jun. 2023. Disponível em: <<https://www.cnnbrasil.com.br/economia/novo-pac-tera-mais-de-mil-projetos-e-exigencia-de-conteudo-nacional-em-obras/>>. Acesso em: 02 jul. 2023.

SALLABERRY, C. R. **Implementação de um sistema ERP em uma empresa construtora: impactos no processo de aquisição de material.** Monografia (Bacharelado em Engenharia Civil) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2009.

SALOMÃO, P.E.A.; PÊGO, D.P.; RHIS, A.R.; COELHO, S.S.F... **O impacto econômico e social da paralisação das obras públicas de infraestrutura.** Research, Society and Development, vol.8, núm 5,2019.

SCHLINDWEIN, E... **Construção Civil.** Centro Universitário Leonardo da Vinci - (UNIASSELVI). – Indaial: Ed. Grupo UNIASSELVI,2009.

SIEGEL, E. **Análise Preditiva: O Poder de Prever Quem Vai Clicar, Comprar, mentir ou Morrer.** Rio de Janeiro: Alta Books, 2017.

SILVA, I.R.S; SILVA, R.O... **Linguagem de Programação Python.** Revista Tecnologias em Projeção, v10, nº1, ano 2019.

SILVA, J.P. **Análise Financeira das Empresas.** 8. ed. São Paulo: Atlas, 2007.

TAVENARD, R.; FAOUZI, J.; VANDEWIELE, G.; DIVO, F.; ANDROZ, G.; HOLTZ, C.; PAYNE, M.; YURCHAK, R.; RUßWURM,M.; KOLAR,K.; WOODS,E... **Tslearn, A Machine Learning Toolkit for Time Series Data.** Journal of Machine Learning Research, 2020.

TAYLOR, H.M.; KARLIN, S. **An Introduction to Stochastic Modeling.** 3rd Edition, Academic Press, San Deigo, CA., 1998.

TURBAN, E.; VOLONINO, L. **Tecnologia da informação para gestão: em busca do melhor desempenho estratégico e operacional.** Tradução de Aline Evers. 8 ed. Porto Alegre: Bookman, 2013.

UDACITY BRASIL. **Como as empresas usam a análise preditiva na tomada de decisão?** Disponível em: <<https://br.udacity.com/blog/2017/06/analise-preditiva-tomada-decisao/>>. Acesso em: 06 jun. 2023.

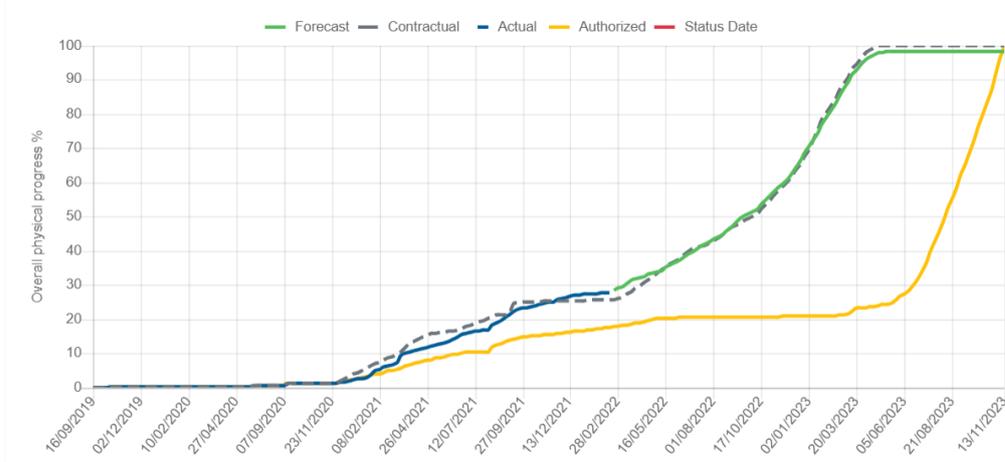
VICENTAINER, D.; MATTEDI, M.; MELLO, B... **Aplicação das bibliotecas Python para tratamento de dados em tempo real.** Outubro, 2020. Disponível em <<https://publicacoes.ifc.edu.br/index.php/metapre/article/view/1392>>. Acesso em 15 mai. 2023

ZDANOWICZ, J.E. **Fluxo de Caixa: uma decisão de planejamento e controle financeiro.** 10.ed. Porto Alegre: Sagra Luzzato, 2004.

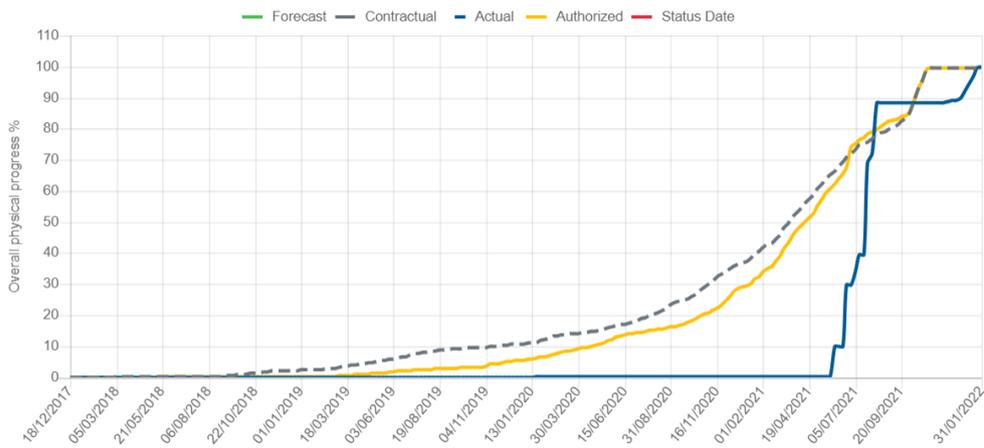


ANEXO A

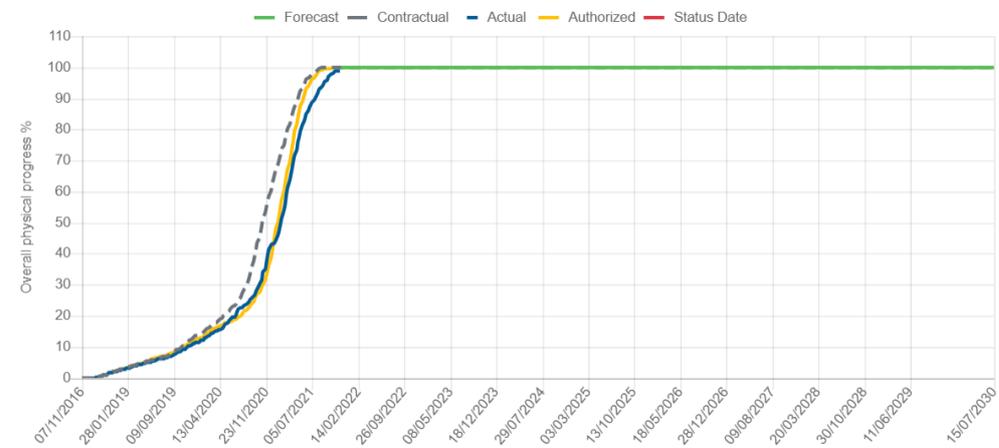
IGUAB5



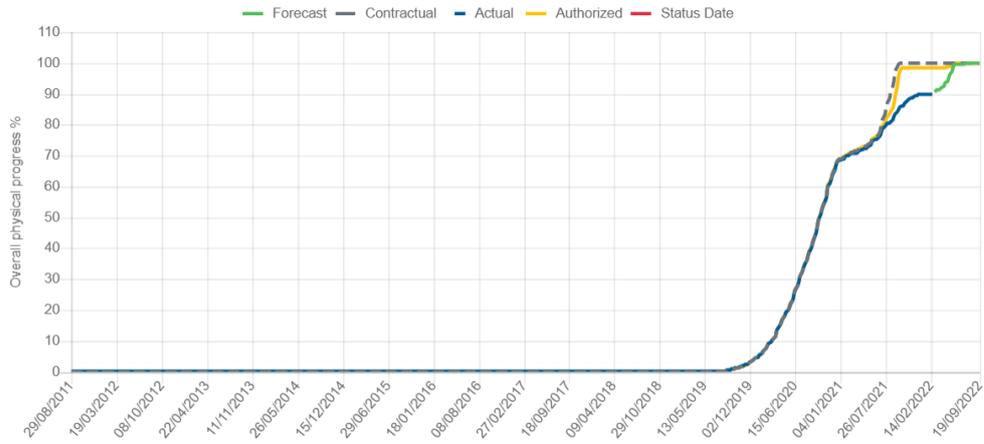
DIAM2



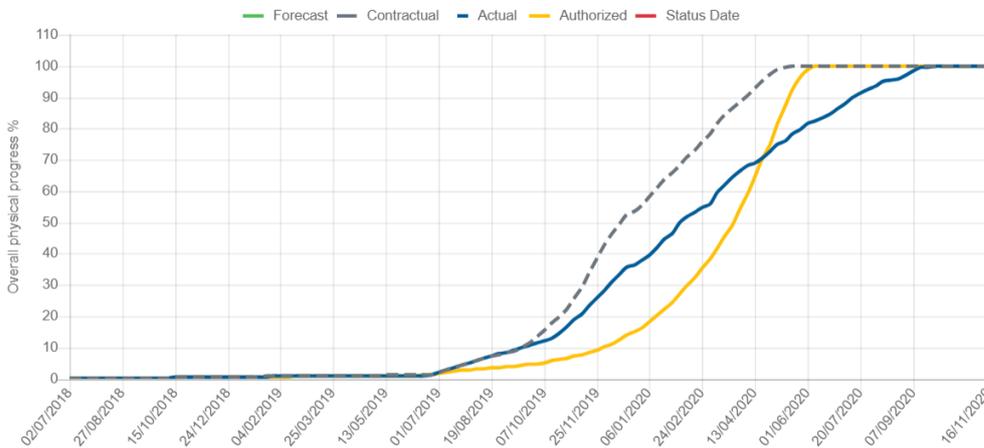
PRAIA1



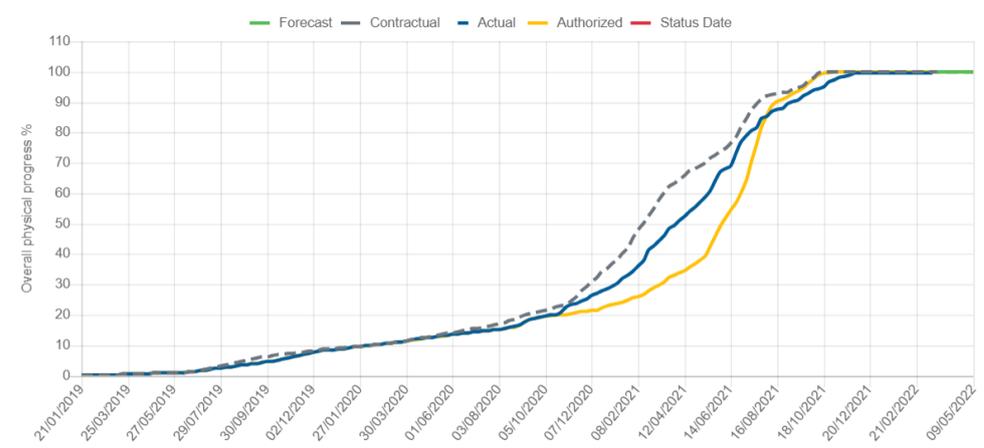
PARTAGE3



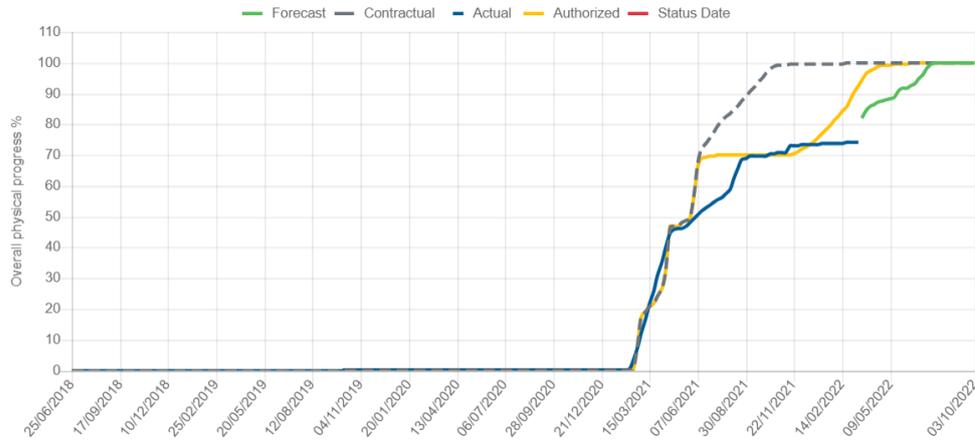
PARTAGE2



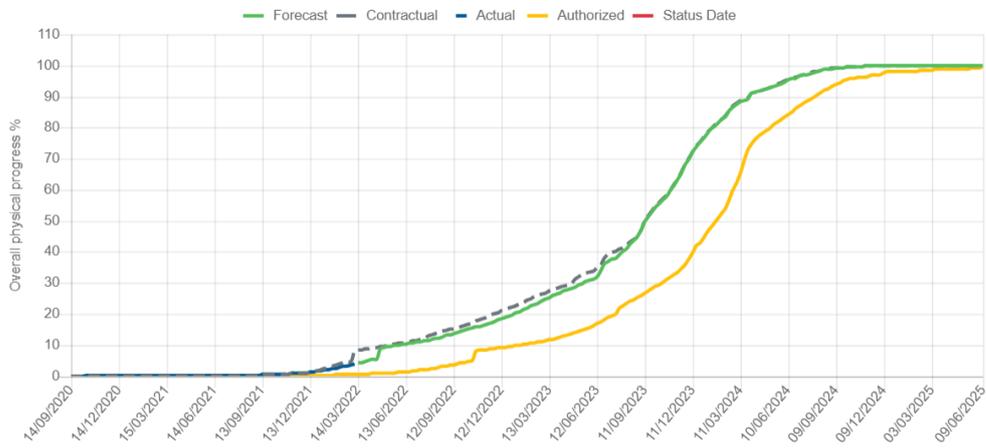
AGUA2



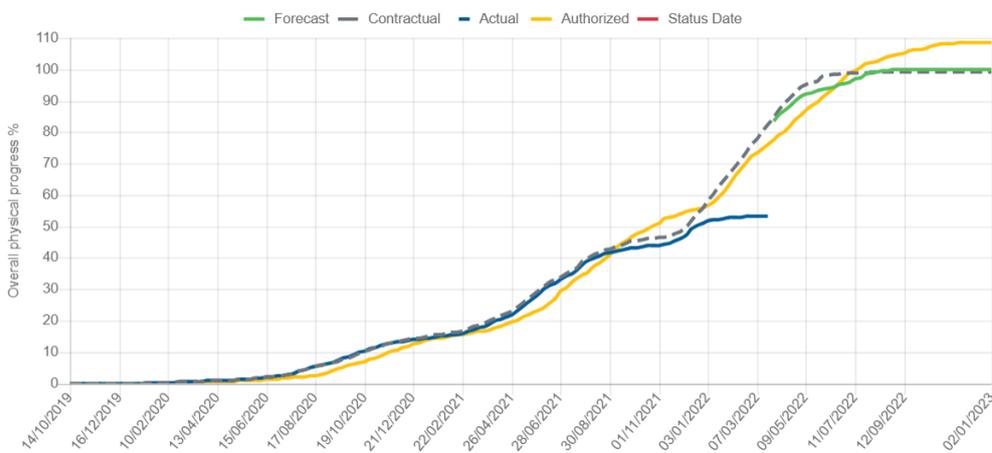
IGUAB3



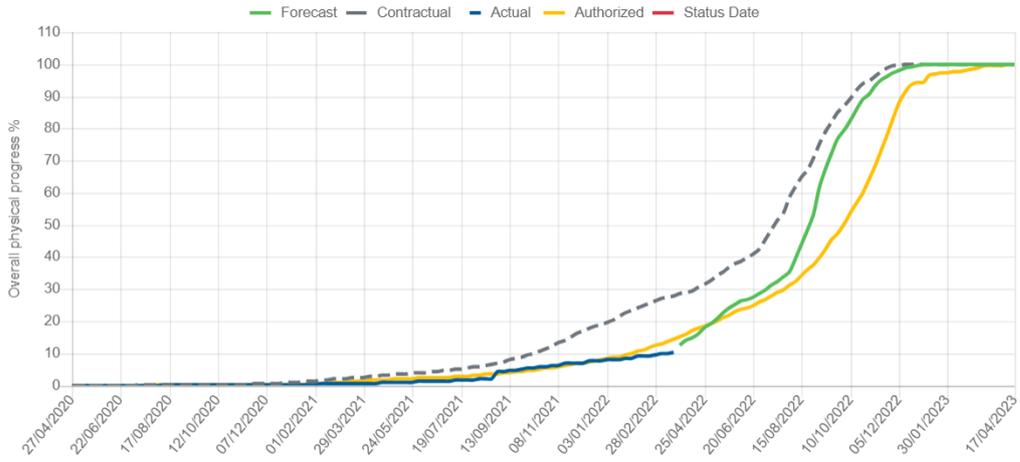
PORTINARI1



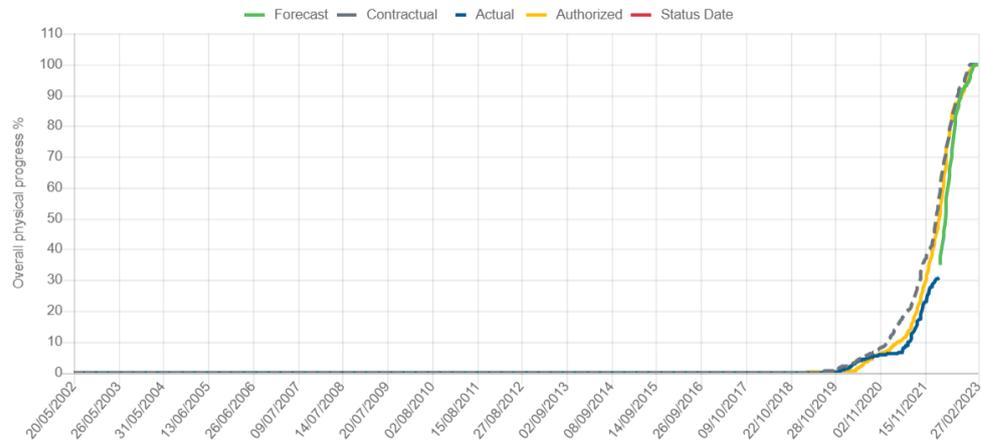
LALOM1



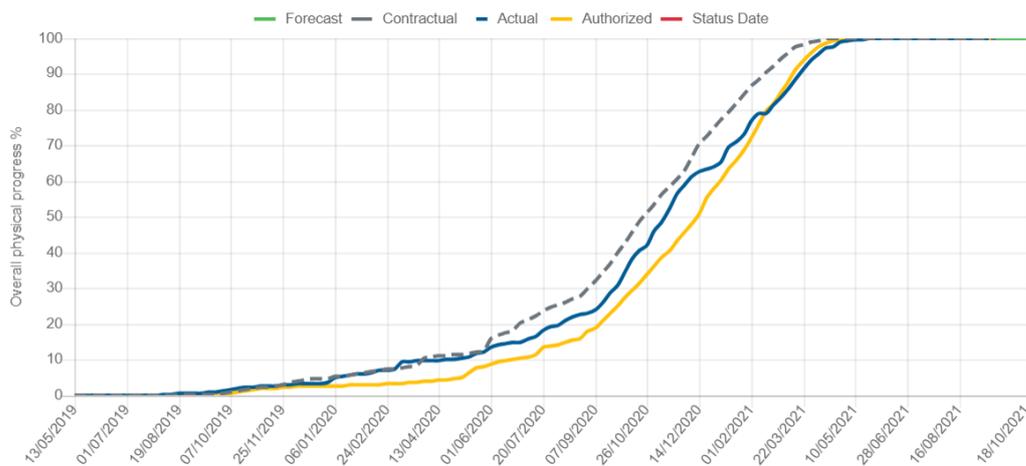
ELMAN1



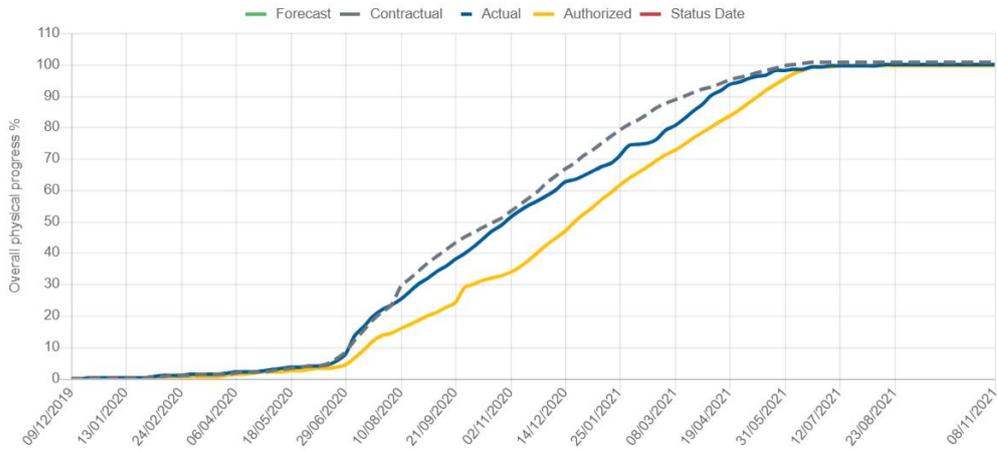
SIGOR1



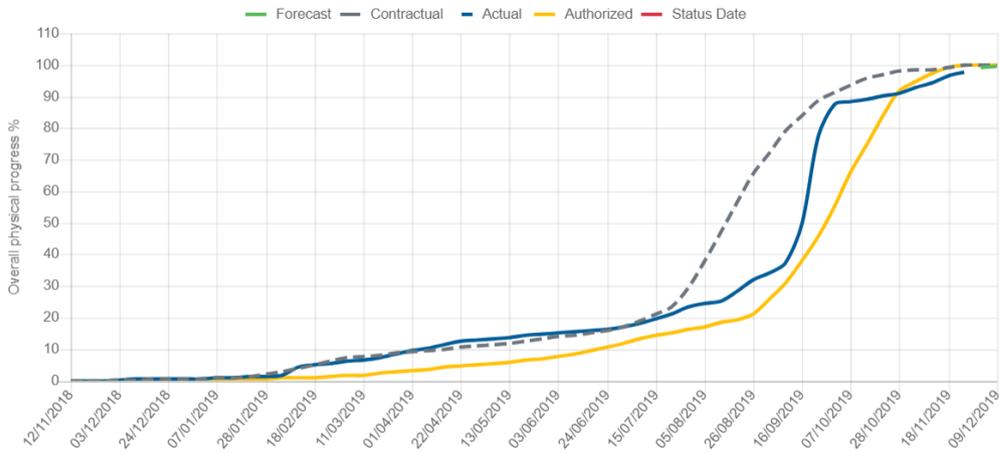
LISOL1



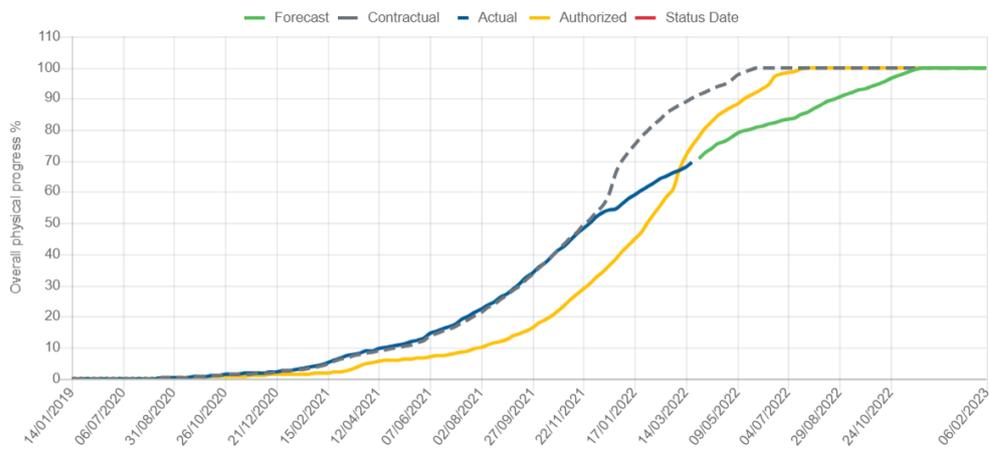
AZURE1



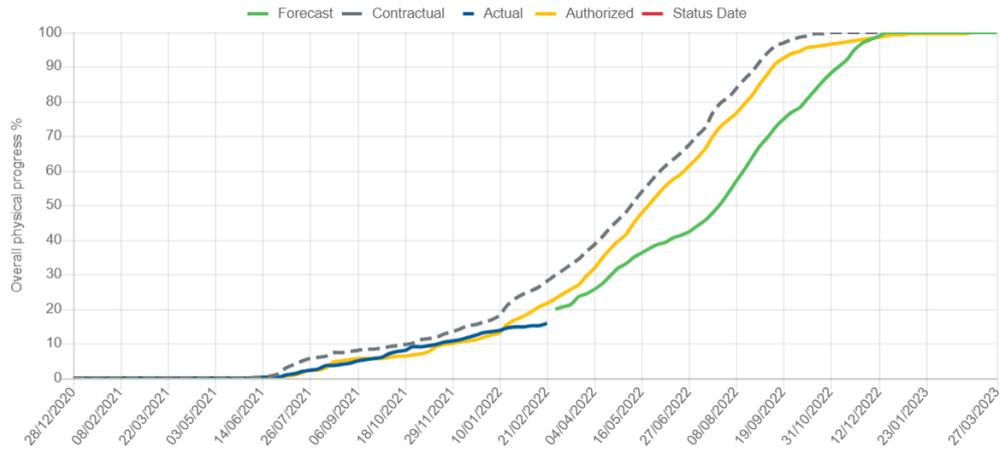
WHILL1



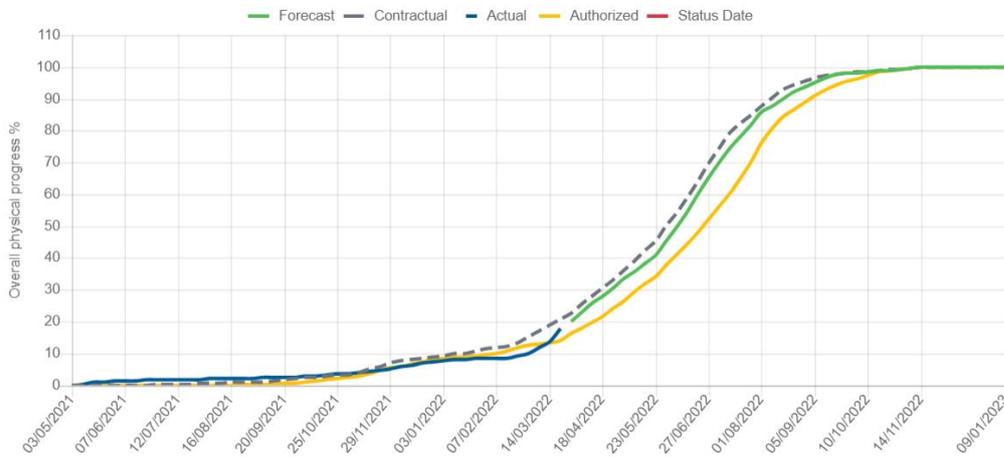
ROSELA1



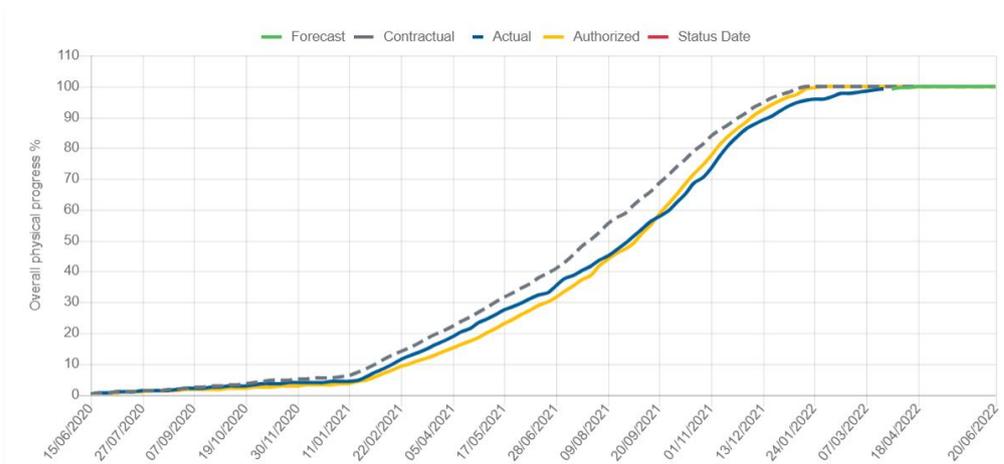
FENCE1



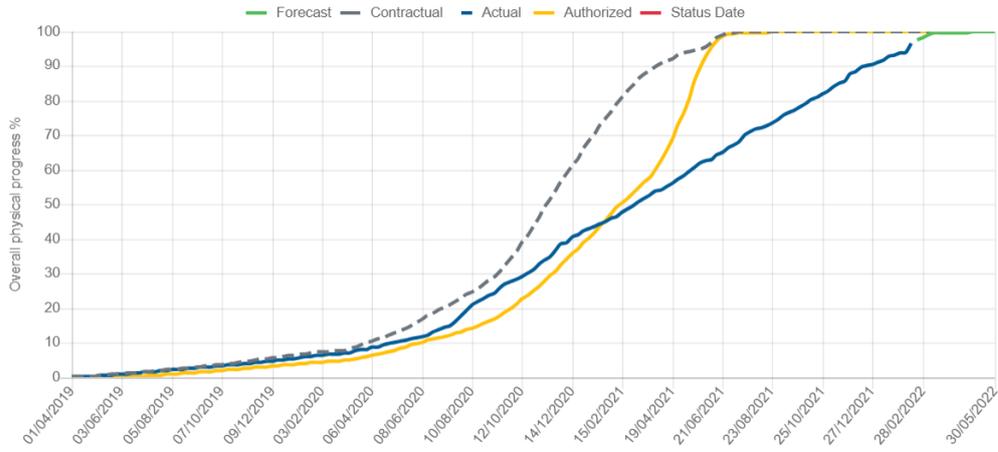
SEVCO1



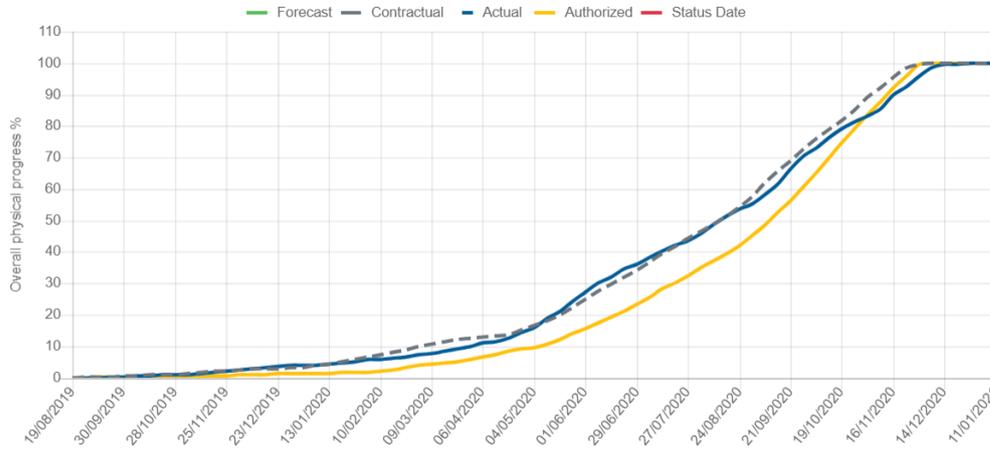
RACHA1



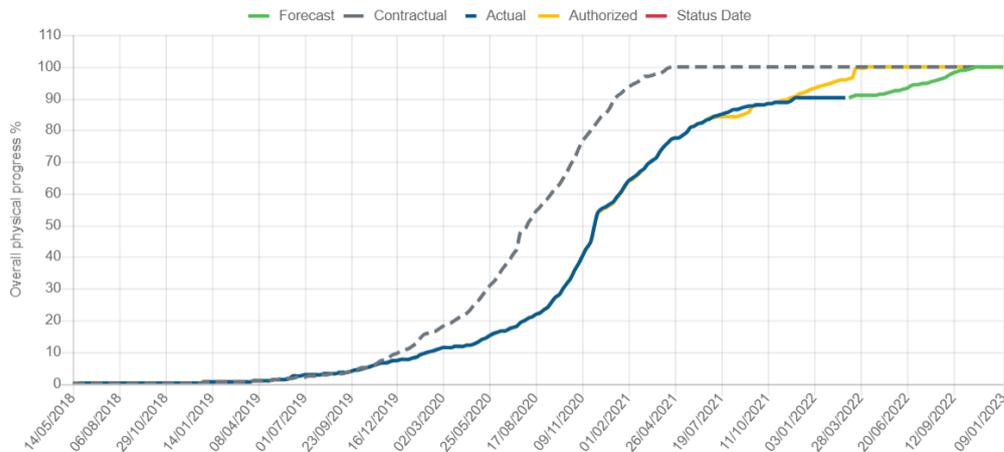
DOMEY2



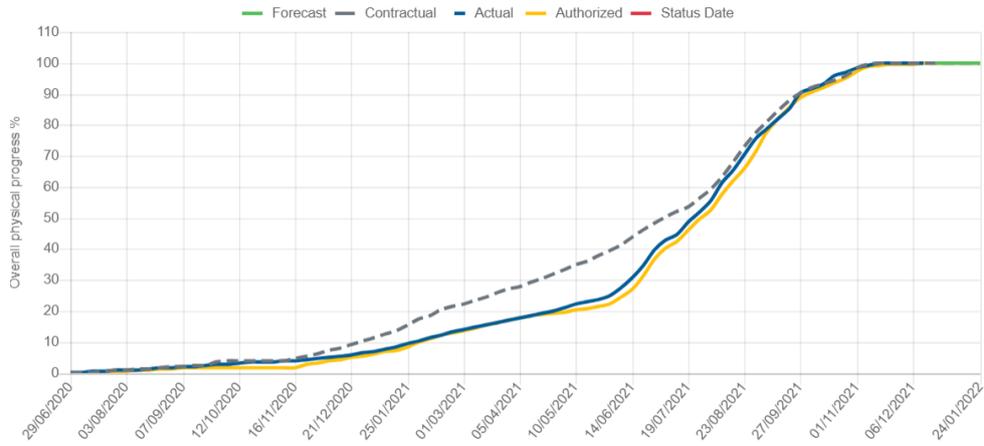
CIMAR3



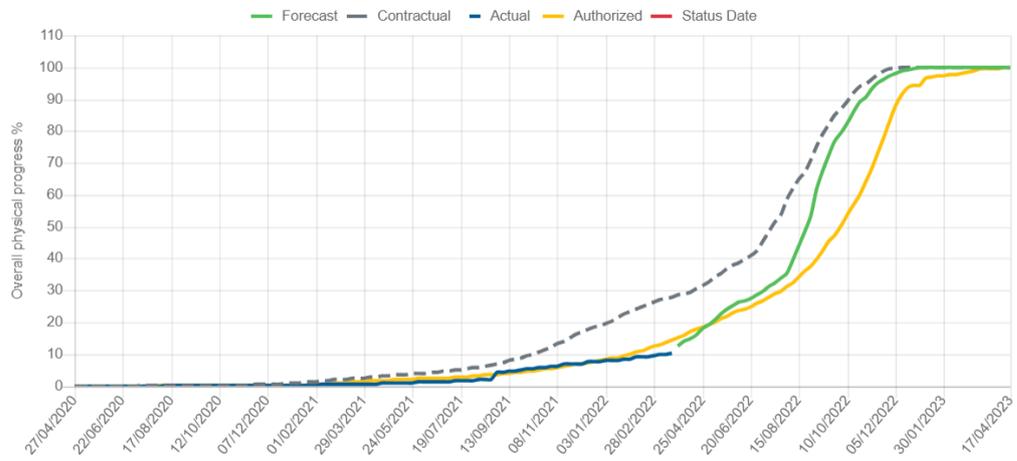
REINO2



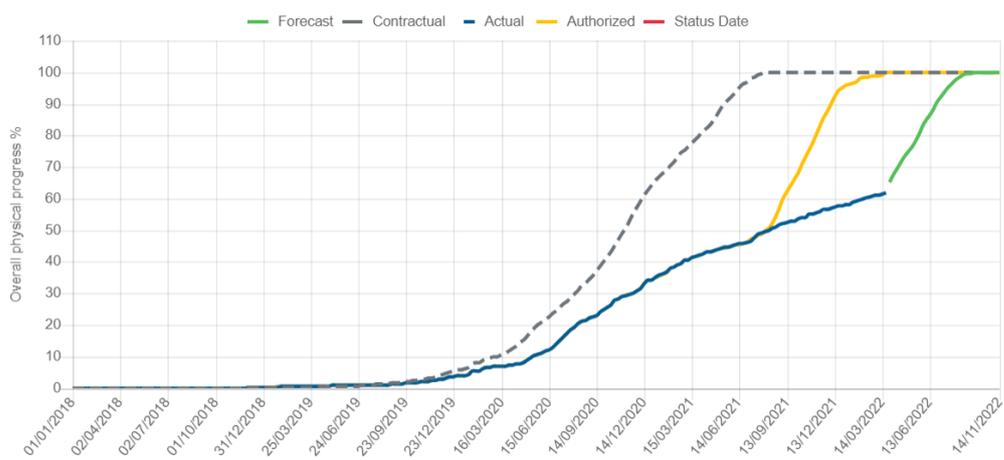
ROCK1



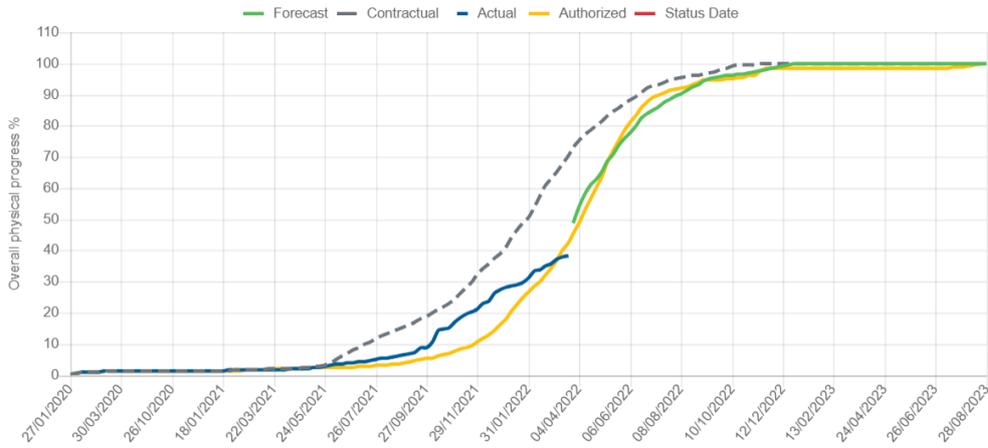
ELMAN1



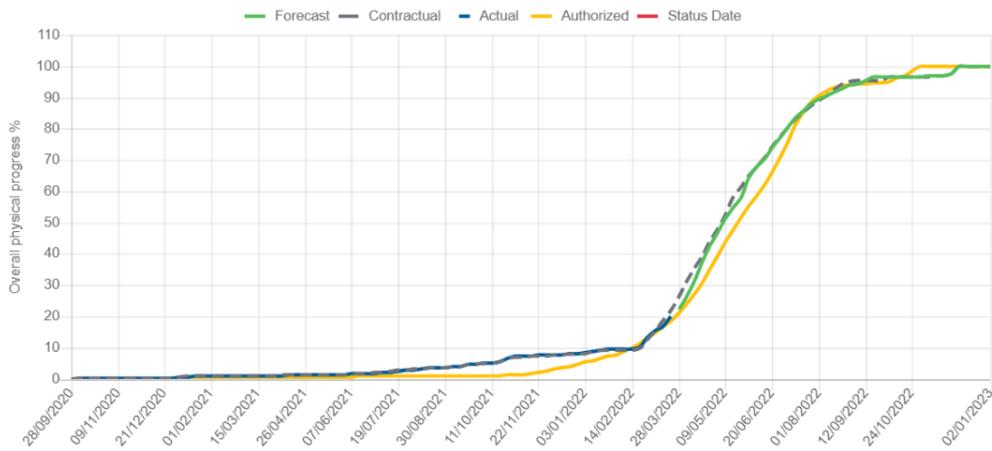
FINTE1



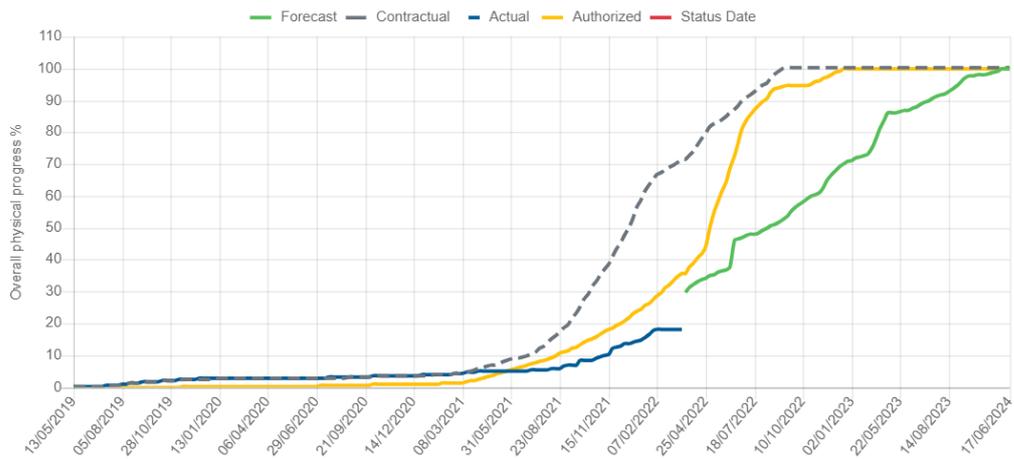
CAMPI2



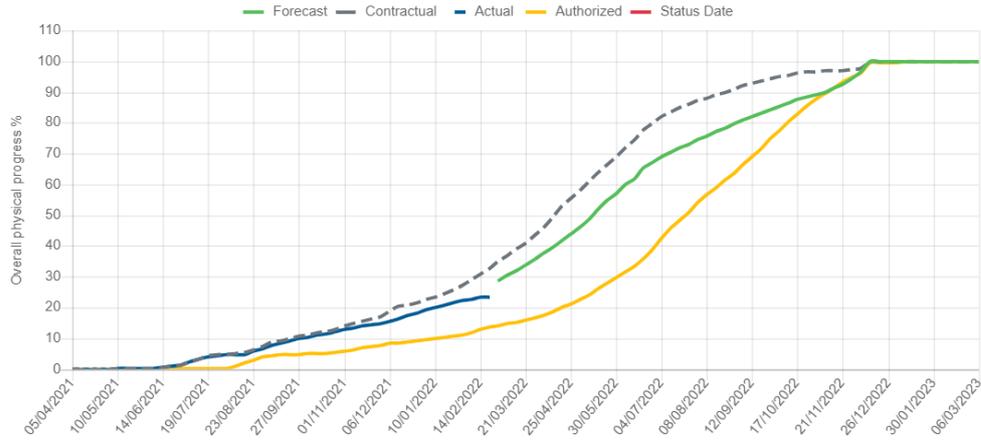
MINGL1



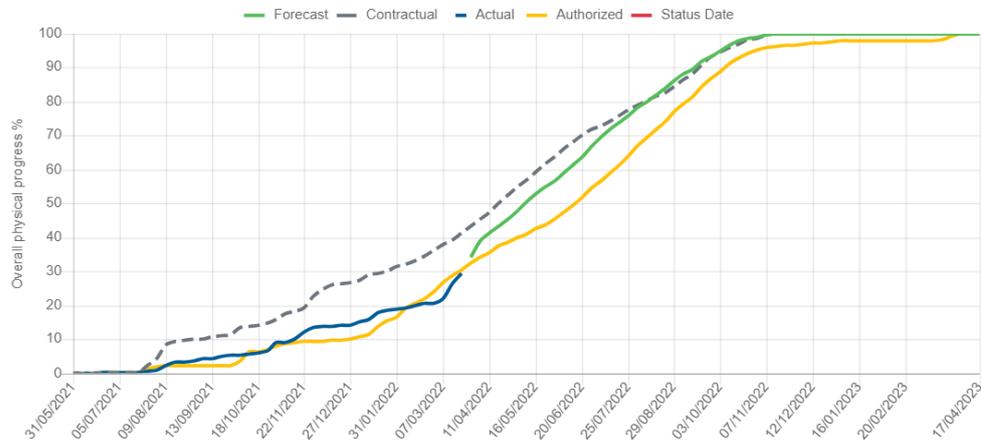
POLIZ1



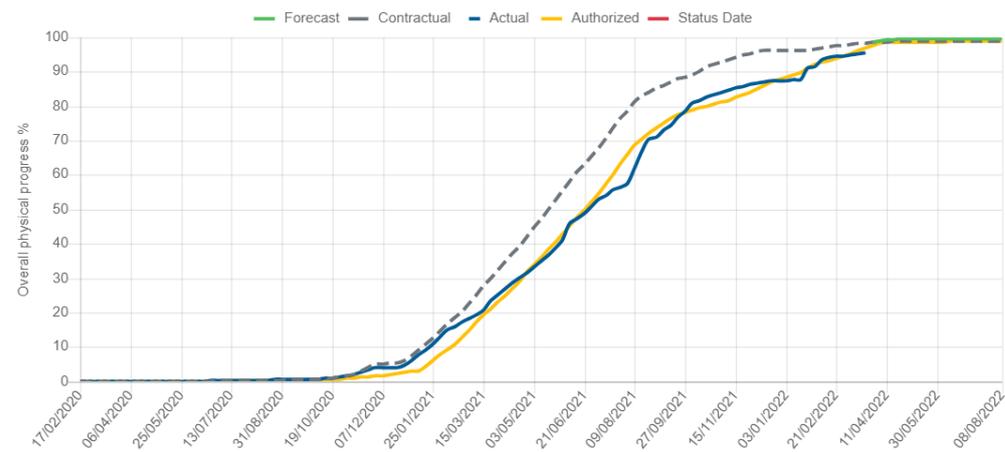
STAMP1



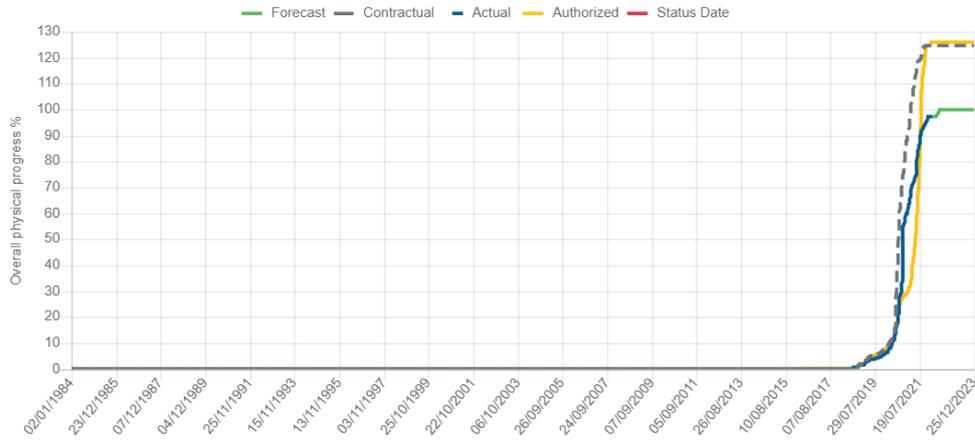
GANAD1



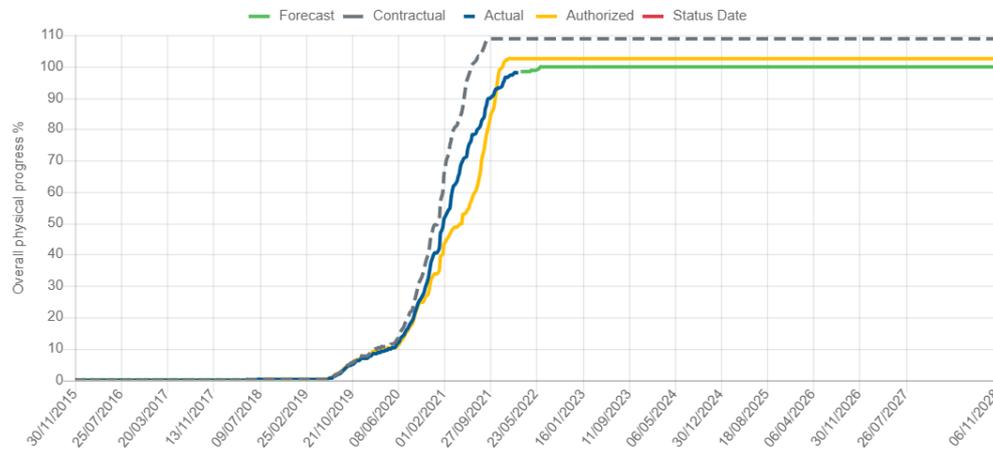
TICOC1



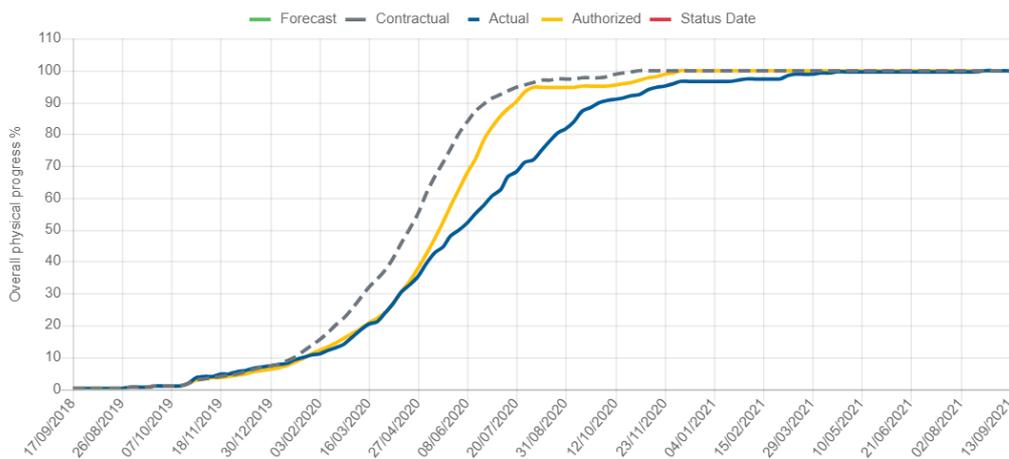
KARUS1



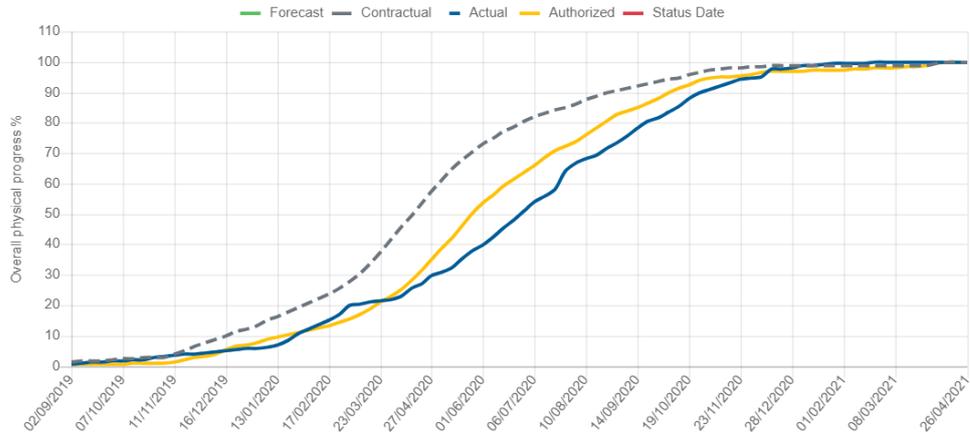
SOETW1



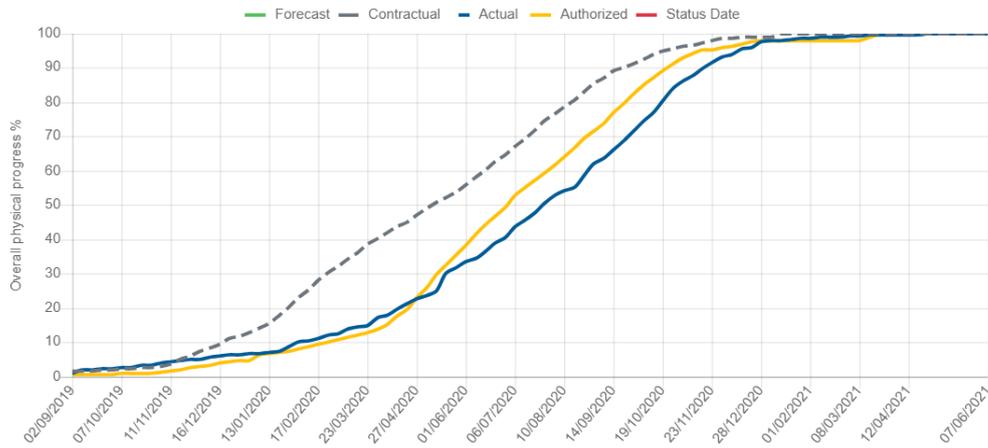
AUGUS1



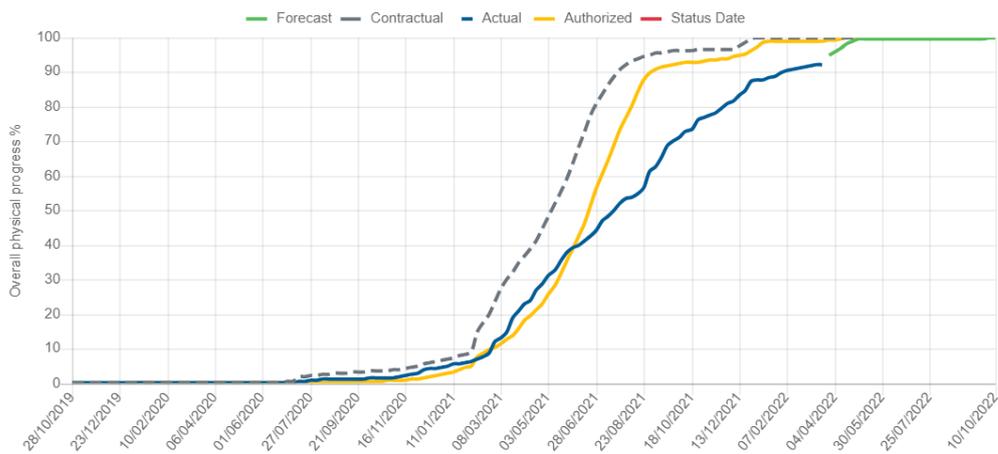
LASCO1



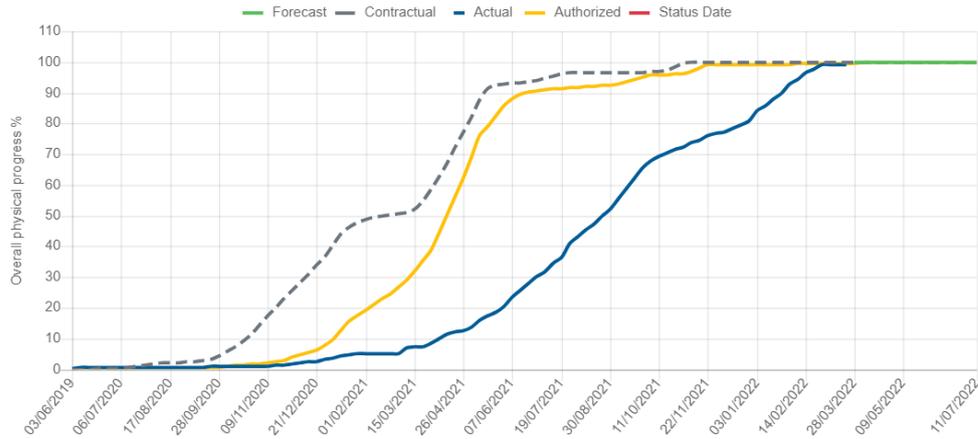
LOSNA1



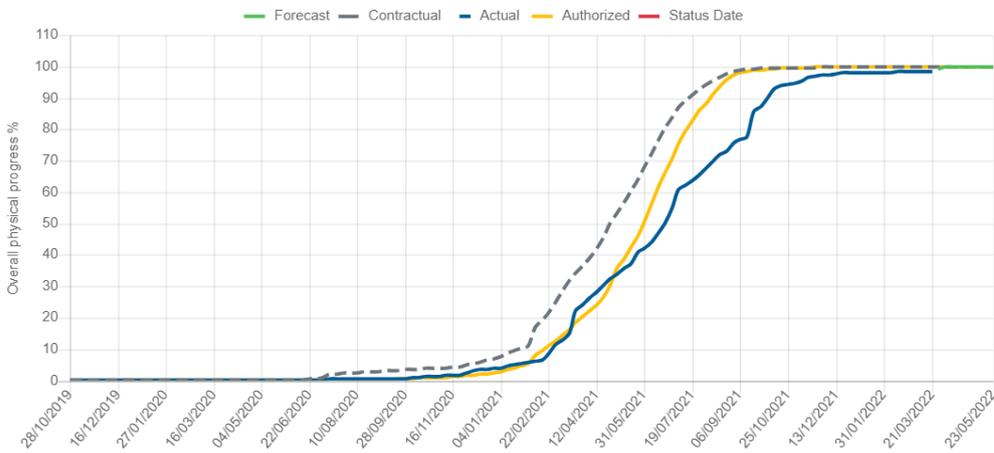
NERTO1



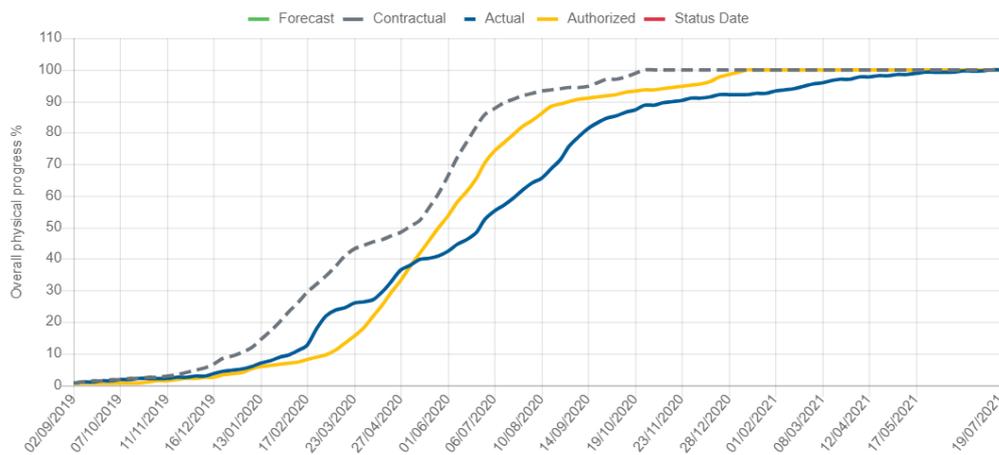
PORTA1



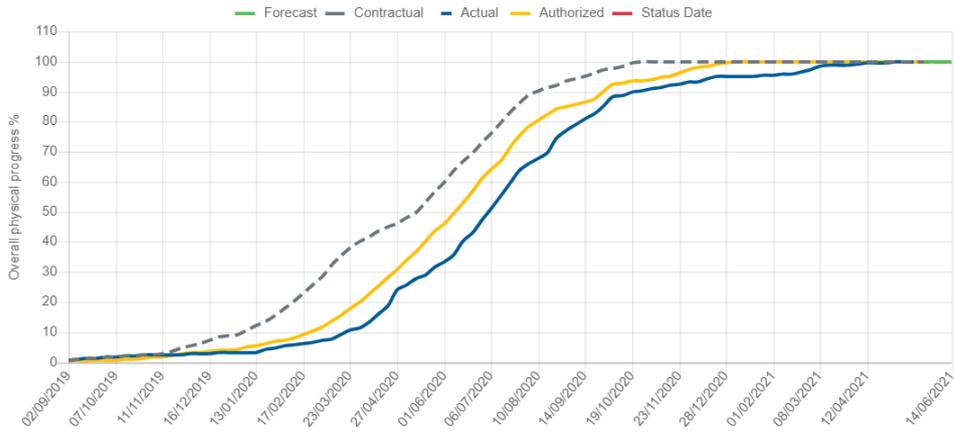
APICI1



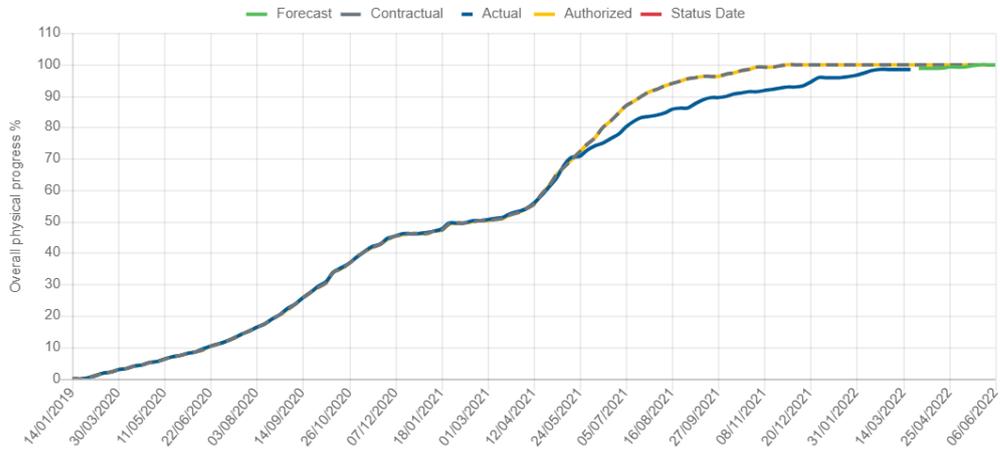
LAVEG1



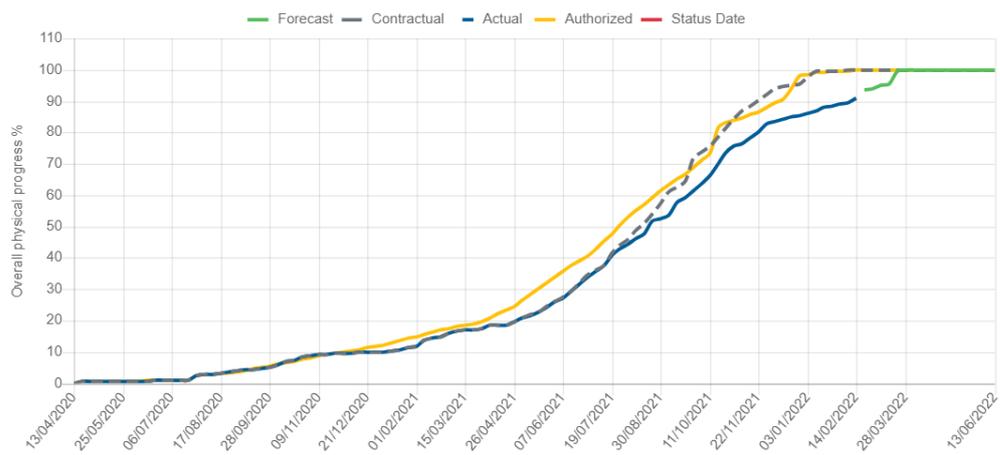
LAVEG2



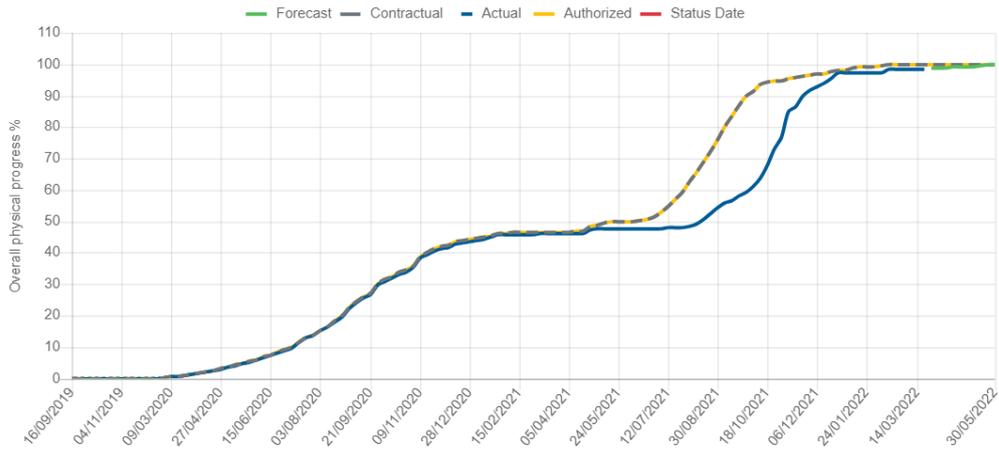
SANTO1



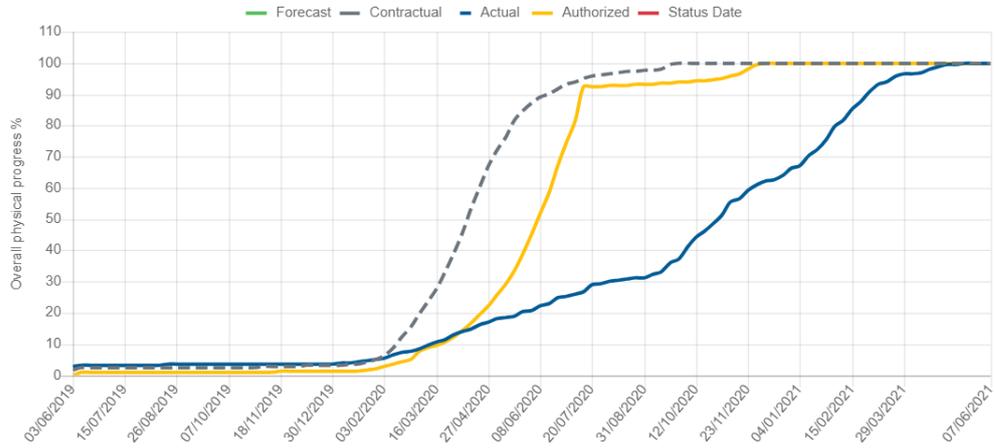
PARAN1



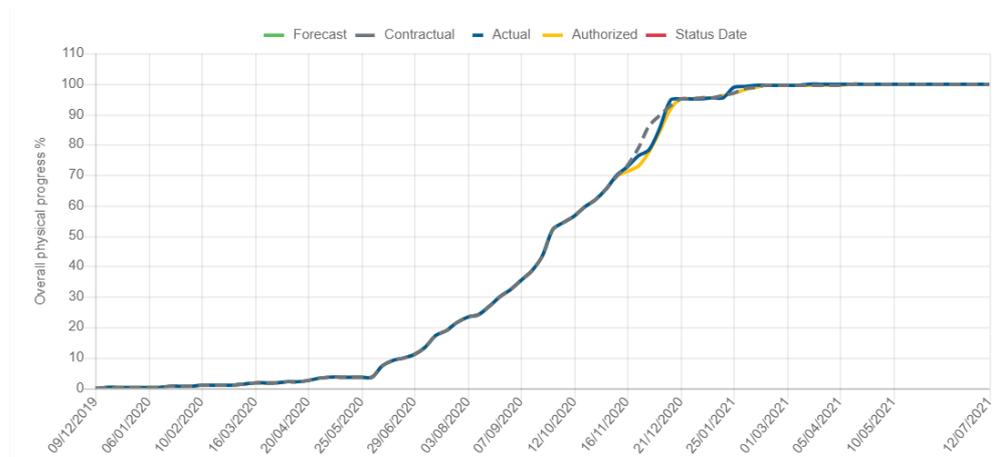
TORRE1



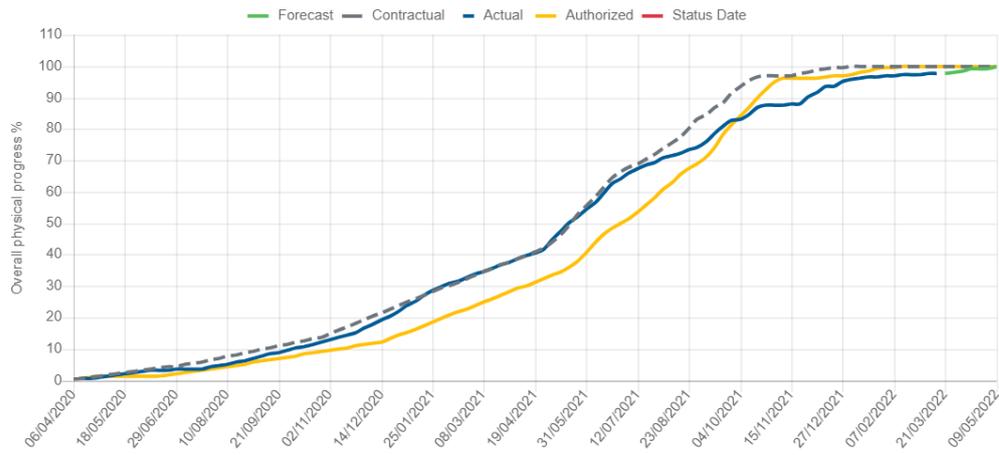
BINIA1



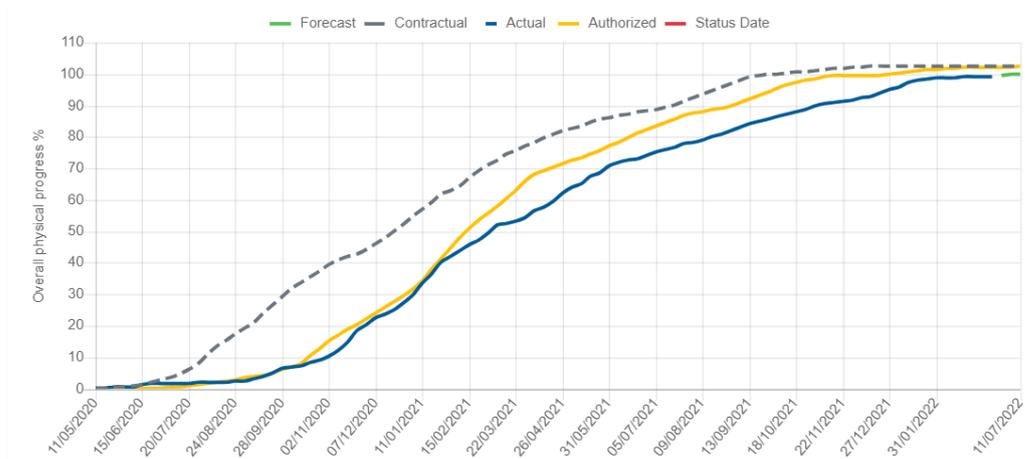
VAMVA1



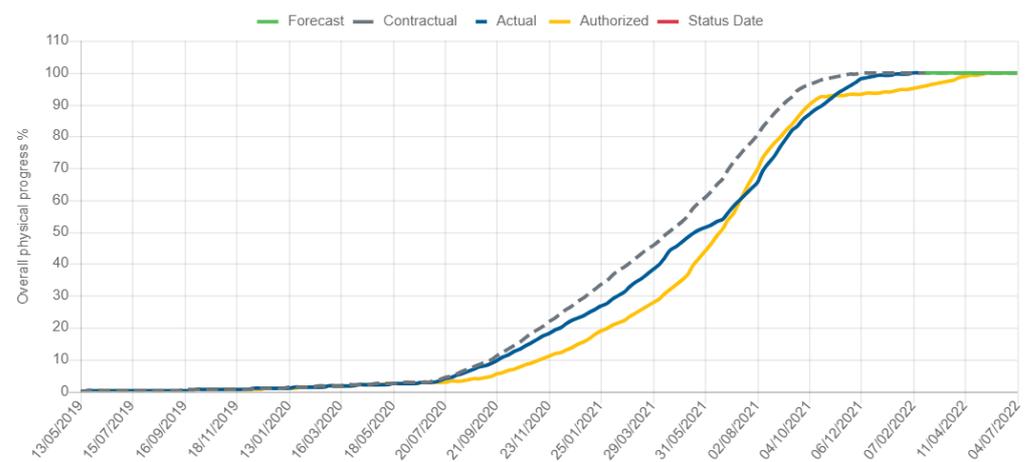
SANAG



LIVEN1



STRET1





ANEXO B

DATA	IGUAB5	DIAM2	PRAIA1	PARTAGE3	PARTAGE2	AGUA2
1	0	-1	0	0	0	0
2	0	-1	0	0	0	0
3	0	-1	0	0	0	0
4	0	-1	0	0	0	0
5	0	-1	0	0	0	0
6	0	-1	0	0	0	0
7	1	-1	0	0	0	0
8	1	-1	0	0	0	0
9	1	-1	0	0	0	0
10	1	-1	0	0	0	0
11	1	-1	-1	0	0	0
12	1	-1	-1	0	0	0
13	1	-1	-1	0	1	0
14	1	-1	-1	0	1	0
15	1	-1	-1	0	1	0
16	1	-1	-1	0	1	0
17	1	-1	-1	0	1	0
18	1	-1	-1	0	1	0
19	1	-1	-1	-1	1	0
20	1	-1	-1	-1	1	0
21	1	-1	-1	-1	1	0
22	1	-1	0	-1	1	1
23	1	-1	1	-1	1	1
24	1	-1	1	-1	1	1
25	1	-1	1	-1	1	1
26	1	-1	1	-1	1	1
27	1	-1	1	-1	1	1
28	1	-1	1	-1	1	1
29	1	-1	1	-1	0	1
30	1	-1	0	-1	-1	0
31	1	-1	-1	-1	-1	-1
32	1	0	-1	-1	-1	-1
33	1	1	-1	-1	-1	-1
34	1	1	-1	-1	-1	-1
35	1	0	-1	-1	-1	0
36	1	-1	-1	-1	-1	0
37	1	-1	-1	0	0	0
38	1	-1	-1	0	0	0
39	1	-1	-1	0	0	0
40	0	0	0	0	0	0



REINO2	ROCK1	ELMAN2	FINTE1	CAMPI2	MINGL1	POLIZ1
0	0	0	0	0	1	0
0	0	1	0	0	1	0
0	0	1	0	0	1	1
0	0	1	0	0	1	1
0	0	1	0	0	1	1
0	0	1	0	0	1	1
0	1	1	0	0	1	1
0	1	1	0	0	1	1
0	1	1	0	0	1	1
0	1	1	0	0	1	1
0	1	0	0	0	1	1
0	1	0	0	1	1	1
0	1	0	0	1	1	1
0	1	-1	0	1	1	1
0	0	-1	0	1	1	1
0	0	-1	0	1	1	1
0	0	-1	0	1	1	0
0	0	-1	0	1	1	-1
0	0	-1	0	1	1	-1
0	1	0	0	-1	0	-1
0	1	1	0	1	0	-1
0	1	1	0	0	1	-1
0	1	1	-1	-1	1	-1
0	1	1	-1	-1	1	-1
1	1	1	-1	-1	1	-1
1	1	1	-1	-1	1	-1
0	1	1	-1	0	1	-1
-1	1	1	-1	1	1	-1
-1	0	1	-1	1	0	-1
-1	1	1	-1	1	-1	-1
-1	1	1	-1	0	0	-1
-1	0	1	-1	1	1	-1
-1	0	1	-1	1	0	-1
-1	0	1	-1	1	-1	-1
0	0	0	0	1	-1	-1
0	0	0	0	0	0	0



STAMP1	GANAD1	TICOC1	KARUS1	SOETW1	AUGUS1	LASCO1
0	0	0	0	0	0	0
0	0	0	0	0	0	1
0	0	0	0	0	0	1
0	-1	0	-1	0	1	1
1	1	1	-1	0	1	1
1	1	1	-1	0	1	1
1	1	1	-1	0	1	1
1	0	1	-1	0	0	-1
1	0	1	-1	0	1	-1
1	1	1	-1	0	1	0
1	1	1	0	0	0	1
1	1	1	0	0	-1	1
1	1	1	0	0	-1	1
1	0	-1	0	0	-1	0
1	-1	-1	0	0	-1	-1
1	-1	0	1	0	-1	-1
1	1	-1	1	0	-1	-1
1	1	-1	1	1	-1	-1
1	1	-1	1	1	-1	-1
1	1	-1	1	1	-1	-1
1	1	0	1	1	-1	-1
1	1	1	1	1	-1	-1
1	1	1	1	1	-1	-1
1	1	1	0	1	-1	-1
1	1	1	-1	1	-1	-1
1	1	-1	-1	1	-1	-1
1	1	-1	-1	1	-1	-1
1	1	0	-1	1	-1	-1
1	1	-1	-1	1	-1	-1
1	1	0	-1	1	-1	-1
1	1	0	-1	1	-1	-1
1	1	0	-1	0	-1	0
1	1	0	-1	-1	-1	1
0	1	0	-1	-1	0	1
0	1	0	-1	-1	0	1
0	1	0	-1	-1	0	1
0	1	0	-1	-1	0	1
0	0	0	-1	-1	0	1
0	0	0	-1	-1	0	0



ANEXO C

Modelo: [0.11881362929570444, 0.12282164753014266]

Simulada_1:
MSE: 5.527822742681606
MAE: 1.890131281953733
SMAPE: 78.23210194342931
Correlation: 0.9535948283243955

Simulada_5:
MSE: 5.419857408367271
MAE: 2.0587204588636587
SMAPE: 77.5007165017328
Correlation: 0.9553625736099749

Simulada_2:
MSE: 7.492285812962924
MAE: 2.44753682935083
SMAPE: 81.53249360075506
Correlation: 0.9563744702291896

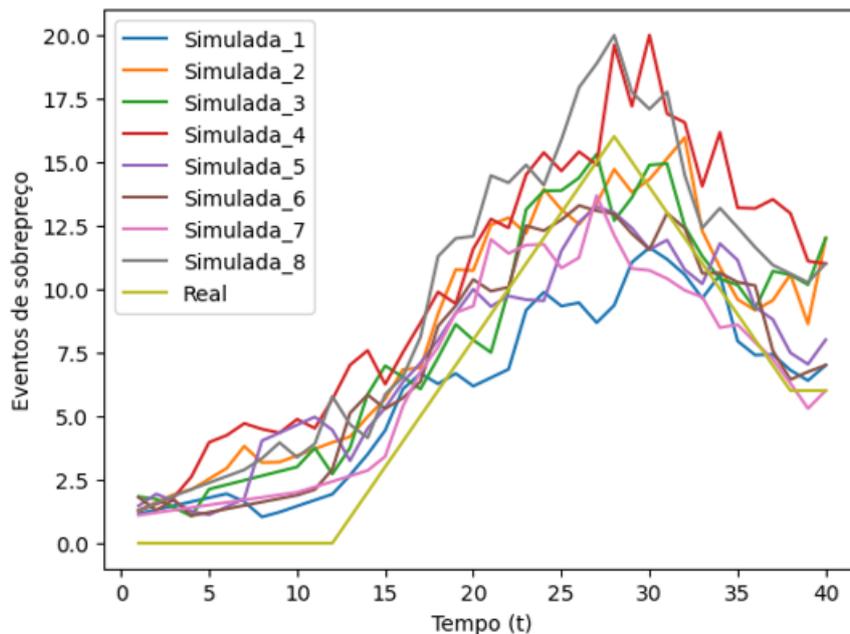
Simulada_6:
MSE: 3.3828661114734486
MAE: 1.5574997606550025
SMAPE: 75.87956256544578
Correlation: 0.9710912641480236

Simulada_3:
MSE: 6.009913576645781
MAE: 2.036661101266076
SMAPE: 79.1778803114932
Correlation: 0.9505734904858473

Simulada_7:
MSE: 3.4462737110297907
MAE: 1.5868342095367611
SMAPE: 73.65290775713792
Correlation: 0.9625109506104625

Simulada_4:
MSE: 17.32820303527061
MAE: 3.860249767099414
SMAPE: 90.91770553502727
Correlation: 0.9564667540621916

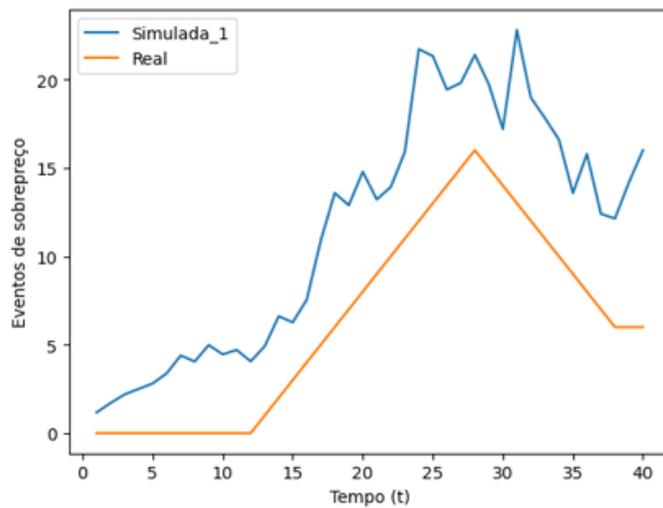
Simulada_8:
MSE: 12.868067037408736
MAE: 3.4102725565492973
SMAPE: 88.37249168982868
Correlation: 0.9813091211015792





Modelo: [0.31437127602021914, 0.27318183003278473]

Simulada_1:
 MSE: 31.793276434308893
 MAE: 5.22011605521096
 SMAPE: 99.74780844409293
 Correlation: 0.9617578416647182



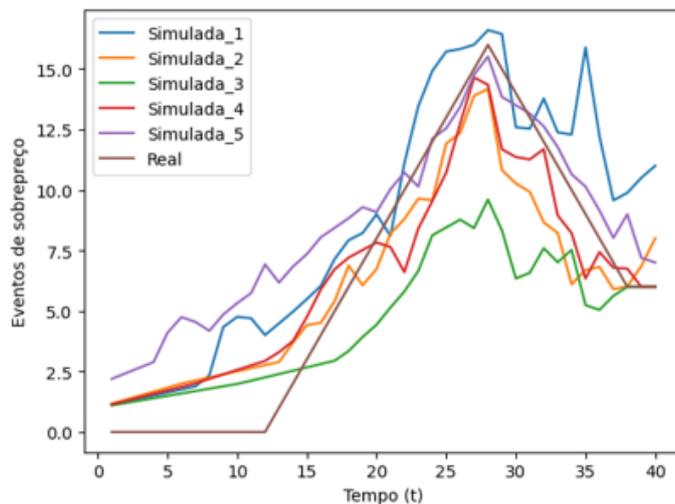
Modelo: [0.1050502084165468, 0.07076115620958508]

Simulada_1:
 MSE: 7.978510368909494
 MAE: 2.4335024205858278
 SMAPE: 81.66682177569854
 Correlation: 0.9518553101755257

Simulada_3:
 MSE: 12.00719869734661
 MAE: 2.8327244142225467
 SMAPE: 89.9894425412546
 Correlation: 0.9549562421048459

Simulada_2:
 MSE: 4.234148874405735
 MAE: 1.8254912151681004
 SMAPE: 76.55582818545197
 Correlation: 0.968003484632956

Simulada_4:
 MSE: 3.5788164071925737
 MAE: 1.6634111353823857
 SMAPE: 75.4137454438619
 Correlation: 0.9719197280081107



Simulada_5:
 MSE: 9.234774545387769
 MAE: 2.387438966136313
 SMAPE: 77.95124706820813
 Correlation: 0.9633927433248685

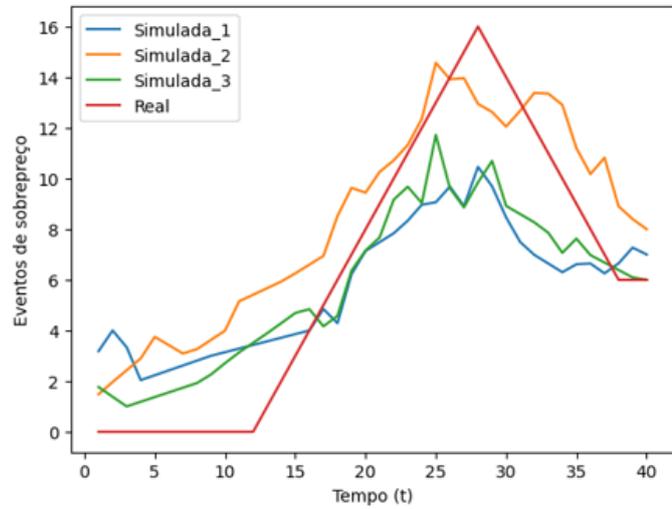


Modelo: [0.0896317243767274, 0.06976351811352355]

Simulada_1:
 MSE: 10.15700392576015
 MAE: 2.7555863582923257
 SMAPE: 82.99328809849244
 Correlation: 0.9549316123361185

Simulada_2:
 MSE: 7.699116289244546
 MAE: 2.4663266471942817
 SMAPE: 80.49389422811437
 Correlation: 0.9609690220637064

Simulada_3:
 MSE: 7.069821602834198
 MAE: 2.1547914501192134
 SMAPE: 79.94544598188416
 Correlation: 0.9575472613575624



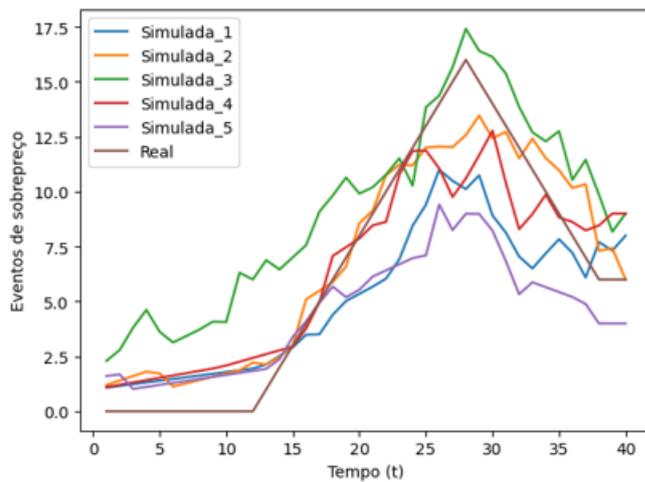
Modelo: [0.10039500615769674, 0.10333319421298007]

Simulada_1:
 MSE: 7.758771041680445
 MAE: 2.3591031722052973
 SMAPE: 83.31466993053314
 Correlation: 0.9502780737865837

Simulada_3:
 MSE: 10.983617619850682
 MAE: 2.957663633076286
 SMAPE: 83.64362532238208
 Correlation: 0.9659640238260278

Simulada_2:
 MSE: 2.352439772442271
 MAE: 1.288418713989722
 SMAPE: 70.16981899295452
 Correlation: 0.9746235044169209

Simulada_4:
 MSE: 4.3421053891211265
 MAE: 1.6351063700497381
 SMAPE: 73.71114955013847
 Correlation: 0.9505286080181236



Simulada_5:
 MSE: 12.163924919578715
 MAE: 2.815390085051108
 SMAPE: 89.19112329047269
 Correlation: 0.9753495331775403



Modelo: [0.05727385475501244, 0.0659949712713608]

Simulada_1:
 MSE: 5.245370319074786
 MAE: 1.9926290609571893
 SMAPE: 76.29186204125828
 Correlation: 0.9647067231246318

Simulada_5:
 MSE: 7.34026184960832
 MAE: 2.1935907016951015
 SMAPE: 80.16329746696307
 Correlation: 0.9528892364032268

Simulada_2:
 MSE: 12.825593660752137
 MAE: 2.995218123442305
 SMAPE: 95.252129893712
 Correlation: 0.963922873333658

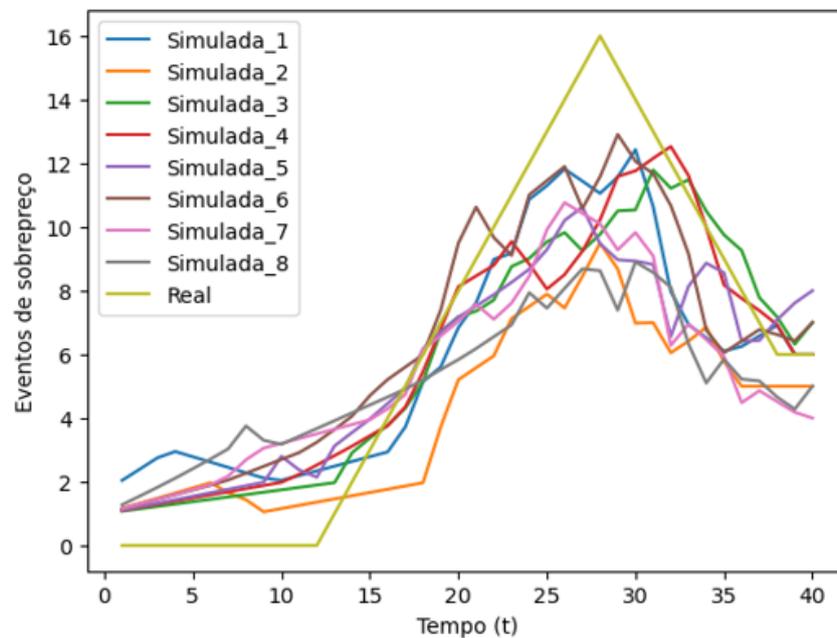
Simulada_6:
 MSE: 4.149465608961007
 MAE: 1.7698320616481016
 SMAPE: 75.87717214527558
 Correlation: 0.9669078111253235

Simulada_3:
 MSE: 4.857767826666487
 MAE: 1.6990366405235817
 SMAPE: 74.61754161463652
 Correlation: 0.9522673149161417

Simulada_7:
 MSE: 8.873659314028941
 MAE: 2.5951958201298084
 SMAPE: 84.97603540287407
 Correlation: 0.9517534082491024

Simulada_4:
 MSE: 4.906736921898807
 MAE: 1.591456598227444
 SMAPE: 73.245602387891
 Correlation: 0.9503588533864062

Simulada_8:
 MSE: 13.049603796222238
 MAE: 3.148212493470483
 SMAPE: 90.0473592715014
 Correlation: 0.9518541516932885





Modelo: [0.06883955767623993, 0.025218936132029596]

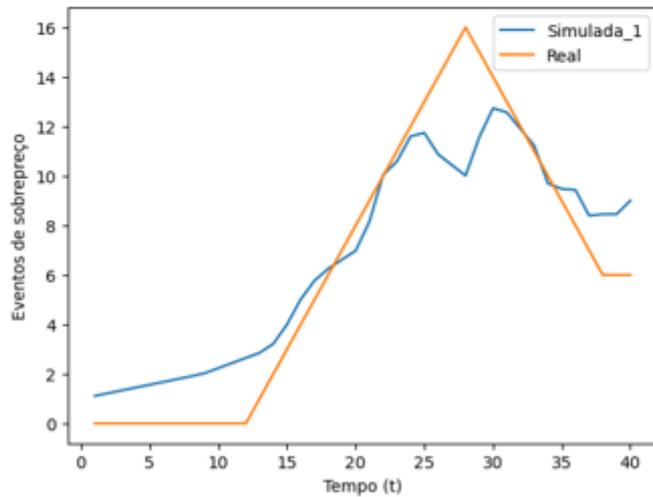
Simulada_1:

MSE: 3.94237325034184

MAE: 1.5637228647025876

SMAPE: 73.96821790346195

Correlation: 0.956746960910223



Modelo: [0.06400264536200635, 0.036477968567365036]

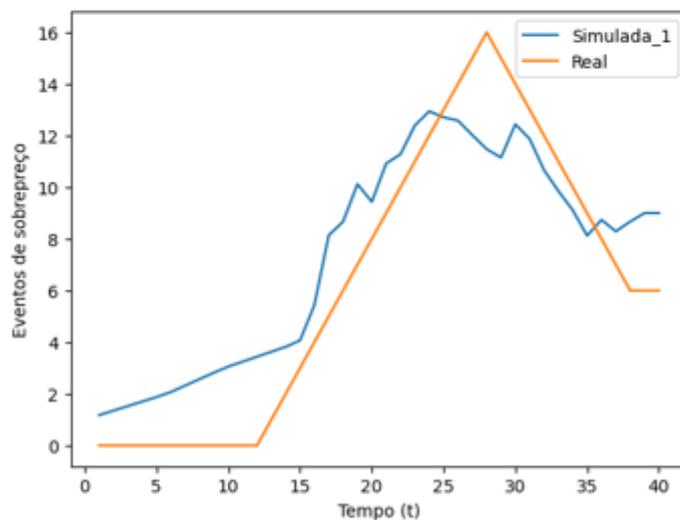
Simulada_1:

MSE: 5.002694540319967

MAE: 2.0169867917667825

SMAPE: 78.18200583317909

Correlation: 0.9505853180521827

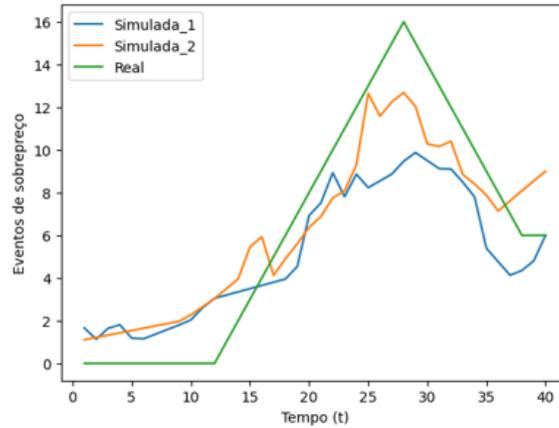




Modelo: [0.06604965459144374, 0.04978964757176377]

Simulada_1:
 MSE: 8.336826565071036
 MAE: 2.44074132411126
 SMAPE: 84.8335063112583
 Correlation: 0.9654309207510918

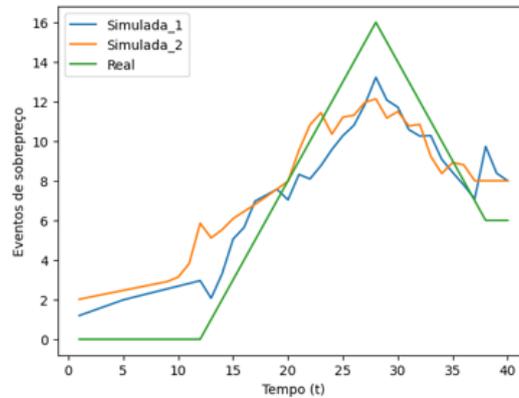
Simulada_2:
 MSE: 4.555801030313512
 MAE: 1.9888679028800325
 SMAPE: 79.96557169014218
 Correlation: 0.958050017298913



Modelo: [0.09531108562577362, 0.042044077601490044]

Simulada_1:
 MSE: 4.342510799558051
 MAE: 1.8928763090288705
 SMAPE: 76.35373519857073
 Correlation: 0.9692469429901482

Simulada_2:
 MSE: 6.2340790716768595
 MAE: 2.1908576709853502
 SMAPE: 78.53033873641466
 Correlation: 0.9639119940366251



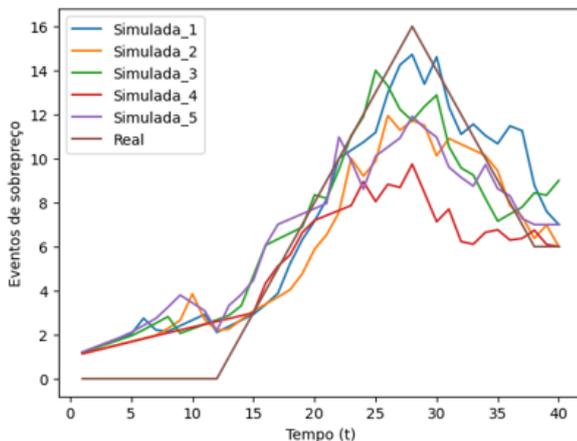
Modelo: [0.09957151766252241, 0.042876157728558795]

Simulada_1:
 MSE: 3.0317361708496557
 MAE: 1.479782046413445
 SMAPE: 71.80213764097428
 Correlation: 0.9639211697831641

Simulada_3:
 MSE: 3.7224968370473
 MAE: 1.690231507813246
 SMAPE: 75.13109469619108
 Correlation: 0.9670164866270309

Simulada_2:
 MSE: 4.402196079447319
 MAE: 1.7621443957533536
 SMAPE: 74.698030737694
 Correlation: 0.9656834367315958

Simulada_4:
 MSE: 10.062961139990735
 MAE: 2.452109646489127
 SMAPE: 81.79564616368373
 Correlation: 0.9534201239665622



Simulada_5:
 MSE: 5.46569453201819
 MAE: 2.032939696189215
 SMAPE: 77.11955815478004
 Correlation: 0.9693092254122081



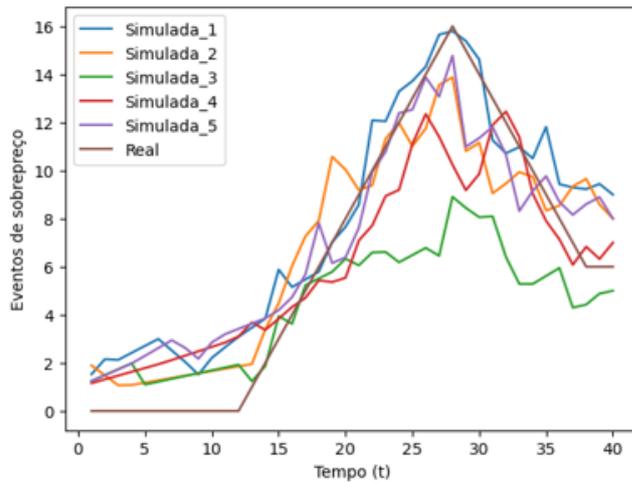
Modelo: [0.09283226597057208, 0.07533463846694832]

Simulada_1:
 MSE: 3.695283605638788
 MAE: 1.6215501588159058
 SMAPE: 74.33210127467035
 Correlation: 0.9767648598260732

Simulada_3:
 MSE: 13.568081052844112
 MAE: 2.8756822561164177
 SMAPE: 88.21332128595756
 Correlation: 0.9559208157794206

Simulada_2:
 MSE: 3.9266353327965917
 MAE: 1.722271304154079
 SMAPE: 76.87116266090818
 Correlation: 0.9503825517932253

Simulada_4:
 MSE: 5.151330797237908
 MAE: 1.8713175432976166
 SMAPE: 76.34438880119772
 Correlation: 0.9555826478411605

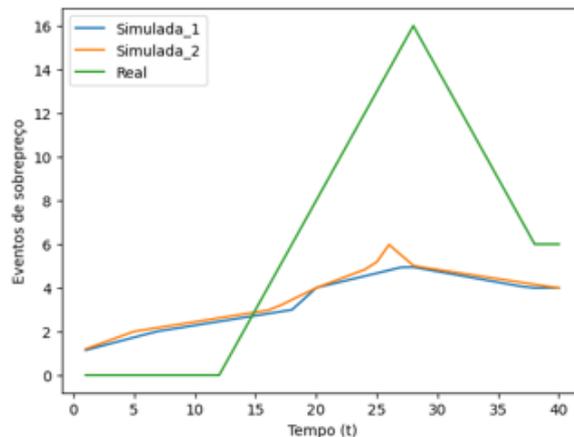


Simulada_5:
 MSE: 3.8978766835873215
 MAE: 1.7138446520514905
 SMAPE: 74.86477190680546
 Correlation: 0.9667211047443595

Modelo: [0.017399115799055242, 0.03410513855659888]

Simulada_1:
 MSE: 26.747629399489437
 MAE: 4.174540782215817
 SMAPE: 109.95890828630344
 Correlation: 0.9507860215851588

Simulada_2:
 MSE: 25.201163948825766
 MAE: 4.088997967622113
 SMAPE: 107.36436894792405
 Correlation: 0.9519099562148707





Modelo: [0.1087103089156497, 0.08259916618989965]

Simulada_1:
 MSE: 13.917105030293829
 MAE: 3.053124129911647
 SMAPE: 92.58411064643765
 Correlation: 0.9618745322065743

Simulada_5:
 MSE: 7.02322777920624
 MAE: 2.3468001987498273
 SMAPE: 79.64871780420755
 Correlation: 0.9548556576450996

Simulada_2:
 MSE: 7.778193095929325
 MAE: 2.3280792570497075
 SMAPE: 82.5190160788997
 Correlation: 0.9520733244421054

Simulada_6:
 MSE: 10.890680790895942
 MAE: 2.900298632520061
 SMAPE: 86.71981626985912
 Correlation: 0.9516539445266918

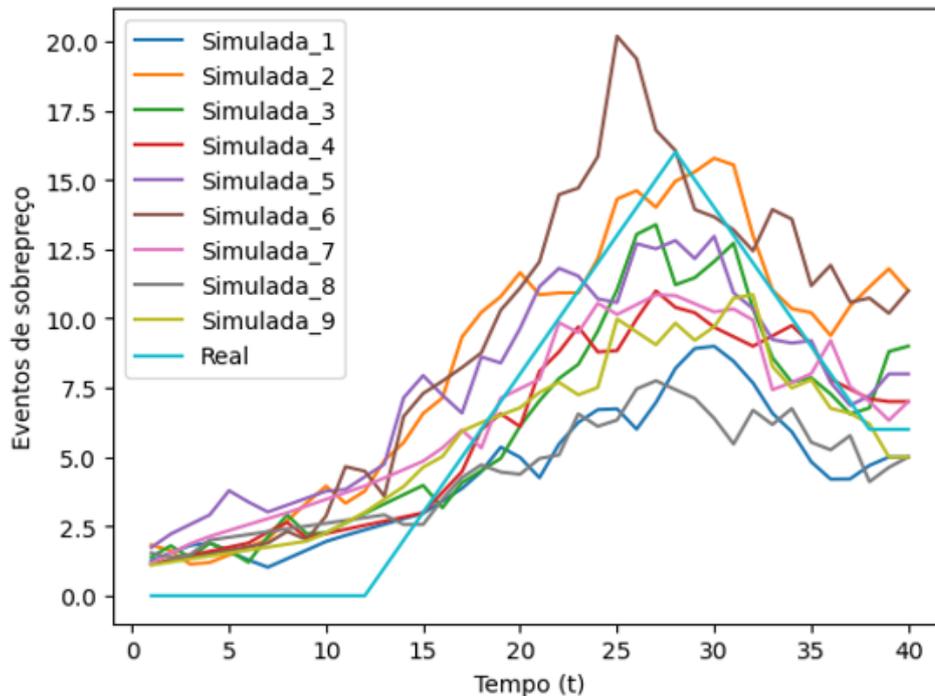
Simulada_3:
 MSE: 4.391364767590571
 MAE: 1.9028499643506258
 SMAPE: 78.10541221232882
 Correlation: 0.9557946141012578

Simulada_7:
 MSE: 6.37588738684519
 MAE: 2.1735799043996438
 SMAPE: 78.10928730687239
 Correlation: 0.9669500220194213

Simulada_4:
 MSE: 5.5276955800345835
 MAE: 1.8722958478150495
 SMAPE: 75.03956833287847
 Correlation: 0.9696611755042888

Simulada_8:
 MSE: 16.49906790351202
 MAE: 3.3531607531635585
 SMAPE: 94.2441949682937
 Correlation: 0.9619361516275118

Simulada_9:
 MSE: 7.041970723136717
 MAE: 2.183023309235626
 SMAPE: 80.7162743502326
 Correlation: 0.9651927638016464





Modelo: [0.0830877016997886, 0.08304582639195181]

Simulada_1:
 MSE: 5.757182482840703
 MAE: 1.9996863641558744
 SMAPE: 79.3773316016874
 Correlation: 0.9581565389214227

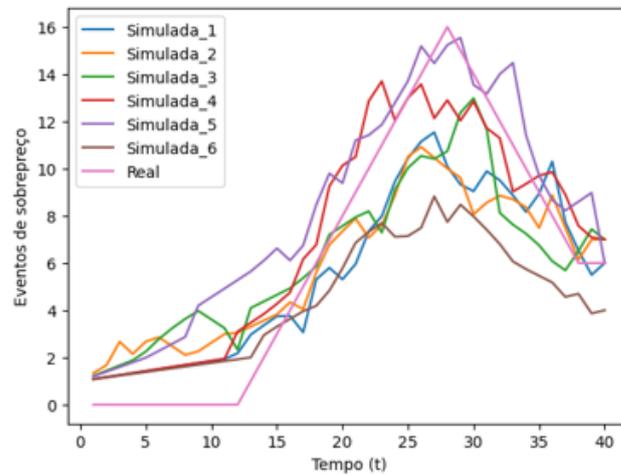
Simulada_4:
 MSE: 3.087702629729957
 MAE: 1.5698735303509341
 SMAPE: 74.98069889452997
 Correlation: 0.9676951823813797

Simulada_2:
 MSE: 7.329655800357733
 MAE: 2.2670523649369962
 SMAPE: 78.8947310221645
 Correlation: 0.962877659339741

Simulada_5:
 MSE: 6.158245280778711
 MAE: 2.05644659426396
 SMAPE: 77.83422009584295
 Correlation: 0.9686727228071151

Simulada_3:
 MSE: 6.598235705865259
 MAE: 2.2409138229232637
 SMAPE: 79.33672411799586
 Correlation: 0.9529013404036623

Simulada_6:
 MSE: 12.24555908360865
 MAE: 2.8605607337508503
 SMAPE: 90.25528510148796
 Correlation: 0.9825472125479563



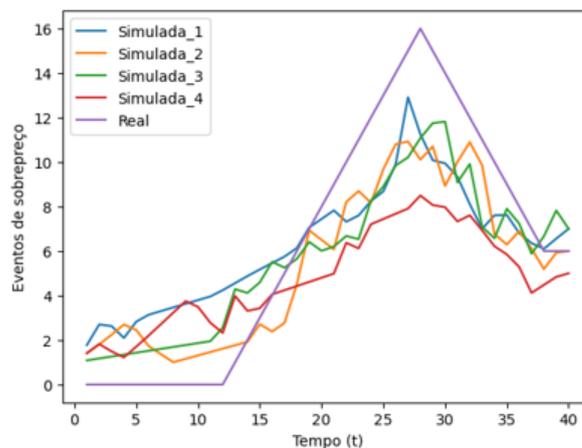
Modelo: [0.08785105179872042, 0.09859211451800363]

Simulada_1:
 MSE: 8.623880982270261
 MAE: 2.57952646946945
 SMAPE: 81.17502606610451
 Correlation: 0.9603547977234888

Simulada_2:
 MSE: 5.76221362104685
 MAE: 1.9767345448579292
 SMAPE: 78.05128849773205
 Correlation: 0.9703837103434693

Simulada_3:
 MSE: 6.531561543928727
 MAE: 2.212249222464481
 SMAPE: 82.49273264516397
 Correlation: 0.9562059261580959

Simulada_4:
 MSE: 13.85854556057277
 MAE: 3.198328342037679
 SMAPE: 91.93747196309685
 Correlation: 0.9632222959618184





Modelo: [0.07546314441346764, 0.06963222359510693]

Simulada_1:
 MSE: 37.323385052204436
 MAE: 5.627059077004084
 SMAPE: 102.9869583775898
 Correlation: 0.9515765430992168

Simulada_4:
 MSE: 5.87128275595086
 MAE: 2.1272141359462102
 SMAPE: 81.96461953609017
 Correlation: 0.956022194009313

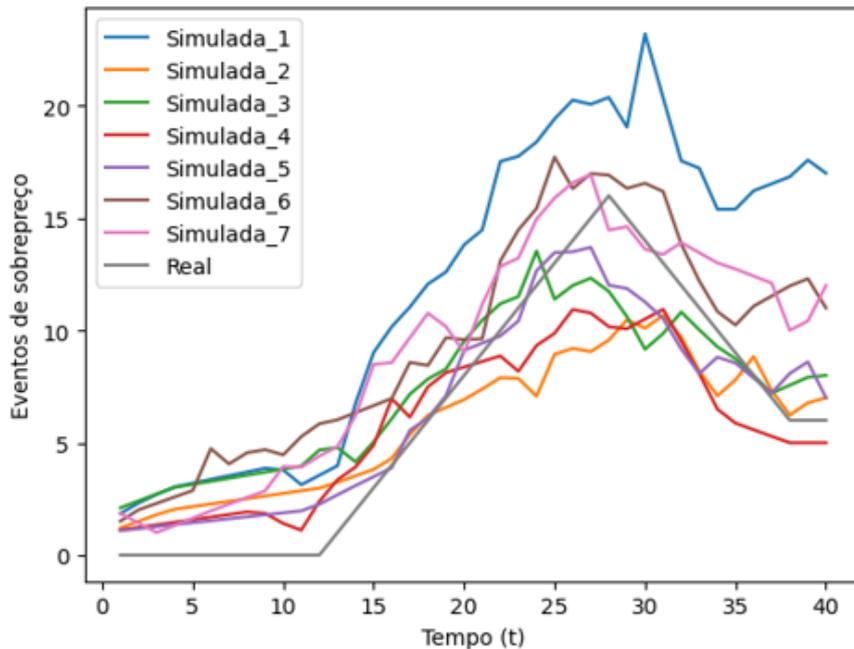
Simulada_2:
 MSE: 7.451681387573212
 MAE: 2.25544804782741
 SMAPE: 79.04877768294176
 Correlation: 0.9625254131979332

Simulada_5:
 MSE: 2.756247559541812
 MAE: 1.3555481012707216
 SMAPE: 71.68890496765027
 Correlation: 0.9724978665768946

Simulada_3:
 MSE: 7.014821451315166
 MAE: 2.3271868217809946
 SMAPE: 78.15730878564413
 Correlation: 0.9512575739877073

Simulada_6:
 MSE: 12.672961017176858
 MAE: 3.219903635010906
 SMAPE: 85.7819865549616
 Correlation: 0.9606693966074644

Simulada_7:
 MSE: 10.543959488468051
 MAE: 2.90495562352852
 SMAPE: 87.20308545551033
 Correlation: 0.9510377382152678



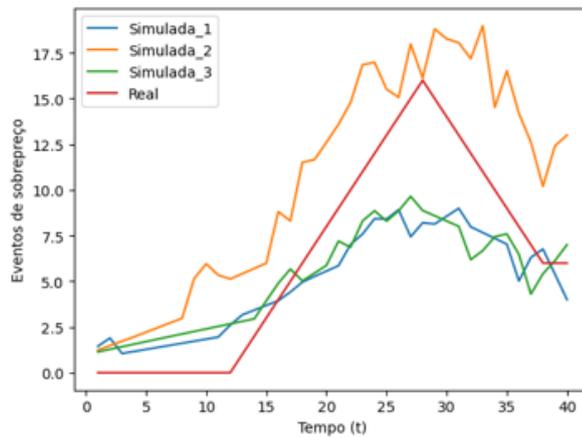


Modelo: [0.08611783212949338, 0.09355277736970888]

Simulada_1:
 MSE: 10.108050979110036
 MAE: 2.576304686862133
 SMAPE: 85.90480830378375
 Correlation: 0.9653199819166747

Simulada_2:
 MSE: 20.678740635586184
 MAE: 4.175995900130266
 SMAPE: 93.91369911354637
 Correlation: 0.952818196602588

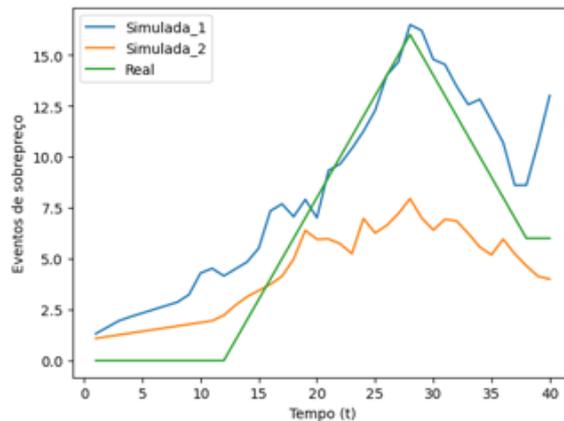
Simulada_3:
 MSE: 9.735242677439766
 MAE: 2.576797795881971
 SMAPE: 84.56284416423657
 Correlation: 0.9604872323846166



Modelo: [0.09269246127912027, 0.09531108495641614]

Simulada_1:
 MSE: 6.721503760410092
 MAE: 2.1454706552489586
 SMAPE: 78.95111701450666
 Correlation: 0.9538627988342372

Simulada_2:
 MSE: 15.282376201251271
 MAE: 3.0967157068798996
 SMAPE: 92.96249089348241
 Correlation: 0.9622004832261344



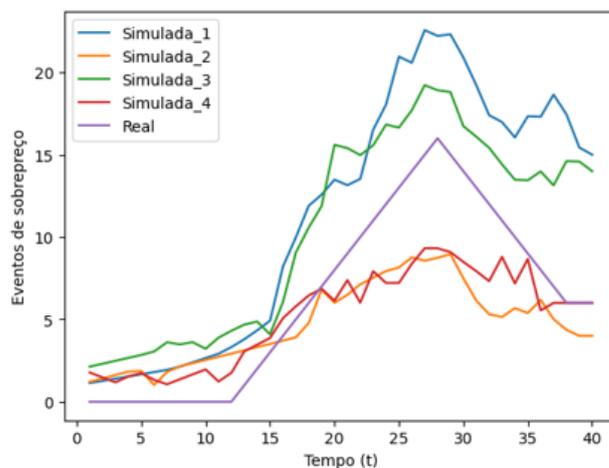
Modelo: [0.1062603737644523, 0.10410598591914483]

Simulada 1:
 MSE: 33.55886249131847
 MAE: 5.048051974273186
 SMAPE: 99.89980791628763
 Correlation: 0.9584542955425178

Simulada 2:
 MSE: 12.639515523809596
 MAE: 2.9546188210267026
 SMAPE: 90.28609421794714
 Correlation: 0.9508514474902707

Simulada 3:
 MSE: 20.104289520111593
 MAE: 4.134271133112589
 SMAPE: 92.71012294696263
 Correlation: 0.9583562915797287

Simulada 4:
 MSE: 9.163907096315183
 MAE: 2.3500248450647208
 SMAPE: 82.97305795621156
 Correlation: 0.9554574876892189





Modelo: [0.07987343161815483, 0.04990865437932415]

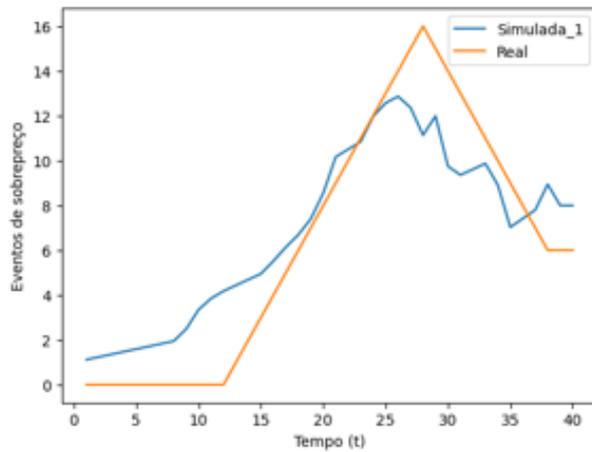
Simulada_1:

MSE: 5.005608965009219

MAE: 1.8807631664831064

SMAPE: 77.29840054559871

Correlation: 0.9505024494251495



Modelo: [0.05034814480209794, 0.01611496160787683]

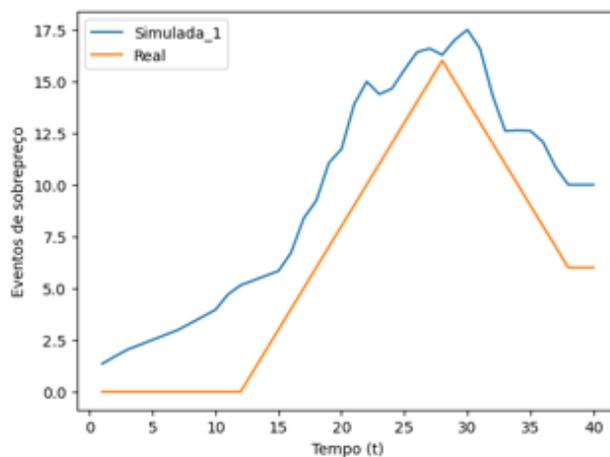
Simulada_1:

MSE: 11.058487236284106

MAE: 3.149669806031249

SMAPE: 86.9181855689537

Correlation: 0.9800840894906411

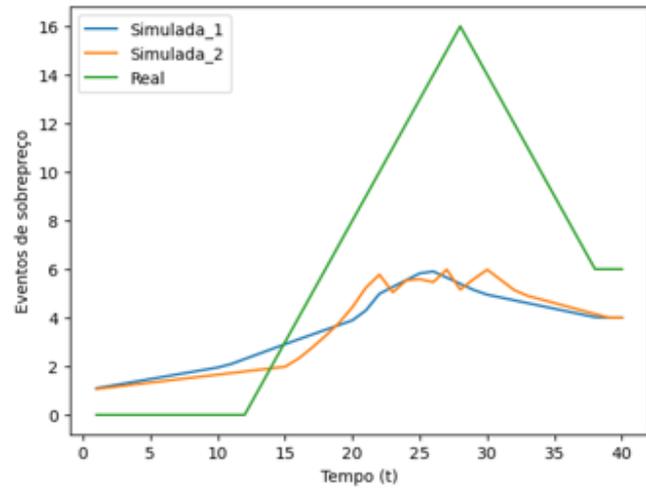




Modelo: [0.057946978501754044, 0.017455602406577404]

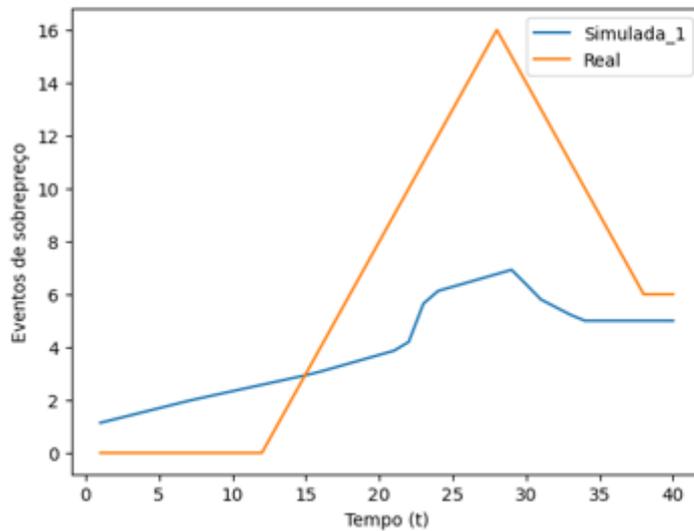
Simulada_1:
 MSE: 23.461776047602434
 MAE: 3.875559834454741
 SMAPE: 105.72155263854324
 Correlation: 0.9619304765280996

Simulada_2:
 MSE: 21.801783668869184
 MAE: 3.70773692088117
 SMAPE: 104.20988798568365
 Correlation: 0.9654187670858045



Modelo: [0.04928101978936014, 0.014645524593899336]

Simulada_1:
 MSE: 19.44883093955712
 MAE: 3.609260768087405
 SMAPE: 100.29747857995103
 Correlation: 0.9534796206379232





ANEXO D

Código Python

```

import birdepy as bd
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import time
from sklearn.metrics import silhouette_samples, silhouette_score
from tslearn.clustering import TimeSeriesKMeans
from tslearn.preprocessing import TimeSeriesScalerMeanVariance

def fluxo_mais_provavel(lista_p_dados): #Obtém um fluxo médio dentre um conjunto de
fluxos fornecidos
    cont = max(len(l) for l in lista_p_dados)
    p_medio = []
    for i in range(cont):
        z = 0
        n = 0
        p = 0
        for lista in lista_p_dados:
            if len(lista) > i:
                if lista[i] == 0:
                    z += 1
                elif lista[i] == 1:
                    p += 1
                elif lista[i] == -1:
                    n += 1
        if z >= p and z >= n:
            val = 0
        elif p > n:
            val = 1
        else:
            val = -1
        p_medio.append(val)

    return p_medio

def converte_para_serie_acumulada(p_data): #Converte fluxos que estão binarizados para
uma série temporal
    lista_dados = []
    val_0 = 0
    for i in p_data:
        val_0 += i
        lista_dados.append(val_0)
    return lista_dados

```



```

def estimativa_de_parametros(lista_dados, t_data, range_i, range_j, dnm = True, em = False,
abc = False, lse = False): #estima os parâmetros para uma série temporal fornecida
    range1 = list(range(range_i))
    range2 = list(range(range_j))

    result_dnm = []
    result_em = []
    result_abc = []
    result_lse = []

    for i in range1:
        for j in range2:
            if j > i:
                if dnm:
                    try:
                        est_dnm = bd.estimate(t_data, lista_dados, [0, 0], [[i, j], [i, j]], framework='dnm',
model='linear')
                        result_dnm.append([est_dnm.p, i, j])
                    except:
                        est_dnm = 'error'
                        result_dnm.append([est_dnm, i, j])
                if em:
                    try:
                        est_em = bd.estimate(t_data, lista_dados, [0, 0], [[i, j], [i, j]], framework='em',
model='linear')
                        result_em.append([est_em.p, i, j])
                    except:
                        est_em = 'error'
                        result_em.append([est_em, i, j])
                if abc:
                    try:
                        est_abc = bd.estimate(t_data, p_data, [0, 0], [[i, j], [i, j]], framework='abc',
model='linear')
                        result_abc.append([est_abc.p, i, j])
                    except:
                        est_abc = 'error'
                        result_abc.append([est_abc, i, j])
                if lse:
                    try:
                        est_lse = bd.estimate(t_data, lista_dados, [0, 0], [[i, j], [i, j]], framework='lse',
model='linear')
                        result_lse.append([est_lse.p, i, j])
                    except:
                        est_lse = 'error'
                        result_lse.append([est_lse, i, j])

    return [result_dnm, result_em, result_abc, result_lse]

def parametros_estimados(p_fluxo, t_data, i_range, j_range):

```



```

estimations = estimativa_de_parametros(p_fluxo, t_data, i_range, j_range)
return estimations

def parametro_medio(estimations):
    nasc = 0
    mort = 0
    for l in estimations:
        if type(l[0]) == list:
            nasc+= l[0][0]
            mort+= l[0][1]
        else:
            nasc+= l[0]
            mort+= l[1]
    return [nasc/len(estimations),mort/len(estimations)]

def simulate_series(param, t_max, valor_inicial, t_data):
    jump_times = []
    cont = 0
    while len(jump_times) < (len(t_data)*0.75):
        jump_times, pop_sizes = bd.simulate.continuous(param, 'linear', valor_inicial,
t_max=t_max)
        cont+=1
        if cont > 50:
            break
    interp_pop_sizes = np.interp(t_data, jump_times, pop_sizes)
    return interp_pop_sizes

def calibração_do_modelo(t_data_teste, p_fluxo_teste, num_iterations, param, start):
    results = []
    correlação = []
    best_samples = []

    for i in range(num_iterations-1):
        simulated_series = simulate_series(param, t_max=len(p_fluxo_teste), valor_inicial = 1,
t_data = t_data_teste)

        results.append(simulated_series)

        # Métricas de proximidade
        mse = np.mean((p_fluxo_teste - simulated_series) ** 2)
        mae = np.mean(np.abs(p_fluxo_teste - simulated_series))
        mape = np.mean(np.abs((p_fluxo_teste - simulated_series) / p_fluxo_teste)) * 100
        smape = np.mean(2 * np.abs(p_fluxo_teste - simulated_series) / (np.abs(p_fluxo_teste) +
np.abs(simulated_series))) * 100

        correlation = np.corrcoef(p_fluxo_teste, simulated_series)[0, 1]
        correlação.append(correlation)
        if (correlation > 0.95) and (mse < 100) and (mae < 10) and (smape < 110):
            best_samples.append([simulated_series,mse,mae,mape,smape,correlation,param])
        end = time.time()

```



```

if (end - start) > 30:
    break
return results, correlation, best_samples

```

```

def simulação_de_modelos_calibrados(amostras_desejadas, t_data_teste, p_fluxo_teste,
num_iterations, parametro):
    start_sim = time.time()
    best_samples_modelos = []
    results_modelos = []
    correlation_modelos = []
    cont = 0
    while len(best_samples_modelos) < amostras_desejadas:
        cont += 1
        results, correlation, best_samples = calibração_do_modelo(t_data_teste, p_fluxo_teste,
num_iterations, parametro, start_sim)
        for sample in best_samples:
            best_samples_modelos.append(sample)
            results_modelos.append(results)
            correlation_modelos.append(correlation)
        if cont > 49:
            break
        end_sim = time.time()
        if (end_sim - start_sim) > 15:
            break
    return best_samples_modelos, results_modelos, correlation_modelos, cont

```

```

def plot_modelos(modelos, p_data, t_data, exibir_errores_e_correlação = True):
    cont = 0
    for sample in modelos:
        cont += 1
        simulated_series = sample[0]
        mse = sample[1]
        mae = sample[2]
        mape = sample[3]
        smape = sample[4]
        correlation = sample[5]
        param = sample[6]

        if exibir_errores_e_correlação:
            print(f'Simulada_{cont}:')
            print("MSE:", mse)
            print("MAE:", mae)
            #print("MAPE:", mape)
            print("SMAPE:", smape)
            print("Correlation:", correlation)
            print('\n')

        # Plotagem das séries
        plt.plot(t_data, simulated_series, label=f'Simulada_{cont}')

```



```

plt.plot(t_data, p_data, label='Real')
plt.ylabel('Eventos de sobrepreço')
plt.xlabel('Tempo (t)')
plt.legend()

plt.show()

def validação_do_modelo(best_samples, p_fluxo_validação):
    modelo_final = []
    for amostra in best_samples:
        # Métricas de proximidade
        simulated_series = amostra[0]
        param = amostra[6]
        mse = np.mean((p_fluxo_validação - simulated_series) ** 2)
        mae = np.mean(np.abs(p_fluxo_validação - simulated_series))
        mape = np.mean(np.abs((p_fluxo_validação - simulated_series) / p_fluxo_validação)) *
100
        smape = np.mean(2 * np.abs(p_fluxo_validação - simulated_series) /
(np.abs(p_fluxo_validação) + np.abs(simulated_series))) * 100

        correlation = np.corrcoef(p_fluxo_validação, simulated_series)[0, 1]
        if (correlation > 0.95) and (mse < 100) and (mae < 10) and (smape < 110):
            modelo_final.append([simulated_series,mse,mae,mape,smape,correlation,param])

    return modelo_final

arquivo = r"amostras_obras.xlsx" #Arquivo com as séries de cada obra
df_obras = pd.read_excel(arquivo)
dict_obras = df_obras.to_dict('list')
obras = list(dict_obras.keys())
qnt = len(obras)-1
p_datas_cross_validation = []
p_datas_validação = []
t_data =[]
cont = 0
for obra in obras:
    if obra == 'DATA':
        t_data = dict_obras[obra]
    elif cont >= qnt*0.9:
        p_datas_validação.append(dict_obras[obra])
        cont+=1
    else:
        p_datas_cross_validation.append(dict_obras[obra])
        cont+=1

s_media_validação = fluxo_mais_provavel(p_datas_validação)
SAM_validação = converte_para_serie_acumulada(s_media_validação)

t_data_validação = list(range(len(SAM_validação)))

```



```

start0 = time.time()

num_iterations_param = 10
num_iterations = 50

parametros_finais = []

BestSamples_SAM = []
results_SAM = []
correlation_SAM = []
cont_SAM = 0

BestSamples_PM = []
results_PM = []
correlation_PM = []
cont_PM = 0

BestSamples_treino = []

contador = 1
for p in p_datas_cross_validation:
    start = time.time()

    base_treino = p_datas_cross_validation.copy()
    base_treino.remove(p)

    s_media_treino = fluxo_mais_provavel(base_treino)
    SAM_treino = converte_para_serie_acumulada(s_media_treino)
    base_teste = converte_para_serie_acumulada(p)

    conjunto_treino = []
    for p in base_treino:
        l = converte_para_serie_acumulada(p)
        conjunto_treino.append(l)

    estimations = []
    for l in conjunto_treino:
        est = parametros_estimados(l,t_data,1,num_iterations_param)
        estimations.append(est[0])

    parametros = []
    for lista in estimations:
        parametros.append(parametro_medio(lista))

    estimations_SAM = parametros_estimados(SAM_treino,t_data,1,num_iterations_param)[0]
    param_SAM = parametro_medio(estimations_SAM)
    param_PM = parametro_medio(parametros)

    parametros_finais.append(parametros)
    parametros_finais.append(param_SAM)

```



```

parametros_finais.append(param_PM)

endP = time.time()
tempoP = round((endP - start),2)

BestSampSAM, resultsSAM, correlationSAM, contSAM =
simulação_de_modelos_calibrados(5, t_data, base_teste, num_iterations, param_SAM)
BestSampPM, resultsPM, correlationPM, contPM = simulação_de_modelos_calibrados(5,
t_data, base_teste, num_iterations, param_PM)
BestSamples_p = []

for param in parametros:
    if param[0] > 1 or param[1] > 1: #Filtra os parâmetros invalidos que podem prejudicar a
qualidade das simulações
        print(f'o parâmetro {param} foi desconsiderado')
        continue
    BestSamp, results_p, correlation_p, cont_p = simulação_de_modelos_calibrados(5,
t_data, base_teste, num_iterations, param)
    BestSamples_p.append((BestSamp, cont_p))

for s in BestSampSAM:
    BestSamples_SAM.append(s)

results_SAM.append(resultsSAM)
correlation_SAM.append(correlationSAM)
cont_SAM += contSAM

for s in BestSampPM:
    BestSamples_PM.append(s)

results_PM.append(resultsPM)
correlation_PM.append(correlationPM)
cont_PM += contPM

for bs in BestSamples_p:
    if len(bs[0])>0:
        BestSamples_treino.append(bs[0])

end = time.time()
tempo = round((end - start),2)
print(f'Tempo para tentativa {contador}: {tempo} segundos')

contador+=1

end0 = time.time()
tempo0 = round((end0 - start0),2)
print(f'Tempo para o cross-validation: {tempo0} segundos')

modelo_SAM = validação_do_modelo(BestSamples_SAM, SAM_validação)
modelo_pm = validação_do_modelo(BestSamples_PM, SAM_validação)

```



```

modelos_p = []
for serie in BestSamples_treino:
    modelo_p = validação_do_modelo(serie,SAM_validação)
    if len(modelo_p) > 0:
        modelos_p.append(modelo_p)

modelo_final = []
for model in modelo_SAM:
    modelo_final.append(model)
for model in modelo_pm:
    modelo_final.append(model)
for l_models in modelos_p:
    for model in l_models:
        modelo_final.append(model)

for model in modelo_final:
    cont+=1
    plt.plot(t_data, model[0], label=f'Série_{cont}')
plt.ylabel('Eventos de sobrepreço')
plt.xlabel('Tempo (t)')
plt.show()

parametros_finais = []
for s in modelo_final:
    param_f = s[6]
    if param_f not in parametros_finais:
        parametros_finais.append(param_f)

dic_parametros = {}
dic_parametros['Nascimento'] = []
dic_parametros['Morte'] = []

for par in parametros_finais:
    dic_parametros['Nascimento'].append(par[0])
    dic_parametros['Morte'].append(par[1])

df_parametros = pd.DataFrame(dic_parametros)

caminho = 'resultados_modelos.xlsx' #Arquivo Excel onde serão salvos os parâmetros
validados
with pd.ExcelWriter(caminho, engine='openpyxl', mode='a', if_sheet_exists='overlay') as
writer:
    df_parametros.to_excel(writer, sheet_name='Parametros Teste 2', header=False)

dic_model = {}
for serie in modelo_final:
    param = str(serie[6])
    if param not in dic_model.keys():
        dic_model[param] = []
    dic_model[param].append(serie)

```



```

for param in parametros_finais: #Plotagem das séries para cada parâmetro vaidado
    print(f'Modelo: {param}\n')
    plot_modelos(dic_model[str(param)],SAM_validação,t_data)

l_smp_modelos = [] #Cálculo das séries médias finais, por parâmetro, para a análise por
clusterização
contador=0
dic_parametros_seriesM = {}
for parametro in parametros_finais:
    l_series = []
    l = dic_model[str(parametro)]
    for sl in l:
        l_series.append(sl[0])

    media_series_temporais = np.mean(l_series, axis=0)
    l_smp_modelos.append(media_series_temporais)
    contador+=1
    dic_parametros_seriesM[f'Parâmetro {contador}: {str(parametro)}'] =
media_series_temporais

df_parametros_e_series = pd.DataFrame(dic_parametros_seriesM)
caminho = 'resultados_modelos.xlsx' ##Arquivo Excel onde serão salvas as séries médias e seus
respectivos parâmetros
with pd.ExcelWriter(caminho, engine='openpyxl', mode='a', if_sheet_exists='overlay') as
writer:
    df_parametros_e_series.to_excel(writer, sheet_name='Parametros Finais e Séries',
index=False)

data = l_smp_modelos.copy()

# Converter a lista para um array numpy
data_array = np.array(data)

# Pré-processamento
scaler = TimeSeriesScalerMeanVariance()
data_scaled = scaler.fit_transform(data_array)

# Clusterização
n_clusters = 3
kmeans = TimeSeriesKMeans(n_clusters=n_clusters, metric= 'dtw', max_iter=500,
random_state=0)
kmeans.fit(data_scaled)

# Obter rótulos dos clusters
labels = kmeans.labels_
print(labels)

# Plotar os resultados da clusterização
fig, ax = plt.subplots(n_clusters, sharex=True, sharey=True, figsize=(10, 15))

```



```
cont = 0
for i in range(n_clusters):
    cluster_data = data_array[labels == i]
    for j, ts in enumerate(data_array):
        if ts in cluster_data:
            ax[i].plot(ts, label=f'Parâmetro_{j+1}') # Adicionando rótulo à legenda

    ax[i].set_ylabel('Eventos de sobrepreço')
    ax[i].set_title(f'Cluster {i+1}')
    ax[i].legend(loc = 'upper left') # Adicionando a legenda no subplot correspondente

plt.xlabel('Tempo')
plt.suptitle('Resultados da Clusterização\n')
plt.tight_layout()

plt.show()
```