Assessing the Potential of Large Language Models (LLMs) for the Automatic Generation of Solutions for Programming Exercises

André N. Alcantara*, Mateus P. Silva*, Hugo N. Oliveira*, Julio C. S. Reis*
*Deparment of Informatics (DPI), Universidade Federal de Viçosa (UFV), Brazil
{andre.n.alcantara,mateus.p.silva,hugo.n.oliveira,jreis}@ufv.br

ABSTRACT

Since several AI chatbots have become available to the general public, the popularity of Large Language Models (LLMs) has risen, especially among students, who have used them as educational support tools. This directly impacts the reproduction of scientific knowledge and the learning process. In this context, it is relevant to evaluate the performance and limitations of these tools in providing correct answers. This article discusses the results of C++ code solutions requested from the Meta Llama 3 model for practical programming tasks of Olimpíada Brasileira de Informática (OBI). It was demonstrated that the method still needs some correctness checker, presenting only 30.4% of correct answers in general. In addition, it was evidenced that the model's performance deteriorates in questions with images, since the model does not support them as input, presenting a 20.3% drop in correct answers. Finally, the model appears to work better the shorter the question text, with 31.1% more correct answers for questions with up to 200 words compared to questions with more than 400 words.

KEYWORDS

LLMs, Programming Exercises, Automatic Generation of Code

1 INTRODUÇÃO

Modelos de Linguagem de Grande Escala (LLMs) têm emergido como ferramentas promissoras no contexto educacional [12], sendo capazes de gerar texto de maneira coerente e com forte potencial de sumarização e associação de informações, quando treinados com vastos conjuntos de dados. Por isso, são explorados no apoio ao processo de ensino [11] e aprendizagem e na personalização do ensino [2].

A adoção de LLMs na educação superior tem crescido significativamente [1]. Em uma pesquisa realizada pela Associação Brasileira de Mantenedoras de Ensino Superior (ABMES)[1], constatou-se que 92% dos estudantes e potenciais alunos consideram as ferramentas de Inteligência Artificial (IA) eficazes ou muito eficazes para a resolução de problemas e esclarecimento de dúvidas. Ainda mais, 84% dos entrevistados acreditam que a IA substituirá parcialmente (62%) ou totalmente (22%) os professores. O fato é que tais ferramentas são utilizadas em uma série de tarefas para as quais não se conhece, exatamente, o seu potencial para realizá-las. É neste contexto que está inserido o objetivo deste trabalho.

Particularmente, neste estudo, investigamos a capacidade de um LLM referência no mercado de IA, o Llama 3, na geração de soluções funcionais para exercícios de programação. A hipótese central é de que LLMs geram conteúdo com pouco ou nenhum compromisso com corretude, gerando grande impacto no processo de ensino-aprendizagem. Para isso, coletou-se 460 exercícios de programação da Olimpíada Brasileira de Informática (OBI), considerando diferentes níveis de dificuldade. Depois disso, simula-se a interação de um usuário com um LLM de código aberto na solicitação de um código solução para determinado exercício. A solução gerada é então submetida ao corretor automático da OBI (i.e., Juiz Online) para avaliação da corretude.

Este trabalho se distingue de *benchmarks* puramente técnicos (como *HumanEval*, *SWE-Bench* ou *Mostly Basic Python Problems –MBPP*)[2], amplamente utilizados para medir a corretude funcional de código gerado [3], ao focar no contexto educacional aplicado e realista (i.e., OBI). Essa abordagem confere uma avaliação de corretude mais estrita e contextualizada, uma vez que utiliza o Juiz Online como corretor automático, que exige não apenas a lógica correta, mas o cumprimento de restrições de tempo e memória de execução, elementos essenciais no ambiente de programação competitiva e real de ensino no Brasil.

Quanto aos resultados obtidos, o LLM demonstrou inadequação para uso não supervisionado, com apenas 30,4% de acertos em geral. O desempenho do modelo foi significativamente impactado por fatores contextuais: a omissão intencional do contexto visual no *input* causou uma queda absoluta de 20,3% nos acertos, e o modelo apresentou resultados 31,1% melhores para enunciados mais curtos do que longos.

2 TRABALHOS RELACIONADOS

LLMs têm sido amplamente investigados no contexto educacional, incluindo tarefas como personalização do aprendizado [2], apoio ao ensino e à aprendizagem [11], bem como no uso em ferramentas educacionais [10]. Silva *et al.* [8], por exemplo, avaliaram a capacidade de modelos do estado-da-arte (incluindo GPT-4o e Gemini-1.5-pro) para gerar *feedback* em problemas de programação, constatando que 63% das dicas eram precisas e completas, mas que uma taxa significativa de 37% continha erros, como falhas de alucinação ou incorreções na identificação de linhas.

Outro estudo aponta limitações significativas na resolução de problemas com enunciados incompletos ou que não explicitam casos de borda – i.e., entradas que trazem casos menos triviais comumente responsáveis por erros –, além dos que não fornecem exemplos de respostas corretas (casos de teste) [6]. Por outro lado, uma abordagem orientada por casos de teste também foi explorada

[1]https://abmes.org.br/blog/detalhe/18833/ia-na-educacao-positivo-e-operante

[2]https://github.com/openai/human-eval, https://www.swebench.com/, https://github.com/google-research/google-research/tree/master/mbpp

In: V Concurso de Trabalhos de Iniciação Científica (CTIC 2025). Anais Estendidos do XXXI Simpósio Brasileiro de Sistemas Multimídia e Web (CTIC'2025). Rio de Janeiro/RJ, Brasil. Porto Alegre: Brazilian Computer Society, 2025.

ISSN 2596-1683

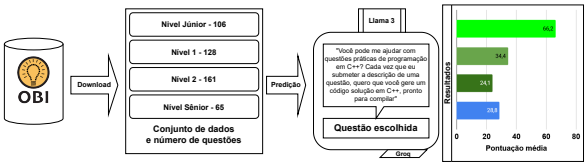

Figura 1: Visão geral da metodologia.

[5], destacando o uso de código incorreto de alunos para aprimorar o aprendizado.

Savelka *et al.* [7] identificaram que modelos atuais não conseguem aprovação em cursos de programação de linguagem Python, obtendo menos de 70% de acurácia em módulos básicos. Apesar disso, o *feedback* desses modelos permite que os aprendizes tenham mais de 55% da pontuação em cursos introdutórios e intermediários.

Por fim, Souza *et al.* [9] avaliaram LLMs na resolução de problemas de lógica da Olimpíada Brasileira de Informática (OBI). Mesmo nesta modalidade destinada ao ensino fundamental, os resultados foram limitados, com acertos de apenas 22,2% e 27,5% em questões dos níveis Júnior e 1, respectivamente, e nenhum acerto no nível 2. O trabalho aqui apresentado é complementar ao citado, diferindo-se no uso de modelo aberto e de questões de outra modalidade e linguagem (i.e., C++).

3 METODOLOGIA

A metodologia detalhada é apresentada nesta seção, incluindo os dados explorados e a estratégia (LLM) para gerar os resultados. Uma visão geral é apresentada na Figura 1, em etapas: 1) coleta das questões da OBI; 2) *prompts* de solução fornecidos ao LLM; e 3) resultados do juiz.

3.1 Dados da OBI

A OBI[3] é uma competição anual organizada pela Sociedade Brasileira de Computação (SBC), projetada para estudantes do Ensino Fundamental até o Ensino Superior, com o objetivo de estimular o interesse pela Ciência da Computação. Em resumo, ela é dividida em duas modalidades: de Iniciação – que traz problemas de lógica computacional para alunos sem conhecimentos em programação – e de Programação, da qual trata este artigo, que demanda a elaboração de algoritmos em uma linguagem de programação.

A modalidade de Programação da OBI se divide em níveis de dificuldade crescente: Júnior, 1, 2 e Sênior, pelos quais os participantes podem optar. Dado um nível de dificuldade, a prova sucede-se em 3 fases consecutivas, também de dificuldade crescente, como processo eliminatório para o pódio. Uma questão da modalidade Programação contém um enunciado textual que descreve o problema, podendo incluir elementos auxiliares como imagens. O enunciado especifica, ainda, os formatos de entrada e saída textuais, além de disponibilizar exemplos de testes para validação.

Em seguida, o programa submetido ao Juiz Online da OBI é executado em subtarefas com conjuntos de entrada. A pontuação é dada pela porcentagem de subtarefas que executaram com a saída esperada. O Juiz impõe limites máximos de tempo e memória para a execução, essenciais para penalizar soluções ineficientes.

A OBI disponibiliza abertamente na página "Pratique", uma série de questões - de edições passadas - com Juiz Online acoplado, em seu *Website*[4]. Nesta página, é permitida a submissão de soluções individuais que seguem o fluxo ilustrado na Figura 2, que consiste na compilação do programa, que, em caso de sucesso, é seguida da execução, que também pode ser interrompida por erros de Tempo Limite Excedido (TLE) ou de Memória Limite Excedida (MLE). Após a execução completa, a pontuação da solução é fornecida.

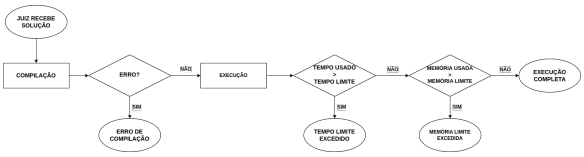

Figura 2: Etapas do Juiz Online na avaliação de soluções.

3.2 Coleta dos Exercícios

Os enunciados foram coletados de forma automatizada (utilizando um "*Web Crawler*") no índice de questões da página "Pratique" do *website* da OBI, abrangendo as edições de 2006 a 2021. O volume inicial de 494 enunciados foi reduzido para 460 questões para análise, devido à impossibilidade de submissão de código de 34 exercícios na plataforma da OBI, por erros de conexão persistentes.

É importante ressaltar que este conjunto é uma amostra disponibilizada pela OBI para treinamento, em que não há uniformidade na quantidade de questões por ano, e em que o conjunto de questões para cada categoria (edição, nível ou fase) não representa a totalidade destas nas provas originais. Neste estudo, a escolha da linguagem de programação C++ como padrão para as soluções segue a convenção da OBI na seleção de competidores para olimpíadas de programação internacionais e a popularidade de tal linguagem no meio de programação competitiva.

3.3 Geração Automática de Código

O LLM utilizado é o Llama 3 (llama3-70b-8192), da Meta AI, reconhecido como o modelo de código aberto com melhor desempenho em benchmarks de raciocínio e geração de código (e.g., *HumanEval* e *MBPP*), reconhecido como estado-da-arte na comunidade *open-source* [4] para realização de tarefas diversas. A escolha deste modelo teve como finalidade estabelecer um limite superior de desempenho esperado de modelos abertos no contexto educacional, sendo a versão mais recente e disponível através da API da Groq[5](no período dos experimentos em Nov/2024). Esta abordagem prioriza ferramentas acessíveis ao público no contexto educacional, facilitando a reprodutibilidade dos resultados, em contraste com modelos proprietários (e.g., ChatGPT). Sua arquitetura é focada no processamento de texto, sem suporte para entrada direta de arquivos de imagem. Portanto, imagens foram omitidas nos enunciados fornecidos ao modelo. O acesso por API (Groq) proporcionou a infraestrutura necessária para a execução eficiente e a interação automatizada. Cada solicitação foi feita em uma nova instância de *chat*, sem janela de contexto de *prompts* ou respostas para outras questões já solicitadas, todas geradas sob mesma condição inicial.

Foi anexado ao início dos *prompts* a solicitação: "*Você pode me ajudar com questões práticas de programação em C++? Cada vez que eu submeter a descrição de uma questão, quero que você gere um código solução em C++, pronto para compilar.*", seguida pelo enunciado da

[3]http://www.sbc.org.br/obi

[4]https://olimpiada.ic.unicamp.br/info/

[5]https://groq.com/

questão. Os códigos gerados em resposta foram então submetidos ao Juiz Online da OBI presente na página Web "Pratique", cujos resultados foram armazenados para análise.

3.4 Terminologia e Convenções

Para padronizar a interpretação dos resultados, os vereditos apresentados pelo Juiz Online foram categorizados e rotulados da seguinte forma:

- **Acerto:** Pontuação total;
- **Zero:** Execução completa e pontuação zero;
- **Parcial:** Execução completa e pontuação no intervalo (0, 100);
- **Não compila:** Execução impossível devido a falha de compilação;
- **TLE:** Execução interrompida por tempo limite excedido;
- **MLE:** Execução interrompida por memória limite excedida.

Ademais, quando o termo "erro" é usado de forma isolada, refere-se ao conjunto dos vereditos **Zero**, **Não compila**, **TLE** e **MLE**. Quando atrelado a outro termo, como em "erro de compilação", mantém seu significado usual.

As proporções (%) de solução com veredito **TLE** são calculadas em relação ao total de soluções compiláveis (410), dado que programas executados, mesmo com interrompimento, foram previamente compilados. Assim, as estatísticas se traduzem de forma mais balanceada e relevante. Já a análise da ocorrência de veredito **MLE** não será apresentada, uma vez que sua ocorrência foi ínfima, impossibilitando apresentar qualquer tendência relevante. Finalmente, as porcentagens de acertos e as pontuações médias foram arredondadas para a primeira casa decimal. Os resultados obtidos são apresentados nas seções subsequentes.

4 RESULTADOS EXPERIMENTAIS

No geral, o LLM investigado (i.e., Llama 3) obteve acerto em 140 de 460 soluções (30,4%), registrando erro em 258 casos (56,2%), conforme detalhado na Figura 3. A pontuação média geral foi 37,3, com a média entre soluções compiláveis em 41,9. A alta taxa de falhas e a pontuação relativamente baixa indicam limitações na interpretação dos desafios e na execução correta das soluções.

4.1 Impacto do Nível de Dificuldade e Fase

O corpo de questões extraídas pode ser dividido por nível ou fase da prova à qual pertencem, conforme apresentado respectivamente nas Tabelas 1 e 2. Conforme pode ser visto na Tabela 1, há uma

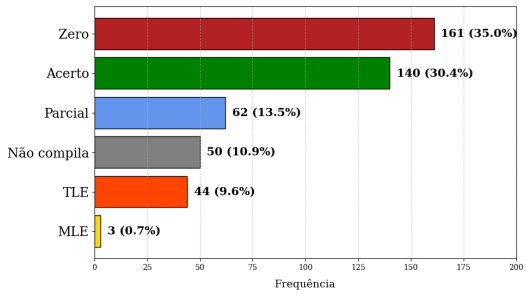

Figura 3: Distribuição de vereditos por categoria.

Tabela 1: Frequência, pontuação média e acertos estratificados por nível de dificuldade.

	Frequência	Pontuação média	Acertos (%)
Nível Júnior	106	66,2	58,5
Nível 1	128	34,4	28,1
Nível 2	161	24,1	15,5
Nível Sênior	65	28,8	26,2

Tabela 2: Frequência, pontuação média e acertos estratificados pelas fases da prova da OBI.

	Frequência	Pontuação média	Acertos (%)
Fase 1	190	43,3	36,3
Fase 2	198	34,9	28,3
Fase 3	72	28,1	20,8

queda próxima de 50% na pontuação média e na taxa de acertos ao elevar-se do Nível Júnior (o único destinado ao ensino fundamental), evidenciando inadequação a propostas menos triviais. A tendência de queda no desempenho é similarmente observada entre as fases da prova, porém menos acentuada.

As porcentagens de vereditos **Não compila** em relação aos níveis foram, respectivamente: 3,8, 11,7, 13,7 e 13,8. Já em relação às fases, foram: 9,5, 13,1 e 8,3. As porcentagens de vereditos **TLE** em relação aos níveis foram, respectivamente: 4,9, 8,8, 17,3 e 8,9. Já em relação às fases, foram: 9,9, 11,6 e 10,6. Por outro lado, a tendência de queda no desempenho se mantém consistente até o penúltimo nível e fase, possivelmente efeito da amostragem reduzida de questões de **Nível Sênior** ou **Fase 3**, que formam um corpo pouco representativo em relação ao seu todo.

4.2 Resultados para Enunciados com e sem Imagens

Entre as 460 questões analisadas, houve 298 (64,9%) ocorrências de enunciados sem imagem e 162 (35,2%) com imagem. Experimentos considerando a omissão das imagens para 162 questões simulam a prática de estudantes que utilizam LLMs como ferramenta de apoio, tipicamente através de operações de copiar e colar o texto do enunciado desconsiderando imagens, visando testar a resiliência do modelo na geração de soluções para um cenário de *input* incompleto.

A Tabela 3 apresenta a tendência destas variações sobre os vereditos, e a Figura 4 estabelece um comparativo de eficácia entre suas soluções. As diferenças causadas pelo contexto faltante são acentuadas, especialmente na taxa de vereditos **Zero**, **Acerto** e **TLE**, que tiveram variações próximas de 50%. A discrepância é melhor ilustrada na Figura 4 ao agrupar todas as categorias de erro. Isso indica que a falta de contexto visual foi um grande diferencial na capacidade de resolução correta. Entretanto, mesmo para as questões sem imagem, das quais foi fornecido o contexto completo para o problema, a taxa de acerto (37,6%) não se sobressaiu à taxa geral do estudo (30,4%).

4.3 Impacto das Características dos Enunciados

Por fim, considerando a contagem de palavras e o número de exemplos de saída correta no enunciado de questão, o tamanho médio

Tabela 3: Distribuição de vereditos por categoria e presença de imagem no enunciado apresentados em porcentagem.

	Zero	Acerto	Parcial	Não compila	TLE	MLE
Sem imagem	29,5	37,6	14,8	10	7	1
Com imagem	45,1	17,3	11,1	12,3	14,2	0

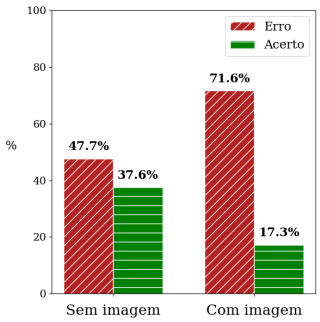

Figura 4: Porcentagens de erros e acertos por presença de imagem no enunciado.

Tabela 4: Frequência, pontuação média e acertos por tamanho de enunciado.

	Frequência	Pontuação média	Acertos (%)
Até 200 palavras	54	50,6	44,4
Entre 201 e 400 palavras	316	39,6	32,9
Mais que 400 palavras	90	21,5	13,3

foi de 320,5 palavras, com uma média de 2,7 exemplos de saída. Entre todos os enunciados, houve contagens de 122 até 745 palavras, enquanto o número de exemplos se limitou de 1 a 6. Não foi identificada discrepância no tamanho ou número de exemplos do enunciado entre os níveis e fases, o que poderia impactar a avaliação de pontuação/acertos ao agrupar por estas características da questão.

Os tamanhos médios dos enunciados em relação aos níveis foram, respectivamente: 283,8, 315,4, 346,3 e 326,3 palavras. Já em relação às fases, foram: 304,9, 334,3 e 323,4 palavras. Todos estes valores se encontram próximos da média geral, não indicando tendências. As médias de exemplos para cada nível e fase foram uniformes, dentro do intervalo (2.6, 2.7), permitindo avaliação mais clara das métricas de interesse.

Em relação aos 3 grupos de tamanho de enunciado que constam na Tabela 4, a divisão foi definida para isolar os extremos (enunciados curtos e longos) e caracterizar o grupo intermediário, que concentra a maioria dos casos (316 de 460). As porcentagens de vereditos **Não compila** para estes grupos foram, respectivamente: 7,4, 9,8 e 16,7. Quanto às porcentagens de vereditos **TLE**, temos: 2, 8,4 e 25,3. Considerando os grupos de número de exemplos que constam na Tabela 5, as porcentagens de vereditos **Não compila** foram, respectivamente: 16,7, 12,6, 10,6, 3,7, 0 e 0. Quanto às porcentagens de vereditos **TLE**, observa-se: 20, 12,7, 9,2, 3,8, 22,2 e 0. Finalmente, o tamanho do enunciado foi um parâmetro de impacto considerável, com tendência unânime e expressiva de proporcionalidade inversa ao desempenho. Quanto ao número de exemplos, pouco pode ser inferido, especialmente devido à acentuada diferença de frequência entre os grupos.

Tabela 5: Frequência, pontuação média e acertos por número de exemplos.

	Frequência	Pontuação média	Acertos (%)
1 ex.	6	33,3	33,3
2 ex.	198	37,4	32,9
3 ex.	218	38,5	28,9
4 ex.	27	38,9	33,3
5 ex.	9	11,1	11,1
6 ex.	2	10	0

5 CONCLUSÃO E TRABALHOS FUTUROS

Este estudo avaliou o Llama 3 para problemas de programação da OBI, utilizando o Juiz Online como métrica estrita para validar o modelo em um contexto educacional aplicado. Demonstrou-se que, apesar do potencial dos LLMs, o modelo investigado obteve apenas 30,4% de acertos, revelando inadequação para uso não supervisionado. O desempenho foi inversamente proporcional ao tamanho do enunciado e impactado pela ausência de contexto visual (-20,3% de acertos). A análise qualitativa revelou que as falhas se concentram na ineficiência (TLE) e na incorreção para casos de borda, indicando fragilidade no raciocínio algorítmico complexo. Para fins de reprodutibilidade do trabalho, os códigos implementados neste estudo estão disponíveis publicamente[6].

Trabalhos futuros podem realizar uma análise qualitativa das respostas geradas, em busca de uma engenharia de *prompt* que auxilie a interpretação do modelo e que otimize os resultados. Ademais, a experimentação com diferentes LLMs, incluindo modelos multimodais, projetados com diferentes propósitos, pode indicar melhores potenciais no ambiente educacional.

AGRADECIMENTOS

CAPES, FAPEMIG, INCT-TILD-IAR e PIBEN/UFV–FUNARBEN.

REFERÊNCIAS

[1] Celso Candido de Azambuja and Gabriel Ferreira da Silva. 2024. Novos desafios para a educação na Era da Inteligência Artificial. *Filosofia Unisinos* 25, 1 (2024), e25107.
[2] Aditi Bhutoria. 2022. Personalized education and Artificial Intelligence in the United States, China, and India: A systematic review using a Human-In-The-Loop model. *Computers and Education: Artificial Intelligence* 3 (2022).
[3] Mark Chen, Jerry Tworek, Heewoo Jun, Uri Shoham, Jeremiah Chung, Charles Lasecki, Sheng He, Lawrence Stock, and Barret Zoph. 2021. Evaluating Large Language Models Trained on Code. *arXiv preprint arXiv:2107.03374* (2021). arXiv:2107.03374 [cs.LG]
[4] Polat Ersoy and Mahmut Erşahin. 2024. Benchmarking Llama 3 70B for Code Generation: A Comprehensive Evaluation. *Orclever Proceedings of Research and Development* 4, 1 (2024), 52–58.
[5] Tung Phung, Victor-Alexandru Pădurean, Anjali Singh, Christopher Brooks, José Cambronero, Sumit Gulwani, Adish Singla, and Gustavo Soares. 2024. Automating human tutor-style programming feedback: Leveraging gpt-4 tutor model for hint generation and gpt-3.5 student model for hint validation. In *Proc.of the Learning Analytics and Knowledge Conference*. 12–23.
[6] Sami Sarsa, Paul Denny, Arto Hellas, and Juho Leinonen. 2022. Automatic generation of programming exercises and code explanations using large language models. In *Proc. of the ACM Conference on International Computing Education Research-Volume 1*. 27–43.
[7] Jaromir Savelka, Arav Agarwal, Christopher Bogart, Yifan Song, and Majd Sakr. 2023. Can generative pre-trained transformers (gpt) pass assessments in higher education programming courses?. In *Proc. of the Conference on Innovation and Technology in Computer Science Education V. 1*. 117–123.
[8] Priscylla Silva and Evandro Costa. 2025. Assessing large language models for automated feedback generation in learning programming problem solving. *arXiv preprint arXiv:2503.14630* (2025).
[9] João Vitor de Melo Cavalcante Souza. 2024. Avaliando modelos de linguagem grande na resolução de problemas de lógica da OBI. (2024).
[10] Yoshija Walter. 2024. Embracing the future of Artificial Intelligence in the classroom: the relevance of AI literacy, prompt engineering, and critical thinking in modern education. *International Journal of Educational Technology in Higher Education* 21, 1 (2024), 15.
[11] Shan Wang, Fang Wang, Zhen Zhu, Jingxuan Wang, Tam Tran, and Zhao Du. 2024. Artificial intelligence in education: A systematic literature review. *Expert Systems with Applications* 252 (2024), 124167.
[12] Lixiang Yan, Lele Sha, Linxuan Zhao, Yuheng Li, Roberto Martinez-Maldonado, Guanliang Chen, Xinyu Li, Yueqiao Jin, and Dragan Gašević. 2024. Practical and ethical challenges of large language models in education: A systematic scoping review. *British Journal of Educational Technology* 55, 1 (2024), 90–112.

[6] https://github.com/andre-n-alcantara/IC