

Coding Dojo no aprendizado de programação: Uma Revisão sistemática

Paulo Cezar de Oliveira¹, Adolfo Gustavo Serra Seca Neto¹

Av. Sete de Setembro, 3165. Curitiba - Paraná

Abstract

O aprendizado de programação de computadores se mostra, por vezes, de difícil compreensão. Para minimizar essa dificuldade são empregadas algumas técnicas de programação ágil como por exemplo, programação pareada e desenvolvimento orientado a testes. Coderetreat e coding dojo são dois eventos que também utilizam métodos ágeis para, de forma colaborativa, compartilhar conhecimento entre seus participantes, além de permitir o aprendizado de uma linguagem de programação; Coding Dojo, tem se mostrado promissor na proposta de ensino de programação de computadores, por ter uma característica democrática onde os desenvolvedores apresentam passo a passo o desenvolvimento do código, esse código não precisa ser complexo, e nem existe a necessidade de concluí-lo, coding dojo também não é uma competição entre programadores. seu objetivo é apenas compartilhar novas formas de desenvolver e se divertir.

Keywords: Coding Dojo, Test-Driven Development - TDD, pair programming, teaching, Coderetreat

1. Introdução

O aluno pode encontrar muitos atrativos para uma aprendizagem eficiente, entretanto, ao se deparar com um ambiente acadêmico esses atrativos nem sempre são evidentes, por vezes não existem.

¹Universidade Tecnológica Federal do Paraná.

5 Toda a metodologia de ensino ainda é focada na escola tradicional, onde há um professor que expõe seu conteúdo de forma direta, e o aluno absorve aquele conteúdo fazendo suas anotações [1], sempre baseadas no que foi exposto, para em seguida, de alguma forma, praticar.

Esse método de ensino alcança uma parcela de alunos, porém, a falta de
10 atrativo e desafios leva o mesmo a se limitar ao que é exposto pelo professor, sem procurar desenvolver sua própria habilidade de aprendizagem, sem auto impor desafios e sem conhecer outras formas de desenvolver suas capacidade intelectual.

As práticas no ensino de programação têm se mostrado trabalhosas para
15 alguns docentes e difícil de entender para muitos discentes. Diante dessas circunstâncias alguns alunos avançam em seu curso sem a base necessária. É importante motivar o aprendizado, e novas metodologias são sempre vistas com bons olhos, entretanto, empregar novas metodologias se baseado apenas em uma suposição não traz, necessariamente, bons resultados.

20 O método de programação chamada de Coding Dojo tem sido aplicado em ambientes de desenvolvimento de software com o intuito de motivar o uso de novas linguagens de programação bem como apresentar maneiras de chegar a um resultado satisfatório no código fonte do programa em questão. Com esse estudo pretendemos elencar as técnicas empregadas em Coding Dojo, para analisar sua
25 aplicação no aprendizado de programação de computadores.

O objetivo deste trabalho é *analisar* se a aplicação de Coding Dojo no aprendizado de programação de computadores no ambiente acadêmico é viável.

OBJETIVOS ESPECÍFICOS

Coderetreats é um evento cujo propósito é reunir programadores para que
30 durante um dia inteiro, em um ambiente informal, realizem práticas intensiva em desenvolvimento de software, o objetivo do evento é se concentrar nos princípios básicos do design sem levar em consideração a pressão para concluir o projeto. Outro objetivo do evento é o de incentivar a prática reflexiva ao abordar um único problema de programação com diversos parceiros de desenvolvimento [2].

35 Popularizado por Corey Haines, que também criou o dia mundial do Codere-

treat, é um evento anual onde programadores se reúnem em diversas partes do mundo para participar de eventos relacionados ao programa, toda a interação entre os participantes pode acontecer pelas redes sociais e todos trocam experiências sobre melhores práticas de desenvolvimento. [3],

40 Com características semelhantes ao Coding Dojo, sua principal diferença está na sistemática de codificação além de ocorrer durante um dia inteiro.

Na sua estrutura, o dia consiste em 5 a 6 sessões de aprendizado cujo período da manhã é dedicado à compreensão do problema a ser solucionado, quebrando velhos hábitos e propondo a autodescoberta. no período da tarde, então iniciam-
45 se os desafios, onde programadores colocam em prática suas habilidades de programação [4].

O formato original do coderetreat consiste dos seguintes tópicos [4]:

- Problema a ser solucionado: Conway's Game Life;
- Duração de cada sessão: 45 minutos;
- 50 • Duração do evento: das 8h30min até às 17 ou 18 horas;
- A programação pareada é fundamental para que haja a troca de conhecimento;
- É importante utilizar Test-Driven Development (TDD) durante o processo de desenvolvimento;
- 55 • Após cada sessão o código criado deve ser deletado.

Mesmo sendo um evento para programadores experientes, Parsons [5] e [6], levanta a preocupação sobre um fenômeno que vem ocorrendo nos eventos ao longo do tempo. Ele afirma que esse tipo de atividade vem atraindo programadores com um amplo espectro de idade e experiência. Isso além do surgimento de novas ferramentas de programação bem como a evolução de outras
60 já existentes, Então ele sugere uma abordagem mais suscetível a esse público através eventos mais eficazes para esse perfil de desenvolvedor.

De acordo com [7], Test-Driven Development (TDD) e programação pareada são dois conceitos centrais da Extreme Programming (XP).

65 TDD é uma forma de desenvolver testes de unidade automatizados em códigos de programas, esses testes podem tanto ser escritos pelo próprio programador, quanto por outro profissional com a função exclusiva de testar códigos.

Sua execução pode ser manual ou automatizada. Usualmente, esses testes ocorrem após uma unidade ser codificada, e esse processo pode levar desde 70 poucos minutos até alguns meses [8].

O uso de TDD ajuda no desenvolvimento orientado a testes e na explicação da estrutura hierárquica de uma funcionalidade básica do código em desenvolvimento, esse teste é construído e aplicado antes de ser implantado no código original, dessa forma, é possível saber se tal funcionalidade é viável ou não para 75 ser utilizado no código original.

Caso algum teste falhe, o código é corrigido antes de dar continuidade no desenvolvimento uma vez testado e aprovado o código pode ser incluído na sua origem, e novos testes podem ser feitos. Com esse processo é possível identificar e corrigir rapidamente possíveis falhas antes da implementação efetiva [9].

80 Quando o desenvolvimento orientado para testes começa a ficar longo e de difícil compreensão, é necessário simplificar o código utilizando um passo-a-passo, conhecido como *baby steps*, seu objetivo é fazer mudanças sutis no desenvolvimento do código de forma que, além de obter uma melhor solução, também seja possível que outros compreendam todo o processo de desenvolvimento, sendo inclusive, possível sua replicação posterior [10], [11]. 85

Programação pareada (PP) do inglês *pair programming*, é uma das principais práticas de *Extreme Programming* (XP). Seu sistema colaborativo de desenvolvimento de software consiste em dois programadores trabalhando juntos, em um mesmo computador e com um único teclado e mouse trocando experiências e 90 sugestões sobre o trabalho em desenvolvimento [12].

Em PP existem duas funções distintas, porém, mutuamente colaborativas, o piloto, que assume o teclado e aplica as ideias ao código e o copiloto, que acompanha todo o processo de desenvolvimento orientando e dando sugestões sobre

como melhorar o código. Essas funções não são permanentes elas podem ser
95 trocada entre os envolvidos a qualquer momento de acordo com o entendimento
de ambos [13].

Em estudos realizados foram constatados que o desenvolvimento em pares
tende a ter menor índice de erros do que o mesmo trabalho realizado por um
único programador. Mais pessoas estão familiarizadas com o código, além de re-
100 duzir o tempo decorrido no processo de desenvolvimento. Outro fator relevante
para essa técnica de programação é que a troca de conhecimento traz maior
satisfação com o trabalho. [14], [15].

Segundo [11], a programação pareada é uma boa maneira de motivar a pro-
dutividade, enquanto o programador não percebe pequenos erros, uma segunda
105 pessoa poderá facilmente perceber-los e sugerir melhorias no código.

Mesmo sendo um conceito com pouco mais de uma década, ao pesquisar o
tema de estudo desse trabalho, constatou-se que ainda é pouco explorado, não
são muitos os pesquisadores que se dedicam ao assunto. Todas as buscas foram
realizadas nas bases de pesquisas da *IEEE Software Magazine*, *Information and*
110 *Software Technology*, *Association for Computing Machinery – ACM* e *Sociedade*
Brasileira de Computação - SBC.

Uma outra forma de pesquisa da expressão coding dojo foi o acréscimo de
aspas duplas na frase, dessa forma as bases de pesquisas listadas a seguir apre-
sentaram os seguintes resultados para a expressão "coding dojo":

- 115 • Scopus = 10 resultados.
- ACM DL = 3 resultados.
- IEEE Explorer = 4 resultados.

As atividades para a elaboração do artigo estão representadas no cronograma de
atividades, apresentado na figura 2. Esta figura demonstra com maior detalhe
120 o andamento de cada etapa do trabalho (anexo 3).

2. Estado da Arte

O ensino de programação utilizando Coding Dojo, permite ao que o aluno entender que não existe apenas um método ou maneira para o desenvolvimento de software. O mesmo problema pode ter outras formas de ser abordado.

125 É fundamental associar ao Coding Dojo as técnicas de programação pareada, e TDD, já que a Programação Orientada a Testes se mostrou eficiente em estudo elaborado por [16].

2.1. Coding Dojo

O Kata Dojo, foi o precursor do coding dojo, proposto por Dave Thomas
130 ². Então, baseado na proposta do kata code, Laurent Bossavit propôs o coding dojo, que teve sua primeira apresentação em Paris [3].

Coding Dojo se trata de um encontro organizado, preferencialmente com poucos participantes, ([17] e [18]), onde programadores e interessados em aprender se reúnem com o objetivo de trocar experiência, se divertir, discutir temas
135 como design, testes, refatoração, escolha do editor de código e ferramentas. [3].

Essa sistemática permite encontrar soluções simplificadas para problemas também simples além de permitir que o público acompanhe a linha de raciocínio dos desenvolvedores podendo inclusive fazer sugestões quando permitido ou solicitado.

140 Mesmo que interessados em aprender alguma linguagem de programação possam participar do Coding Dojo, seu propósito não é ensinar programação, e sim mostrar formas práticas e outras ideias de como chegar a um resultado.

O conhecimento de uma linguagem de programação acontece por inferência em um ambiente favorável e aberto a novas ideias [19]. De acordo com Luz [20],
145 a intenção da metodologia é que, ao fim, se tenha um software com uma boa cobertura de testes.

Segundo Bache [3], mesmo sendo flexível e de fácil adaptação, ainda há

²<http://codekata.cm/>

alguns poucos elementos obrigatórios, e são esses que realmente definem um Coding Dojo de outra reunião de programadores:

- 150 • *introdução e retrospectiva*: Tanto a introdução, quanto a moderação do Dojo devem ser feitas de forma que os participantes sintam-se a vontade para experimentar e aprender. E a retrospectiva, deve provocar a reflexão dos participantes sobre o aprendizado naquele Dojo.
- 155 • *Codificação e testes*: Os testes são um mecanismo eficiente para demonstrar se o código realmente está de acordo com o definido.
- 160 • *Mostre seu trabalho* Demonstrando como está sendo desenvolvido o código é possível que outros possam entender o processo, além de permitir opiniões sobre uma melhor maneira de desenvolvimento.
- 160 • *Tenha um moderador ou facilitador* É a figura que vai orientar e dirigir o Dojo, além de conduzir todo o processo no Coding Dojo.

Como Coding Dojo é um evento aberto a uma plateia, todos podem participar e ao término do encontro é importante saber dos participantes suas impressões e opiniões, coletadas através de algumas perguntas e exposta no mesmo instante.

- 165 • O que aprendemos?
- 165 • O que gostamos?
- 165 • O que poderia melhorar?
- 165 • Discussões ou comentários.

O que é preciso para o encontro.

- 170 • Piloto, copiloto e plateia.
- 170 • Ambiente espaçoso.
- 170 • Projetor multimídia e tela computador.
- 170 • Quadro para definir o problema.

Pontos importantes do Coding Dojo.

- 175 • Não é uma competição;
- Os problemas não devem ser complexos;
- Não há a necessidade achar a solução para o problema proposto;
- O problema não precisa ser resolvido.

Usualmente o Coding Dojo acontece se utilizando de práticas ágeis de desenvolvimento[20]
180 e possui três formatos:

Randori. A solução de um problema é realizada em tempo real, em um computador conectado a um projetor multimídia e dois programadores, o piloto e o copiloto que interagem entre si. Nesse formato é empregado TDD e baby step.

Enquanto o piloto assume o teclado digitando suas ideias o copiloto, por sua
185 vez, sugere outras maneiras que possam ser aplicadas ao código. O tempo de duração de cada dupla varia entre 5 a 10 minutos, terminado esse tempo o piloto deixa o teclado, o copiloto assume a posição de programador e um integrante da plateia assume a posição de copiloto [17].

Kata. Diferente do Randori, a solução de um problema já foi resolvida anteriormente pelo apresentador (ou por um grupo), os resultados da solução são
190 apresentados aos participantes e são desafiados a alcançar o mesmo resultado sem conhecer o código original.

Nesse formato a plateia pode interagir com os programadores fazendo sugestões. O objetivo é que todos sejam capazes de reproduzir os passos e resolver
195 o problema [17].

Take. Suas diferenças em relação ao Randori são que, várias duplas trabalham simultaneamente e todos devem ter conhecimento avançado de programação. A cada período as duplas são trocadas, permitindo que os participantes se envolvam entre si.

200 3. Conclusão

O ensino de programação pode ser considerado um desafio aos professores, como discutido anteriormente neste trabalho, métodos de ensino tradicionais nem sempre alcançam uma parcela de alunos, é necessário descobrir e testar novas técnicas, com o objetivo de atingir um grupo de estudantes que apresentam maior dificuldade na compreensão da programação de computadores. Algumas técnicas ágeis de programação têm sido amplamente discutidas em vários estudos científicos, existem muitos trabalhos propondo a utilização de Test-Driven Development e pair programming no aprendizado. Neste trabalho, foram apresentadas as práticas de programação ágil através do Coding Dojo, mesmo sendo um evento voltado para a troca de experiência, essa prática se mostra promissora, pois, suas características dinâmica, inclusiva e favorecendo o aprendizado, permitem que alunos troquem experiências de forma colaborativa [10].

Segundo Sato *et al* [17], o principal objetivo do Coding Dojo, é o aprendizado através da prática. Neste caso, ao observar e praticar, o aluno adquire habilidades que pouco provavelmente não desenvolveria no método tradicional de aprendizado. com exceção do Kake, tanto Randori, quanto Kata tem boas condições de serem utilizados no ensino de programação de computadores.

Referências

- [1] D. Saviani, Escola e Democracia, 4th Edition, Cortez Editora, 1991.
- 220 [2] D. Parsons, A. Mathrani, T. Susnjak, A. Leist, Coderetreats: Reflective practice and the game of life, IEEE Software 31 (4) (2014) 58–64. doi: 10.1109/MS.2014.25.
- [3] E. Bache, The Coding Dojo Handbook a practical guide to creating a space where good programmers can become great programmers, 2013.
- 225 [4] C. Haines, Structure of a coderetreat, Coderetreat Community Network. URL <http://coderetreat.org/>

- [5] D. Parsons, T. Susnjak, A. Mathrani, The software developer cycle: Career demographics and the market clock: Or, is sql the new cobol?, in: Proceedings of the ASWEC 2015 24th Australasian Software Engineering Conference, ASWEC '15 Vol. II, ACM, New York, NY, USA, 2015, pp. 86–90. doi:10.1145/2811681.2811698.
URL <http://doi.acm.org/10.1145/2811681.2811698>
- [6] D. Parsons, T. Susnjak, A. Mathrani, Design from detail: Analyzing data from a global day of coderetreat, Information and Software Technology 75 (2016) 39–55. doi:10.1016/j.infsof.2016.03.005.
- [7] M. M. Müller, A. Höfer, The effect of experience on the test-driven development process, Springer Science + Business Media doi:10.1007/s10664-007-9048-2.
- [8] D. Janzen, H. Saiedian, Test-driven development: Concepts, taxonomy, and future direction, IEEE Computer Society.
- [9] J. Pitt-Francis, P. Pathmanathan, M. O. Bernabeu, R. Bordas, J. Cooper, A. G. Fletcher, G. R. Mirams, P. Murray, J. M. Osborne, A. Walter, S. J. Chapman, A. Garny, I. M. van Leeuwen, P. K. Maini, B. Rodríguez, S. L. Waters, J. P. Whiteley, H. M. Byrne, D. J. Gavaghan, Chaste: A test-driven approach to software development for biological modelling, Computer Physics Communications.
- [10] R. B. Luz, A. Neto, Usando dojos de programação para o ensino de desenvolvimento dirigido por testes, Anais do 23º Simpósio Brasileiro de Informática na Educação (SBIE 2012) (ISSN 2316-6533).
- [11] K. Beck, Test-Driven Development By Example, Addison Wesley, 2002.
- [12] K. Beck, C. Andres, Extreme Programming Explained: Embrace Change, Addison-Wesley, 2005.

- [13] L. Plonka, J. Segal, H. Sharp, , J. van der Linden, Collaboration in pair programming: Driving and switching, Springer-Verlag Berlin Heidelberg LNBIP 77 (2011) 43–59.
- [14] I. Fronza, A. Sillitti, G. Succi, J. Vlasenko, Analysing the usage of tools in pair programming sessions, Springer-Verlag Berlin Heidelberg LNBIP 77 (2011) 1–11.
- [15] F. Zieris, L. Prechelt, On knowledge transfer skill in pair programming, in: Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM '14, ACM, New York, NY, USA, 2014, pp. 11:1–11:10. doi:10.1145/2652524.2652529. URL <http://doi.acm.org/10.1145/2652524.2652529>
- [16] J. Boydens, P. Cordemans, H. Hallez, On using test-driven development to tutor novice engineering students using self-assessment, 43rd Annual SEFI Conference June 29 - July 2, 2015 Orléans, France.
- [17] D. T. Sato, H. Corbucci, M. V. Bravo, Coding dojo: An environment for learning and sharing agile practices, in: Agile, 2008. AGILE '08. Conference, 2008, pp. 459–464. doi:10.1109/Agile.2008.11.
- [18] K. Heinonen, K. Hirvikoski, M. Luukkainen, A. Vihavainen, Learning agile software engineering practices using coding dojo, in: Proceedings of the 14th Annual ACM SIGITE Conference on Information Technology Education, SIGITE '13, ACM, New York, NY, USA, 2013, pp. 97–102. doi:10.1145/2512276.2512306. URL <http://doi.acm.org/10.1145/2512276.2512306>
- [19] M. Aniche, Test-Driven Developement. Teste e Design no Mundo Real, Casa do Código, 2012.
- [20] R. B. da Luz, A. G. S. S. Neto, R. V. Noronha, Teaching tdd, the coding dojo style, in: 2013 IEEE 13th International Conference on Advanced Learning Technologies, 2013, pp. 371–375. doi:10.1109/ICALT.2013.114.

Apêndice

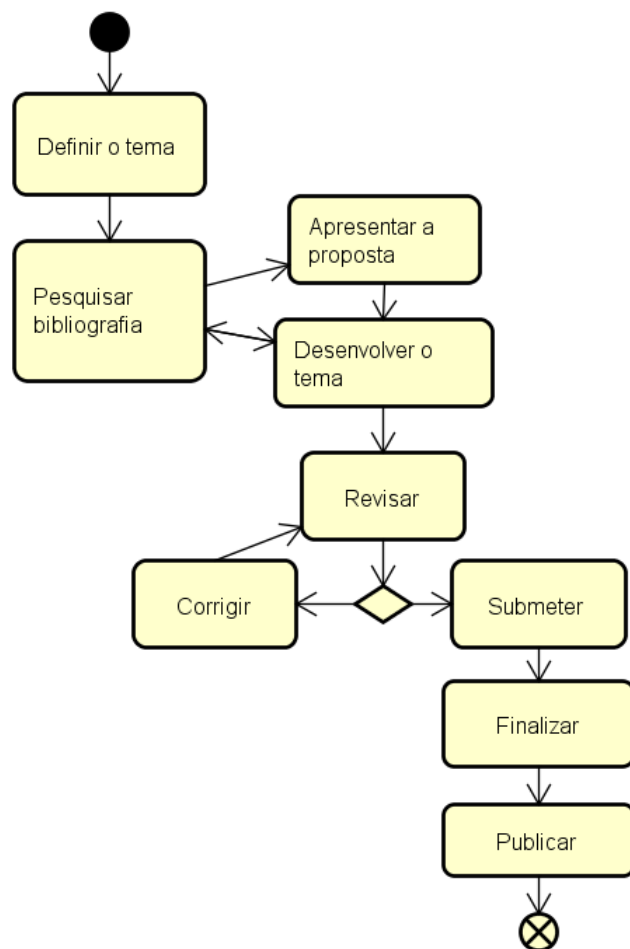


Figure 1: Diagrama de atividades

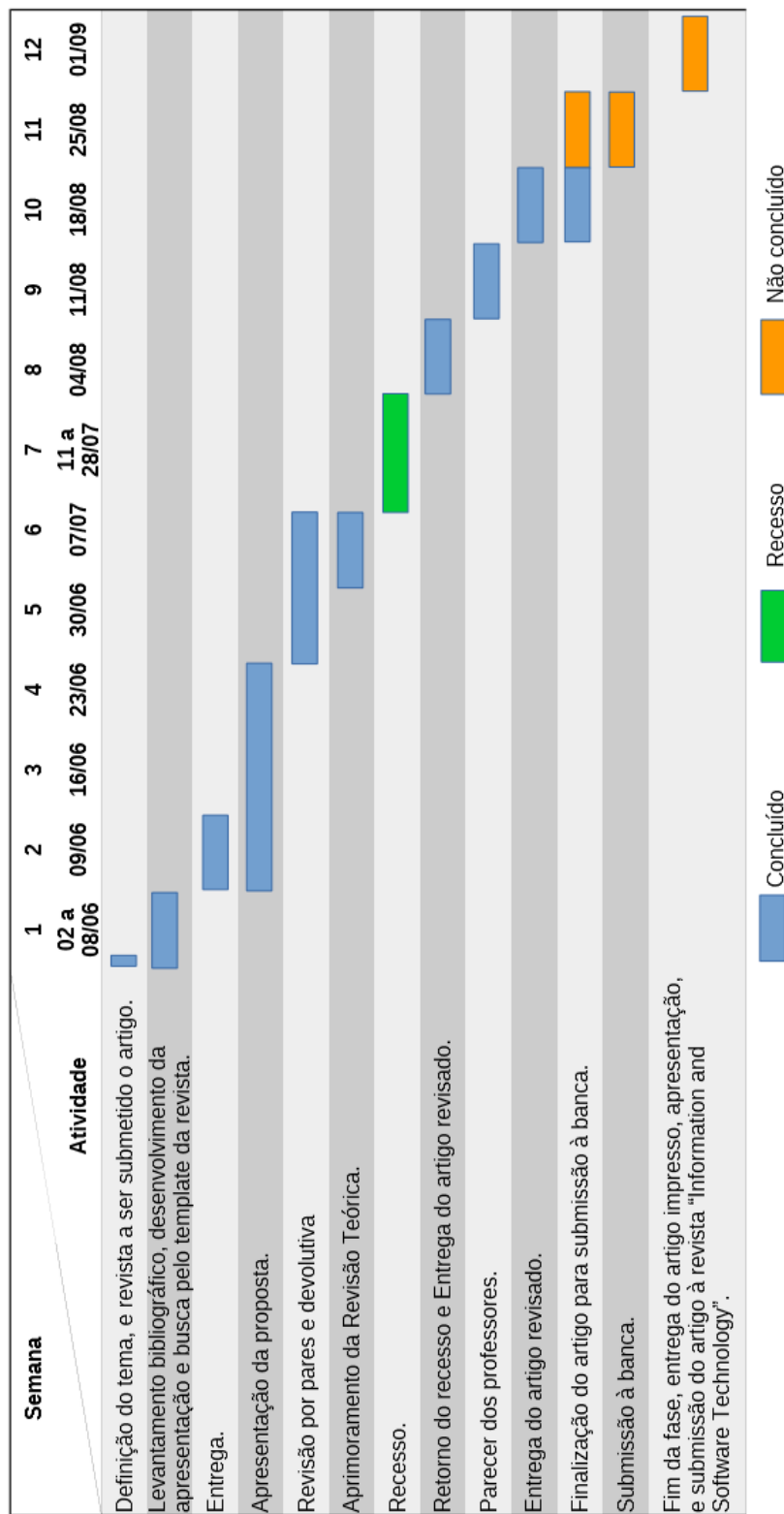


Figure 2: Cronograma de atividades