Título do Projeto

Some Author[1], Another Author[2] & Yet Another Author[2]
[1]Departamento Acadêmico de Computação (DACOM)
Universidade Tecnológica Federal do Paraná (UTFPR)

Abstract

This the article template for the journal "Constructions". Its goal is to make it as easy as possible for authors to typeset their papers in LaTeX. For users with limited LaTeX experience, we recommend using Overleaf (https://www.overleaf.com). To start working on your paper in Overleaf, just make a copy of this template and replace the text of this tutorial by the text of your paper.

Resumo

This the article template for the journal "Constructions". Its goal is to make it as easy as possible for authors to typeset their papers in LaTeX. For users with limited LaTeX experience, we recommend using Overleaf (https://www.overleaf.com). To start working on your paper in Overleaf, just make a copy of this template and replace the text of this tutorial by the text of your paper.

1 Instruções

Este é o *template* LaTeX para ser utilizado como formato para o trabalho.

"Constructions"is a platinum open-access journal: We do not charge author processing fees, and we do not pursue any commercial interests. However, this also means that the journal is run on a voluntary basis, and we do not have money to pay professional typesetters. We therefore have to ask the authors to do the typesetting of their papers. To facilitate this task as much as possible, we provide this LaTeX template. Just open the template in the LaTeX editor of your choice and start working in the document main.tex by replacing the text of this short tutorial with the text of your own paper. You can just work as would in a simple text editor; if you have never worked with LaTeX, there are a few commands you should know:

- If you want to use *italics*, you have to wrap the text to be italicized in the command `\textit{...}`
- If you want to use **boldface**, use the command `\textbf{...}`
- For adding 'single' and "double"quotes, please use a single or double grave accent (` or ``) for opening the quote and the apostrophe ' or the regular double quote " to close it.
- To start a new **section**, use `\section{...}` If the section should not be numbered (which is usually the case for Acknowledgment sections and the references), add an asterisk: `\section*{...}`
- If you want to make a bulletpoint list like the one you're reading right now, use:

```
\begin{itemize}
    \item first item
    \item second item
    \item etc.
\end{itemize}
```

Corresponding author(s): Random Author, University of Somewhere, random.author@uni-somewhere.edu

- For linguistic examples, see below.
- Footnotes can be inserted using `\footnote{...}`.[1]
- LaTeX hyphenates automatically, and usually the hyphenation feature is pretty good. If you want to override the default hyphenation of a word, add it to the `\hyphenation{}` word list towards the beginning of the main.tex document – that's what I've done for the word *numb-ered* a few bulletpoints above (just for expository reasons of course, not because I want to change English hyphenation rules...). If you have very long words, I recommend to insert a **soft hyphen**, i.e. a hyphen that is not shown unless there is a line break, asin-thisverylongword. You can insert a soft hyphen by typing `\-`, e.g. `hy\-phe\-na\-tion`.

1.1 Figuras

Inserting figures is easy, while it can be a bit harder to place them where you would like to see them. The Overleaf tutorial provides a good starter. Usually you can just work with the code in this template. Please make sure that you have high-resolution figures (at least 300 dpi). Like the linguistic examples discussed above, you can label figures and then refer to them using `\ref{the_label}` or, if you want to add the word "Figure"(or "Table", in the case of tables) automatically, `\autoref{the_label}`.

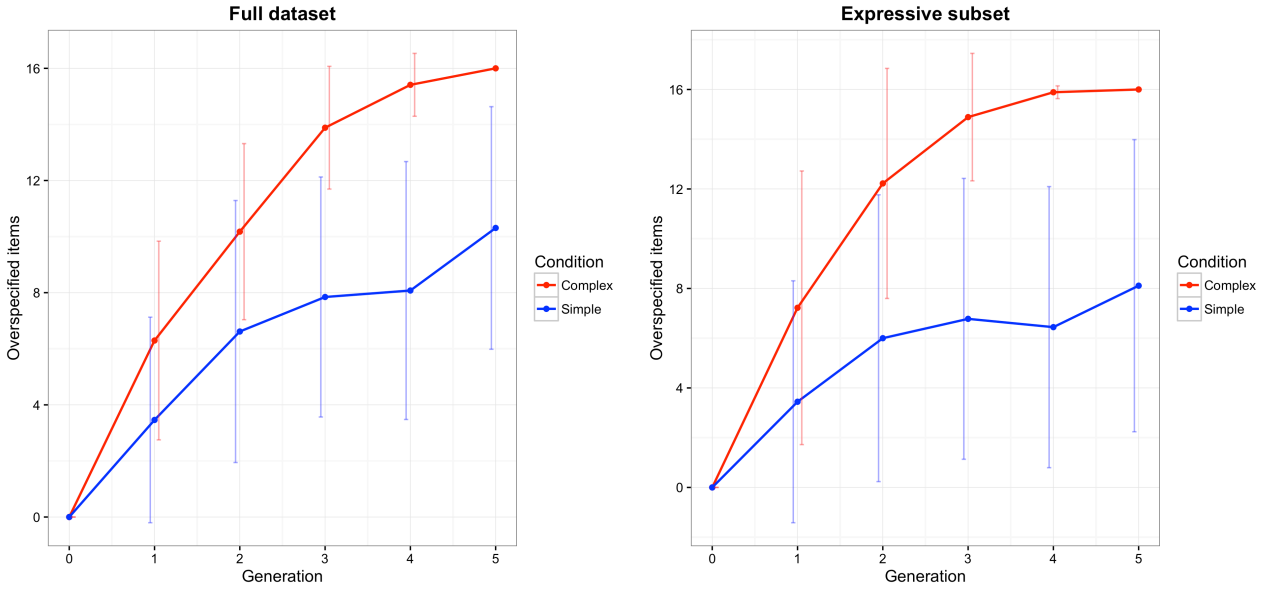

Figura 1: A sample figure

The float package can be used to customize the position of figures and tables. For example, Figura 1 and Figura 2 are forced to appear at a specific position in the text using the `[H]` command. If there is no reason to position a figure or table at a specific point in the text, you can also use a less strict configuration and let LaTeX choose the position automatically. If you use `[H]` and start a new paragraph afterwards, please make sure to start the paragraph with the command `\noindent` to avoid indentation.

[1] Duh!

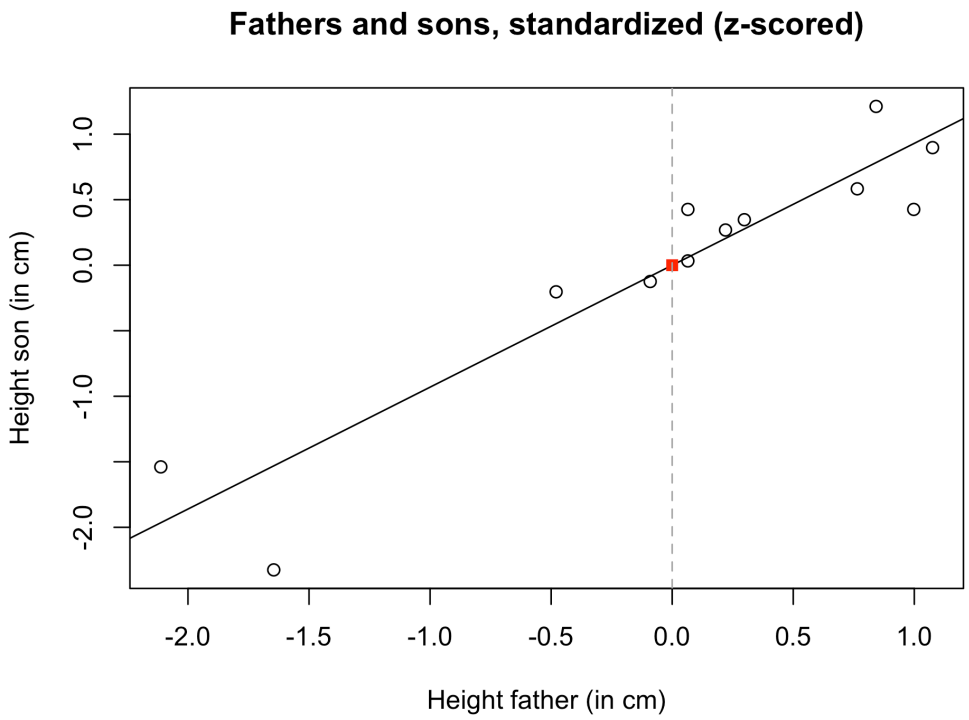

Figura 2: Another sample figure

1.2 Tabelas

Adding tables is, admittedly, one of the more challenging aspects of LaTeX typesetting. Luckily, there are a few resources that help you generate tables, such as https://www.tablesgenerator.com/.

Tabela 1: A very simple table

bla	blubb
blah	blub

I recommend to use the tabularx package, which is a bit more complicated to use than the normal "tabular"environment, but it gives you more flexibility in customizing the tables, and makes it easier to print it in text width. In the localstuff.tex file, I have defined the column types L (left-aligned), C (centered), and R (right-aligned) so you don't have to use the full commands that tabularx usually requires. In the simplest case, therefore, constructing a table is as easy as in the source code of Tabela 1. For more elaborate tables in which some rows and columns are merged, check the source code of Tabela 2.

Tabela 2: A more complicated table

Col1	Col2	Col3
blubb	blibb	2
Lorem ipsum dolor sit amet,		bla
Lorem ipsum dolor sit amet,		
bla	blabb	22

1.3 Códigos

Para apresentar algum trecho ou código completo iremos utilizar o pacote *Listings* do Latex.

Lista de Códigos https://www.overleaf.com/project/6499cc2d5a298ea86cde0f74 ou https://www.overleaf.com/learn/latex/Code_Highlighting_with_minted

O Código 1 apresenta um exemplo de código de contrato inteligente.

```
pragma solidity ^0.8.28;

contract Addition {
   uint8 x;

   function addx(uint8 y, uint8 z ) public {
      x = y + z;
   }
   function retrievex() view public returns (uint8) {
      return x;
   }
}
```

Código 1: Teste de Caption

1.4 Citações e referências

"Constructions"follows the Unified Style Sheet for Linguistics Journals. Working with LaTeX and BibTeX, you don't have to do the formatting manually, though. All references you cite should be in the bibliography.bib file in BibTeX format. We strongly recommend using a reference manager like Zotero. Zotero allows you to copy and paste entries in BibTeX format, thus making it much easier to work with LaTeX.

For inserting citations, use LaTeX's citation syntax:

- Use `\citep{...}` for inserting citations in parantheses, e.g. (Antonopoulos & Wood 2018).
- Use `\citet{...}` to insert citations in the format Author/s (Year), as in Antonopoulos & Wood (2018).
- Use `\citeauthor{...}` and `\citeyear{...}` if you just want to cite the author or the year, without parantheses. This can be helpful if you cite something within parantheses (as in this example, see Antonopoulos & Wood 2018).
- If you want to use a possessive 's in references, you can use the custom command `\citegen{...}` defined in the localstuff.tex file, as in: Antonopoulos & Wood's (2018) definition of constructions.

Troubleshooting

If there are problems with the bibliography, this172.16.255.69 can have several reasons:

- Some Zotero plugins (like Better BibTeX) change the "Year"field in the BibTeX entry to "Date". In this case, you'll have to change all instances of "Date"to "Year"in the bibliography file.
- If you want to keep sentence-internal capitals in the citations (e.g. when quoting German titles, such as `? ?`), make sure that the relevant words are enclosed in curly brackets. If you export the BibTeX entries from Zotero, you usually won't have to worry because it does so automatically.

2 Seções Esperadas

Aqui apresentamos uma sugestão de estrutura para o trabalho, com seções esperadas.

1. **Introdução:** Falar de Tecnologias Blockchain de forma geral. Contextualizar e falar que irá trabalhar com Ethereum. Dar uma introdução sobre Tokenização.
2. **Ethereum:** Falar sobre `EVM`. Redes: principal e testes.
3. **Contratos Inteligentes:** Falar o que são Contratos. Seu uso e utilidade. Citando o artigo original Szabo (1997).
4. **Solidity e implementação de Contratos Inteligentes:** Falar sobre a linguagem de desenvolvimento. Falar sobre a Implementação.
5. **Tokenização:** O que são tokens. Padrões de Implementação. Funcionamento e utilidade.
6. **Ferramentas de Desenvolvimento:** Falar sobre as ferramentas que utilizou no projeto. Redes de Teste, Ganache ou Locais. Truffle ou Hardhat para a automatização do processo de compilação e deploy.
7. **Especificação e Desenvolvimento do Projeto:** Ambiente de desenvolvimento. Preparação das ferramentas. Configurações para o projeto. Telas do Fluxo de Execução do Projeto.
8. **Considerações Finais:** Apresentar as considerações finais, conclusões...
9. **Referências:** Apresenta as referências de artigos, livros e outras fontes utilizadas no estudo e no desenvolvimento do trabalho.

Referências

Antonopoulos, A.M. & G. Wood. 2018. *Mastering ethereum: Building smart contracts and dapps*. O'Reilly Media, Incorporated. https://books.google.com.br/books?id=SedSMQAACAAJ.

Szabo, Nick. 1997. Formalizing and securing relationships on public networks. *First Monday* 2(9). http://dblp.uni-trier.de/db/journals/firstmonday/firstmonday2.html#Szabo97.