



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA  
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**RAPHAEL DE PAULA FONSECA**

**SOL PROCEEDINGS:  
EXTENSÃO CHROME PARA A CRIAÇÃO DE PROCEEDINGS  
PARA A SOL**

**CAMPINA GRANDE - PB**

**2024**

**RAPHAEL DE PAULA FONSECA**

**SOL PROCEEDINGS:  
EXTENSÃO CHROME PARA A CRIAÇÃO DE PROCEEDINGS  
PARA A SOL**

**Trabalho de Conclusão Curso  
apresentado ao Curso Bacharelado em  
Ciência da Computação do Centro de  
Engenharia Elétrica e Informática da  
Universidade Federal de Campina  
Grande, como requisito parcial para  
obtenção do título de Bacharel em  
Ciência da Computação.**

**Orientador: Professor Dr. Rohit Gheyi**

**CAMPINA GRANDE - PB**

**2024**

**RAPHAEL DE PAULA FONSECA**

**SOL PROCEEDINGS:  
EXTENSÃO CHROME PARA A CRIAÇÃO DE PROCEEDINGS  
PARA A SOL**

**Trabalho de Conclusão Curso  
apresentado ao Curso Bacharelado em  
Ciência da Computação do Centro de  
Engenharia Elétrica e Informática da  
Universidade Federal de Campina  
Grande, como requisito parcial para  
obtenção do título de Bacharel em  
Ciência da Computação.**

**BANCA EXAMINADORA:**

**Professor Dr. Rohit Gheyi  
Orientador – UASC/CEEI/UFCG**

**Professor Dr. Hyggo Oliveira de Almeida  
Examinador – UASC/CEEI/UFCG**

**Professor Dr. Francisco Vilar Brasileiro  
Professor da Disciplina TCC – UASC/CEEI/UFCG**

**Trabalho aprovado em: 15 de MAIO de 2024.**

**CAMPINA GRANDE - PB**

## RESUMO

A Sociedade Brasileira de Computação (SBC) promove diversos eventos científicos e alguns deles são publicados na biblioteca digital SOL (SBC OpenLib). Após o processo de avaliação dos artigos ser concluído, os coordenadores de eventos (*chairs*) precisam preparar a documentação para enviar para a publicação dos anais dos eventos (*proceedings*) na biblioteca digital. No entanto, a quantidade de artigos e o trabalho manual envolvido na extração dos campos para a criação dos *proceedings* demandam muito tempo, tornando o processo lento e sujeito a erros humanos. Além disso, não existe uma ferramenta que extraia todos os campos de forma automatizada. Dessa forma, o objetivo deste trabalho é desenvolver uma ferramenta que reduza o máximo possível esse trabalho manual no contexto de trabalhos publicados na plataforma SOL e que seguem o estilo da SBC. Com isso, a maior parte da informação será extraída a partir do conteúdo da versão final dos artigos em formato PDF. E a *SOL Proceedings* é uma extensão do navegador Google Chrome devido a facilidade para o usuário que ela fornece. Alguns testes com *proceedings* já publicados foram realizados, e a ferramenta consegue reduzir o esforço na preparação dos *proceedings*.

# **SOL PROCEEDINGS: CHROME EXTENSION FOR CREATING SOL PROCEEDINGS**

## **ABSTRACT**

The Brazilian Computing Society (SBC) organizes various scientific events, some of which are published in the SOL digital library (SBC OpenLib). After the article evaluation process is completed, event coordinators (chairs) need to prepare the documentation for publication of the event proceedings in the digital library. However, the quantity of articles and the manual work involved in extracting fields for proceedings creation consume a lot of time, making the process slow and prone to human errors. Additionally, there is no tool that extracts all fields automatically. Therefore, the aim of this work is to develop a tool that minimizes manual work as much as possible in the context of works published on the SOL platform and following the SBC style. Consequently, most of the information will be extracted from the content of the final version of the articles in PDF format. SOL Proceedings is a Google Chrome browser extension due to the user-friendly interface it provides. Some tests with already published proceedings have been conducted, and the tool successfully reduces the effort in preparing the proceedings.

# SOL Proceedings: extensão Chrome para a criação de proceedings para a SOL

Raphael de Paula Fonseca  
Universidade Federal de Campina Grande  
Campina Grande, Paraíba  
raphael.fonseca@ccc.ufcg.edu.br

Rohit Gheyi  
Universidade Federal de Campina Grande  
Campina Grande, Paraíba  
rohit@dsc.ufcg.edu.br

## RESUMO

A Sociedade Brasileira de Computação (SBC) promove diversos eventos científicos e alguns deles são publicados na biblioteca digital SOL (SBC OpenLib). Após o processo de avaliação dos artigos ser concluído, os coordenadores de eventos (*chairs*) precisam preparar a documentação para enviar para a publicação dos anais dos eventos (*proceedings*) na biblioteca digital. No entanto, a quantidade de artigos e o trabalho manual envolvido na extração dos campos para a criação dos *proceedings* demandam muito tempo, tornando o processo lento e sujeito a erros humanos. Além disso, não existe uma ferramenta que extraia todos os campos de forma automatizada. Dessa forma, o objetivo deste trabalho é desenvolver uma ferramenta que reduza o máximo possível esse trabalho manual no contexto de trabalhos publicados na plataforma SOL e que seguem o estilo da SBC. Com isso, a maior parte da informação será extraída a partir do conteúdo da versão final dos artigos em formato PDF. E a *SOL Proceedings* é uma extensão do navegador Google Chrome devido a facilidade para o usuário que ela fornece. Alguns testes com *proceedings* já publicados foram realizados, e a ferramenta consegue reduzir o esforço na preparação dos *proceedings*.

## PALAVRAS-CHAVE

SOL, Proceedings, Google Chrome, Extensão, SBC, Artigos, PDF.

## 1. INTRODUÇÃO

A Sociedade Brasileira de Computação (SBC) promove diversos eventos científicos todos os anos e alguns são publicados na biblioteca digital aberta SOL (SBC OpenLib). Esta biblioteca vem se popularizando bastante nos últimos anos devido ao pesquisador e eventos não precisarem arcar com custos elevados para fazer sua publicação. Assim, quando ocorre o anúncio de um evento, artigos são submetidos para a avaliação de revisores, e após serem aceitos, os coordenadores de eventos (*chairs*) precisam submetê-los na biblioteca para disponibilizar os anais (*proceedings*). A submissão para a SOL consiste no envio de 4 arquivos em formato CSV (artigos, autores, referências e seções) que contêm diversas informações sobre os artigos aceitos e o evento. A extração dessas informações para o CSV demanda trabalho manual e está sujeita a erros humanos quando o *proceedings* tem uma grande quantidade de artigos. Atualmente, não existe uma ferramenta aberta e gratuita para realizar esse trabalho.

Diante desta situação, as extensões do Google Chrome mostraram-se uma forma de fácil acesso para usuários comuns, pois o usuário precisa apenas adicioná-las ao seu navegador e o código será executado na máquina do usuário, não necessitando assim de uma infraestrutura. Além disso, elas abstraem todo o trabalho necessário de uma instalação de um software. Outrossim, possuem uma forma fácil de distribuição através da Web Store do Chrome e possuem um custo associado baixo, pois precisa-se apenas pagar a taxa de inscrição da conta *Google Developer*.

Também, é mais fácil para os usuários interagirem com uma *UI* (User Interface) do que com uma *CLI* (Command Line Interface), sendo a *CLI* a forma comumente utilizada por scripts. A alternativa mais comum de prover uma *UI* sem precisar de instalação é prover um *website*, porém, seria necessário hospedar e isso acarretaria em custos. Através da extensão, podemos prover *UI* semelhante a sites normais, sem a necessidade de hospedar, reforçando assim a escolha do modelo de extensão Chrome para a ferramenta.

Portanto, este trabalho tem como objetivo desenvolver uma extensão para Google Chrome chamada de *SOL Proceedings* [1]. A partir de bibliotecas Javascript já existentes e estratégias de extração baseadas no template da SBC, a ferramenta extrai os campos necessários para o preenchimento dos dados do *proceedings* no formato da biblioteca digital SOL. O objetivo é reduzir o trabalho manual, diminuir erros e acelerar o processo de submissão na biblioteca SOL. Para o desenvolvimento da *SOL Proceedings*, foram usados artigos do evento CTIC-ES 2023 (Concurso de Trabalho de iniciação científica em engenharia de software) [2], enquanto para os testes foram usados os artigos da MSSIS (V Workshop em Modelagem e Simulação de Sistemas Intensivos em Software) e analisada qual a porcentagem dos campos que foram preenchidos corretamente pela ferramenta. A nossa ferramenta processa apenas artigos que seguem o formato da SBC [3]. Este formato ajuda a extrair alguns campos requisitados pela SOL.

O artigo está organizado da seguinte forma: A Seção 2 descreve a SOL. Já a Seção 3 descreve a extensão em seu funcionamento, a Seção 4 sua arquitetura. A Seção 5 apresenta detalhes da implementação da *SOL Proceedings*. A Seção 6 descreve uma avaliação inicial sobre a performance da ferramenta. Por fim, a Seção 7 apresenta os trabalhos futuros enquanto que a Seção 8 conclui com as lições aprendidas deste trabalho.

## 2. SOL

A SOL (SBC Open Lib) é uma biblioteca digital aberta com informação especializada em computação e é hoje o maior repositório de acesso aberto da América Latina para a publicação de artigos científicos da área de Computação [4]. Ela vem se popularizando justamente por sua característica de ser aberta e o pesquisador não ter que arcar com custos bastante elevados, já que muitos acervos desse tipo cobram um valor bem alto, neles temos como exemplo a Association for Computing Machinery (ACM). Ao final de 2021 a SOL contava com 12.383 artigos de 112 eventos e chegava a ter cerca de 100 mil visualizações em média por mês [4]. Além disso, a SBCOpenLib oferece acesso a todo o conteúdo produzido no âmbito dos eventos da SBC [5]. E também, a SOL utiliza o *Open Journal Systems* [6] (OJS 3.0.2.0), sistema de código livre gratuito para a administração e a publicação de publicações, desenvolvido com suporte e distribuição pelo *Public Knowledge Project* sob a licença *GNU General Public License*.

Para a publicação dos *proceedings* de um evento na biblioteca digital da SOL, os *chairs* devem preencher dados sobre o evento, e sobre cada artigo como nomes dos autores, referências bibliográficas, dentre outras informações. Esses dados devem ser preenchidos em arquivos em formato comma-separated values (CSV). Esse trabalho manual demanda tempo e é tendencioso a erros. É possível solicitar à SBC que se realize esse serviço, mas será necessário pagar uma taxa por cada artigo publicado.

## 3. EXTENSÃO GOOGLE CHROME

A *SOL Proceedings* é uma extensão para o navegador Google Chrome, que está disponível em repositório aberto no *Github* [22] e estará disponível gratuitamente na *Chrome Store*. Após adicioná-la, basta clicar no ícone da extensão para iniciá-la. Uma nova aba será aberta, apresentando um formulário no qual o usuário pode preencher com as informações sobre o evento e os artigos em PDF (ver Figura 1).

Baseado nisso, nesta seção será detalhado melhor o uso da extensão, quais os passos para utilizá-la e as decisões tomadas referente a experiência do usuário.

**SOL Proceedings**

Nome do evento

Nome do evento em inglês

Sigla do evento

Blind: 1- Single Blind, 2- Double Blind

Número de artigos submetidos

Número de artigos aceitos

Data: Paper submission  
mm/dd/yyyy

Date: Notification of acceptance  
mm/dd/yyyy

Date: Camera-ready paper submission  
mm/dd/yyyy

Arquivos pdf dos artigos aceitos  
 No file chosen

**Como usar:**

1. Preencha os campos acerca do seu evento.
2. Submeta a pasta com os artigos, sendo o nome de cada arquivo o IdJEMS dele.
3. Serão feitos automaticamente o download dos 4 CSVs (artigos, referências, seções, autores).
4. Importe os 4 arquivos na ferramenta preferida de planilha, revise os campos e preencha os faltantes.

• Por favor revise os documentos que serão gerados.  
• Em caso de muitos artigos esse processo pode demorar alguns minutos.

Figura 1 - UI da extensão Google Chrome.

### 3.1 Submissão dos Artigos

O usuário deve preencher as informações solicitadas no formulário e submeter os artigos em formato PDF, seguindo a orientação de nomear o arquivo com o seu idJEMS (ver Figura 2). Após isso, basta clicar em "submitter" (ver Figura 3). A ferramenta então realizará automaticamente o *download* dos CSVs para o dispositivo do usuário após alguns segundos. Os CSVs podem ser importados para a ferramenta de planilha preferida do usuário, onde é possível editar ou preencher os campos que a ferramenta não conseguiu preencher automaticamente.

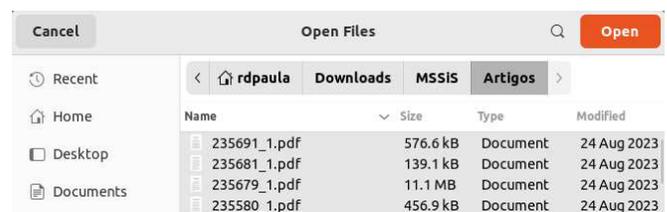


Figura 2 - Selecionando arquivos para submeter.

Figura 3 - Campos preenchidos e arquivos selecionados.

### 3.2 Arquivos Gerados

A ferramenta gerará quatro arquivos (ver Figura 4) em formato comma-separated values (CSV), nomeados como "artigos", "secoes", "autores" e "referencias", cada um contendo informações específicas seguindo as diretrizes para a publicação dos *proceedings* na biblioteca digital da SOL. Nos casos em que a ferramenta não conseguir preencher determinados campos, eles serão marcados com uma interrogação, indicando ao usuário que deverá preenchê-los manualmente. Devido às informações retiradas dos artigos poderem conter vírgulas, o que é comum em textos, todos os campos do arquivo CSV estarão entre aspas, seguindo o padrão de escape usual em arquivos CSV.

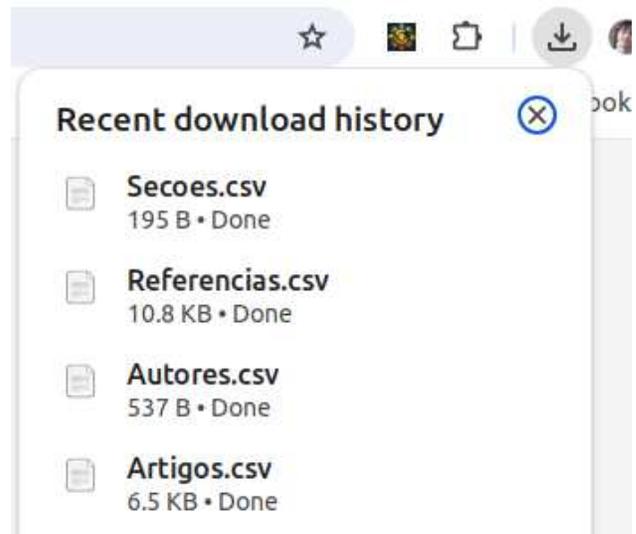


Figura 4 - Arquivos gerados pela extensão.

### 3.3 Importação do CSV

Após baixar os arquivos, o usuário deve importá-los para uma ferramenta de edição de planilhas. Durante o desenvolvimento, utilizamos o *Google Sheets* para testar a importação e edição dos arquivos gerados. Nas Figuras 5 e 6, podemos observar o processo de importação e o resultado na planilha, utilizando como exemplo o arquivo "artigos", respectivamente.



Figura 5 - Importando arquivo CSV gerado para *Google Sheets*.

	A	B	C	D	E	F
1	seq	language	sectionAbbrev	titleOrig	titleEn	AbstractOrig
2	1	en	MSSIS	Tracking Events	Tracking Events	The Routed DEV
3	2	pt	MSSIS	SoSML: Rumo a	SoSML: Rumo a	Sistemas de Sisi
4	3	en	MSSIS	A review on the	A review on the	Automation is a
5	4	pt	MSSIS	Utilizando Teoria	Utilizando Teoria	A gest'ao de pr.

Figura 6 - Planilha "Artigos" gerada pela ferramenta e importada no *Google Sheets*.

## 4. ARQUITETURA

A SOL Proceedings foi desenvolvida utilizando JavaScript clássico, que é a linguagem padrão para construir extensões para o Google Chrome. Utilizamos a ferramenta *chrome-extension-cli* [7] para criar o esqueleto da nossa extensão, incluindo o código base

e facilitando a geração dos artefatos para teste. Na Figura 7, é possível visualizar o fluxo do código na extensão.

Com base nisso, nesta seção será discutido sobre a estrutura inicial do código da extensão e a arquitetura presente nele.

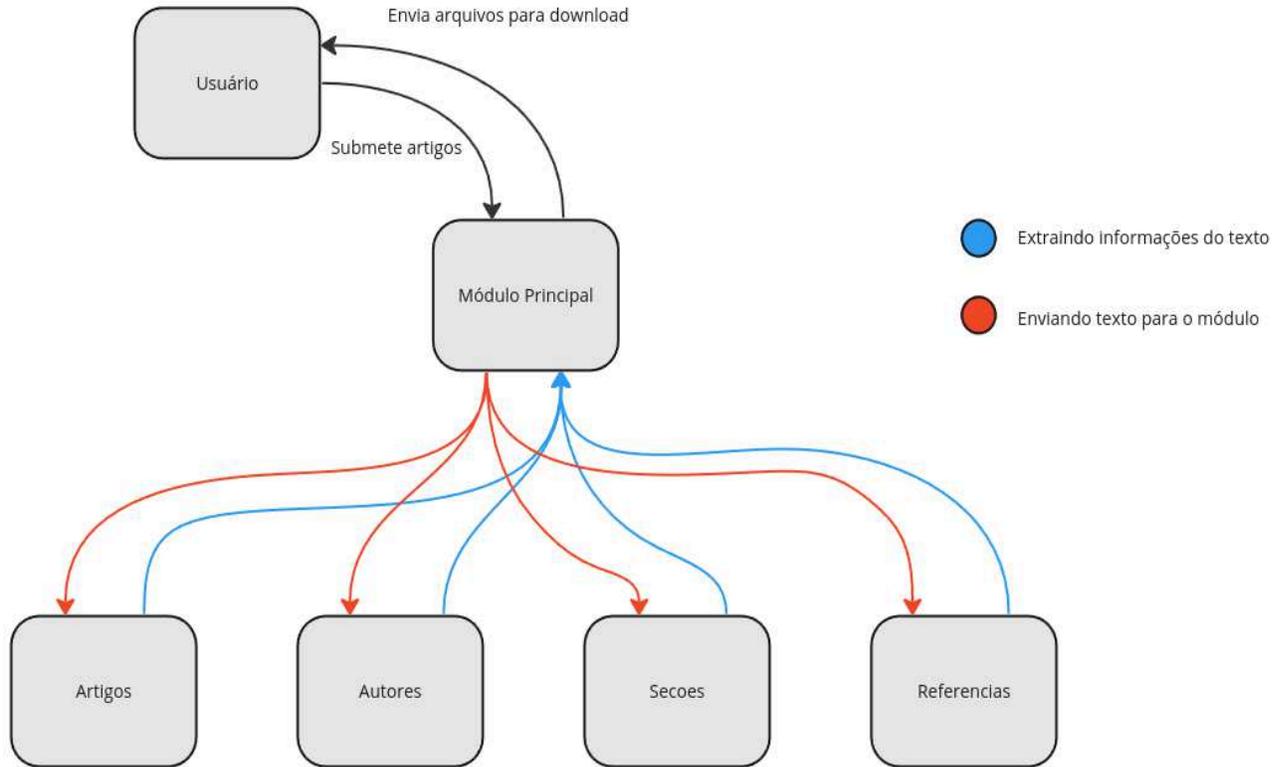


Figura 7 - Fluxo do código na *SOL PROCEEDINGS*.

### 4.1 Scaffolding da Extensão e Configurações

Conforme mencionado anteriormente, a extensão foi construída com base no *scaffold* da *chrome-extension-cli*, que gera a seguinte estrutura de pastas apresentada na Figura 8.

```
my-extension
├─ README.md
├─ node_modules
├─ package.json
├─ .gitignore
├─ config // Webpack with minimal configurations
│  └─ paths.js
│  └─ webpack.common.js
│  └─ webpack.config.js
├─ public
│  └─ icons
│     └─ icon_16.png
│     └─ icon_32.png
│     └─ icon_48.png
│     └─ icon_128.png
│  └─ *.html // HTML files will vary depending on extension type
│  └─ manifest.json
└─ src
   └─ *.css // CSS files will vary depending on extension type
   └─ *.js // JS files will vary depending on extension type
```

Figura 8 - Estrutura de arquivos criada pela *chrome-extension-cli*.

Na pasta "config", encontramos a configuração do Webpack, que empacota os arquivos para permitir a construção da extensão. Na pasta "public", estão os ícones e imagens utilizados pela extensão.

No nosso caso, nossos ícones foram gerados a partir do Chat GPT-4 [8]. O "manifest.json" contém as configurações da extensão. A Figura 9 apresenta as configurações que estamos utilizando na SOL Proceedings.

```
{
  "manifest_version": 3,
  "name": "SOL Proceedings",
  "version": "0.1.0",
  "description": "Extensão para automatizar criação de Proceedings para a SOL",
  "icons": {
    "16": "icons/icon_16.png",
    "32": "icons/icon_32.png",
    "48": "icons/icon_48.png",
    "128": "icons/icon_128.png"
  },
  "web_accessible_resources": [{
    "resources": ["pdfjs/web/viewer.html"],
    "matches": ["<all_urls>"],
    "use_dynamic_url": true
  }],
  "background": {
    "service_worker": "background.js"
  },
  "action": {},
  "permissions": [
    "downloads"
  ]
}
```

Figura 9 - Configuração da extensão.

Na seção de *Icons*, definimos o ícone da extensão em diferentes tamanhos para exibição em várias telas, como a barra de extensão do usuário e a *Chrome Store*. A seção *web\_accessible\_resources* é utilizada para permitir que o código da biblioteca pdf.js [9] seja acessado pela aba da nossa extensão. O *script* em *background* fica ouvindo o clique do usuário no ícone da extensão para criar uma nova aba no navegador com a interface da extensão (ver Figura 10). Além disso, nossa extensão requer a permissão de downloads, pois ela baixa os arquivos gerados no dispositivo do usuário.

```
'use strict';
chrome.action.onClicked.addListener(function(tab) {
  chrome.tabs.create({ url: "form.html" });
});
```

Figura 10 - Script em *background* que escuta *click* do usuário.

## 4.2 Separação de Responsabilidades

Devido às diferenças nas lógicas de extração em cada arquivo, optamos por adotar uma abordagem modular [10], separando as responsabilidades (ver Figura 11). Designamos as lógicas de extração do arquivo "artigos" para o módulo "artigos", "secoes" para "secoes" e assim por diante. Dessa forma, a parte principal da extensão é responsável por ler os arquivos em formato PDF e formatar o texto extraído pelos módulos em CSV, permitindo o *download* no dispositivo do usuário. Entretanto, nossa configuração de *webpack* não permitiu separarmos em diferentes arquivos os módulos, por isso a diferenciação se dá por serem funções/"caminhos" distintos que chamamos para executar a tarefa de cada arquivo.

```
articlePreCSV.push(await lerArticlePDF(base64Data, i+1, files[i].name));
authorsPreCSV.push(...(await lerAuthorsPDF(base64Data, i+1)));
referencesPreCSV.push(...(await lerReferencesPDF(base64Data, i+1)));
```

Figura 11 - Separação em módulos para cada arquivo.

## 5. IMPLEMENTAÇÃO

Para extrair os dados dos artigos, utilizamos a biblioteca pdf.js [9], desenvolvida pela Mozilla para analisar e renderizar arquivos PDF. Ao ler um arquivo PDF, essa biblioteca retorna uma lista de objetos que contêm os seguintes atributos conforme descrito na Figura 12.

```
▼ 2:
  dir: "ltr"
  fontName: "g_d0_f1"
  hasEOL: false
  height: 12
  str: "on functionalities as part of DE"
  transform: (6) [12, 0, 0, 12, 106.34, 731.62]
  width: 169.59600000000003
  ▶ [[Prototype]]: Object
  ▶ 3: {str: 'VS', dir: 'ltr', width: 15.311999999999998, height: 12, transform: Array(6), ...}
  ▶ 4: {str: '-', dir: 'ltr', width: 3.996, height: 12, transform: Array(6), ...}
  ▶ 5: {str: 'based formalisms using a design pattern', dir: 'ltr', width: 211.58399999999995, height: 1}
  ▶ 6: {str: '.', dir: 'ltr', width: 3.336, height: 12, transform: Array(6), ...}
```

Figura 12 - Estrutura do objeto que a biblioteca PDF.js trabalha.

A seguir descrevemos cada um dos campos:

- Dir: Direção de escrita do texto;
- fontName: Nome da fonte usada;
- hasEOL: True caso tenha um espaço após o texto;
- height: Tamanho do texto;
- str: Texto em si;
- Transform: Representa um array com as coordenadas geométricas do texto no pdf;
- width: Largura do texto.

Com esses atributos e sabendo das regras que um artigo em formato da SBC tem que seguir, foi possível extrair grande parte dos campos.

A partir disso, nesta seção será discutido para cada arquivo que campos foi possível extrair e quais não, assim como alguns exemplos de como a extração foi feita.

### 5.1 Artigos

O arquivo CSV *artigos* contém os seguintes campos: *seq*, *language*, *sectionAbbrev*, *titleOrig*, *titleEn*, *abstractOrig*, *abstractEn*, *keywordsOrig*, *keywordsEn*, *pages* e *idJEMS*. Dos campos mencionados, não conseguimos extrair os campos *keywordsOrig* e *keywordsEn* já que nem todos os artigos colocam esse campo no formato da SBC. Para extrair o campo de título, por exemplo, utilizamos a estratégia de considerar a primeira parte do artigo, com o tamanho da fonte sempre constante, como apresentado na Figura 13.

```
function getTitle(text) {
  var title = "";
  var heightTitle = text[0].height;
  var ind = 0
  while(text[ind].height == heightTitle || text[ind].height == 0){
    title += text[ind].str + (text[ind].hasEOL ? " " : "");
    ind++;
  }
  return title;
}
```

Figura 13 - Função para extrair o campo título.

Para extrair o resumo dos artigos, utilizamos a estratégia de buscar a palavra "Resumo" em diferentes idiomas. Atualmente, oferecemos suporte para "Abstract" em inglês, "Resumen" em espanhol e "Resumo" em português. A Figura 14 apresenta o código para extrair os resumos em inglês.

```
function getAbstractEn(text) {
  var abstract = "";
  var ind = 0
  while(ind < text.length && !text[ind].str.includes("Abstract")){
    ind++;
  }

  ind+=2;
  var fontAbstract = text[ind].fontName;
  while(ind < text.length && text[ind].fontName == fontAbstract){
    abstract += text[ind].str + (text[ind].hasEOL ? " " : "");
    ind++;
  }
  return abstract;
}
```

Figura 14 - Função para extrair resumo em inglês.

## 5.2 Autores

O arquivo CSV *autores* possui os seguintes campos: *article*, *authorFirstname*, *authorMiddlename*, *authorLastname*, *authorAffiliation*, *authorAffiliationEn*, *authorCountry*, *authorEmail* e *orcid*. Não conseguimos preencher a filiação do autor, o país e o *orcid*. No caso do *orcid*, ele não é informado nos artigos seguindo o formato da SBC. Para extrair o campo de e-mail dos autores, utilizamos a estratégia de localizá-lo imediatamente antes do "Resumo", utilizando uma modificação do algoritmo de pilha que verifica sequências de colchetes balanceadas [11]. Isso se deve ao fato de encontrarmos diferentes formatos para os e-mails, como "deborasouza@ccc.ufcg.edu.br, rohit@dsc.ufcg.edu.br" [12] e "{bruno.carvalho1,rodrigo.nascimento}@discente.ufma.br,davi.viana@ufma.br" [13].

Para esse caso, adotamos uma estratégia diferente em vez de simplesmente dividir o texto. No segundo caso, precisamos extrair cada prefixo presente nas chaves e concatená-lo com o sufixo após as chaves. Para fazer isso, mantemos uma pilha com todos os prefixos presentes nas chaves. Ao encontrar o fechamento das chaves, coletamos o sufixo e o concatenamos com todos os prefixos presentes na pilha. As Figuras 15 e 16 apresentam trechos de código que ilustram essa abordagem.

```
function extractEmailsFromText(text) {
  var emails = [];
  var ind = 0
  while(ind < text.length && (!text[ind].str.includes("Abstract") &&
    !text[ind].str.includes("Resumo") && !text[ind].str.includes("Resumen"))){
    ind++;
  }

  ind--;
  if(text[ind].height == 0) ind--;
  var heightEmail = text[ind].height
  while(ind >= 0 && (text[ind].height == heightEmail || text[ind].height == 0)){
    emails.push(text[ind].str);
    ind--;
  }
  emails.reverse()
  let emailString = ""
  for (let i = 0; i < emails.length; i++) {
    emailString += emails[i]
  }
  return emailString;
}
```

Figura 15 - Função para extrair a parte do artigo que contém os emails.

```
function extractEmails(input) {
  input = extractEmailsFromText(input);
  let emails = []
  var stack = []
  var actStr = ""
  for(var i = 0; i < input.length; i++){
    if(input[i] == "{"){
      i++;
      while(input[i] != "}") {
        if(input[i] != ',') actStr += input[i];
        else stack.push(actStr), actStr = "";
        i++;
      }
      i++;
      stack.push(actStr), actStr = "";
      let suffix = ""
      while(i < input.length && input[i] != ","){
        suffix += input[i];
        i++;
      }
      for(var j = 0; j < stack.length; j++){
        emails.push(stack[j] + suffix);
      }
      stack = [];
    }
    else{
      if(input[i] != ',') actStr += input[i];
      if(input[i] == ',' || i == input.length - 1){
        emails.push(actStr)
        actStr = "";
      }
    }
  }
  for(var i = 0; i < emails.length; i++){
    emails[i] = emails[i].trim();
  }
  return emails;
}
```

Figura 16 - Função para fazer *split* dos emails.

## 5.3 Referências

O arquivo CSV *referencias* possui os campos *article*, *description*, *doi*, *link* e *accessed*. Conseguimos preencher os campos *article* e *description*. Para preencher o campo de descrição da referência, utilizamos a informação de que no formato da SBC, em cada referência, a primeira linha começa sem indentação e as demais linhas são indentadas. Com base nisso, utilizamos as coordenadas geométricas fornecidas pela biblioteca que usamos para analisar

os PDFs. A Figura 17 apresenta o trecho de código que ilustra essa estratégia.

```

let beginRefPosition
let curStr = ""
for (let i = indPage; i <= numPages; i++) {
  let text = await getPageText(i, file)
  let size = text.length;
  let ind = i == indPage ? indArray : 0;
  if (i == indPage) beginRefPosition = Math.round(text[ind].transform[4] * 100);
  while (ind < size) {
    if (beginRefPosition != Math.round(text[ind].transform[4] * 100) || text[ind].str == "") {
      text[ind].str = "";
      curStr += text[ind].str + (text[ind].hasEOL ? "\n" : "");
    }
    else {
      referencesResult.push({
        article: index.toString(),
        description: curStr,
        doi: "?",
        link: "?",
        accessed: "?",
      });
      curStr = text[ind].str + (text[ind].hasEOL ? "\n" : "");
    }
    ind++;
  }
  if (i == numPages && curStr != "") {
    referencesResult.push({
      article: index.toString(),
      description: curStr,
      doi: "?",
      link: "?",
      accessed: "?",
    });
  }
}
referencesResult = referencesResult.slice(1, referencesResult.length)
for (var i = 0; i < referencesResult.length; i++) {
  referencesResult[i].description = referencesResult[i].description.trim();
}
return referencesResult;

```

Figura 17 - Código para extrair as referências dos artigos.

Não conseguimos separar os *DOIs* e *links* devido à falta de padronização nas referências dos artigos testados. Embora existam diretrizes no formato da SBC, os autores de artigos nem sempre o seguem. Decidimos manter a referência completa na descrição, deixando para o usuário separá-la manualmente, quando for necessário.

## 5.4 Seções

O arquivo CSV *secoes* contém os campos *sectionTitle*, *sectionTitleEn*, *sectionAbbrev*, *blind*, *numSubmitted*, *numAccepted*, *dateSub*, *dateResult* e *dateReady*, os quais se referem às informações do evento. Para preenchê-los, foi criado um formulário na interface do usuário (UI) para que o usuário possa inserir os dados (ver Figura 1), os quais serão exportados para o CSV normalmente.

## 6. AVALIAÇÃO PRELIMINAR

Como avaliação preliminar de desempenho da SOL Proceedings iremos usar o critério de tempo para gerar os arquivos CSV. Além disso, para testar e desenvolver a SOL Proceedings, utilizamos artigos do evento CTIC-ES 2023. Entretanto, optamos por avaliar o desempenho da ferramenta com artigos do evento V Workshop em Modelagem e Simulação de Sistemas Intensivos em Software (MSSIS) [14].

Sendo assim, a Tabela 1 apresenta o tempo necessário para a ferramenta extrair informações com base no número de artigos submetidos. Esses tempos foram medidos usando 7 artigos diferentes do evento MSSIS, que foram replicados e suas réplicas selecionadas aleatoriamente para análise. O tempo foi calculado com base na diferença entre o horário de início e o horário de término da execução da *SOL Proceedings*.

Artigos	Tempo em segundos
1	0,54
5	2,47
10	4,95
20	8,30
40	14,06

Tabela 1 - Tempo que a ferramenta demora para extrair os campos de acordo com a quantidade de artigos submetidos.

## 7. TRABALHOS FUTUROS

O objetivo da *SOL Proceedings* é minimizar o esforço necessário ao submeter artigos para a SOL. Portanto, um dos principais focos diz respeito a aumentar o nível de preenchimento automático da ferramenta, reduzindo ainda mais a necessidade de intervenção manual. Posteriormente, uma avaliação de correteude será realizada para validar o nível de preenchimento. Outro ponto chave é aprimorar a UI, tornando-a mais intuitiva e esteticamente agradável. Uma solução seria migrar para um modelo em ReactJS [15], o que proporcionaria maior customização e permitiria o uso de bibliotecas como a MaterialUI [16], que oferece um design mais atraente.

Além disso, seria interessante adicionar suporte para outros templates de artigos, como o formato da ACM [17], o que ampliaria a gama de artigos suportados e os casos de uso. Por exemplo, o evento SBES publica na SOL com esse formato. No entanto, uma grande diferença em relação ao formato da ACM é o *layout* de duas colunas, o que exigiria ajustes significativos nas estratégias atualmente utilizadas no modelo da SBC, que é baseado em uma única coluna.

Outra melhoria seria a capacidade de diferenciar entre artigos completos (*full*) e resumidos (*short*), que se distinguem principalmente pelo número de páginas. Esse foi um dos fatores que impactaram significativamente a porcentagem de preenchimento ao testarmos com o MSSIS, pois a ferramenta foi desenvolvida e testada com base no CTIC-ES, que não apresentava essa distinção entre os tipos de artigos.

Por fim, também como trabalhos futuros iremos melhorar a configuração do *webpack*, para poder termos o código melhor estruturado e possibilitar *refactorings* mais fáceis.

## 8. EXPERIÊNCIAS

Nesta seção será discutido como se deu o processo de desenvolvimento e quais os principais desafios encontrados.

### 8.1 Processo de Desenvolvimento

O processo de desenvolvimento seguiu uma abordagem semelhante ao modelo TDD (Test-Driven Development) [18]. Nesse método, o orientador forneceu casos de teste para validar o resultado da extensão, permitindo avançar com base nos erros e acertos identificados. Além disso, foram realizadas reuniões periódicas para acompanhar o progresso da extensão e garantir que estivesse seguindo na direção correta. Esse acompanhamento foi fundamental, pois o orientador também enfrentou as

dificuldades que a ferramenta se propôs a resolver e atuou como um *stakeholder* durante todo o processo de desenvolvimento.

## 8.2 Desafios Encontrados

Em relação aos desafios enfrentados, a falta de experiência prévia com extensões do Chrome foi um obstáculo inicial, já que esse paradigma difere um pouco do que foi apresentado no curso. Contudo, a seção "*Get Started*" no site do *Google Developer* [19] foi extremamente útil para nos familiarizarmos com a estrutura de uma extensão. Além disso, nos inspiramos na extensão *MouseTooltipTranslator* [20], que consegue traduzir textos em PDFs ao passar o mouse sobre eles.

Essa extensão foi especialmente útil devido à recente exigência do Google de migrar para a versão 3 de sua plataforma de extensões [21], que restringe várias permissões anteriormente disponíveis para bibliotecas e extensões. A maioria das extensões que encontramos como referência ainda utilizava a versão 2 e dependia da permissão para usar "eval" em seu código, o que não é mais permitido na versão 3.

## AGRADECIMENTOS

Gostaria primeiramente de expressar minha gratidão a Deus por me guiar ao longo de toda a jornada do curso. Também desejo agradecer imensamente à minha família e à minha namorada pelos seus apoios incondicionais e pela fé em mim. Ao meu orientador, sou profundamente grato por sua orientação paciente e comprometida ao longo de todo o projeto, assim como por abrir portas e criar oportunidades para mim não apenas durante este projeto, mas ao longo de toda a minha faculdade. Aos demais professores, agradeço sinceramente por todos os ensinamentos e conhecimentos transmitidos ao longo destes anos, os quais, sem dúvida, contribuíram para o meu crescimento profissional e pessoal.

## REFERÊNCIAS

- [1] Fonseca, R. SOL Proceedings. Disponível em: <https://drive.google.com/file/d/1oVQftpEUzajjr7iT4OVq3tcJ07XuAYr/view?usp=sharing>. Acesso em: Maio 2024
- [2] [n. d.]. Concurso de Trabalho de Iniciação Científica em Engenharia de Software. Disponível em: [https://sol.sbc.org.br/index.php/cbsoft\\_estendido/issue/view/1175](https://sol.sbc.org.br/index.php/cbsoft_estendido/issue/view/1175). Acesso em: Maio 2024.
- [3] [n. d.]. Template de Artigos SBC. Disponível em: <https://www.sbc.org.br/documentos-da-sbc/summary/169-templat-es-para-artigos-e-capitulos-de-livros/878-modelosparapublicaodeartigos>. Acesso em: Maio 2024.
- [4] [n. d.]. SOL oferece nova ferramenta de busca integrada. Disponível em: <https://www.sbc.org.br/noticias/2379-sol-oferece-nova-ferramenta-de-busca-integrada>. Acesso em: Maio 2024.
- [5] [n. d.]. Sobre a SBC OpenLib. Disponível em: <https://sol.sbc.org.br/index.php/anais/about>. Acesso em: Maio 2024.
- [6] [n. d.]. Open Journal Systems. Disponível em: <https://pkp.sfu.ca/software/ojs/>. Acesso em: Maio 2024.
- [7] [n. d.]. Chrome-extension-cli. Disponível em: <https://github.com/dutiyesh/chrome-extension-cli>. Acesso em: Maio 2024.

- [8] [n. d.]. ChatGPT-4. Disponível em: <https://chatgpt.com/>. Acesso em: Maio 2024.
- [9] [n. d.]. PDF.js biblioteca. Disponível em: <https://github.com/mozilla/pdf.js>. Acesso em: Maio 2024.
- [10] Toru, C. What is Modular Architecture? Benefits & Implementation Methods. Disponível em: <https://medium.com/geekculture/what-is-modular-architecture-benefits-implementation-methods-8c272ebc05eb>. Acesso em: Maio 2024
- [11] Kogler, J. Balanced validation. Disponível em: [https://cp-algorithms.com/combinatorics/bracket\\_sequences.html#balance-validation](https://cp-algorithms.com/combinatorics/bracket_sequences.html#balance-validation). Acesso em: Maio 2024.
- [12] Débora Souza e Rohit Gheyi. 2023. Estudo de caso: uso do ChatGPT para resolução de problemas de programação. Em Anais Estendidos do XIV Congresso Brasileiro de Software: Teoria e Prática, setembro 25, 2023, Campo Grande/MS, Brasil. SBC, Porto Alegre, Brasil, 80-89. DOI: [https://doi.org/10.5753/cbsoft\\_estendido.2023.235666](https://doi.org/10.5753/cbsoft_estendido.2023.235666).
- [13] Bruno Silva, Rodrigo Nascimento, e Davi Viana. 2023. ReqMLSCity: Uma ferramenta de análise de requisitos utilizando aprendizado de máquina para aplicações de cidades inteligentes. Em Anais Estendidos do XIV Congresso Brasileiro de Software: Teoria e Prática, setembro 25, 2023, Campo Grande/MS, Brasil. SBC, Porto Alegre, Brasil, 120-129. DOI: [https://doi.org/10.5753/cbsoft\\_estendido.2023.235923](https://doi.org/10.5753/cbsoft_estendido.2023.235923).
- [14] [n. d.]. 2023: Anais do V Workshop em Modelagem e Simulação de Sistemas Intensivos em Software. Disponível em: <https://sol.sbc.org.br/index.php/mssis/issue/view/1174>. Acesso em: Maio 2024.
- [15] [n. d.]. Quick Start. Disponível em: <https://react.dev/learn>. Acesso em: Maio 2024.
- [16] [n. d.]. Material UI - Overview. Disponível em: <https://mui.com/material-ui/getting-started/>. Acesso em: Maio 2024
- [17] [n. d.]. ACM Conference Proceedings Primary Article Template. Disponível em: <https://www.overleaf.com/latex/templates/acm-conference-proceedings-primary-article-template/wbvngjhjzwpwpc>. Acesso em: Maio 2024
- [18] [n. d.]. Test-driven development. Disponível em: [https://pt.wikipedia.org/wiki/Test-driven\\_development](https://pt.wikipedia.org/wiki/Test-driven_development). Acesso em: Maio 2024
- [19] [n. d.]. Extensions / Get started | Chrome for Developers. Disponível em: <https://developer.chrome.com/docs/extensions/get-started>. Acesso em: Maio 2024
- [20] [n. d.]. MouseTooltipTranslator. Disponível em: <https://github.com/ttop32/MouseTooltipTranslator>. Acesso em: Maio 2024
- [21] [n. d.]. Manifest V2 support timeline. Disponível em: <https://developer.chrome.com/docs/extensions/develop/migrate/mv2-deprecation-timeline>. Acesso em: Maio 2024
- [22] Fonseca, R. SOL-Proceedings. Disponível em: <https://github.com/Rdpaula/SOL-Proceedings>. Acesso em: Maio 2024