

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

RAPHAEL MALINSKI VIEIRA

**Solução para integração do JEMS3 à SOL:
exportando artigos científicos de eventos
para a biblioteca digital da SBC**

Monografia apresentada como requisito parcial
para a obtenção do grau de Bacharel em Ciência
da Computação

Orientador: Prof. Dr. Lisandro Zambenedetti
Granville

Co-orientador: Marcus Vinícius Brito Da Silva

Porto Alegre
2025

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Prof^a. Marcia Cristina Bernardes Barbosa

Vice-Reitor: Prof. Pedro de Almeida Costa

Pró-Reitora de Graduação: Prof^a. Nádyá Pesce da Silveira

Diretor do Instituto de Informática: Prof. Luciano Paschoal Gaspary

Coordenador do Curso de Ciência de Computação: Prof. Marcelo Walter

Bibliotecário-Chefe do Instituto de Informática: Alexander Borges Ribeiro

RESUMO

O JEMS é um sistema de submissão de artigos científicos disponibilizado pela SBC à comunidade científica, principalmente da área de Computação. Sistemas como o JEMS são plataformas digitais desenvolvidas para receber, avaliar e selecionar artigos de eventos e periódicos. Atualmente, o JEMS está na sua terceira versão, chamada de JEMS3. Uma biblioteca digital, por sua vez, é um repositório online que organiza, armazena e disponibiliza recursos digitais, como livros, artigos, imagens e multimídia, para acesso e consulta via Internet. A SBC OpenLib (SOL) é uma destas bibliotecas, também gerida pela SBC, que tem como principal objetivo oferecer acesso a conteúdos especializados em Ciência da Computação, atendendo às necessidades da comunidade de pesquisadores e profissionais da área. Atualmente, a importação na SOL dos artigos dos eventos organizados pela SBC e gerenciados no JEMS3 é realizada de forma manual. Esse processo é muito trabalhoso, demanda muito tempo e é passível de muitos erros. Este trabalho de conclusão de curso tem como objetivo automatizar parte deste processo visando reduzir o tempo e o trabalho necessários para transferir os artigos de um evento entre JEMS3 e SOL. Para isso, é apresentada uma nova funcionalidade desenvolvida no JEMS3 em conjunto com a modificação do *template* de artigo utilizado nos eventos da SBC. Essa funcionalidade gera de forma automática as 4 planilhas necessárias para criar o XML que é submetido na SOL. Para testar as implementações, um evento real foi selecionado (a Escola Regional de Redes de Computadores, edição 2024 - ERRC 2024) e testes foram realizados em ambiente de homologação, onde foi possível avaliar a solução desenvolvida, identificando tanto seus benefícios quanto limitações.

Palavras-chave: Sistema de submissão de artigos científicos. JEMS. SOL. SBC.

Solution for integrating JEMS3 with SOL: exporting scientific articles from events to SBC's digital library

ABSTRACT

JEMS is a scientific paper submission system provided by SBC (Brazilian Computer Society) to the scientific community, primarily in the field of Computing. Systems like JEMS are digital platforms designed to receive, review, and select papers for conferences and journals. Currently, JEMS is in its third version, known as JEMS3. A digital library, on the other hand, is an online repository that organizes, stores, and provides access to digital resources, such as books, articles, images, and multimedia, for consultation over the Internet. The SBC OpenLib (SOL) is one such library, also managed by SBC, whose main goal is to offer access to specialized content in Computer Science, meeting the needs of researchers and professionals in the field. Currently, importing articles from SBC-organized events managed in JEMS3 into SOL is performed manually. This process is very labor-intensive, time-consuming, and prone to many errors. This undergraduate thesis aims to automate part of this process, seeking to reduce the time and effort required to transfer event articles between JEMS3 and SOL. To achieve this, a new functionality developed in JEMS3 is presented, along with modifications to the article template used in SBC events. This functionality automatically generates the four spreadsheets needed to create the XML file submitted to SOL. To test the implementations, a real event was selected (the Regional School of Computer Networks, 2024 edition - ERRC 2024), and tests were conducted in a staging environment, allowing the evaluation of the developed solution, identifying both its benefits and limitations.

Keywords: Scientific paper submission system. JEMS. SOL. SBC.

LISTA DE FIGURAS

Figura 2.1 Fluxograma do processo de exportação de artigos do JEMS3 para a SOL. ...	13
Figura 2.2 Tela inicial ao acessar página do administrador no JEMS3.	14
Figura 2.3 Página inicial de um evento no JEMS3.	15
Figura 2.4 Tela do JEMS3 com formulário de download de arquivos.	15
Figura 2.5 Exemplo de preenchimento da planilha de seções.	16
Figura 2.6 Exemplo de preenchimento da planilha de artigos.	17
Figura 2.7 Exemplo de preenchimento da planilha de autores.	18
Figura 2.8 Exemplo de preenchimento da planilha de autores.	19
Figura 2.9 Primeira parte do "Formulário de geração de arquivo XML - Anais de Evento"	20
Figura 2.10 Segunda parte do "Formulário de geração de arquivo XML - Anais de Evento"	21
Figura 2.11 Página de envio de XML para exportação dos artigos do evento para a SOL.	21
Figura 3.1 Capítulo de referências do <i>template</i> da SBC com palavras-chave ocultas	25
Figura 3.2 Trecho do <i>template</i> que ocorre problema no parser.	26
Figura 3.3 Palavras-chave ocultas no início e no fim de cada referência.	27
Figura 3.4 Formulário de criação/edição de uma trilha no JEMS3.	28
Figura 3.5 Cards de submissão de arquivo no JEMS3.	28
Figura 3.6 Formulário da geração das planilhas no JEMS3 não preenchido.	29
Figura 3.7 Formulário da geração das planilhas no JEMS3.	30
Figura 3.8 Lista artigos que não estão seguindo o <i>template</i> da SBC.	30
Figura 4.1 Exemplo de inclusão de metadado em \LaTeX utilizando o comando <i>hy-persetup</i>	31
Figura 4.2 Código em \LaTeX da inclusão dos metadados de autores.	33
Figura 4.3 Exemplo do uso do comando <i>resumo</i> em \LaTeX	33
Figura 4.4 Código em \LaTeX da inclusão dos metadados do artigo em inglês.	35
Figura 4.5 Código em \LaTeX da inclusão dos metadados do artigo.	35
Figura 4.6 Código em \LaTeX da implementação do comando <i>SBCprintbibliography</i>	36
Figura 4.7 Código em \LaTeX da implementação dos comandos <i>SBCprintInitKeyBibliography</i> e <i>SBCprintEndKeyBibliography</i>	36
Figura 4.8 Código em \LaTeX da implementação da redefinição do comando <i>begentry</i> . ..	37
Figura 4.9 Código em \LaTeX da implementação da redefinição do comando <i>finentry</i>	37
Figura 5.1 Implementação do ícone de validação do <i>template</i> da SBC.	40
Figura 5.2 Implementação do formulário para geração das planilhas da SOL.	43
Figura 5.3 Implementação do HTML do componente <i>ArticlesListCardDialogComponent</i>	44
Figura 5.4 Implementação do método usado para abrir uma submissão a partir do identificador.	44
Figura 6.1 Tela de visão de calendário da programação de um evento no JEMS3.	51
Figura 6.2 Novo item com nome padrão no calendário da programação de um evento no JEMS3.	51
Figura 6.3 Aba "Configurações gerais" do menu de uma sessão no JEMS3.	52
Figura 6.4 Aba "Submissões" do menu de uma sessão no JEMS3.	52

Figura 6.5 Passo a passo para acessar página da geração das planilhas da SOL no JEMS3.....	53
Figura 6.6 Passo "Conteúdo" na página de geração de anais de um evento no JEMS3..	54
Figura 6.7 Passo "Conteúdo" com ST's inclusas no conteúdo dos anais do evento.	54
Figura 6.8 Passo "Arquivo de trilha" na página de geração de anais de um evento no JEMS3.....	55
Figura 6.9 Passo "Seções da SOL" com os dados preenchidos.	55

LISTA DE TABELAS

Tabela 3.1	Tabela de desempenho do extrator <i>singlethread</i>	24
Tabela 3.2	Tabela de desempenho do extrator <i>multithread</i>	24
Tabela 3.3	Tabela comparação extratores <i>singlethread</i>	27
Tabela 3.4	Tabela comparação extratores <i>multithread</i>	28

LISTA DE ABREVIATURAS E SIGLAS

SOL	SBC-OpenLib
JEMS	Journal and Event Management System
XML	Extensible Markup Language
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
MTV	Model-Template-View
ERRC	Escola Regional de Redes de Computadores
ST	Sessão técnica

SUMÁRIO

1 INTRODUÇÃO	10
1.1 Sistemas de submissão de artigos científicos	10
1.2 Bibliotecas digitais	10
1.3 Problema e objetivo do trabalho.....	11
2 PROBLEMA	13
2.1 Processo atual para a exportação dos artigos na SOL	14
3 SOLUÇÃO	22
3.1 Conceitos Fundamentais e Tecnologias Envolvidas	22
3.2 <i>Template</i> padrão para os artigos	23
3.3 Extrator de referências com biblioteca externa	23
3.4 Extrator de referências com implementação própria.....	24
3.5 Validação do <i>template</i> no JEMS3.....	27
3.6 Geração das planilhas no JEMS3.....	28
4 QUESTÕES TÉCNICAS DA IMPLEMENTAÇÃO DO <i>TEMPLATE</i> EM \LaTeX	31
4.1 Dados da planilha de autores	32
4.2 Dados da planilha de artigos.....	33
4.3 Alterações para uso do extrator de referências.....	34
4.4 Metadado para verificar o uso do <i>template</i>	38
5 QUESTÕES TÉCNICAS DAS IMPLEMENTAÇÕES NO JEMS3	39
5.1 Implementações de telas no <i>frontend</i>	39
5.1.1 Validação do uso do <i>template</i> da SBC	39
5.1.2 Formulário de geração das planilhas da SOL	41
5.1.3 Componente com a lista de submissões com <i>template</i> inválido	42
5.2 Implementações no <i>backend</i> para geração das planilhas	44
5.2.1 <i>UpdateSubmissionTrackFiles</i>	45
5.2.2 <i>SolSections</i>	46
5.2.3 <i>SolSpreadsheets</i>	46
6 TESTES EM HOMOLOGAÇÃO DA GERAÇÃO DAS PLANILHAS PARA A 21 ERRC	50
6.1 Inclusão da programação do evento no JEMS3	50
6.2 Geração das planilhas para a ERRC 2024	53
6.3 Avaliação dos testes e dos resultados obtidos	56
6.3.1 Análise da geração das planilhas e arquivos para a SOL.....	56
6.3.2 Limitações encontradas e possíveis melhorias futuras	57
7 CONCLUSÃO	59
7.1 Trabalhos futuros	61
REFERÊNCIAS	63

1 INTRODUÇÃO

Nos últimos anos, o avanço das bibliotecas digitais e dos sistemas de submissão de artigos científicos transformou significativamente o acesso e a gestão do conhecimento acadêmico. No entanto, embora essas ferramentas promovam a democratização e a organização da informação, muitas ainda operam de forma independente, o que dificulta a interconexão entre os sistemas e a fluidez no compartilhamento de conteúdos. Este trabalho tem como foco a implementação de uma solução que integre duas dessas plataformas: a biblioteca digital SOL e o sistema de submissão de artigos científicos JEMS3.

1.1 Sistemas de submissão de artigos científicos

Os sistemas de submissão de artigos científicos são plataformas digitais desenvolvidas para receber, avaliar e selecionar artigos de eventos e periódicos. Esses sistemas permitem que os autores realizem a submissão dos seus trabalhos para que sejam avaliados e revisados por especialistas da área antes que sejam publicados. Dentre os sistemas de submissão de artigos estão o JEMS (SBC, 2024d), EDAS (EDAS, 2024), EasyChair (LTD, 2024) e HotCRP (SOFTWARE, 2024), com cada um contendo diferentes focos e públicos-alvo.

O JEMS (Journal and Event Management System) é uma plataforma de gerenciamento de submissões amplamente utilizada para eventos científicos. O JEMS possui 3 versões, sendo a mais atual denominada JEMS3. A primeira versão do JEMS foi uma solução mais simples, desenvolvida para permitir a submissão e o acompanhamento de artigos em eventos científicos. A segunda versão, chamada de JEMS 2, trouxe melhorias em relação à interface e nela foi introduzida uma maior robustez no controle de prazos e de alocação de revisores. A versão mais atual, o JEMS3, trouxe avanços em usabilidade e segurança.

1.2 Bibliotecas digitais

As bibliotecas digitais estão, com o grande aumento no acesso à tecnologia e à Internet, se popularizando bastante nas instituições de ensino superior. Elas formam acervos online com uma vasta gama de conteúdos digitais, tais como livros, artigos, imagens

e vídeos. Esses sistemas também permitem um fácil acesso às pesquisas através de mecanismos de busca, indexação por palavras-chave e filtros por áreas do conhecimento.

Além de permitirem a preservação digital de documentos, as bibliotecas digitais trouxeram diversas outras vantagens em relação às bibliotecas convencionais. Elas permitem que não exista limitação de espaço para a ampliação do acervo; possibilitam que vários usuários consigam acessar simultaneamente determinado título, sem a necessidade de se aguardar a devolução dele por outra pessoa; aumentam a democratização do acesso ao conhecimento, permitindo que qualquer pessoa possa acessar as áreas mais remotas ou em instituições com recursos limitados. Outra contribuição importante que as bibliotecas digitais trouxeram é o aumento na disseminação de conhecimento científico. Elas permitem que pesquisadores possam encontrar trabalhos relevantes, acompanhar o progresso em suas áreas de estudo e contribuir com o avanço do conhecimento quando também compartilham suas próprias descobertas.

Dentre as bibliotecas digitais existentes, algumas oferecem acesso gratuito a seus conteúdos; todavia, outras possuem os seus acervos restritos a usuários registrados ou que realizam a assinatura da plataforma. Dentre alguns exemplos dessas bibliotecas, podemos citar a IEEE Xplore (XPLORE, 2024), a ACM Digital Library (LIBRARY, 2024), a Biblioteca Digital Mundial (MUNDIAL, 2024), a Biblioteca Brasileira (USP, 2024) e a Biblioteca Digital da Sociedade Brasileira de Computação (SOL) (SBC, 2024c).

A SBC-OpenLib (SOL) é uma biblioteca digital gerida pela Sociedade Brasileira de Computação (SBC). O objetivo central dessa biblioteca é possibilitar o acesso a conteúdos especializados em Ciência da Computação para atender à comunidade de profissionais e pesquisadores do setor. A maioria do material científico produzido pelos eventos da SBC é oferecida pela SOL, incluindo artigos selecionados e apresentados nos eventos organizados ou apoiados pela sociedade. Além desses, os conteúdos publicados em periódicos da SBC e livros técnicos e didáticos publicados pela Editora SBC também são disponibilizados.

1.3 Problema e objetivo do trabalho

Atualmente, as novas edições dos eventos científicos organizados pela Sociedade Brasileira de Computação (SBC) utilizam o sistema JEMS3 para gerenciar as submissões de artigos científicos. Os autores submetem seus artigos diretamente no JEMS3, e revisores designados para cada submissão são responsáveis pela avaliação que subsidia a

decisão de aprovação ou rejeição dos mesmos. Após a revisão de todos os artigos, o organizador do evento deve consolidar os trabalhos aceitos para submetê-los na biblioteca digital da SBC, a SOL. Esse processo é realizado seguindo as etapas abaixo:

1. Acessar uma lista completa dos artigos aceitos, contendo título, autores e o arquivo final em PDF;
2. Revisar os metadados (como título, autores, afiliações e palavras-chave) para assegurar a consistência e conformidade com o padrão do evento;
3. Inserir os metadados em planilhas a serem submetidas na SOL;
4. Submeter as planilhas na SOL para exportar os artigos que foram aprovados para o evento.

Atualmente, essas etapas são realizadas manualmente, o que gera dois grandes desafios: o tempo excessivo necessário, especialmente para eventos com um grande número de artigos a serem publicados; e a alta propensão a erros decorrentes da transferência manual de informações de PDFs para planilhas.

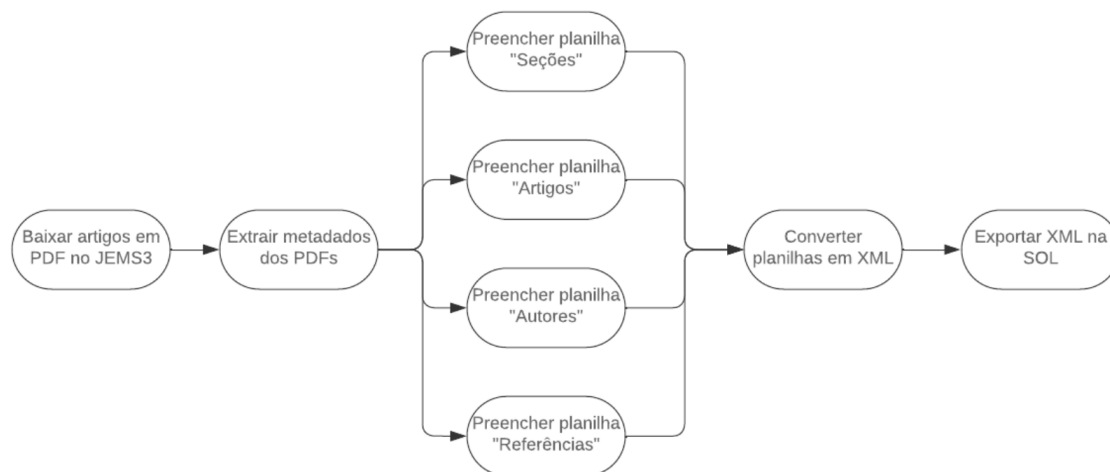
Este trabalho de conclusão irá apresentar uma solução que visa resolver esses desafios nos eventos organizados pela SBC e gerenciados pelo JEMS3. A solução implementada automatiza a parte do processo que mais demanda tempo e é propensa a erros, que é a passagem dos metadados dos artigos para as planilhas que são submetidas na SOL. Para conseguir essa automatização, foi necessário buscar uma solução que buscasse os metadados nos artigos de forma rápida, sem a necessidade de um *parser* que leia todos os artigos individualmente na busca das informações.

O restante deste trabalho de conclusão de curso está organizado da seguinte forma. No capítulo 2, é realizada uma contextualização e explicação mais aprofundada do porquê os problemas citados precisam de alguma solução. No capítulo 3, é realizada uma explicação em alto nível de qual foi a solução encontrada e como ela funciona. No capítulo 4, são descritas as implementações feitas no *template* da SBC de forma mais técnica, incluindo informações das abordagens de desenvolvimento usadas na solução. O capítulo 5 também descreve informações mais técnicas das implementações feitas no JEMS3, descrevendo as abordagens e tecnologias utilizadas na solução. No capítulo 6 será descrito o uso na prática (em ambiente de homologação) da solução no evento da 21ª ERRC, descrevendo o processo para o uso e os problemas enfrentados.

2 PROBLEMA

A Sociedade Brasileira de Computação (SBC) é responsável pela organização de diversos eventos científicos que contam com a presença de autores nacionais e internacionais. Para realizar o gerenciamento das submissões dos artigos científicos do evento, a SBC utiliza o sistema JEMS3. Nele, os autores realizam a submissão dos seus artigos, que irão passar por uma revisão feita em pares por revisores especialistas. Esses revisores podem aceitar ou rejeitar o artigo revisado. Após passada a data limite para submissão dos artigos pelos autores e todos os revisores terem feito as suas avaliações, o coordenador de publicação precisa exportar esses artigos para a biblioteca digital SBC-OpenLib (SOL). Nessa exportação é enviado um arquivo XML que contém as informações dos artigos. Esse XML é gerado a partir de um conversor que recebe 4 planilhas contendo os metadados dos artigos selecionados pelo coordenador de publicação. O processo de exportação dos artigos do JEMS3 para a SOL pode ser visualizado na figura 2.1.

Figura 2.1: Fluxograma do processo de exportação de artigos do JEMS3 para a SOL.



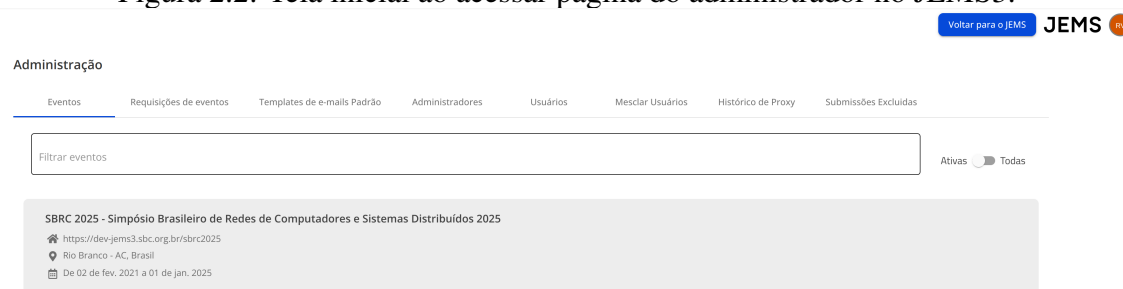
O problema identificado atualmente nesse processo de publicação dos artigos dos eventos na SOL é o tempo que demora para passar os metadados dos artigos para as planilhas usadas na geração do XML. Essa etapa é, atualmente, feita de forma manual, copiando e colando as informações de cada artigo nas planilhas. Além de ter um grande tempo demandado para realizar esse processo, existe uma grande probabilidade de acontecerem erros nessas transferências manuais dos metadados para as planilhas.

2.1 Processo atual para a exportação dos artigos na SOL

No sistema JEMS3, apenas usuários administradores têm acesso à área de administrador. Na página inicial dessa área (figura 2.2), irá constar a lista de todos os eventos em que aquele usuário é considerado administrador. Esse usuário pode conter os seguintes papéis no evento:

- Coordenador de evento: é quem gerencia o evento;
- Coordenador de publicação: é responsável pela exportação dos artigos na biblioteca digital;
- Coordenador de sessão: é o responsável por uma sessão do evento.

Figura 2.2: Tela inicial ao acessar página do administrador no JEMS3.



O usuário que está diretamente relacionado com o problema descrito neste trabalho é aquele que possui o papel de coordenador de publicação. Para esse usuário conseguir realizar o download da lista de artigos aprovados, e então exportá-los para a SOL, primeiramente ele deve selecionar o evento em questão que consta na lista de eventos descrita anteriormente. Ao selecionar o evento desejado, irá abrir a página inicial daquele evento (figura 2.3). Nela constam algumas informações gerais deste evento. Para continuar com o processo de exportação dos artigos, é preciso clicar no botão "Gerenciar evento".

A página de gerenciamento de um evento é o local onde o usuário pode realizar e visualizar todos os processos relacionados àquele evento. Nela, o usuário pode visualizar um dashboard das submissões, visualizar o calendário com as datas importantes do evento, visualizar quem são os coordenadores do evento, entre outros processos. Dentre esses processos, está o download de arquivos do evento (figura 2.4), que é o próximo passo para realizar o download dos artigos necessários para a exportação para a SOL. Para acessar essa página, é preciso clicar em Publicação → Download de arquivos.

Um formulário contendo dois campos de seleção será aberto, aonde o usuário irá selecionar primeiramente uma trilha. Todo evento pode conter diversas trilhas, que

Figura 2.3: Página inicial de um evento no JEMS3.

JEMS 🔔 Eventos 📄 Minhas submissões 📄 Minhas revisões 👤 TPCs que participo 🌟 Eventos que coordeno

SBRC 2025 - Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos 2025

🏠 <https://dev-jems3.sbc.org.br/sbrc2025>

📍 Rio Branco - AC, Brasil

📅 De 02 de fev. 2021 a 01 de jan. 2025

🔔 Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos 2025. Este é um evento teste.

Coordenador de TPC:

- Raphael Malinski Vieira <raphaelmalinski@gmail.com>
- Lisandro Granville2 <lisandro.granville@gmail.com>

[⚙️ Gerenciar evento](#)

Trilhas do evento

Full paper Trilha está fechada

- 🕒 Data limite de registro: 01 de ago. 2021 00:00 UTC-3 🌐
- 🕒 Manuscript - Data limite de envio: 28 de out. 2021 15:17 UTC-3 🌐
- 🕒 Final Version - Data limite de envio: 30 de set. 2021 13:08 UTC-3 🌐

[Submeter](#)

Short Paper Trilha está fechada

- 🕒 Data limite de registro: 31 de mai. 2021 22:38 UTC-3 🌐
- 🕒 Trackfile Nova - Data limite de envio: 20 de mai. 2021 14:33 UTC-3 🌐
- 🕒 track diferente - Data limite de envio: 03 de jun. 2024 14:23 UTC-3 🌐

[Submeter](#)

Figura 2.4: Tela do JEMS3 com formulário de download de arquivos.

JEMS 🔔 Evento 📄 Submissões 📄 Revisões 📧 Mensagens 📅 Programa 📄 Publicação RV

Controles de publicação

Status de publicação

Anais do evento

Download de arquivos

Exportar lista BibTeX

Download de arquivos

Selecione a trilha ▼

Selecione track files ▼

[Baixar arquivos](#)

podem ser, por exemplo, artigos curtos, artigos completos, entre outros. Após selecionada a trilha, o usuário deverá selecionar qual ou quais arquivos de trilha ele deseja incluir no download. As trilhas podem possuir um conjunto de arquivos, que podem ser, por exemplo, artigo submetido, versão final, comprovante de inscrição, entre outros. Com os campos selecionados corretamente, basta o usuário clicar no botão "Baixar arquivos" que o download será iniciado. O arquivo baixado é uma pasta compactada no formato ZIP, onde nela contém todos os artigos que fazem parte daquela trilha e de algum arquivo de trilha selecionado. Outra informação importante é que os artigos baixados são apenas aqueles que contêm o status de aceito ou pendente. O status "aceito" é quando o artigo foi aceito pelos revisores e o status "pendente" é quando o artigo submetido não possui PDF. Agora que o coordenador de publicação possui os artigos, ele precisa passar os metadados para as quatro planilhas necessárias para realizar a exportação para a SOL.

A primeira planilha que será descrita é a que contém as informações das seções

(Figura 2.5). As seções da SOL são equivalentes às trilhas do JEMS3. Essa planilha é a única entre as 4 que não utiliza as informações dos artigos. Ela é preenchida a partir de dados retirados do JEMS3 e da SOL. Cada linha da planilha representa uma seção e ela possui 9 colunas, sendo elas:

- *sectionTitle*: contém a informação do título da seção;
- *sectionTitleEn*: contém a tradução do título da seção para inglês. Caso o título da seção já esteja no referido idioma, ele deve ser repetido nesta coluna;
- *sectionAbbrev*: contém a abreviação do título da seção;
- *blind*: essa coluna recebe apenas o valor 1 ou 2. Quando for 1, significa que os artigos da seção correspondente foram selecionados por um processo de revisão *single-blind*. Quando for 2 caso os artigos foram selecionados por um processo de revisão *double-blind*;
- *numSubmitted*: contém o número de submissões de trabalhos que cada seção recebeu;
- *numAccepted*: contém o número de trabalhos que foram aceitos em cada seção;
- *dateSub*: contém a data da submissão dos artigos no formato mm/dd/yyyy;
- *dateResult*: contém a data da aceitação dos trabalhos no formato mm/dd/yyyy;
- *dateReady*: contém a data do envio da versão final trabalhos no formato mm/dd/yyyy.

Figura 2.5: Exemplo de preenchimento da planilha de seções.

	A	B	C	D	E	F	G	H	I
1	<i>sectionTitle</i>	<i>sectionTitleEn</i>	<i>sectionAbbrev</i>	<i>blind</i>	<i>numSubmitted</i>	<i>numAccepted</i>	<i>dateSub</i>	<i>dateResult</i>	<i>dateReady</i>
2	Artigos Completos	Full Papers	ART	1	25	11	8/5/2013	8/20/2013	8/27/2013
3	Artigos Curtos	Short Papers	SHORT	1	13	4	8/5/2013	8/20/2013	8/27/2013

A próxima planilha é a que contém as informações dos artigos (figura 2.6). Cada linha da planilha representa um artigo e ela possui 11 colunas, sendo elas:

- *seq*: representa o número sequencial do artigo. Ela deve ser iniciada em 1 e será usada para relacionar cada artigo incluído nessa planilha com a informações das outras planilhas;
- *language*: contém a informação do idioma original do artigo. Podem ser: "en", "es" ou "pt";
- *sectionAbbrev*: contém a abreviatura da seção a qual o artigo pertence. É obrigatório que os artigos devam estar agrupados e listados na mesma ordem definida na

Figura 2.6: Exemplo de preenchimento da planilha de artigos.

	A	B	C	D	E	F	G	H	I	J	K
1	seq	language	section	titleOrig	titleEn	abstractOrig	abstractEn	keywordsOrig	keywordsEn	pages	idJEMS
2	1	pt	ART	Técnicas de Do-It Yourself	Este artigo apresenta técnicas de do-it yourself	Técnicas do-it yourself	Do-it yourself	6	163791		
3	2	en	ART	SOI Based \ SOI Based	Sistemas de Recomendação	Recommendation Systems	Sistemas de Recomendação	7	163789		
4	3	pt	ART	GovMobile GovMobi	O avanço nas tecnologias móveis	The advancement of mobile technologies	Governo eletrônico	Electronic government	7	163875	
5	4	pt	ART	Padrões de Design Pa	Linguagens e padrões de design	Languages and design patterns	Compartilhamento de experiências	Experience sharing	7	162813	
6	5	pt	SHORT	Investigando o engajamento	The engagement	Gamificação; Colaboração	Gamification; Collaboration	6	163876		
7	6	en	SHORT	An Ontology for Sensemaking	Sensemaking	Collaborative and	Collaborative and	5	162925		
8	7	pt	SHORT	Do Facebook From Facebook	Em junho de 2013	In June 2013,	Comunidade Virtual	Virtual Community	12	163891	

planilha de seções. Além disso, artigos de seções diferentes não devem ser intercalados nessa planilha;

- *titleOrig*: contém o título no idioma original do artigo;
- *titleEn*: contém a tradução do título do artigo para inglês. Caso o título do artigo já esteja no referido idioma, ele deve ser repetido nesta coluna;
- *abstractOrig*: contém o resumo no idioma original do artigo;
- *abstractEn*: contém o resumo do artigo em inglês. Caso o artigo contenha apenas o resumo em inglês, ele deve ser informado novamente nesta coluna;
- *keywordsOrig*: as palavras-chave no idioma original do artigo. Elas devem estar separadas por ponto e vírgula;
- *keywordsEn*: as palavras-chave do artigo em inglês. Caso o artigo contenha apenas palavras-chave em inglês, elas devem ser informadas novamente neste campo, separadas por ponto e vírgula;
- *pages*: o número de páginas do artigo;
- *idJEMS*: a ID atribuída à submissão na qual o artigo pertence no JEMS3.

As próximas duas planilhas contêm um aumento considerável no número de informações a serem transferidas dos artigos para elas. A primeira delas é a planilha de autores (figura 2.7). Ela recebe as informações dos autores dos artigos separadas em 9 colunas, sendo elas:

- *article*: deve receber o número correspondente ao valor da coluna *seq* do artigo consonante na planilha de artigos;
- *authorFirstname*: contém o primeiro nome do(a) autor(a) de forma idêntica à apresentada no artigo;
- *authorMiddlename*: contém o nome do meio do(a) autor(a) de forma idêntica à apresentada no artigo;
- *authorLastname*: contém o sobrenome do(a) autor(a) de forma idêntica à apresen-

Figura 2.7: Exemplo de preenchimento da planilha de autores.

	A	B	C	D	E	F	G	H	I
1	article	authorFirstname	authorMiddlename	authorLastname	authorAffiliation	authorAffiliationEn	authorCountry	authorEmail	orcid
2	1				IFSP	IFSP	BR	edilson.torre@gmail.com	
3	1				IFSP	IFSP	BR	marcelo@gmail.com	
4	1				UFBA	UFBA	BR	edvaldo@acm.org	
5	2				UFRGS	UFRGS	BR	edvaldo@ufpr.br	
6	2				UFRGS	UFRGS	BR	edvaldo@ufpr.br	https://orcid.org/0000-0001-8991-4444
7	2				UDESC	UDESC	BR	luciano.gasperin@udesc.br	
8	3				UFRJ	UFRJ	BR	marcelo@im.ufrj.br	
9	3				MPOG	MPOG	BR	marcelo.chaves@planejamento.gov.br	
10	3				UFRJ	UFRJ	BR	edvaldo@im.ufrj.br	
11	3				UFRJ	UFRJ	BR	edvaldo@im.ufrj.br	
12	3				UFRJ	UFRJ	BR	edvaldo@im.ufrj.br	
13	4				UDESC	UDESC	BR	alexandre.matos@udesc.br	
14	4				UDESC	UDESC	BR	edvaldo@im.ufrj.br	
15	4				UDESC	UDESC	BR	luciano.gasperin@udesc.br	
16	4				UDESC	UDESC	BR	edvaldo@im.ufrj.br	
17	5				UDESC	UDESC	BR	luciano.gasperin@udesc.br	
18	5				UDESC	UDESC	BR	luciano.gasperin@udesc.br	https://orcid.org/0000-0001-8991-4444
19	5				UDESC	UDESC	BR	edvaldo@im.ufrj.br	
20	6				UFF	UFF	BR	edvaldo@im.ufrj.br	
21	6				ADDLabs / UFF	ADDLabs / UFF	BR	edvaldo@im.ufrj.br	
22	6				ADDLabs / UFF	ADDLabs / UFF	BR	edvaldo@im.ufrj.br	
23	7				UFF	UFF	BR	edvaldo@im.ufrj.br	
24	7				UFMT	UFMT	BR	edvaldo@im.ufrj.br	
25	7				UFF	UFF	BR	edvaldo@im.ufrj.br	

tada no artigo;

- *authorAffiliation*: contém a afiliação do(a) autor(a) preferencialmente a instituição, sem detalhamento de departamento interno ou programa. Se houver uma abreviatura padrão do nome da instituição, esta deve ser usada. Caso algum autor seja afiliado a mais de uma instituição de ensino, elas devem estar escritas separadas por uma barra "/", por exemplo, 'UFRGS / UFSC';
- *authorAffiliationEn*: contém a tradução da afiliação do(a) autor(a) para inglês. Caso o título do artigo já esteja no referido idioma, seja uma sigla ou não tenha tradução, ela deve ser repetido nesta coluna;
- *authorCountry*: contém o código do país em que está localizada a instituição correspondente à afiliação do(a) autor(a). Por exemplo, 'BR' para Brasil, 'DE' para Alemanha e 'FR' para França;
- *authorEmail*: contém o email do(a) autor(a) conforme consta no artigo;
- *orcid*: contém o ORCID do(a) autor(a) no formato URL. Por exemplo, "<https://orcid.org/0000-0002-8094-9261>".

Apesar do número de colunas ser menor que o da planilha de artigos, o número de linhas a ser preenchido pode aumentar de forma considerável. Cada linha da planilha representa um autor, onde eles estão agrupados pelo artigo no qual estão relacionados. No exemplo mostrado na figura 2.6, é possível ver que existem 7 linhas preenchidas na planilha (total de 77 células), mas na planilha mostrada na figura 2.7 o número de linhas salta para 24 (216 células). Mesmo que as colunas *authorMiddlename* e *orcid* não tenham o preenchimento obrigatório, o número de células aumenta de forma considerável.

A planilha que pode possuir o maior número de células a serem preenchidas é a

Figura 2.8: Exemplo de preenchimento da planilha de autores.

	A	B	C	D	E
1	article	description	doi	link	accessed
2	1	Cardoso, A.M.P.; Silva, S.A.A. Web Social: Aspectos culturais e interações de estudante	10.1145/2072609.2072625		
3	1	Creswell, J.W. Research Design: Qualitative, Quantitative, and Mixed Methods Approac			
4	1	Ericsson, K., & Simon, H. Protocol Analysis: Verbal Reports as Data. 2nd ed., MIT Press,			
5	1	Furtado, E.; Santiago, L.; Furtado, V. Uma estratégia para análise da adoção de sistema			
6	1	Krug, S. Don't Make Me Think: A Common Sense Approach to Web Usability. 2nd Editi			
7	1	Krug, S. Rocket Surgery Made Easy: The Do-It-Yourself Guide to Finding and Fixing Usa			
8	1	Lettner, F.; Holzmann, C. Heat maps as a usability tool for multi-touch interaction in mc			
9	1	Meira, S.R.L.; Costa, R.A.; Jucá, P.M.; Silva, E.M. Redes sociais. In: Fuks, H.; Pimentel, M			
10	1	Nicolaci-da-Costa, A.M.; Pimentel, M. Sistemas colaborativos para uma nova sociedade			
11	1	Revista Veterinária. Brasil dispara no ranking do mercado de animais de estimação.		http://www.revistaveterinaria.com.br/	1 set. 2013
12	1	Rosemberg, C.; Schilling, A.; Bastos, C.; Araripe, R. Prototipação de software e design p			
13	1	Santaella, L. Navegar no Ciberespaço: o perfil cognitivo do leitor imersivo. 2a. ed., Paul			
14	1	van Someren, M.W.; Barnard, Y.F.; Sandberg, J.A.C. The think aloud method: a practical			
15	2	Kittur, A, Chi, E.H., Sun, B. Crowdsourcing user studies with Mechanical Turk, In Proc. I			
16	2	Adomavicius G., Tuzhilin, A. Toward The Next Generation of Recommender Systems: A			
17	2	Brabham, D. Crowdsourcing as a Model for Problem Solving: An Introduction and Case			
18	2	Chakoo, N., Gupta, R., Hiremath, J. Towards Better Content Visibility in Video Recomm			
19	2	Chaves, A. P., Steinmacher, I., Vieira, V. Social Networks and Collective Intelligence App			

de referências (figura 2.8). Ela possui apenas 5 colunas, sendo elas:

- *article*: deve receber o número correspondente ao valor da coluna *seq* do artigo consonante na planilha de artigos;
- *description*: contém as referências de forma idêntica à informada no artigo;
- *doi*: contém os links DOI para as referências que possuem esse registro;
- *link*: contém a URL da referência informada em sua descrição. Caso não possua, deverá ficar em branco;
- *accessed*: contém a informação da data do último acesso ao site informado na URL da descrição da referência. Essa informação deve ser preenchida se a referência estiver disponível apenas em meios eletrônicos.

Apesar de possuir apenas 5 colunas e ainda as colunas *doi*, *link* e *accessed* serem preenchidas em poucos casos, o número de células que devem ser preenchidas nessa planilha tende a ser maior que o de autores. Isso ocorre porque, nessa planilha, cada referência dos artigos representa uma linha. No exemplo mostrado na figura 2.8, apenas o artigo com o número de sequência 1 já possui 13 referências, ou seja, de 26 a 65 células desta planilha deverão ser preenchidas para apenas 1 artigo submetido no evento.

O preenchimento das planilhas é a etapa mais trabalhosa e que demanda mais tempo no processo de exportação dos artigos para a SOL. Nesse pequeno exemplo mostrado nas figuras, com apenas 7 artigos, foi possível ver que o número de células que devem ser preenchidas manualmente com informações majoritariamente vindas dos artigos é enorme. Nos eventos científicos, é comum o número de artigos submetidos e aceitos passar dos 50. Com isso, o número de células a serem preenchidas deve passar dos 2000, deixando a probabilidade de alguma informação errada ser colocada nas planilhas quase certa.

Figura 2.9: Primeira parte do "Formulário de geração de arquivo XML - Anais de Evento".

SBC OPEN LIB

FORMULÁRIO DE GERAÇÃO DE ARQUIVO XML - ANAIS DE EVENTO

Edição do evento: **Sigla do evento:**

Cidade do evento: **Início do evento:**

Título dos Anais (Português):

Título dos Anais (Inglês):

Descrição (Português):

Descrição (Inglês):

Apesar do preenchimento das planilhas ser a etapa mais trabalhosa nesse processo, ela não é a última. Tendo as 4 planilhas preenchidas e a lista de artigos, o coordenador de publicação poderá gerar o arquivo XML que será, então, submetido na SOL.

Esse XML é gerado através do preenchimento de um formulário da SOL, que é chamado de "Formulário de geração de arquivo XML - Anais de Evento". Como pode ser visto na figura 2.9, esse formulário contém primeiramente campos referentes às informações do evento. Após os campos de informações, o formulário contém campos referentes às planilhas (figura 2.10). Nesses campos, o coordenador de evento precisa preencher a quantidade de itens em cada planilha, marcar a opção que indica o tipo de paginação e, no final, fazer a submissão do diretório que contém as 4 planilhas e todos os artigos que estão incluídos nelas. Tendo preenchido todos os campos de forma correta e não indicado nenhum erro nas planilhas, o download do arquivo XML será realizado.

Com o arquivo XML gerado corretamente, falta apenas realizar a submissão dele na SOL. Para a realização dessa etapa, o coordenador do evento precisa, primeiramente, realizar o login na página do evento na SOL. Com o login feito, ele conseguirá visualizar a página de submissões. Para acessar o local onde é feito o envio do XML, é necessário ir em Ferramentas → Importar/Exportar e depois clicar em "Plugin XML Nativo". A

Figura 2.10: Segunda parte do "Formulário de geração de arquivo XML - Anais de Evento".

The screenshot shows a web form with three main sections:

- Indique as quantidades de itens em cada planilha:** This section contains four input fields labeled "Artigos:", "Autores:", "Seções:", and "Referências:".
- Marcar a opção que indica o tipo de paginação:** This section has two radio button options:
 - Paginação pós-definida - A planilha contém apenas o número de páginas de cada artigo
 - Paginação pré-definida - A planilha contém uma string com a página inicial e a página final de cada artigo
- Selecione o diretório contendo os arquivos necessários:** This section includes a "Diretório" label, an "Escolher arquivos" button, and the text "Nenhum arquivo escolhido".

At the bottom of the form is a "Processar" button.

Figura 2.11: Página de envio de XML para exportação dos artigos do evento para a SOL.

The screenshot shows a web interface for uploading XML files. On the left is a sidebar menu with the SBC OPEN LIB logo and navigation items: "Tarefas 0", "Submissões", "Edições", "Configurações", "Usuários & Papéis", and "Ferramentas". The main content area is titled "Plugin XML Nativo" and has an "Importar" button. Below this, there is a text instruction: "Por favor, selecione um arquivo XML para carregar e pressione 'Iniciar Envio'. Quando o upload for concluído, pressione 'OK' para importar o documento no OMP." A dashed box contains a file upload area with the text "Arraste um arquivo aqui para iniciar o upload" and an "Enviar arquivo" button. At the bottom of the main area is another "Importar" button.

página de envio do XML será aberta (figura 2.11) e então basta o usuário clicar em "Enviar arquivo", selecionar o arquivo desejado e depois clicar em "Importar". Se nenhum erro for identificado no XML, então o processo de exportação dos artigos do evento para a SOL está finalizado.

Essa descrição de todo o processo para a exportação dos artigos de um evento gerenciado pelo JEMS3 para a SOL mostrou o quão trabalhoso ele é atualmente. Foi possível observar que existem etapas simples e rápidas dentro deste processo, mas outras que podem demandar um tempo muito grande para a sua conclusão. A etapa de preenchimento das planilhas é, claramente, a mais trabalhosa delas e que mais necessita de uma automatização, mas também existem outras que, apesar de não serem tão trabalhosas, possuem uma fácil automatização. E é isso que veremos no próximo capítulo, quais dessas etapas foram automatizadas e como esse processo foi feito.

3 SOLUÇÃO

No capítulo anterior, foi apresentado quão trabalhoso é o processo de exportação dos artigos de um evento para a SOL, feito pelo coordenador do evento. Dentre todo o processo, é possível notar que existem passos que são mais custosos atualmente do que outros. Entretanto, o que ficou muito claro é que a transferência dos metadados dos artigos para as planilhas é o passo mais trabalhoso dentre os descritos. Outros, como por exemplo o download dos artigos, não possui um impacto muito grande no processo. Outra informação que é importante ressaltar é que a transferência de dados para as planilhas é o que mais pode escalar em complexidade dependendo do número de artigos que serão exportados na SOL, quando outros passos não sofrem nenhuma alteração, independente do número de artigos. Com base nestas observações, foi optado pela solução ser focada em automatizar o preenchimento das planilhas a partir dos metadados dos artigos e do próprio JEMS3.

3.1 Conceitos Fundamentais e Tecnologias Envolvidas

Antes de falar da solução, é preciso explicar alguns conceitos teóricos e tecnológicos que serão citados e tratados no decorrer do texto. Os conceitos que serão explicados são: metadados e \LaTeX .

Os metadados já foram citados anteriormente neste texto, e agora serão formalmente introduzidos. Eles são informações estruturadas que descrevem e fornecem dados sobre outros dados. Eles ajudam a organizar, identificar e facilitar a localização de documentos ou objetos digitais. Por exemplo, em sistemas de bibliotecas digitais, metadados como título, autor, data de publicação e palavras-chave auxiliam na indexação e recuperação de conteúdo relevante.

O \LaTeX é um sistema de preparação de documentos amplamente utilizado em publicações acadêmicas e científicas. Ele permite a criação de documentos de alta qualidade tipográfica, como artigos, teses e relatórios, oferecendo controle preciso sobre elementos como equações matemáticas, tabelas e referências bibliográficas. Existem discussões que buscam identificar corretamente o que o \LaTeX é: uma linguagem de programação ou uma linguagem de marcação para construção de documentos. Nesse texto ela será tratada como uma linguagem de programação.

3.2 *Template* padrão para os artigos

O primeiro problema que foi pensado ao planejar a captura dos dados dos artigos e dos autores foi "*como padronizar o local em que os dados estão para facilitar o acesso a eles?*". A conclusão obtida foi a de tentar incluir o máximo de informações possível nos metadados do arquivo PDF do artigo. A SBC possuía um novo *template* padrão em \LaTeX (LATEX, 2024) que estava sendo desenvolvido pelo professor José Viterbo da Universidade Federal Fluminense (UFF) e pelo Nelson Silva Lago da Universidade de São Paulo (USP). Para essa solução funcionar foi preciso modificar esse *template* e tornar o uso deles obrigatório nos eventos da SBC. Ao modificar esse *template* padrão e fazer com que seja obrigatório o uso dele, conseguimos garantir que as informações fossem sempre passadas nos mesmos lugares e a extração desses metadados ficaria muito mais simples e segura. Com essa abordagem, quase todas as informações das planilhas de artigos e autores já estavam disponíveis para serem facilmente manipuladas.

Mesmo com essa abordagem escolhida, as informações das referências ainda não eram possíveis de serem acessadas. Esse problema foi identificado pelo fato de as referências ficarem armazenadas em um arquivo com a extensão `.bib` do pacote BibLaTeX (KIME; WEMHEUER; LEHMAN, 2023). Esse arquivo pode conter quantas referências o autor do artigo quiser, mesmo que ele não vá utilizá-las no texto. Com isso, não era possível garantir que apenas as referências incluídas no texto estavam armazenadas no arquivo `.bib`. Com esse problema identificado, foi optado por utilizar um *parser* que iria extrair as informações das referências diretamente do arquivo PDF do artigo.

3.3 Extrator de referências com biblioteca externa

Todos os dados utilizados na planilha que contém as informações das referências (exceto a coluna *seq*) estão no texto completo da referência no artigo. Então, foi necessário ir em busca de uma biblioteca que realizasse a extração de cada uma, pois, tendo o texto da referência, a busca pelos outros dados era trivial. Para esse processo, foi escolhida a biblioteca em Python *refextract* (HEP, 2025), especializada em extrair referências de textos, PDFs e strings.

Essa biblioteca se mostrou muito eficaz em relação à extração das referências. O uso dela foi muito simples e retornou corretamente todas as referências incluídas no artigo. Com isso, esses dados estavam acessíveis para transferência para as planilhas.

Entretanto, a biblioteca se mostrou quase inviável à medida que o número de artigos aumentava. Com um pequeno aumento de 3 para 6 artigos (com o mesmo número de referências de cada artigo) utilizados para a extração, o tempo necessário para a extração foi aumentando consideravelmente (tabela 3.1).

Tabela 3.1: Tabela de desempenho do extrator *singlethread*.

<i>Número de artigos</i>	<i>Referência por artigo</i>	<i>Tempo para gerar planilhas</i>
3	12	18 segundos
6	12	39 segundos
12	12	64 segundos

Por causa desse aumento gradativo no tempo necessário para extrair as referências com o aumento no número de artigos, foi optado por continuar utilizando a biblioteca, mas com o uso de *multithread*. Após repetir os mesmos testes feitos anteriormente com apenas uma thread, ficou claro que a melhora não foi significativa (tabela 3.2).

Tabela 3.2: Tabela de desempenho do extrator *multithread*.

<i>Número de artigos</i>	<i>Referência por artigo</i>	<i>Tempo para gerar planilhas</i>
3	12	18 segundos
6	12	26 segundos
12	12	50 segundos

Como pode ser visto nas tabelas 3.1 e 3.2, o tempo necessário para extrair as referências ficaria muito grande em eventos reais que podem conter mais de 50 artigos. Mesmo sendo um tempo menor do que o utilizado para realizar manualmente a transferência dos dados para as planilhas, ainda não era um tempo desejável. A partir desses dados, que foi optado por criar o próprio extrator de referências, focado em extraí-las a partir do padrão do *template*.

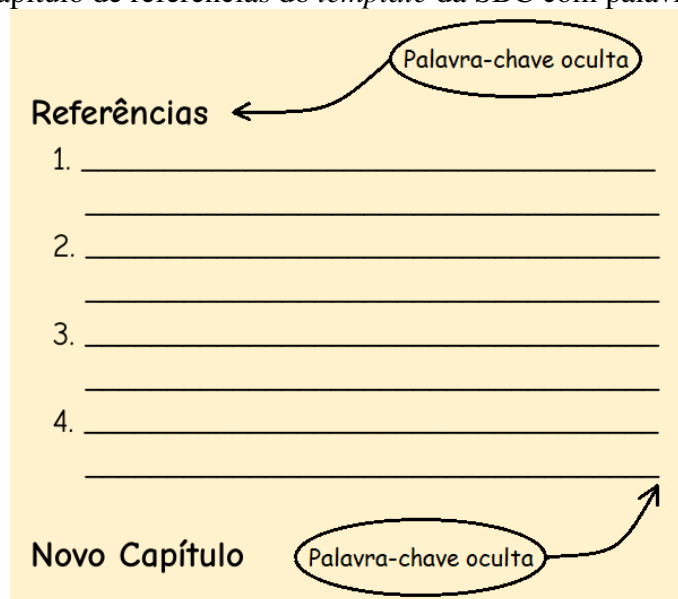
3.4 Extrator de referências com implementação própria

O extrator de referências que fosse implementado deveria fazer exatamente o que o usado anteriormente fazia, entretanto com uma eficiência maior. Para que isso fosse viável, o extrator deveria verificar a menor quantidade de texto possível para localizar o padrão do capítulo de referências de um artigo. A solução pensada para isso foi: incluir no início e no fim do capítulo de referências uma palavra-chave oculta (figura 3.1). Essa palavra-chave é um texto renderizado em branco e com o tamanho da fonte sendo o menor possível, ficando, na realidade, pseudo-oculto no texto. Dessa forma, o leitor

do artigo não veria essas palavras-chave no texto, mas ao passar o artigo pelo parser, ele identificaria elas. Incluindo essas duas palavras-chave, seria possível realizar a extração das referências em duas etapas:

1. Localiza no texto a palavra-chave de início do capítulo de referências;
2. Realiza o parse de cada referência até encontrar a palavra-chave de fim do capítulo de referências.

Figura 3.1: Capítulo de referências do *template* da SBC com palavras-chave ocultas



Entretanto, essa abordagem, apesar de ter melhorado a eficiência da extração, ainda não estava extraíndo as referências da forma correta. Ela esbarrava no problema em que as figuras e tabelas de um texto criado em \LaTeX podem ficar posicionadas no meio de um capítulo que não tem relação com elas. Ou seja, quando uma figura ou tabela ficava no meio do capítulo de referências, o parser não conseguiria diferenciar o texto de uma referência do da legenda de uma figura ou dos dados de uma tabela. Outro problema identificado foi o cabeçalho de uma página. Quando o capítulo da referência ficava entre duas páginas, o cabeçalho confundia o parser, assim como as figuras e tabelas. Um exemplo em que esses dois problemas ocorrem pode ser visualizado na figura 3.2. Para contornar o problema, sem que fosse necessário solicitar ao autor do artigo que não deixasse nenhuma outra informação ficar no meio do capítulo de referências, foi necessário mexer no *template* mais uma vez.

A nova abordagem usada é uma evolução da que estava sendo utilizada anteriormente. Foram mantidas no *template* as palavras-chave no início e no fim do capítulo de referências, e então incluídas novas palavras-chave no início e no fim de cada refe-

Figura 3.2: Trecho do *template* que ocorre problema no parser.

gráficas ou com texturas devem ser convertidas para o formato JPEG. Uma figura ou tabela muito grande (Figura 3, Tabela 2) pode excepcionalmente ocupar a largura das duas colunas de texto, mas apenas no início ou no final da página. Todas as figuras e tabelas devem ser acompanhadas das respectivas legendas explicativas numeradas logo abaixo delas.

6 Conclusion

- @Book: [1].

Referências

- 1 Knuth, D. E.; Larrabee, T.; Roberts, P. M. *Mathematical Writing*. The Mathematical Association of America, set. 1996.
- 2 Mittelbach, F. How to influence the position of float environments like figure and table in \LaTeX ? *TUGboat: Communications of the \TeX Users Group, \TeX Users Group, Portland (OR), USA, v. 35, n. 3, 2014*. Disponível em: tug.org/TUGboat/tb35-3/tb111mitt-float.pdf. Acesso em: 9 jan. 2020.

Exemplo de artigo em Português para a SOL

Viterbo et al. 2025

Experimento número:	1		Data:				jan 2017
Título:	Medições iniciais						
Tipo:	Levantamento quantitativo						
Locais	São Paulo	Rio de Janeiro	Porto Alegre	Recife	Manaus	Brasília	Rio Branco
Valores obtidos	0.2	0.3	0.2	0.7	0.5	0.1	0.4 ^a

^a This is an example of table footnote with a reference
This is an example of table footnote without a reference

Tabela 2. A table spanning both text columns.

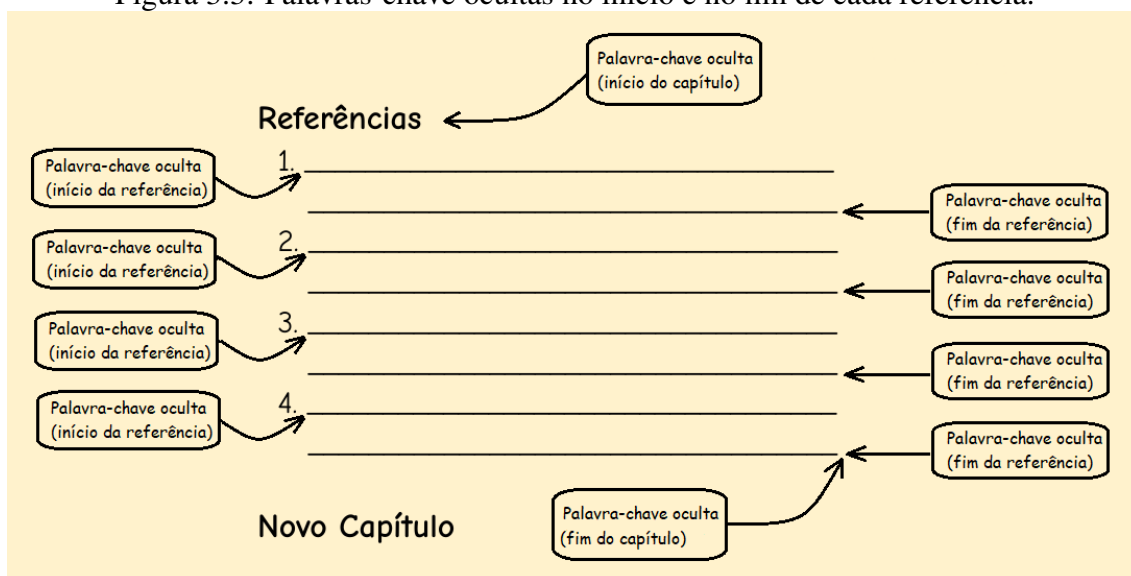
- 3 Alves, C. E. R. et al. A Parallel Wavefront Algorithm for Efficient Biological Sequence Comparison. In: ICCSA'03: The 2003 International Conference on Computational Science and its Applications. Springer-Verlag, mai. 2003. P. 249–258.

- 9 Allcock, W. *GridFTP Protocol Specification. Global Grid Forum Recommendation (GFD.20)*. 2003.
- 10 Fowler, M. *Is Design Dead?* Mai. 2004. Disponível em: martinfowler.com/articles/designDead.html. Acesso em: 30 jan. 2010.

rência no capítulo (figura 3.3). Com esse novo método, foi possível identificar sempre aonde uma referência inicia e termina, visto que cada uma não pode ser dividida entre duas páginas e separada por figuras ou tabelas. Essa nova alteração incluída no *template* permitiu com que as referências fossem acessadas sem erros pelo extrator de referências implementado.

Com o *template* pronto e o extrator de referências com implementação própria modificado para lê-las corretamente, foram realizados novos testes. Foi possível perceber uma melhora muito significativa no desempenho da extração das informações em comparação ao extrator usando a biblioteca *refextract*. Os testes foram realizados com *multithread* e *singlethread*, e os resultados se mostraram muito semelhantes entre si (tabelas 3.3 e 3.4). Devido a essa proximidade no desempenho, foi optado por manter apenas uma *thread*, pois a pequena melhora que o uso de mais *threads* trazia não justificava o aumento na complexidade da implementação.

Figura 3.3: Palavras-chave ocultas no início e no fim de cada referência.

Tabela 3.3: Tabela comparação extratores *singlethread*.

Número de artigos	Referência por artigo	Tempo para gerar planilhas (biblioteca)	Tempo para gerar planilhas (próprio)
3	12	18 segundos	1.5 segundos
6	12	39 segundos	2.8 segundos
12	12	64 segundos	5 segundos

3.5 Validação do *template* no JEMS3

Para garantir o correto funcionamento da geração das planilhas diretamente pelo JEMS3, a implementação de algumas outras funcionalidades foi necessária. A primeira delas é a validação do artigo submetido pelo autor, visando garantir que o *template* correto está sendo utilizado. Como o JEMS3 gerencia vários eventos organizados por diferentes instituições, foi necessária a possibilidade do coordenador de evento indicar se o *template* da SBC deve ser seguido pelos autores. Por isso, foi incluído o campo do tipo *checkbox* "Uso do *template* da SBC" no formulário de criação/edição de uma trilha (figura 3.4). Esse campo permite que o sistema saiba quando deve verificar se o artigo submetido está utilizando o *template* da SBC.

A validação do artigo é realizada quando o autor realiza a submissão no JEMS3. Para o autor estar ciente de que o seu artigo submetido está seguindo ou não o *template* da SBC, foi incluído um ícone logo abaixo do campo de submissão de artigo que indica o resultado da validação (figura 3.5). Os ícones à direita em cada *card* de submissão indicam se o arquivo submetido está seguindo ou não o *template*. No exemplo da figura

Tabela 3.4: Tabela comparação extratores *multithread*.

Número de artigos	Referência por artigo	Tempo para gerar planilhas (biblioteca)	Tempo para gerar planilhas (próprio)
3	12	18 segundos	1.2 segundos
6	12	26 segundos	2.2 segundos
12	12	50 segundos	4.6 segundos

Figura 3.4: Formulário de criação/edição de uma trilha no JEMS3.

Manuscript

Título do arquivo de trilha *

Manuscript

Número máximo de páginas (0: sem limite)

3 Permitir que o arquivo exceda o número máximo de páginas

Tornar público em *

14/07/2022, 09:00:00 (UTC-03:00)

Envio obrigatório: Obrigatório

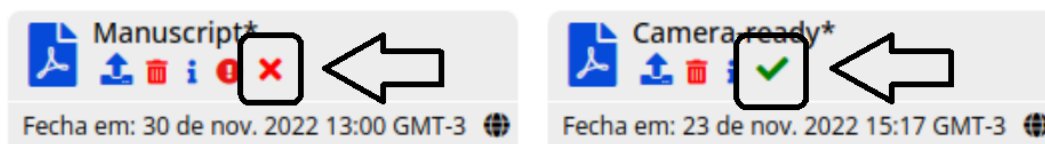
Visível para: Coordenador de Sessão

Uso do template da SBC: Obrigatório

3.5, o arquivo submetido no arquivo de trilha "Manuscript" não está seguindo o *template*, enquanto o submetido no arquivo de trilha "Camera-ready" está. Quando a trilha não está marcada para utilizar o *template*, nenhuma validação é feita e os ícones não são mostrados.

Figura 3.5: Cards de submissão de arquivo no JEMS3.

Arquivos (Arras e solte arquivos para upload)



3.6 Geração das planilhas no JEMS3

Para o JEMS3 está sendo desenvolvida uma funcionalidade de realizar a geração de anais de um evento, aonde esse processo era dividido em 6 etapas:

1. ISBN/ISSN: etapa na qual podem ser informados os códigos que irão identificar os anais gerados;
2. Conteúdo: nela são informadas as sessões técnicas que serão incluídas nos anais do evento, podendo ordená-las conforme preferência do usuário;
3. Arquivo de trilha: nela são informados quais arquivos de trilha serão incluídos nos anais;
4. Índices: nela podem ser informados os índices dos anais do evento;
5. Pré-visualização: nessa etapa o usuário pode realizar uma pré-visualização dos anais que serão gerados;
6. Gerar: a última etapa, aonde a geração do arquivo é finalmente realizada.

Essa nova funcionalidade ainda não está finalizada, mas as suas telas estão implementadas. A funcionalidade de gerar as planilhas da SOL foi incluída como a quarta etapa neste processo. Essa nova etapa não irá interferir na geração dos anais pelo JEMS3, podendo ser pulada sem ser preenchida. A geração das planilhas foi incluída nessa posição no processo porque, dessa forma, o coordenador de publicação pode selecionar os artigos e arquivos de trilha que serão incluídos nas planilhas nas etapas 2 e 3, não sendo mais necessário baixar todos os artigos como era feito antes.

A etapa de geração das planilhas da SOL conta, inicialmente, apenas com um campo de seleção (figura 3.6). Nele, o coordenador de publicação deve, primeiramente, selecionar qual série da SOL será utilizada. Essa informação é necessária para poder saber quais seções estão disponíveis para serem incluídas nas planilhas. Ao selecionar uma série, será aberto um mini formulário para cada trilha selecionada na etapa 3. Em cada um, o usuário irá selecionar apenas um arquivo de trilha e qual a seção da SOL o arquivo de trilha está associado. Com o formulário preenchido corretamente (figura 3.7), o botão para geração das planilhas ficará disponível.

Figura 3.6: Formulário da geração das planilhas no JEMS3 não preenchido.

A imagem mostra a interface do usuário do JEMS3. No topo, há o menu de navegação com ícones para Evento, Submissões, Revisões, Mensagens, Programa e Publicação. À esquerda, há um menu lateral com opções como 'Controles de publicação', 'Status de publicação', 'Anais do evento', 'Download de arquivos' e 'Exportar lista BibTeX'. O conteúdo principal exibe o progresso do processo em sete etapas numeradas: 1. ISBN/ISSN, 2. Conteúdo, 3. Arquivo de trilha, 4. Seções da SOL (destacada), 5. Índices, 6. Pré-visualização e 7. Gerar. Abaixo do progresso, há um campo de seleção rotulado 'Selecione a série da SOL' com uma seta para baixo. Na base da tela, há três botões: 'Voltar' (azul), 'Próximo' (azul) e 'Gerar planilhas da SOL' (cinza).

Nessa etapa do processo de geração das planilhas, todos os dados necessários para

Figura 3.7: Formulário da geração das planilhas no JEMS3.

Anais do evento

1 ISBN/ISSN — 2 Conteúdo — 3 Arquivo de trilha — 4 Seções da SOL — 5 Índices — 6 Pré-visualização — 7 Gerar

Selecione a série da SOL
Anais da Brazilian Conference on Intelligent Systems (BRACIS)

Full paper

Manuscript Camera-ready Letter

Selecione a seção associada
Abertura

Short paper

Trackfile 1

Selecione a seção associada
Artigos

[Voltar](#) [Próximo](#) [Gerar planilhas da SOL](#)

Figura 3.8: Lista artigos que não estão seguindo o *template* da SBC.

Anais do evento

1 ISBN/ISSN — 2 Conteúdo — 3 Arquivo de trilha — 4 Seções da SOL — 5 Índices — 6 Pré-visualização — 7 Gerar

Submissões que contém artigos que não estão seguindo o template da SBC

ID	Título	Título do Arquivo de Trilha
#1	Binary Decision Diagrams on a Cluster Linux/Beowulf	Camera-ready

[Ok](#)

Short paper

Trackfile 1

Selecione a seção associada
Artigos

o preenchimento delas já estão disponíveis. Quando for clicado no botão "Gerar planilhas da SOL", a extração desses dados começará a ser realizada nos artigos e no JEMS3. Entretanto, uma validação do *template* é feita a cada novo artigo que terá seus dados extraídos. Caso algum dos artigos incluídos não esteja utilizando o *template*, a geração das planilhas para e apenas a validação de todos os artigos continua sendo realizada até todos serem verificados. Ao término desse processo, uma lista com todos os artigos que não estão utilizando o *template* (figura 3.8) é exibida na tela para o usuário ficar ciente do porquê as planilhas não foram geradas. Caso todos os artigos estejam seguindo o *template* e nenhum outro erro seja identificado, as 4 planilhas são geradas já preenchidas.

4 QUESTÕES TÉCNICAS DA IMPLEMENTAÇÃO DO *TEMPLATE* EM \LaTeX

Foi possível perceber que a solução apresentada para mitigar o tempo necessário para a exportação dos artigos de um evento para a SOL necessitou de abordagens diferentes na sua implementação. Para o seu sucesso, foi primeiro preciso modificar o *template* desenvolvido pela SBC. Esse *template* foi desenvolvido em \LaTeX pois ele permite o desenvolvimento de textos mais complexos, como artigos e teses. O \LaTeX foi criado baseado na linguagem de marcação em arquivos \TeX (KNUTH, 1984). Ele é muito versátil e suporta, nos seus textos, a inclusão de fórmulas matemáticas, tabelas, gráficos, entre outros. Possui diversas bibliotecas de pacotes que permitem a personalização e trazem maior poder de desenvolvimento nos documentos. Além disso, é compatível com diversos sistemas operacionais e também pode ser desenvolvido em editores online, como o Overleaf (OVERLEAF, 2024). Devido a essas vantagens e facilidades de desenvolvimento, o \LaTeX foi escolhido para o desenvolvimento do *template*.

Em \LaTeX , a forma utilizada para incluir metadados em um documento PDF é através do comando *hypersetup*. Esse comando faz parte do pacote *hyperref*, que é usado para criar links e referências interativas em documentos PDF. Esse comando permite configurar várias opções relacionadas a *hyperlinks*, como cor, comportamento e as informações de metadados do documento. A inclusão dos metadados é feita através da passagem da opção de configuração *pdfinfo*. Essa opção de configuração é um objeto que irá receber os metadados que serão incluídos no PDF. As chaves do objeto serão os nomes dos metadados e o que estiver após o símbolo de igualdade será o conteúdo de cada metadado. Na figura 4.1 pode ser visto um exemplo de código \LaTeX onde o metadado *foo* irá receber o conteúdo *bar*.

É importante ressaltar que o *template* foi implementado de forma que nenhum metadado incluído no PDF do *template* será passado pelo usuário diretamente pelo comando *hypersetup*. Todos os metadados irão acessar as informações indiretamente através dos comandos usados para incluir as informações do artigo e dos autores no texto.

Figura 4.1: Exemplo de inclusão de metadado em \LaTeX utilizando o comando *hypersetup*.

```

1 ▾ \hypersetup {
2 ▾   pdfinfo={
3     foo=bar
4   }
5 }
```

4.1 Dados da planilha de autores

No *template*, os dados dos autores são todos incluídos no texto através do comando *author*, onde cada autor incluído no texto é informado por uma chamada desse comando, ou seja, se o artigo tiver dois autores, o comando será chamado duas vezes. Nesse comando, o usuário pode informar as seguintes opções de configuração:

- *firstName*: primeiro nome do autor;
- *middleName*: nome do meio do autor;
- *lastName*: sobrenome do autor;
- *institutionID*: sigla da instituição na qual o autor está representando. Pode ser informada mais de uma instituição, nesse caso as siglas devem ficar separadas por vírgula;
- *country*: abreviação do país que está localizada a instituição informada;
- *email*: e-mail do autor;
- *orcid*: ORCID do autor.

É possível perceber que essas opções de configuração são as mesmas existentes na planilha de autores. Então, todas as informações dessa planilha (exceto coluna *seq*) serão retiradas desse comando. Cada metadado que contém alguma informação de autor será uma lista contendo aquela informação de todos os autores. Por exemplo, o metadado *authorListFirstName* é aquele que possui uma lista com todos os primeiros nomes dos autores informados separados por ponto e vírgula. Esse mesmo padrão será utilizado para todas as outras informações de autor.

Para montar essas listas, foram criados comandos para cada informação, aonde eles seguem o mesmo padrão. Para o primeiro nome dos autores, por exemplo, foi criado o comando *simpleAuthorFirstNameList*. Esses comandos pegam a informação correspondente de cada chamada do comando *author* e juntam, separando por ponto e vírgula. Com isso, o código de criação dos metadados de autores foi implementado conforme pode ser visto na figura 4.2.

Figura 4.2: Código em \LaTeX da inclusão dos metadados de autores.

```

2293 ...
2294 % Informações de autores
2295 authorListFirstName={\simpleAuthorFirstNameList},
2296 authorListMiddleName={\simpleAuthorMiddleNameList},
2297 authorListLastName={\simpleAuthorLastNameList},
2298 authorAffiliation={\simpleAuthorInstitutionList},
2299 authorCountry={\simpleAuthorCountryList},
2300 authorEmail={\simpleAuthorEmailList},
2301 orcid={\simpleAuthorOrcidList}
2302 ...

```

Figura 4.3: Exemplo do uso do comando *resumo* em \LaTeX .

```

243 ...
244 \resumo
245 {
246     Este texto, em formato de artigo científico, tem por objetivo demonstrar o uso do novo
    template para artigos da SBC para anais de eventos da SOL, descrevendo suas principais
    características e explicando como deve ser utilizado para artigos em português.
    Sugerimos que o resumo em português tenha entre 500 e 750 palavras, mas os
    organizadores de cada evento podem estabelecer outros limites.
247 }
248 ...

```

4.2 Dados da planilha de artigos

A planilha que contém os dados dos artigos possui sete das 11 colunas que serão preenchidas via metadados no *template*. Essas colunas são: *language*, *titleOrig*, *titleEn*, *abstractOrig*, *abstractEn*, *keywordsOrig* e *keywordsEn*. Diferentemente da planilha de autores, as informações do artigo não serão buscadas apenas por um comando. Os comandos que contêm quase todas as informações do artigo que irão para a planilha são:

- *title*: usado para incluir o título no idioma original do texto e a tradução em inglês;
- *resumo*: usado para incluir o resumo do texto em português;
- *abstract*: usado para incluir o *abstract* do texto;
- *palavraschave*: usado para incluir as palavras-chave do texto em português;
- *keywords*: usado para incluir as palavras-chave do texto em inglês.

A partir desses cinco comandos é que serão criados seis metadados que, posteriormente, irão preencher as mesmas seis colunas na planilha. Com exceção do comando *title*, todos os comandos são simples e contêm apenas a informação como conteúdo, como pode ser visto na figura 4.3 um exemplo do comando *resumo*. O comando *title* possui duas opções de configuração que podem ser informadas, sendo uma delas obrigatória. São elas:

- *mainlanguagetitle*: deve conter o título do artigo no mesmo idioma em que foi escrito, ou seja, se o artigo for em português, o campo deve conter o título em português, se o artigo for em inglês, o campo deve conter o título em inglês;
- *translatedtitle*: deve conter uma versão do título em inglês, mas somente se o artigo for em português ou espanhol. Se o artigo for em inglês, o campo deve permanecer vazio.

O conteúdo informado em cada um destes comandos na criação do texto foi armazenado na implementação prévia do *template* em variáveis denominadas "*clean*". Por exemplo, o conteúdo do comando *resumo* foi incluído na variável *cleanresumo*. Essas variáveis foram denominadas dessa forma, pois na sua criação foi incluído um tratamento para remover as quebras-de-linha para poderem ser incluídas nos metadados. Os conteúdos dos comandos *keywords* e *palavraschave* não recebiam nenhum tratamento, então foi implementada uma função para separar as palavras-chave dos dois comandos por ponto e vírgula. Esse tratamento foi incluído para ficar conforme o padrão esperado na planilha e, com isso, foram criadas variáveis "*clean*" para ambos os comandos.

A única coluna da planilha de artigos que não possui o seu conteúdo retirado destes comandos é a coluna *language*. O metadado criado para conter a informação dessa coluna será preenchido através da variável *language* que já estava previamente implementada no *template*. Nesse caso, foi necessário apenas incluí-la na criação do metadado.

Com isso, todas as informações necessárias para criar os metadados do artigo estão acessíveis e sendo retiradas diretamente da informação incluída no texto. Isso garante que não exista divergência entre o que está no texto e o que está sendo incluído nos metadados. Como o artigo criado pelo autor pode ser em inglês, o *template* irá verificar se o idioma definido do texto é em inglês antes de criar os metadados. Caso seja, os metadados criados terão suas informações repetidas nos metadados usados para tradução do conteúdo original (figura 4.4). Caso o texto não seja em inglês, terá uma variável diferente para cada metadado (figura 4.5).

4.3 Alterações para uso do extrator de referências

As referências do *template* da SBC foram implementadas com o uso do pacote BibLaTeX. Ela é uma ferramenta amplamente usada no L^AT_EX para gerenciar e formatar referências bibliográficas de maneira automatizada, semelhante ao BibTeX (BIBTEX,

Figura 4.4: Código em \LaTeX da inclusão dos metadados do artigo em inglês.

```

2252 ...
2253 % Informações de artigos
2254 language={\@language},
2255 titleOrig={\@cleantitle},
2256 titleEn={\@cleantitle},
2257 abstractOrig={\@cleanabstract},
2258 abstractEn={\@cleanabstract},
2259 keywordsOrig={\@cleankeywords},
2260 keywordsEn={\@cleankeywords},
2261 ...

```

Figura 4.5: Código em \LaTeX da inclusão dos metadados do artigo.

```

2283 ...
2284 % Informações de artigos
2285 language={\@language},
2286 titleOrig={\@cleantitle},
2287 titleEn={\@cleantranslatedtitle},
2288 abstractOrig={\@cleanresumo},
2289 abstractEn={\@cleanabstract},
2290 keywordsOrig={\@cleanpalavraschave},
2291 keywordsEn={\@cleankeywords},
2292 ...

```

2024). Ela permite armazenar referências em um arquivo separado com extensão `.bib` e as formata de acordo com estilos bibliográficos predefinidos. O `BibLaTeX` é mais flexível e poderoso que o `BibTeX` tradicional, permitindo personalizar praticamente todos os aspectos da formatação de citações e referências, o que será ideal para esse *template*. O comando que é chamado para incluir no texto as referências é o `printbibliography`. Ao chamar esse comando no texto, o capítulo de referências é criado, contendo todas as referências que foram citadas no artigo.

Para alterar o capítulo de referências visando permitir que o extrator funcionasse corretamente, foi necessário criar um novo comando que foi chamado de `SBCprintbibliography` (figura 4.6). Esse comando tem o objetivo de incluir as palavras-chave ocultas no início e no fim do capítulo de referências. A implementação dele consiste em chamar o `printbibliography` entre dois outros comandos: `SBCprintInitKeyBibliography` e `SBCprintEndKeyBibliography`. O primeiro inclui o texto `###init_bibliography###` no início do capítulo de referências e o segundo inclui o texto `###end_bibliography##` no fim do capítulo. Os textos incluídos ficam, na realidade, "pseudo-ocultos" porque as palavras-chave são incluídas com a cor branca e o tamanho da fonte 0.1pt (figura 4.7). Com isso, eles ficam invisíveis no texto, mas estão presentes na leitura do PDF por um leitor de arquivo PDF.

A outra alteração necessária no *template* para permitir o correto funcionamento

Figura 4.6: Código em \LaTeX da implementação do comando *SBCprintbibliography*.

```

2072 ...
2073 \newcommand\SBCprintbibliography[1]{%
2074   \protected@xdef\@SBCprintbibliography{\par
2075     \SBCprintInitKeyBibliography\par
2076     \printbibliography\par
2077     \SBCprintEndKeyBibliography\par
2078     \trim@spaces@noexp{#1}\par
2079   }%
2080 }
2081 ...

```

Figura 4.7: Código em \LaTeX da implementação dos comandos *SBCprintInitKeyBibliography* e *SBCprintEndKeyBibliography*.

```

1700 ...
1701 \NewDocumentCommand\SBCprintInitKeyBibliography{}{
1702   \vspace{2mm}
1703   \textcolor{white}{
1704     {\fontsize{0.1pt}{5pt}\selectfont\#\#init_bibliography\#\#}
1705   }
1706   \vspace{-7mm}
1707 }
1708
1709 \NewDocumentCommand\SBCprintEndKeyBibliography{}{
1710   \vspace{-5mm}
1711   \textcolor{white}{
1712     {\fontsize{0.1pt}{5pt}\selectfont\#\#end_bibliography\#\#}
1713   }
1714 }
1715 ...

```

do extrator foi a inclusão de mais palavras-chave ocultas, mas agora no início e no fim de cada referência. O primeiro desafio para isso foi tentar localizar uma forma de poder editar as referências de forma individual, pois o comando que realiza a impressão delas no texto é de um pacote externo. Felizmente, o *template* já continha um comando que foi criado com o intuito de editar as referências conforme a preferência da SBC; esse comando é o *SBCbibConfig*.

Para personalizar uma referência implementada com o uso do BibLaTeX, foi preciso redefinir os comandos *begentry* e *finentry*, que, neste contexto, são chamados de *bibmacros*. Os *bibmacros* são semelhantes a funções ou modelos que encapsulam partes específicas do formato bibliográfico. O *begentry* e o *finentry* são comandos do BibLaTeX usados para executar ações no início e no final de cada entrada bibliográfica, respecti-

vamente. Para redefinir esses comandos, foi utilizado outro comando do BibLaTeX, o *renewbibmacro*.

A implementação da redefinição desses comandos é bem semelhante a feita nos comandos usados para incluir as palavras-chave ocultas no início e fim do capítulo. Na redefinição do *begentry* foi incluído o texto "[begin-ref-item]" com o tamanho de fonte de 0.1pt e cor branca. Na do *finentry* foi incluído o texto "[end-ref-item]" com o mesmo tamanho e cor de fonte do *begentry*, mas incluindo um ponto final. Esse ponto final foi incluído porque, ao redefinir esse comando, o ponto final não era mais incluído no final de cada referência, então foi necessário colocá-lo na redefinição. A implementação final da redefinição destes comandos pode ser vista nas figuras 4.8 e 4.9.

Figura 4.8: Código em L^AT_EX da implementação da redefinição do comando *begentry*.

```

911 ...
912 ▾ \renewbibmacro*{begentry}{%
913 ▾   \textcolor{white}{%
914 ▾     {\fontsize{0.1pt}{5pt}\selectfont%
915 ▾       \printtext[brackets]{%
916 ▾         \begingroup%
917 ▾           begin-ref-item%
918 ▾         \endgroup%
919 ▾       }%
920 ▾     }%
921 ▾   }%
922 ▾ }%
923 ...

```

Figura 4.9: Código em L^AT_EX da implementação da redefinição do comando *finentry*.

```

923 ...
924 ▾ \renewbibmacro*{finentry}{%
925 ▾   \hspace{-1.5mm}
926 ▾   \textcolor{black}{.}
927 ▾   \textcolor{white}{%
928 ▾     {\fontsize{0.1pt}{5pt}\selectfont%
929 ▾       \printtext[brackets]{%
930 ▾         \begingroup%
931 ▾           end-ref-item%
932 ▾         \endgroup%
933 ▾       }%
934 ▾     }%
935 ▾   }%
936 ▾ }%
937 ...

```

4.4 Metadado para verificar o uso do *template*

Para que o JEMS3 pudesse verificar se o artigo submetido está seguindo o *template* correto da SBC, optou-se por incluir um metadado que foi utilizado para validar esse uso. O metadado incluído para este fim foi o *templateVersion*. O conteúdo dele é a versão do *template*, então sempre que uma nova versão for lançada, esse valor será alterado e, conseqüentemente, deverá ser alterado no JEMS3. O valor incluído no *template* para a implementação desta solução foi "1.0".

5 QUESTÕES TÉCNICAS DAS IMPLEMENTAÇÕES NO JEMS3

Além da modificação do *template* da SBC, foi preciso implementar novos componentes no *frontend* do JEMS3 para interação do usuário com a solução e implementar toda a lógica da solução no *backend*, que consistia em: implementar o extrator de referências, incluir novos campos no banco de dados, abrir e ler os metadados dos artigos, validar o *template* usado nos artigos e, por fim, escrever todas as informações nas planilhas. Neste capítulo, irei detalhar de forma mais técnica e aprofundada tudo o que foi implementado no JEMS3 para que essa solução funcionasse corretamente, falando sobre as tecnologias que foram testadas e utilizadas.

5.1 Implementações de telas no *frontend*

O *frontend* do JEMS3 é implementado utilizando o *framework* Angular (GOOGLE, 2024). Ele é um *framework frontend* de código aberto, desenvolvido e mantido pelo Google, projetado para construir aplicações web dinâmicas. Essas aplicações são compostas de componentes reutilizáveis, que encapsulam lógica, layout e estilo de uma parte específica da interface. O Angular utiliza *TypeScript* (MICROSOFT, 2010), uma linguagem de programação tipada que é considerada um "*superset*" do *JavaScript* (FLANAGAN; TORTELLO; NEDEL, 2012), o que significa que tudo o que funciona em *JavaScript* também funciona em *TypeScript*.

Um componente no Angular é uma das unidades fundamentais de construção de uma aplicação. Ele encapsula a lógica, o layout e o estilo de uma parte específica da interface do usuário. Ele é composto por três partes principais:

1. Classe (*TypeScript*): Define a lógica do componente;
2. Template (HTML): Define o layout ou estrutura do componente;
3. Estilo (CSS/SCSS): Define os estilos aplicados ao componente.

5.1.1 Validação do uso do *template* da SBC

Para implementar o recurso de validação do *template* da SBC, foi necessário realizar alterações em alguns componentes do JEMS3. O primeiro deles foi o *EventTrack-EditComponent*. Esse componente implementa o formulário de criação/edição de uma

trilha de um evento no JEMS3 (figura 3.4). O Angular permite criar componentes reutilizáveis no restante do código. Para incluir o campo do tipo checkbox "Uso do *template* da SBC" no formulário, foi preciso apenas utilizar o componente *CheckboxInputComponent* previamente existente em conjunto com algumas tags *div* em HTML (DUCKETT, 2011) para o correto posicionamento do campo no formulário. Para o campo incluído funcionar corretamente com a lógica do componente *EventTracksEditComponent*, foi necessária a inclusão do atributo do tipo booleano *useOfSbcTemplate*, que irá ter o seu valor de acordo com o que for selecionado no checkbox.

Outro componente que necessitou de alterações foi o *FileRulesCardComponent*. Esse componente implementa os *cards* de submissão de arquivo no JEMS3 (figura 3.5). O Angular permite a inclusão de lógica de programação no HTML através dos atributos das tags que são chamadas de diretivas. Para a correta implementação da exibição dos ícones de acordo com o resultado da validação do *template* foi utilizada a diretiva *ngIf*. Ela é uma diretiva estrutural do Angular usada para condicionalmente incluir ou excluir elementos com base na avaliação de uma expressão booleana. Nessa implementação (figura 5.1) a diretiva foi utilizada de forma encadeada, utilizando os métodos *needSbcTemplate* e *isUsingSbcTemplate* respectivamente. O primeiro método verifica se o *template* é necessário naquela trilha e o segundo verifica se o artigo está realmente utilizando o *template*.

Figura 5.1: Implementação do ícone de validação do *template* da SBC.

```

...
<ng-container *ngIf="needSbcTemplate(slot)">
  <ng-container *ngIf="isUsingSbcTemplate(slot); else dontUse">
    <a class="button-rule-uses-sbc-template">
      <mat-icon fontSet="fas" fontIcon="fa-check" class="file-action"
        [matTooltip]='forms.buttons.is-using-sbc-template' | translate"></mat-icon>
    </a>
  </ng-container>
  <ng-template #dontUse>
    <a class="button-rule-not-uses-sbc-template">
      <mat-icon fontSet="fas" fontIcon="fa-times" class="file-action"
        [matTooltip]='forms.buttons.is-not-using-sbc-template' | translate"></mat-icon>
    </a>
  </ng-template>
</ng-container>
...

```

5.1.2 Formulário de geração das planilhas da SOL

O formulário implementado para gerar as planilhas da SOL foi incluído dentro da funcionalidade ainda não finalizada do JEMS3 que é a geração de anais de um evento. O *frontend* dessa funcionalidade é composto por dois componentes:

- *EventPublicationProceedingsComponent*: componente responsável por implementar a lógica de formulário em etapas e o layout das etapas;
- *ProceedingStepComponent*: componente responsável por implementar o conteúdo de todas as etapas e a lógica de cada uma, realizando a transição de informações passadas de uma etapa à outra.

No componente *EventPublicationProceedingsComponent* a única alteração realizada foi a inclusão da nova etapa chamada *Sections*. Essa etapa representa a etapa de geração das planilhas da SOL. O componente foi implementado para receber no atributo *steps* da sua classe uma lista contendo objetos com as chaves *name* e *label*. O layout do componente, então, percorre essa lista em ordem e cria na tela cada etapa, incrementando cada etapa em 1 para identificar a ordem e o número total de etapas. Então, a implementação necessária nesse componente foi apenas incluir um novo objeto com a chave *name* no atributo *steps*.

O componente *ProceedingStepComponent* foi o que mais recebeu novas implementações no *frontend*. No código HTML do componente foi incluído o formulário que deve ser preenchido antes de gerar as planilhas da SOL. A primeira implementação nesse código é a inclusão da diretiva *ngIf* para verificar se a etapa atual é a da geração das planilhas. Essa diretiva é necessária porque o conteúdo de todas as etapas fica contido no mesmo arquivo HTML, onde a implementação de cada etapa possui no seu início a mesma diretiva e será exibido ao usuário apenas o conteúdo da etapa correspondente. Após essa diretiva, é incluído um campo de seleção implementado pelo componente *SelectInputComponent*. O conteúdo que é apresentado por esse componente está definido no atributo da classe do *ProceedingStepComponent* chamado *solSeriesValues*. Esse atributo é um objeto estático que contém as informações das séries que existem na SOL. A partir de uma série é que se obtêm as informações de uma seção da SOL que serão utilizadas para preencher a planilha de seções. Selecionada a série, uma nova diretiva *ngIf* é utilizada para verificar se foi selecionada uma série válida (através do atributo *solSeriesSelected*) e então exibir o restante do formulário.

O corpo principal do formulário de geração das planilhas é formado por uma lista de mini formulários. Essa lista é composta pelas trilhas selecionadas pelo usuário na etapa "Arquivos de trilha", que ficam armazenadas no atributo *tracks*. Como essa lista possui tamanho variável, é necessário implementar um laço para percorrer essa lista e exibir cada conteúdo dos índices. No Angular, um laço é implementado utilizando a diretiva *ngFor*. Essa diretiva funciona como um comando *for* convencional, onde o conteúdo interno da tag HTML que contém essa diretiva será repetido o número de vezes definido pelo condicional do laço.

Cada mini formulário dessa etapa irá conter campos do tipo *radio-button* e um campo de seleção. Os campos *radio-button* são formados pelos arquivos de trilha selecionados na etapa "Arquivos de trilha", que ficam armazenados no atributo *proceedings-TrackFiles*. Para obter apenas os arquivos de trilha da trilha correspondente, é utilizado o método *sectionsStepListOfOptions* que recebe como parâmetro a trilha atual e a lista de todos os arquivos de trilha selecionados. A lista de informações mostradas no campo de seleção é obtida através da série escolhida anteriormente, onde o conteúdo dela são as seções daquela série, que ficam armazenadas no atributo *sectionList*. Esse campo é igual em todos os mini formulários e a partir dele que serão preenchidas as colunas *sectionTitle*, *sectinoTitleEn* e *sectionAbbrev* da planilha de seções e a coluna *sectionAbbrev* da planilha de artigos.

Por fim, nesse formulário existe o botão de geração das planilhas da SOL, que só é habilitado quando todos os campos estiverem corretamente preenchidos e selecionados. O método que realiza essa verificação é o *canProceed*. Esse botão também possui uma diretiva chamada *click*, que realiza uma ação ao clicar no botão. O método incluído para ser chamado nessa diretiva é o *downloadSolSpreadsheets*, que realiza uma requisição ao *backend* para realizar o download das planilhas da SOL. O código completo do HTML implementado para esse formulário pode ser visualizado na figura 5.2.

5.1.3 Componente com a lista de submissões com *template* inválido

O único componente que foi preciso ser implementado do zero para essa funcionalidade foi o *ArticlesListCardDialogComponent*. Esse componente é a implementação de um *dialog*, que representa uma caixa de diálogo interativa. Ela será apresentada quando o sistema detectar que existe algum PDF que não está seguindo o *template* correto da SBC na geração das planilhas.

Figura 5.2: Implementação do formulário para geração das planilhas da SOL.

```

...
<ng-container>
  <div class="sections" *ngIf="this.step === 'Sections'">
    <div *ngIf="this.seriesForm">
      <app-select-input class="solSeries" [formGroupController]="this.seriesForm" name="solSeries"
        [options]="this.solSeriesValues"
        [placeholder]=" 'admin.event.publication.proceedings.sections.select-sol-series' | translate ">
      </app-select-input>
    </div>
    <div *ngIf="this.solSeriesSelected">
      <div class="main-body">
        <div class="columns sections-list" *ngFor="let track of tracks" [formGroup]="getSectionFormGroup(track)">
          <div class="column">
            <h3>{{ track.name }}</h3>
            <mat-radio-group formControlName="idTrackFile">
              <mat-radio-button *ngFor="let choice of sectionsStepList0fOptions(track, this.proceedingsTrackFiles)"
                value="{{ choice?.id }}" (click)="this.setTrackFile(choice.id, track)">
                {{ choice.value }}
              </mat-radio-button>
            </mat-radio-group>
          </div>
          <div class="column">
            <h3>&nbsp;</h3>
            <ul>
              <li>
                <app-select-input class="sectionList" [formGroupController]="getSectionFormGroup(track)" name="sections"
                  [options]="this.sectionList"
                  [placeholder]=" 'admin.event.publication.proceedings.sections.select-section' | translate ">
                </app-select-input>
              </li>
            </ul>
          </div>
        </div>
      </div>
    </div>
  </ng-container>
  ...
  <div class="buttons">
    ...
    <button *ngIf="step === steps[3]" mat-raised-button (click)="downloadSolSpreadsheets()" [disabled]="!canProceed()" color="primary">
      {{ 'forms.buttons.generate-sol-spreadsheets' | translate }}
    </button>
  </div>
</div>

```

Figura 5.3: Implementação do HTML do componente *ArticlesListCardDialogComponent*.

```

<div>
  <h1 class="information-title">{{ 'admin.event.publication.proceedings.articles-list-card.title' | translate }}</h1>
  <div>
    <table>
      <thead>
        <tr>
          <th class="cell-id">{{ 'admin.event.publication.proceedings.articles-list-card.column-id' | translate }}</th>
          <th>{{ 'admin.event.publication.proceedings.articles-list-card.column-title' | translate }}</th>
          <th>{{ 'admin.event.publication.proceedings.articles-list-card.column-track-file' | translate }}</th>
        </tr>
      </thead>
      <tbody>
        <tr class="table-line" *ngFor="let article of this.articlesList" (click)="openSubmissionById(article.submissionId)">
          <td class="cell-id">#{{ article.submissionId }}</td>
          <td>{{ article.title }}</td>
          <td>{{ article.trackFile }}</td>
        </tr>
      </tbody>
    </table>
  </div>
  <div>
    <button mat-raised-button [mat-dialog-close]="true" color="primary">{{ 'forms.buttons.ok' | translate }}</button>
  </div>
</div>

```

Figura 5.4: Implementação do método usado para abrir uma submissão a partir do identificador.

```

openSubmissionById(submissionId: string) {
  window.open(`/submissions/${submissionId}`, '_blank');
}

```

A implementação deste componente é bem simples, sendo basicamente uma tabela HTML, onde as linhas serão iteradas através da diretiva Angular *ngFor*. O atributo da classe que contém a lista de artigos que não estão seguindo o *template* é o *articlesList*. Outra implementação incluída nessa lista é a possibilidade de acessar diretamente a submissão que contém o artigo fora do padrão. Para isso, foi incluída uma diretiva *click* em cada linha da lista, onde é realizada a chamada do método *openSubmissionById* ao clicá-la. Esse método recebe como parâmetro o campo identificador da submissão e cria um link de acesso a ela a partir desse identificador. A implementação desse método e do código HTML do componente podem ser vistos nas figuras 5.3 e 5.4.

5.2 Implementações no *backend* para geração das planilhas

O *backend* do JEMS3 é um projeto que foi implementado separado do *frontend*. Para essa implementação foi utilizado o Django (FOUNDATION, 2024) que é um *framework web* de alto nível para Python, projetado para facilitar o desenvolvimento de aplicativos web rápidos, seguros e escaláveis. O Django segue o padrão MTV, onde:

- *Model*: Representa os dados e a lógica de acesso ao banco;
- *Template*: Define como a resposta é apresentada ao usuário (HTML);
- *View*: É o intermediário que conecta os modelos e *templates*, processando as requisições e controlando o fluxo de dados.

Para implementação dos novos recursos no backend do JEMS3, foram criados três novas *views* para o *model SubmissionFile*. Todas as *views* são implementadas como classes em Python e estão associadas a uma URL específica, a qual será utilizada pelo *frontend* do JEMS3 para requisitar o recurso correspondente. As três novas *views* implementadas são:

- *UpdateSubmissionTrackFiles*: responsável por verificar se os arquivos de trilha de uma submissão estão utilizando o *template* correto da SBC e atualizá-lo de acordo;
- *SolSections*: responsável por retornar a lista de seções a partir de uma determinada série da SOL;
- *SolSpreadsheets*: responsável por gerar as quatro planilhas utilizadas para exportar os artigos de um evento para a SOL.

O nome do método de uma classe *view* é definido com base na funcionalidade que ele desempenha. Quando se utiliza classes baseadas em *views*, o nome do método geralmente está associado ao verbo HTTP correspondente, como *GET*, *POST*, etc. Por esse motivo será possível notar que os métodos principais das *views* implementadas possuíram os seus nomes baseados neste conceito.

5.2.1 *UpdateSubmissionTrackFiles*

Para verificar se um artigo está utilizando o *template* correto da SBC, foi implementada a *view UpdateSubmissionTrackFiles*. Ela é uma classe que contém um método principal chamado *post* e um método auxiliar definido como *__is_using_sbc_template*. Os dois *underscores* no início do nome do método são utilizados em Python como convenção para indicar que o método é privado, já que não existe o conceito de método privado em Python.

A implementação do método *post* é bem simples. Ela recebe como parâmetro a requisição do *frontend*, onde no corpo dela é informada uma lista com *id's* do *model SubmissionFile*. Ele representa a tabela do banco de dados que contém os arquivos

de uma submissão. Essa lista é iterada e com cada *id* é buscado no banco de dados o registro de um *SubmissionFile*. O arquivo PDF desse registro é aberto utilizando a biblioteca *PyPDF2* (FENNIK, 2024), que irá ler os metadados do PDF. Esses metadados são passados para o método `__is_using_sbc_template` que irá tentar localizar o metadado `templateVersion` e verificar se o seu valor é igual ao da versão atual utilizada (no momento atual a versão 1.0). O retorno desse método auxiliar é um *booleano* que é salvo no campo `is_using_sbc_template` do *model SubmissionFile*. O retorno do método `post` será uma resposta HTTP com o código de *status* 200 e com o corpo sendo um objeto que tem como chaves os *id's* do *SubmissionFile* e o valor de cada registro sendo conteúdo do campo `is_using_sbc_template` daquele registro.

5.2.2 SolSections

Para obter a lista de seções da SOL a partir de uma série, foi implementada a *view SolSections*. Essa classe contém o método principal `get` e mais dois métodos auxiliares privados, sendo eles:

- `__get_text_from_tag`: recebe como parâmetro uma *tag* XML e retorna o conteúdo contido nela;
- `__build_select_list`: cria uma lista de objetos, onde cada objeto contém a abreviação e o título.

A implementação do método `get` consiste em uma requisição à URL da SOL que retorna um XML contendo todas as informações das seções da série passada na URL. Esse XML retorna a abreviação da seção (*tag* XML `setSpec`) e o título (*tag* XML `setName`). Para realizar a manipulação das informações do XML foi usada a biblioteca *BeautifulSoup4* (RICHARDSON, 2024). O retorno do método `get` será uma resposta HTTP com o código de *status* 200 e com o corpo sendo a lista retornada pelo método `__build_select_list`.

5.2.3 SolSpreadsheets

A geração das planilhas da SOL é, das três implementadas, a *view* mais complexa. Essa *view* é responsável não só por criar as quatro planilhas para exportação dos artigos

para a SOL, como também por montar uma pasta que contenha essas planilhas e mais uma pasta contendo todos os artigos incluídos nelas. O método principal dessa *view* é o método *get*. Ele realiza todo o processo de criação da pasta, planilhas e verificação do template. Para executar todas estas ações, a *view* conta com diversos outros métodos auxiliares, sendo alguns mais importantes do que outros. Além do método *get*, apenas os métodos auxiliares mais importantes serão descritos.

Os métodos `__get_text_references` e `__get_parsed_references` são os que foram criados para substituir a biblioteca externa de extração de referências do texto. Ou seja, apenas com dois métodos não muito complexos foi possível realizar a extração das referências do texto de forma muito mais eficiente. O primeiro realiza a primeira filtragem no PDF, localizando as palavras-chave ocultas "`##init_bibliography##`" e "`##end_bibliography##`". O retorno do método é todo o texto em formato de *string* localizado entre as duas palavras-chave. O segundo método irá receber como parâmetro o texto retornado pelo primeiro, localizando as palavras-chave ocultas que delimitam o início e o fim de cada item do capítulo de referências. Ao localizar cada um item, ele é incluído numa lista de referências. Cada item nesta lista é um objeto que possui as chaves *num_seq*, *description* e *doi*, onde cada uma dessas chaves representa as colunas *article*, *description* e *doi* da planilha de referências respectivamente.

Outro método auxiliar importante que foi implementado é o `__open_workbook`. Esse método foi desenvolvido para criar um arquivo XLSX (formato de arquivo do Excel para planilhas). Ele recebe como parâmetro uma lista de títulos, que serão utilizados como os títulos das colunas, e o nome que será dado ao arquivo. Com isso, esse método consegue ser utilizado para criar quantas planilhas forem necessárias. A biblioteca utilizada para a criação de planilhas foi a *XlsxWriter*, que é uma biblioteca Python para escrever arquivos no formato XLSX do Excel 2007.

Outros métodos importantes dessa *view* são os quatro métodos criados para incluir os dados nas planilhas. Cada um desses métodos inclui as informações em planilhas distintas. Todos estes métodos recebem dois parâmetros iguais, o *workbook* e o *line_num*. O primeiro representa um objeto da biblioteca *XlsxWriter* e o segundo a linha da planilha que deve ser preenchida. O método `__put_references_in_spreadsheet` é responsável por incluir as informações na planilha de referências. Além dos dois parâmetros em comum, ele recebe também uma lista de referências no formato retornado pelo `__get_parsed_references` e retorna o novo número de linha da planilha que virá a próxima referência. O método implementado para incluir os autores na sua planilha é o

`__put_authors_in_spreadsheet`. Ele recebe como parâmetros adicionais o objeto *metadata* da biblioteca *PyPDF2* e um parâmetro chamado *num_seq*. O primeiro deles contém as informações dos metadados do artigo e o segundo representa o valor que irá na coluna *article* da planilha. O retorno deste método também é o número da próxima linha que será preenchida na planilha. O método que preenche a planilha de artigos é o `__put_article_in_spreadsheet`. Ele recebe como parâmetros adicionais o mesmo objeto *metadata* e o inteiro *num_seq*, além do número de páginas que o artigo possui, o *id* do *SubmissionFile* daquele artigo e a abreviação da seção na qual aquele artigo está incluso. O número de páginas irá preencher a coluna *pages*, o *id* a coluna *idJEMS* e a abreviação da seção a coluna *sectionAbbrev*. O retorno deste método também será o número da próxima linha a ser preenchida. O último método é o `__put_sections_in_spreadsheet` que implementa o preenchimento da planilha de seções. Ele contém diversos parâmetros adicionais, sendo eles:

- *track*: esse parâmetro é o objeto de uma trilha. A partir dele serão preenchidas as informações das colunas *blind*, *numSubmitted* e *numAccepted*;
- *section_title*: o título da seção da SOL que preencherá a coluna *sectionTitle*;
- *section_title_en*: o título da seção da SOL em inglês que preencherá a coluna *sectionTitleEn*;
- *section_abbrev*: a abreviação do título da seção da SOL que preencherá a coluna *sectionAbbrev*;
- *date_sub*: a data de submissão dos artigos que preencherá a coluna *dateSub*;
- *date_result*: a data de aceitação dos artigos que preencherá a coluna *dateResult*;
- *date_ready*: a data de envio da versão final dos artigos que preencherá a coluna *dateReady*.

Esse método será o único que não irá retornar o número da próxima linha que deve ser preenchida na planilha. Isso porque as informações desta planilha não dependem do número de artigos que ingressarão na SOL.

O método *get* irá chamar todos estes métodos auxiliares dentro de uma lógica para gerar o link de download do arquivo ZIP, que contém as planilhas e PDFs, ou retornar uma lista contendo os artigos que não estão seguindo o *template* da SBC. Esse método irá receber como parâmetro uma lista contendo os *ids* dos conteúdos incluídos nos anais do evento, o *id* do evento e uma lista que contém as relações de um arquivo de trilha com uma seção da SOL. Com o *id* do evento será gerado o nome do arquivo ZIP, seguindo o

padrão "{evento_id}_sol_spreadsheets.zip". A partir da lista de *ids* será obtida a lista de submissões que estão contidas no conteúdo incluído nos anais do evento. Essa lista de submissões será iterada e, para cada submissão, a lista de arquivos contida na submissão será também iterada. A cada iteração na lista de arquivos da submissão será verificado se o arquivo está seguindo o *template* da SBC.

Caso esteja, os dados contidos nele são incluídos nas planilhas e os PDFs são incluídos numa pasta com o nome "files". A iteração continua até terminar todos os arquivos e submissões e, ao término delas, estando todos os arquivos seguindo o *template*, a planilha de seções é preenchida a partir da lista de relações entre as seções da SOL e os arquivos de trilha selecionados no formulário. Depois deste processo, todas as planilhas estarão preenchidas e serão incluídas na pasta que será comprimida em um arquivo ZIP juntamente com a pasta de PDFs. A resposta do método *get* nesse caso será uma resposta HTTP com o código de *status* 200 e com o corpo sendo um objeto que contém a informação do link para download do arquivo ZIP gerado.

Caso não esteja, os dados desse arquivo são incluídos numa lista de arquivos que não estão seguindo o *template* da SBC. A partir do momento em que a lista de arquivos que não seguem o *template* não for vazia, mais nenhum dado é incluído nas planilhas e apenas a verificação do *template* é feita para o restante dos arquivos de todas as submissões. A resposta do método *get* nesse caso será uma resposta HTTP com o código de *status* 200 e com o corpo sendo um objeto que contém a lista de submissões que contém arquivos que não seguem o *template* da SBC.

6 TESTES EM HOMOLOGAÇÃO DA GERAÇÃO DAS PLANILHAS PARA A 21ª ERRC

Idealmente, para conseguir testar o uso do *template* da SBC desenvolvido e também as implementações no JEMS3, seria preciso utilizá-los num evento real. Infelizmente não foi possível realizar esses testes em ambiente de produção, mas um evento real, existente no JEMS3 de produção, foi copiado no ambiente de homologação. Esse evento, realizado pela SBC, foi a XXI Escola Regional de Redes de Computadores (ERRC 2024) (SBC, 2024b). Esse evento foi organizado pelo Centro de Ciências Computacionais (C3) e ocorreu na Universidade Federal do Rio Grande do dia 27 ao dia 29 de novembro de 2024. Dentro do evento ocorreu o 8º Workshop Regional de Segurança da Informação e de Sistemas Computacionais (WRSeg 2024).

O gerenciamento das submissões da ERRC 2024 foi feito pelo JEMS3, assim como o gerenciamento das submissões do WRSeg 2024. Como o workshop ocorreu dentro do evento, no JEMS3, o WRSeg 2024 é considerado um subevento do evento-pai ERRC 2024. Eles serão gerenciados separadamente no JEMS3, tendo as suas próprias submissões e programação. Antes de iniciar os passos para realizar os testes, foi preciso copiar os dados (submissões, artigos, trilhas, etc) para o ambiente de homologação. Então, todos os testes que serão descritos a seguir foram feitos em ambiente de homologação.

6.1 Inclusão da programação do evento no JEMS3

O primeiro passo antes de realizar a geração das planilhas no JEMS3 é incluir a programação do evento no sistema, na página de administrador do evento. A informação da programação completa é possível verificar no site oficial do evento (SBC, 2024a). Para inserir essas informações no JEMS3, é preciso acessar a página de gerenciamento do evento e clicar no botão "Programa". Ao clicar nesse botão, será aberta a visão de calendário (figura 6.1), onde será possível verificar as sessões técnicas inclusas na programação até o momento.

Para incluir uma nova sessão técnica na programação, basta clicar duas vezes no espaço do calendário que representa o dia e horário na qual ela irá iniciar. Ao fazer isso, será incluído um novo item no calendário com o nome padrão "New session" (figura 6.2). Ao clicar duas vezes sobre a nova sessão, irá abrir um menu que permitirá editar as

Figura 6.1: Tela de visão de calendário da programação de um evento no JEMS3.

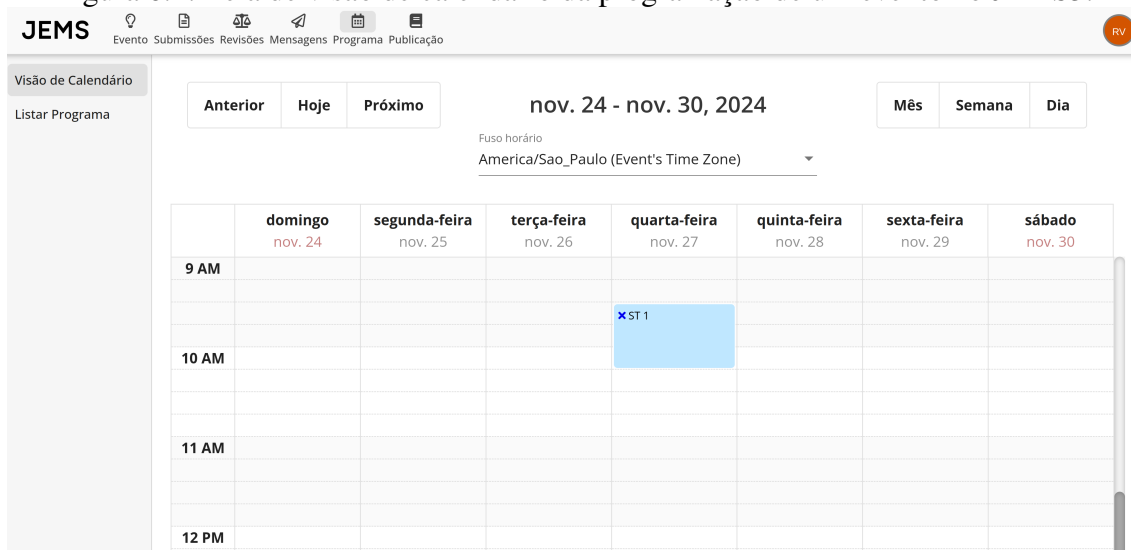


Figura 6.2: Novo item com nome padrão no calendário da programação de um evento no JEMS3.

	domingo nov. 24	segunda-feira nov. 25	terça-feira nov. 26	quarta-feira nov. 27	quinta-feira nov. 28	sexta-feira nov. 29	sábado nov. 30
9 AM							
10 AM				x ST 1			
11 AM							
12 PM							
3 PM							
4 PM				x New session			

informações daquela sessão criada. Este menu terá duas abas, sendo elas: "Configurações gerais" e "Submissões". A primeira é de um formulário (figura 6.3) que contém campos sobre a sessão, que devem ser preenchidos ou editados. A segunda irá conter uma lista com todas as submissões daquele evento. As submissões que irão fazer parte daquela sessão devem ser selecionadas, conforme mostrado na figura 6.4. Ao clicar em confirmar, os dados inseridos no formulário e as submissões selecionadas na aba "Submissões" serão incluídos naquela sessão.

A ERRC 2024 teve 9 sessões técnicas durante os três dias de evento, sendo 5 delas do evento-pai (STs 1, 2, 3, 5 e 9) e 4 da WRSeg (STs 4, 6, 7 e 8). Com a inclusão das sessões no JEMS3 foi possível ver uma limitação da solução desenvolvida nesse trabalho: não seria possível gerar as quatro planilhas para o evento-pai e sub-evento em apenas uma execução, nesse caso sendo preciso gerar 8 planilhas. Isso ocorreu porque o JEMS3 não permite incluir nas sessões de um evento-pai os artigos submetidos para um de seus sub-eventos. Então, cada um dos eventos teria suas sessões separadas.

Figura 6.3: Aba "Configurações gerais" do menu de uma sessão no JEMS3.

The screenshot shows the 'Configurações gerais' tab of a session menu. The interface includes the following elements:

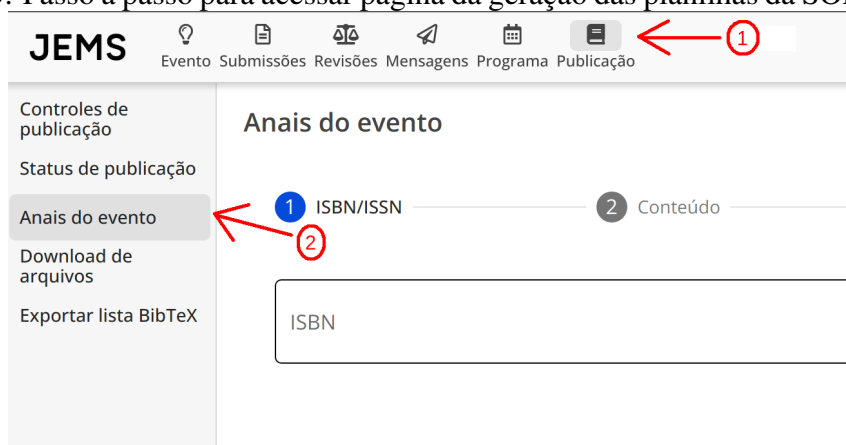
- Configurações gerais** (selected) and **Submissões** tabs.
- Título**: ST 3
- Subtítulo**: (empty)
- URL**: (empty)
- Sala**: (empty)
- início da sessão**: 27 de nov. 2024 15:30 (UTC-03:00)
- Fim da sessão**: 27 de nov. 2024 17:00 (UTC-03:00)
- Coordenador de Sessão**: (empty)
- Pesquisar usuário por nome, e-mail ou ID**: (input field with search icon)
- Cor da sessão**: #ABCF3
- Buttons**: Confirmar (blue), Cancelar (red)

Figura 6.4: Aba "Submissões" do menu de uma sessão no JEMS3.

The screenshot shows the 'Submissões' tab of a session menu. The interface includes the following elements:

- Configurações gerais** and **Submissões** tabs.
- Pesquisa por submissões**: (input field)
- Alocados**: (checkbox)
- Submissions List**:
 - 1520 - Providing smartphone authentication for an existing access control system at the Industrial Technical College of Santa Maria (CTISM)
 - 1523 - A cloud-based platform for MPI execution support in education
 - 1525 - VPN de camada 2 de baixo custo para unidades remotas
- Buttons**: Confirmar (blue), Cancelar (red)

Figura 6.5: Passo a passo para acessar página da geração das planilhas da SOL no JEMS3.



6.2 Geração das planilhas para a ERRC 2024

Com o evento tendo a sua programação incluída corretamente no JEMS3, as planilhas já estão prontas para serem geradas. Para realizar o este processo, estando na página de gerenciamento do evento, basta acessar a aba "Publicação" e depois na opção "Anais do evento" (figura 6.5).

O primeiro passo "ISBN/ISSN" desta página não precisa ser preenchido, então basta seguir para o passo seguinte nomeado como "Conteúdo". Esse passo é muito importante para a geração dos anais, pois é nele que as sessões técnicas criadas na programação do evento serão selecionadas para incluí-las nas planilhas. Ao acessar esse passo, poderá ser visualizada uma lista contendo as sessões técnicas e os artigos aceitos no evento (figura 6.6). Os itens da lista em azul são sessões técnicas criadas na programação do evento e os itens em verde são os artigos que foram aceitos no evento. Para incluir um item na geração das planilhas basta arrastá-lo para a área nomeada como "Atividades dos anais do evento ordenadas". Ao fazer isso, o conteúdo daquele item poderá ser incluso nas planilhas. A ordem dos itens pode ser alterada conforme a preferência do usuário, bastando arrastar o item incluso para a ordem que desejar. Para este teste, todas as sessões técnicas serão incluídas em ordem crescente (figura 6.7). Feito isso, o próximo passo pode ser acessado.

No passo "Arquivo de trilha" (figura 6.8) serão selecionados os arquivos de trilha que devem aparecer no formulário do passo seguinte. Os arquivos de trilha que aparecerem neste passo são todos os que foram identificados no conteúdo incluído no passo "Conteúdo". Para este teste apenas os arquivos de trilha "Trilha Principal (camera ready)" e "exp_demo (camera ready)" foram identificados e ambos devem permanecer seleciona-

Figura 6.6: Passo "Conteúdo" na página de geração de anais de um evento no JEMS3.

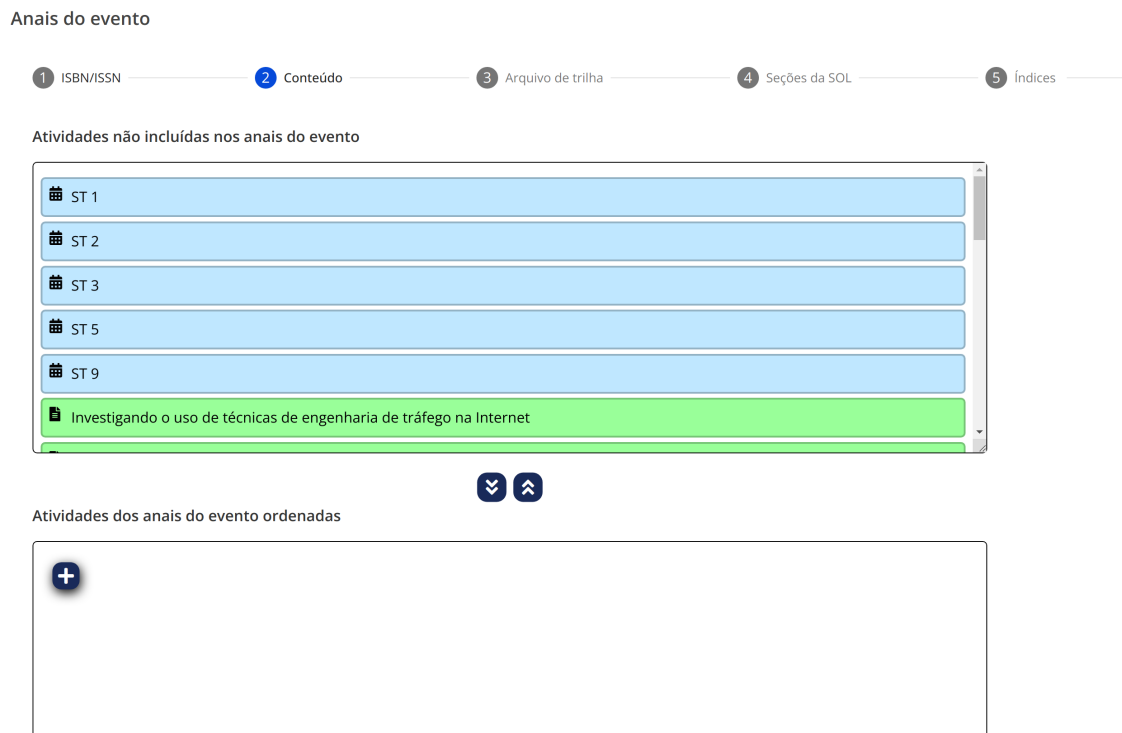


Figura 6.7: Passo "Conteúdo" com ST's inclusas no conteúdo dos anais do evento.

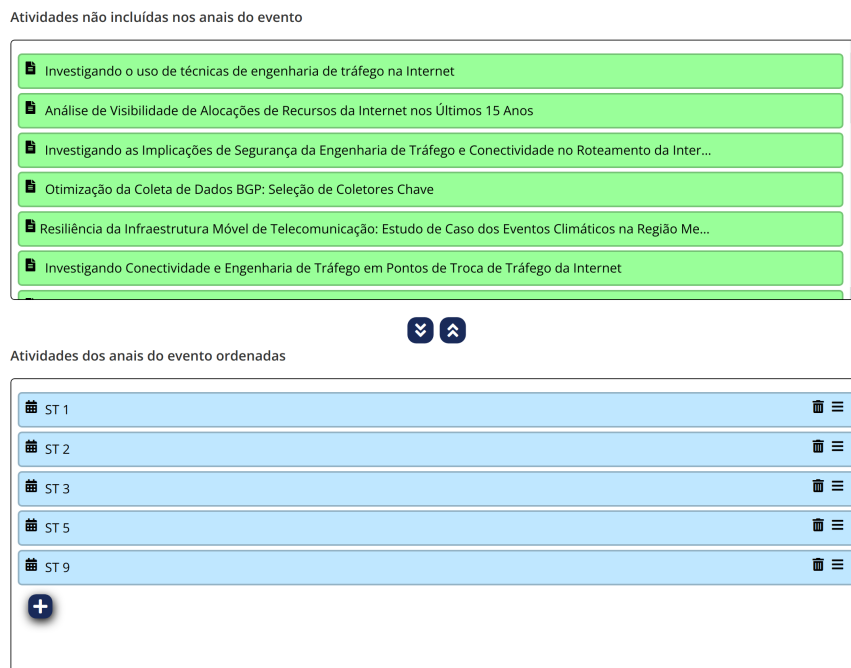


Figura 6.8: Passo "Arquivo de trilha" na página de geração de anais de um evento no JEMS3.

Anais do evento

1 ISBN/ISSN — 2 Conteúdo — 3 Arquivo de trilha — 4 Seções da SOL

Trilha principal

Trilha Principal (camera ready) ☰

Trilha de experiências e demonstrações

exp_demo (camera ready) ☰

[Voltar](#) [Próximo](#)

Figura 6.9: Passo "Seções da SOL" com os dados preenchidos.

Anais do evento

1 ISBN/ISSN — 2 Conteúdo — 3 Arquivo de trilha — 4 Seções da SOL — 5 Índice

Selecione a série da SOL

Anais da Escola Regional de Redes de Computadores (ERRC) ▼

Trilha principal

Trilha Principal (camera ready) Selecione a seção associada

Artigos Completos da ERRC ▼

Trilha de experiências e demonstrações

exp_demo (camera ready) Selecione a seção associada

Artigos Completos do WRSeg ▼

[Voltar](#) [Próximo](#) [Gerar planilhas da SOL](#)

dos. Feito isso, o próximo passo pode ser acessado.

O passo "Seções da SOL" é o último passo para a geração das planilhas da SOL. Nele, primeiramente, deve ser selecionada a série da SOL correspondente a este evento. A série neste caso é a "Anais da Escola Regional de Redes de Computadores (ERRC)". Selecionada esta série, será aberto o restante do formulário para ser preenchido. Nele, devem ser selecionados os mesmos arquivos de trilha do passo anterior e a seção da SOL que será associada àquele arquivo de trilha. Com todas as informações preenchidas corretamente (figura 6.9), basta clicar no botão "Gerar planilhas da SOL". Para a geração das planilhas para o subevento da WRSeg 2024 o mesmo passo a passo foi seguido, inserindo como conteúdo as ST's incluídas na programação.

6.3 Avaliação dos testes e dos resultados obtidos

Os testes da geração das planilhas da SOL no ambiente de homologação para o evento da ERRC 2024 e o seu subevento WRSeg 2024 tiveram um bom resultado. Para o evento da ERRC foram incluídos 20 artigos e na WRSeg 16. As planilhas foram geradas com sucesso, mas o preenchimento dos dados teve um pequeno problema, que foi no preenchimento da coluna "orcid" da planilha "Autores" e no preenchimento de toda a planilha "Referencias". Além disso, durante a realização dos testes foi possível verificar algumas limitações na solução que pode contar com melhorias no futuro. A descrição dos resultados será descrita com mais detalhes nesta seção.

6.3.1 Análise da geração das planilhas e arquivos para a SOL

A análise sobre a geração das planilhas e seu preenchimento será a mesma para o evento da ERRC e da WRSeg porque ambos eventos obtiveram os mesmos resultados. A geração das planilhas ocorreu sem a emissão de erros, ou seja, todas as planilhas foram criadas e os artigos que devem ser incluídos no download numa pasta também ocorreram corretamente.

Ao clicar no botão "Gerar planilhas da SOL", é esperado que seja feito o download de um arquivo ZIP nomeado com o acrônimo do evento concatenado com "SOL_Spreadsheets". Para os dois eventos esse arquivo foi gerado com o nome correto: "ERRCSOL_Spreadsheets" e "WRSegSOL_Spreadsheets". Dentro do arquivo ZIP, é esperado que tenha uma pasta com o nome "*files*" e quatro arquivos XLSX com os nomes: "Artigos", "Autores", "Referencias" e "Secoes". Estes cinco itens estavam presentes com os nomes corretos. Dentro da pasta "*files*" devem estar presentes todos os artigos em PDF que foram utilizados no preenchimento das planilhas, nomeados com o ID do artigo no JEMS3. Todos os artigos foram incluídos e com os nomes corretos.

O preenchimento da planilha "Secoes" foi feita corretamente. Para o evento da ERRC, as duas seções selecionadas no formulário de geração das planilhas foram incluídas e com os seus dados corretos. Essa planilha para o subevento da WRSeg também teve seu preenchimento correto.

A planilha "Artigos" também foi preenchida corretamente para ambos os eventos. A da ERRC foi gerada com os 20 artigos que deveriam estar inclusos e a da WRSeg com os 16 artigos. A coluna "seq" foi preenchida corretamente, seguindo a ordem correta nas

demais planilhas.

A planilha "Autores" teve o preenchimento quase todo correto, tendo apenas um pequeno erro para ambos os eventos. Todos os 68 autores identificados nos artigos foram incluídos na planilha da ERRC e os 69 na planilha da WRSeg. Os seus dados foram todos incluídos de forma correta, mas a coluna "orcid" das planilhas foram preenchidas mesmo quando o autor não passou essa informação no artigo. Com isso, nesses casos, o valor preenchido na coluna foi apenas a URL da página do ORCID (<https://orcid.org/>), sem o ID do autor concatenado a ela.

A última planilha, a "Referencias", foi a única que não foi preenchida. Infelizmente o *template* usado na submissão dos artigos da ERRC e da WRSeg não foi a versão final que tinha a inclusão das palavras-chave ocultas em cada item das referências. Com isso, o extrator de referências implementado no JEMS3 não conseguiu pegar nenhuma referência incluída em cada artigo. Por causa deste fato não foi possível avaliar se o preenchimento seria feito corretamente caso a versão correta do *template* estivesse sendo utilizada. Entretanto, testes pontuais em ambiente de homologação foram realizados com menos artigos e com o uso da versão correta do *template* e o preenchimento se mostrou ser feito de forma correta.

Os testes, apesar de terem sido realizados em ambiente de homologação, se mostraram bastante satisfatórios. O preenchimento automático das planilhas, que era principal problema que buscava ser solucionado, foi feito quase em sua totalidade de forma correta. O único problema identificado foi o preenchimento da coluna "orcid" na planilha "Autores", mas que é um problema com solução de baixa complexidade. O não preenchimento da planilha "Referencias" não se mostrou ser um problema da solução, apenas no uso da versão correta do *template*, que em futuros eventos deve ser corrigido.

6.3.2 Limitações encontradas e possíveis melhorias futuras

Durante a realização dos testes da geração da planilhas para a ERRC 2024 em homologação foram percebidas algumas limitações na solução que podem ser solucionadas futuramente. A primeira delas é o verificador do *template* usado no artigo. Atualmente, o que identifica que o *template* está sendo usado é um metadado chamado *templateVersion*. É um identificador bem simples, que, se não tiver seu valor atualizado constantemente no JEMS3, pode não validar corretamente. Esse verificador pode ser aperfeiçoado para verificar se todas as informações importantes do *template* estão presentes. Por exemplo,

verificar a presença de cada metadado obrigatório, verificar a presença das palavras-chave ocultas, verificar a presença de metadados obrigatórios vazios, entre outros.

Outra limitação importante notada é no passo "Conteúdo" da página de geração de anais do JEMS3 que é utilizada para gerar as planilhas da SOL. Antes de discorrer sobre essa limitação, é importante citar o contexto sobre o padrão do conteúdo das planilhas da SOL para a ERRC. O conteúdo dos artigos e seções devem estar contidos em apenas quatro planilhas para o evento da ERRC e para os seus subeventos, ou seja, devem ser geradas apenas quatro planilhas para o evento principal (ERRC) e o subevento (WRSeg) e não quatro planilhas para cada um. A solução não prevê esse padrão, sendo necessário gerar as planilhas de forma separada e depois junta-las manualmente. Essa limitação se mostrou bastante significativa, pois o objetivo principal da solução é não ser necessário ser feito nenhum trabalho manual no preenchimento das planilhas. Para solucionar essa limitação, seria necessário apenas incluir o conteúdo da programação dos subeventos no passo "Conteúdo" da página de geração de anais do JEMS3. Consequentemente, os arquivos de trilha também seriam incluídos no passo seguinte e no passo "Seções da SOL".

7 CONCLUSÃO

Neste capítulo conclusivo, será apresentada uma breve recapitulação do problema abordado ao longo deste trabalho. Ademais, será revisitada a solução proposta para esse problema, considerando tanto sua concepção abstrata quanto os aspectos inerentes à sua implementação. Também serão discutidos os testes realizados em ambiente de homologação com dados de um evento real, as conclusões decorrentes deles, os desafios enfrentados durante o desenvolvimento e as conclusões extraídas.

No início deste trabalho, foi detalhado o problema que a solução desenvolvida buscou solucionar. O problema, enfrentado pela SBC, estava no processo de publicação dos artigos dos eventos científicos na biblioteca digital SOL, gerenciados pelo sistema JEMS3. Este processo contém vários passos intermediários, que são: baixar os artigos aceitos em PDF, copiar os dados dos artigos para quatro planilhas (artigos, autores, referências e seções), submeter as planilhas e artigos para um conversor e então enviar o arquivo XML criado pelo conversor para a SOL. Essa transcrição dos dados dos artigos aceitos para as planilhas era feita manualmente, isto é, um processo demorado e propenso a erros que exigia um esforço significativo, especialmente em eventos com muitos artigos. Os possíveis erros que podiam ser introduzidos nesse processo manual comprometiam a publicação dos artigos e demandavam retrabalho. Dado esse contexto, o trabalho propõe a automatização do preenchimento das planilhas, utilizando tecnologias que extraiam metadados diretamente dos artigos e sistemas relacionados, mitigando os problemas identificados.

A solução apresentada envolveu, primeiramente, a padronização do *template* utilizado na submissão dos artigos, possibilitando a extração automática das informações que serão utilizadas nas planilhas. Esse *template* foi modificado para incluir metadados essenciais no formato PDF, garantindo consistência na localização e no formato dos dados. Alterações no JEMS3 também foram necessárias, sendo necessário verificar o uso do *template* nas submissões de um evento, extrair as referências do texto, extrair os metadados do PDF, buscar seções da SOL para relacioná-las com os arquivos de trilha do JEMS3 e passar todas essas informações na ordem correta para as planilhas.

O *template* modificado foi desenvolvido em \LaTeX . Para incluir os metadados de forma que os seus dados fossem consistentes com o que estava no texto, foi preciso extrair as informações dos comandos utilizados na inclusão de conteúdo do artigo. Por exemplo, o comando *resumo* foi utilizado para incluir o resumo do artigo no texto. Para extrair

as informações dele conforme apareceria no texto, foi criada a variável *cleanresumo* que armazena o conteúdo do comando *resumo*. Desta forma, bastou incluir o conteúdo da variável no metadado através do comando *hyersetup* do pacote *hyperref*. Além dos metadados, foi necessário incluir palavras-chave ocultas no capítulo de referências do *template*, buscando melhorar a performance na busca das informações das referências que são utilizadas em uma das planilhas. Para isso, foi criado um comando próprio de impressão de referências para incluir as palavras-chave no início e fim do capítulo e foram redefinidos os comandos *begentry* e *finentry* para incluir as palavras-chave no início e fim de cada referência.

As implementações realizadas no JEMS3 foram feitas no *frontend* e *backend*. No *frontend*, alguns componentes foram desenvolvidos e modificados utilizando o *framework* Angular. Esses componentes incluem formulários para entrada de dados e exibição de informações sobre submissões. No *backend*, diversas funcionalidades foram implementadas em Python, como métodos para extração de metadados, criação de planilhas e verificação do *template*. A biblioteca *BeautifulSoup* foi usada para manipulação de dados XML, enquanto a biblioteca *XlsxWriter* foi utilizada para geração de arquivos XLSX. O sistema também passou a realizar verificações automáticas para identificar artigos que não seguem o *template* correto, evitando erros na geração das planilhas. Essas implementações garantiram um fluxo automatizado, desde a validação dos *templates* até a geração final dos arquivos necessários para a publicação na SOL, reduzindo o esforço manual e aumentando a eficiência do processo.

Para testar a solução implementada, foram utilizados dados de um evento real no ambiente de homologação do JEMS3. O evento utilizado foi a 21^a ERRC e o seu subevento WRSeg 2024. Apesar dos testes terem sido executados em um ambiente de homologação, eles se mostraram bastante positivos para a obtenção de resultados e conclusões. A partir destes testes foi possível não só verificar que a solução funcionou muito bem, mas também que ela contém algumas limitações que podem ser solucionadas futuramente para poder aprimorá-la. Uma destas limitações é a impossibilidade de gerar as quatro planilhas necessárias para submeter na SOL para o evento e os seus subeventos de uma só vez. Com a solução implementada, é preciso gerar separadamente as quatro planilhas para o evento e para cada subevento e no fim juntá-las manualmente. Entretanto, para um evento sem subeventos a solução mostrou-se adequada para o problema apresentado, gerando as quatro planilhas preenchidas com praticamente todos os dados estando corretos.

O processo de desenvolvimento deste trabalho foi bastante desafiador, tanto na parte prática de desenvolvimento da solução quanto na escrita deste texto. Iniciei este trabalho tendo praticamente nenhuma experiência em \LaTeX , o que foi um grande desafio inicialmente no aperfeiçoamento do *template*. Apesar disso, após muito tempo buscando entender o desenvolvimento do *template* e como funciona o \LaTeX , essa linguagem de programação acabou se tornando mais natural. A implementação no JEMS3 não foi tão complexa para mim quanto a do *template*. Mesmo o JEMS3 sendo um sistema muito grande e complexo, a minha familiaridade prévia com o desenvolvimento web e com a linguagem de programação Python foram essenciais para entender rapidamente a implementação do sistema. Por fim, a escrita deste texto se mostrou bastante desafiadora no início. Iniciei a escrita deste trabalho sem ter experiência na escrita de textos científicos, inclusive sem ter a prática de escrever textos longos, mas no decorrer da escrita acabou ficando mais fácil escrever cada capítulo.

Apesar dos desafios, com este trabalho foi possível obter avanços na integração entre o JEMS3 e a SOL. Anteriormente não existia nenhum tipo de integração, apenas era possível realizar o download dos artigos no JEMS3 e realizar o restante do processo de submissão deles de forma manual. Com o desenvolvimento deste trabalho, grande parte desse processo pôde ser automatizado para os eventos da SBC. Ainda existem melhorias e correções a serem feitas, mas um grande passo já foi dado.

7.1 Trabalhos futuros

Como dito anteriormente, existem melhorias que devem ser feitas a partir da solução apresentada neste trabalho. Além destas melhorias, outras novas funcionalidades podem surgir para tornar a integração do JEMS3 com a SOL ainda melhor. Atualmente, o *template* modificado neste trabalho possui um tempo de compilação muito grande, o que se mostrou um problema para os autores que utilizam o Overleaf para desenvolver seus artigos. Isso ocorre porque o Overleaf possui um tempo de compilação limitado para usuário que utilizam o modo gratuito, obrigando os usuários a utilizarem a versão paga para modificar esse *template* ou precisando utilizar outra plataforma para isso.

O processo de geração das planilhas foi automatizado neste trabalho, mas após ter as planilhas o usuário ainda precisa passá-las para um conversor que irá gerar um XML para fazer a submissão na SOL. Uma nova solução no JEMS3 que não precise mais gerar as planilhas e sim gerar o arquivo XML diretamente seria mais um avanço

na integração dos dois sistemas. Entretanto, ainda é possível realizar a extinção deste processo intermediário com a criação de uma API que integre o JEMS3 e a SOL, podendo fazer a submissão dos artigos dos eventos da SBC diretamente do JEMS3.

REFERÊNCIAS

- BIBTEX. **Site oficial do BibTeX**. 2024. <<https://www.bibtex.org/>>. Acessado em 4 dez. 2024.
- DUCKETT, J. **HTML and CSS: Design and Build Websites**. [S.l.]: Wiley, 2011.
- EDAS. **Site oficial do EDAS**. 2024. <<https://edas.info/doc>>. Acessado em 4 dez. 2024.
- FENNIK, M. **Site repositório da biblioteca PyPDF2 no Github**. 2024. <<https://github.com/py-pdf/pypdf>>. Acessado em 4 dez. 2024.
- FLANAGAN, D.; TORTELLO, J. E. N.; NEDEL, L. **JavaScript: O Guia Definitivo**. [S.l.]: Bookman, 2012.
- FOUNDATION, D. S. **Site oficial do framework web Django**. 2024. <<https://www.djangoproject.com/>>. Acessado em 4 dez. 2024.
- GOOGLE. **Site oficial do Angular**. 2024. <<https://angular.dev/>>. Acessado em 4 dez. 2024.
- HEP, I. **Site do repositório no GitHub**. 2025. <<https://github.com/inspirehep/refextract>>. Acessado em 11 jan. 2025.
- KIME, P.; WEMHEUER, M.; LEHMAN, P. **The biblatex Package**. 2023. <<https://ctan.math.illinois.edu/macros/latex/contrib/biblatex-ms/doc/biblatex-ms.pdf>>. Acessado em 4 dez. 2024.
- KNUTH, D. E. **The TeXbook**. [S.l.]: Addison Wesley, 1984.
- LATEX. **Site oficial do LaTeX Project**. 2024. <<https://www.latex-project.org/>>. Acessado em 4 dez. 2024.
- LIBRARY, A. D. **Site oficial da ACM Digital Library**. 2024. <<https://dl.acm.org/>>. Acessado em 4 dez. 2024.
- LTD, E. **Site oficial do EasyChair**. 2024. <<https://easychair.org/overview>>. Acessado em 4 dez. 2024.
- MICROSOFT. **Site oficial do TypeScript**. 2010. <<https://www.typescriptlang.org/>>. Acessado em 4 dez. 2024.
- MUNDIAL, B. D. **Site oficial da Biblioteca Digital Mundial**. 2024. <<https://www.wdl.org/pt/>>. Acessado em 4 dez. 2024.
- OVERLEAF. **Site oficial do Overleaf**. 2024. <<https://www.overleaf.com/>>. Acessado em 4 dez. 2024.
- RICHARDSON, L. **Site oficial da biblioteca Beautiful Soup**. 2024. <<https://www.crummy.com/software/BeautifulSoup/>>. Acessado em 4 dez. 2024.
- SBC. **Site com a programação da 21ª ERRC**. 2024. <<https://errc2024.c3.furg.br/Programacao>>. Acessado em 5 dez. 2024.

SBC. **Site oficial da 21ª ERRC**. 2024. <<https://errc2024.c3.furg.br/>>. Acessado em 5 dez. 2024.

SBC. **Site oficial da SOL**. 2024. <<https://sol.sbc.org.br/>>. Acessado em 4 dez. 2024.

SBC. **Site oficial do JEMS 3**. 2024. <<https://jems3.sbc.org.br>>. Acessado em 4 dez. 2024.

SOFTWARE, H. C. M. **Site oficial do HotCRP**. 2024. <<https://hotcrp.com/>>. Acessado em 4 dez. 2024.

USP. **Site oficial da Biblioteca Brasileira**. 2024. <<https://www.bbm.usp.br/pt-br/>>. Acessado em 4 dez. 2024.

XPLORE, I. **Site oficial da IEEE Xplore**. 2024. <<https://ieeexplore.ieee.org/Xplore/home.jsp>>. Acessado em 4 dez. 2024.