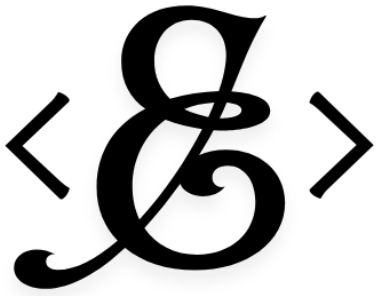

EDICOLAB

Implementação de mecanismos de validação de linguagens de descrição

University of Aveiro
Department of Electronics, Telecommunications and Informatics
Technical Report

André Dora, António Alberto, Gonçalo Sousa,
Hugo Sousa, José Pedro Oliveira

Supervised by
Prof. Dr. Osvaldo Pacheco
Prof. Dr. João Paulo Silvestre

osvaldo.pacheco@ua.pt, jpsilvestre@ua.pt

June 17, 2025

Abstract

Edicolab is a collaborative platform designed to facilitate the digital editing and annotation of historical documents, with a specific focus on medieval Portuguese Royal charters (forais). Developed within the scope of the iForal project, which aims to linguistically and historically study charters granted by the crown until 1279, Edicolab utilizes TEI-XML - a markup language standardized in the digital humanities. However, TEI lacked predefined models for encoding forais, needing customized markup structures. This shortage of flexibility introduces challenges related to consistency, correctness, and efficiency in the editing process.

Keywords: Charters, validation, customization, markup, digital annotation

Contents

Acknowledgments

We would like to acknowledge and express our deepest gratitude to all those who have supported and contributed to the completion of this project.

We thank our Advisors for all the support, input and guidance given throughout the Project in Informatics Course.

We thank Dr[a]. Catarina Coelho, who gave us important insights regarding the Digital Humanities field, and MSc Vasco Sousa, who explained us how Edicolab was implemented and helped us with doubts regarding Software Engineering.

Last, but not least, we would like to thank all of our colleagues and friends for their useful tips and words of encouragement.

Chapter 1

Introduction

Edicolab is a collaborative platform designed to facilitate the digital editing and annotation of historical documents, focusing on medieval Portuguese royal charters (forais), which are official documents that established the legal, administrative, and fiscal rights of portuguese towns and communities, playing a crucial role in the organization of local governance. The platform was developed under the iForal initiative, which focuses on the historical and linguistic study of these charters. To ensure accurate digital representation, Edicolab employs TEI-XML (Text Encoding Initiative), a widely used standard in Digital Humanities for encoding texts with structural and semantic metadata. However, since TEI does not provide a predefined schema for charters, there was a need for customizing markup structures to accommodate their unique characteristics.

The initial version of Edicolab, developed by José Vasco Sousa in his master's thesis, functioned as a specialized TEI-XML editor. It allowed researchers to edit and annotate historical texts, maintain an interactive glossary for consistent terminology, and produce critical editions of charters.

Despite these features, the platform faced limitations, particularly the absence of a validation mechanism to ensure proper TEI markup compliance and a true sense of collaboration when encoding a charter. These gap often led to structural inconsistencies, such as incorrect element hierarchies, missing required fields, or improper tag usage, making the editing process more time-consuming and prone to errors.

To address these challenges, this project focused on enhancing Edicolab by introducing a robust validation module. The validator checks the hierarchical structure of TEI-XML documents, verifies the presence of mandatory elements, and provides real-time or post-editing feedback to users. Additionally, it supports controlled customization, allowing researchers to adapt the validation rules to different document types while maintaining overall TEI compliance. Alongside this, improvements were made to the user experience, including automated TEI-XML tagging and a more streamlined editing workflow.

1.1 Report Structure

Apart from the Introduction, our report is divided into four additional sections: The State-of-the-art, which examines current technologies as well as the context of our project, the Conceptual Modeling, detailing our approach to selecting Personas, Scenarios, Use Cases and Requirements,

the Architecture And Implementation chapter, illustrating the pretended design of our system, and the description of the validation module, as well as some important Conclusions.

Chapter 2

State-of-the-art

2.1 Introduction

For a better understanding of the functionalities our project aimed to implement on the Edicolab platform, as well as their requirements, we analysed some of the most popular technologies for the transcription of ancient texts.

The research made by MSc José Vasco Sousa, responsible for the development of the first version of Edicolab, gave us a starting point for identifying the most popular platforms for historical documents transcription, as well as the most relevant when considering the goal of our project.

We identified some digital platforms with various main features and goals, such as document editing, document visualization and personalized TEI schema definition. We analysed the capabilities and limitations of each one, and compared them based on some characteristics that were relevant to the scope of the problem we were trying to solve, such as autocomplete features or error highlighting.

2.2 Overview of the Technologies

2.2.1 CEED

Cooperative Web-Based Editor for Critical Editions (CEED) was developed with the intent of studying medieval Arabian manuscripts. The main goal of the platform is to facilitate the formulation on Critical Editions (transcribed and/or translated text that tries to resemble the original text and it's author's intent as closely as possible) of those manuscripts, as well as allowing for collaboration with different transcribers for the same document.

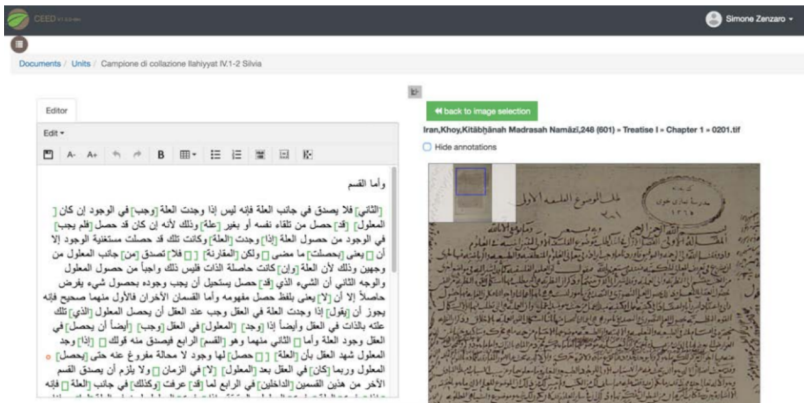

Figure 2.1: CEED interface

2.2.2 Patrimonivm Editor

Patrimonium Editor is a web application designed for studying inscriptions about ancient Roman emperors. It allows for the standard editing of text, but also implements the capability of georeferencing the places quoted on the documents. It also allows for conversion of texts with various markups into TEI markup.

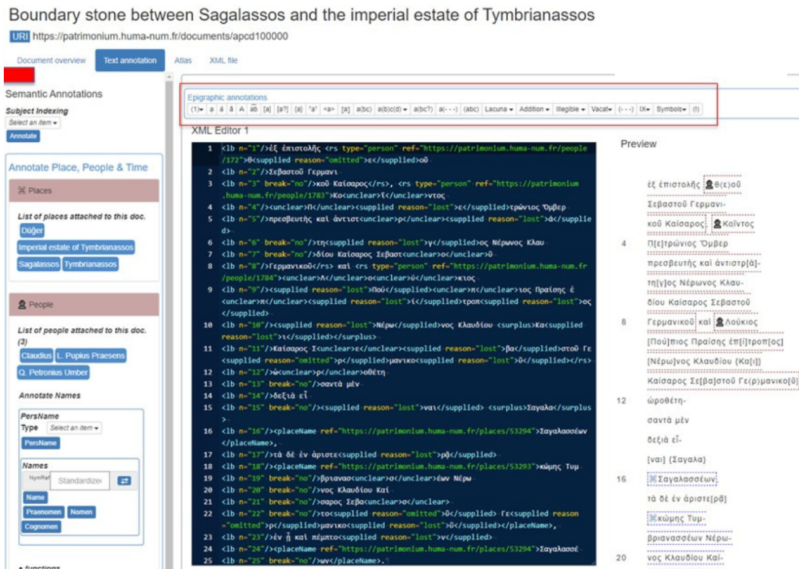

Figure 2.2: Patrinomivm Editor interface

2.2.3 Oxygen Editor

Oxygen Editor is a software based on Java that allows for editing text of various XLM based languages, such as TEI. Its UI resembles a standard IDE and its versatility made it one of the most popular tools for writing using XML languages.

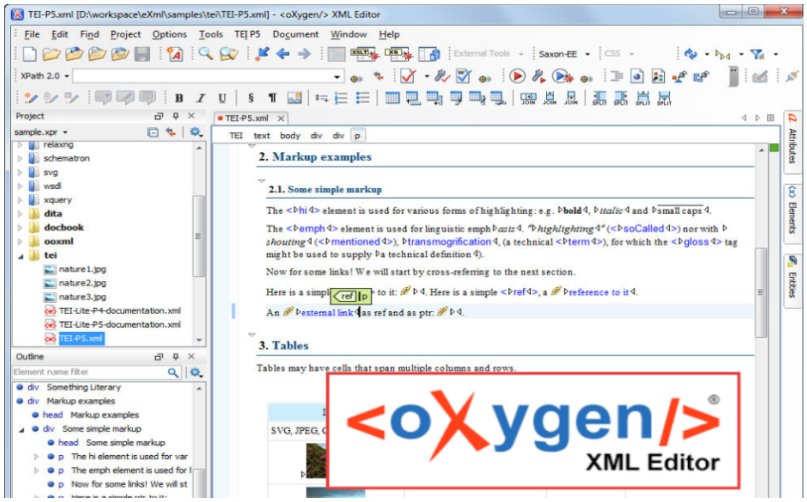

Figure 2.3: Oxygen Editor interface and logo

2.2.4 EVT

Edition Visualization Technology (EVT) is a web application made for the simple goal of visualizing TEI documents. It allows for conversion of TEI documents into HTML pages and also allows for the upload of scans and pictures that can be visualized in high quality. Its most relevant feature is the capability of selecting a line of text on the document image that then gets highlighted on the TEI visualizer.

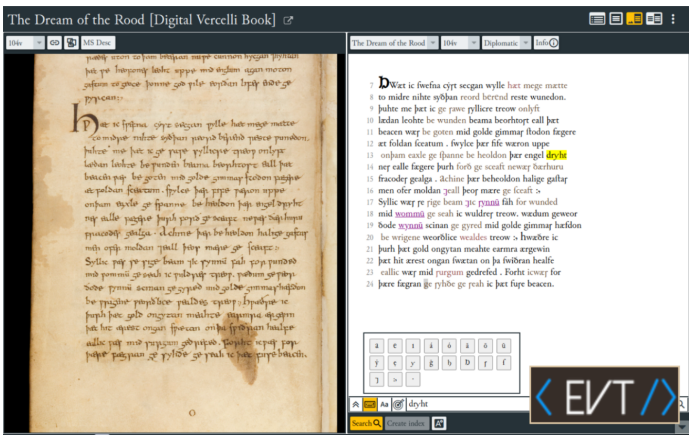

Figure 2.4: EVT interface and logo

2.2.5 ROMA

ROMA was created for allowing customization of the TEI markup. It lets the user define the structure of the TEI schema according to their personal needs and allows for creation of personalized tags, generating the custom schema. It's not an editor nor a visualizer.

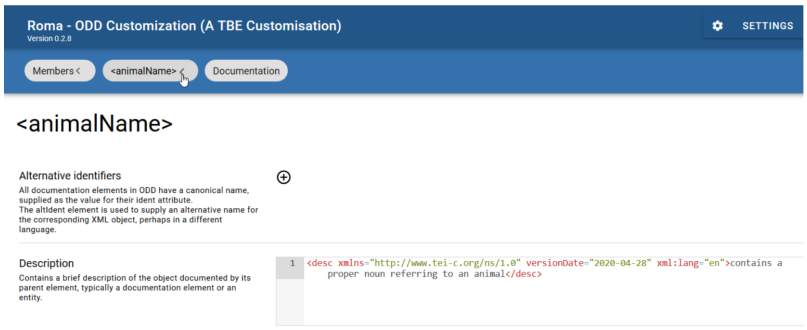

Figure 2.5: ROMA interface

2.3 Conclusion

After analysing the purpose and the available features of each platform, we formulated a table that includes some common useful features for any platform that works with documents that follow the TEI standard. We compared them based on a few simpler features and capabilities, as well as some features that we aimed to implement on our project.

We also included on the table the current version of Edicolab, which helped us identifying its strengths, weaknesses and main features.

	CEED	**Patrimonivm Editor**	**Oxygen Editor**	**EVT**	**ROMA**	**EDICOLAB (first version)**
TEI editor	Yes	No	Yes	No	No	Yes
TEI Visualizer	Yes	Yes	No	Yes	No	Yes
HD Document/image viewer	Yes	Yes	No	Yes	No	Yes
Interface Personalization	No	No	Yes	No	No	Partially
Custom Tags	Partially	Yes	Yes	No	Yes	No
Semantic Analysis	No	No	Partially	No	No	No
Tags' Autocomplete	Partially	No	Yes	No	No	No
Personalized validation Schema	No	No	Yes	No	Yes	No
Validation of TEI's Sintax	Limited	No	Yes	No	No	No
Error Highlighting	No	No	Yes	No	No	No

Figure 2.6: Technologies comparison

Chapter 3

Conceptual Modeling

3.1 Understanding the context and user needs

To develop the solutions to the problem that our project aimed to implement, we needed to better understand how Edicolab worked, but most importantly, who was going to interact with our system and what their goals were.

To gather those requirements, we first discussed these topics with our advisors, who gave us a clearer path towards our goals. After those discussions and consulting some documentation provided by them, we conducted some brainstorming sessions where we developed some personas and scenarios, as well as some user stories and use cases.

3.2 Personas

Joseph, 32 - Digital Humanities Investigator

Joseph Silva is 32 years old and got his Master's Degree in Digital Humanities at the University of Oxford. He always lived in the United Kingdom, but his Portuguese roots led him to integrate an international project where we got in charge of studying and formulating transcriptions of Portuguese Forais.

Miguel, 45 - Linguistics Professor

Miguel Soares, 45 years old, is a Professor of Historical Linguistics at Faculdade de Letras da Universidade de Lisboa and integrates the iForal Project. He is fluent in various languages, such as Portuguese and Latin, which is essential for the transcription of Portuguese forais that were initially written in Latin.

3.3 Scenarios

Based on the personas mentioned above and particularly on their needs and motivations, we came up with a set of scenarios that are illustrative of the goals of the Edicolab Platform, as well as some

additional features we aimed to implement.

Joseph's work on transcriptions

Joseph wants to start formulating Critical Editions (which is a term representative of a transcription that resembles its original text and author's intent as closely as possible) of Portuguese forais.

Even tho he already has experience with transcribing documents that follow the TEI standards, he isn't familiar with the particularities of the Portuguese forais and doesn't have experience with transcribing those types of documents. Edicolab was recommended to him by a fellow Portuguese colleague and friend.

Miguel's work on transcriptions

Professor Miguel is in charge of transcribing some 12th-century forais from the northern part of the country. He already has experience with transcribing these documents manually and digitally, but he isn't familiar with the TEI standards for the encoding of texts. Because he integrates the iForal Project, he was already familiarized with Edicolab, which is the most appropriate platform for the transcription of Portuguese charters.

Conclusions

These Personas and Scenarios helped us better understand who and how the majority of the users would use our platform.

Edicolab has been, since its inception, a platform that allows for text editing that supports the TEI markup, which can be achieved with the current state of the platform. But those transcriptions don't have any kind of validation or error searching and highlighting.

The platform is, without a doubt, a great one for visualizing documents and writing transcriptions at the same time. It has some useful features that help with transcribing text following the TEI markup, such as buttons and menus that automatically write the various tags supported in the TEI standards, as well as a glossary that helps in the transcription of Portuguese forais. But it's evident that the lack of validation features keeps it from being the ideal platform for tasks like the ones described in the scenarios we created:

- Joseph's lack of practice with the transcription of the Portuguese Forais could be fixed with the implementation of a feature that allows for the addition of a custom validation schema that is appropriate for the specific document he wants to transcribe. That would allow him to make sure that the TEI standards were met and that the text is semantically validated according to the defined schema;
- Miguel isn't familiar with the TEI standards, so a feature that allows for error highlighting when using the custom TEI markup tags, as well as an autocomplete feature when writing those tags, would help with reducing errors and the time it takes to formulate those transcriptions

3.4 Use Cases

During his work and research for his dissertation, MSc Vasco Sousa defined three actors for the main Use Cases, these being: Administrator, Encoder and Reader.

He defined Encoder as the users responsible for the insertion and codification of the forais, written with TEI, on the system. As well as the users responsible for filling in the header of each document and uploading the scanned images of each document.

As part of our work, we focused exclusively on implementing functionalities related to the use case that outlines the process of encoding charters (forais).

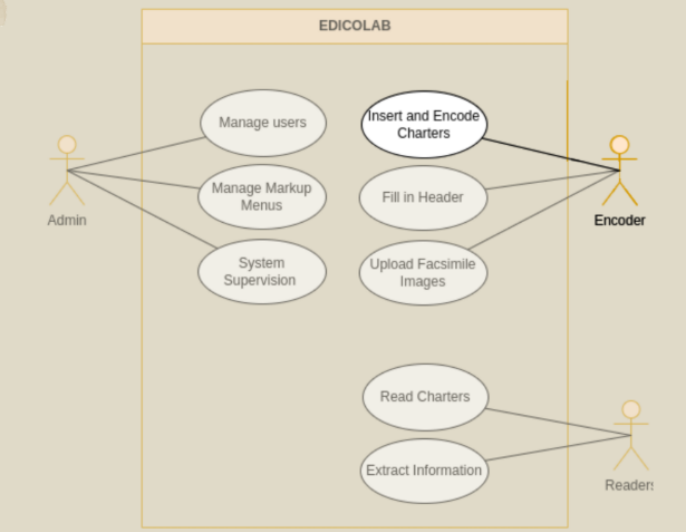

Figure 3.1: Use Case Diagram

3.5 Functional Requirements

During the elaboration of his dissertation, MSc Vasco Sousa also defined some functional requirements that he aimed to integrate on the Edicolab platform. The requirements that he defined that also are applicable in the scope of our project are the following:

- **"RF-7: Suportar a edição direta e indireta do código-fonte de um documento."**
- **"RF-8: Validar sintaticamente o código-fonte de um documento aquando da sua edição."**

Naturally, we also developed some functional requirements specific to the goal of our project:

- **Allow for the construction of different validation schemas that respect the TEI standard, with the option of creating custom tags.**

- **The system should have the ability to automatically insert tags that structurally depend on others while editing a document**
  For example, in HTML, the "HTML" tag should have a "body" tag inside of it.

- **Semantically validate the TEI source code of a document while it is being edited, according to the selected validation schema**
  For example, the system should validate that the tags and their attributes are correctly utilized and positioned

- **Error highlighting while editing a TEI document**

The functional requirements can be summarized in the following table:

Reference	Description
FR-1	Support direct and indirect editing of the source code of a document
FR-2	Syntactically validate the source code of a document while editing
FR-3	Allow for the construction of different validation schemas that respect the TEI standard, with the option of creating custom tags
FR-4	The system should have the ability to automatically insert tags that structurally depend on others while editing a document
FR-5	Semantically validate the TEI source code of a document while it is being edited, according to the selected validation schema
FR-6	Error highlighting while editing a TEI document

Table 3.1: Functional Requirements

3.6 Non-Functional Requirements

The Non-Functional Requirements defined by MSc Vasco Sousa were inserted in four categories: Usability, Performance, Security and Technologies.

Most of them are also applicable to the scope of our project, and some of the most relevant ones are:

- **"RD-2: Assegurar que o tempo de resposta é inferior a 500 ms"** - because the validation module should respond almost instantaneously;

- **"RS-1: Efetuar toda a comunicação entre os componentes do sistema através de uma ligação HTTPS"** - because the communication between the different components of the system should remain secure;

- **"RT-1: Utilizar exclusivamente tecnologias de código aberto cujo projeto não se encontre abandonado"** - to maintain the scalability of the system;

- **"RT-2: Fornecer um container Docker que permita colocar o sistema em produção"** - allowing for future production of the system and also allow for easy addition of new features;

We also decided to include a new non-functional requirement: **The existing API must communicate with the new controllers.**

It's also important to note that, even tho not described in the report, our platform should and will continue to follow the usual non-functional requirements associated with good practices of software engineering, such as reliability and maintainability.

The Non-Functional Requirements can also be summarized in the following table:

Reference	Description
NFR-1	Ensure that the response time is inferior to 500 ms
NFR-2	The communication between all the components of the system should be made through HTTPS connections
NFR-3	Usage of Open Source technologies where its project isn't abandoned
NFR-4	Provide a Docker Container that allows to put the system in production
NFR-5	The existing API must communicate with the new controllers

Table 3.2: Non-Functional Requirements

Chapter 4

Implementation and Architecture

4.1 System's Architecture

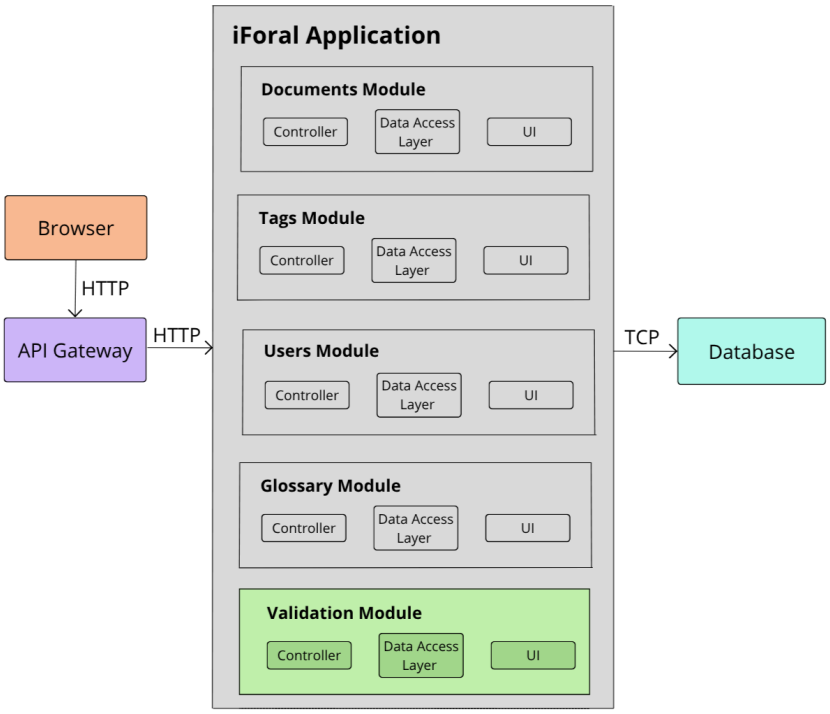

Figure 4.1: Diagram depicting Edicolab's updated architecture.

Instead of restructuring the existing platform, we decided to develop a validation module. This module includes a **TypeScript-based backend** responsible for processing and validating docu-

ments, and a **Svelte-based frontend** for the user interface. In the **Database (PostgreSQL)**, we added the support to store the validation schema associated with each document and incorporated the logic to manage the automatic creation of tags during the editing process.

4.2 Implementation

As said before, to overcome the validation limitations, we developed a validation module integrated into Edicolab. This new feature ensures that TEI-XML annotations follow a coherent and customizable schema, capable of adapting to different document types and editorial guidelines. The validator checks for the presence and structure of required elements, maintains hierarchical integrity, and provides informative feedback - either in real-time during editing or as a post-processing step. It also supports controlled customization, allowing editors to define validation rules tailored to their specific research needs. The key features we added into the system aimed to:

- Validate structural elements and their hierarchy;
- Provide clearer feedback as each document is being written;
- Support custom rules for different documents and editorial guidelines;
- Autocompletion of tags to simplify the writing process for less experienced users.

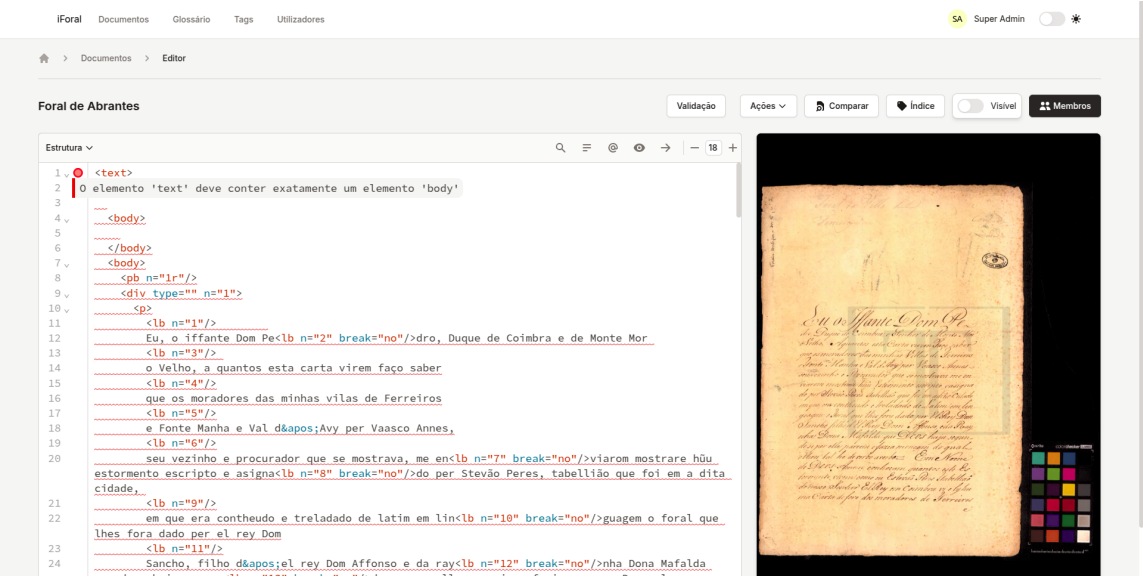

Figure 4.2: Example of an error that affect's the whole document

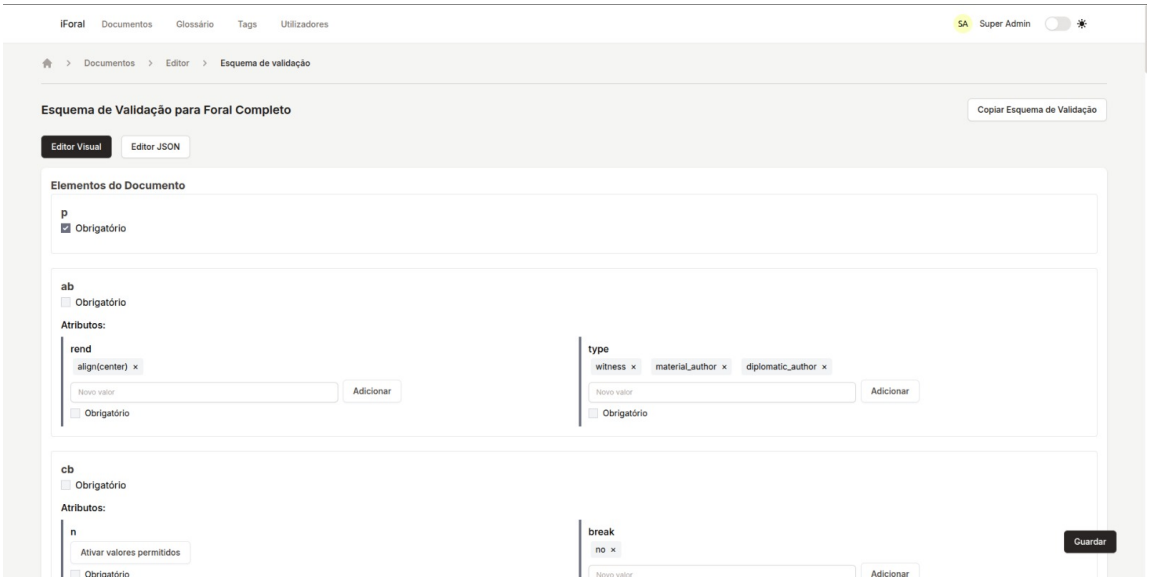

Figure 4.3: Interface to define the schema for each specific transcription

4.3 User's Feedback

After implementing the main features we set out to do in the inception phase of our project, we conducted a feedback session with our Advisor, Prof. Dr. João Paulo Silvestre, and Dr$^{\underline{a}}$ Catarina Coelho, as both of them had knowledge and experience with digitally transcribing ancient documents as well as practice using the TEI Standard.

While discussing the usage of the Edicolab platform following the changes we implemented, they described the new features as an important step-up for the digital transcription of charters and suggested the addition of a button that could turn off the validation functionalities, which we implemented before our final delivery.

4.4 Development Challenges

The implementation of the validation module faced several technical hurdles. Initially, we developed a separate Flask API for validation, but this created unnecessary complexity. The external service made schema customization difficult and introduced performance bottlenecks.

We addressed these issues by integrating validation directly into Edicolab's core architecture. Key improvements included:

- **Document-Specific Validation Rules:** Added a dedicated validationSchema column to enable customized validation per document;
- **Hierarchical Tag Structure:** Redesigned the tags table to support parent-child relationships and automatic tag generation;

- **Validation Customization API:** Created a dedicated **/validation** endpoint for dynamic schema management;
- **Real-Time Error Feedback:** Integrated CodeMirror's Linter to provide instant visual validation during editing.

This integrated approach eliminated the external dependency while improving responsiveness and enabling dynamic schema customization. The solution successfully balanced validation requirements with the platform's existing architecture, demonstrating the value of cohesive system design.

Chapter 5

Conclusion

Even though we acknowledge the advancements made in the platform, we would also like to set some others functionalities as possible future work that would make it more robust.

The integration of Artificial Intelligence could exponentially enhance Edicolab's capabilities in many ways, for example, by allowing for the suggestion of tags based on the structure, content and context of each manuscript.

Another relevant feature we would like to point out is the addition of a module that could register each document's history of editing as well as each transcriber's contribution, allowing for a truly collaborative experience with transcribing charters and other ancient documents.

To address the challenges set at the beginning of our journey through Digital Humanities, we focused on enhancing Edicolab by introducing a robust validation module. The validator now allows for the platform to check the hierarchical structure of TEI-XML documents, verifies the presence of mandatory elements, and provides real-time or post-editing feedback to users. Additionally, it supports controlled customization, allowing researchers to adapt the validation rules to different document types while maintaining overall TEI compliance. Alongside this, improvements were made to the user experience, including automated TEI-XML tagging and a more streamlined editing workflow.