Recetas de las Américas: A Case Study in Hugo Static Site Generator for Bilingual Web Publishing

SARAH TEW AND MELISSA JEROME

Abstract: This article describes *Recetas de las Américas*, a static web project for historic newspaper recipes made with the static site generator Hugo. Static sites offer opportunities for robust online publishing without large computational, financial, or technical resources but are not without their own challenges and constraints. This paper presents *Recetas de las Américas*, discusses the place of static sites in the context of minimal computing, and examines the utility of Hugo for multilingual online publishing, including comparison between Hugo and other static site generators. Static publishing follows minimal computing principles and is a valuable tool for increasing access to digital publishing. Recently, static websites have experienced a renaissance thanks to the development of static site generators such as Jekyll, Hugo, Gatsby, 11ty, and Next.js among others which automate the tedious process of creating each static webpage. Unlike some other popular static site generators, Hugo's multilingual capabilities are built-in, decreasing dependencies and complexity. This further lowers barriers to multilingual scholarly publishing, potentially increasing participation by historically marginalized communities as publishers and readers. *Recetas de las Américas* is a valuable case study in using the static site generator Hugo to publish a bilingual Spanish/English website. By sharing our workflow and lessons learned we hope to help others create their own multilingual static websites.

Keywords: multilingualism, Latin American, Hugo, static websites, Latin American, Hugo, static websites

Static websites hold promise for expanding access to publishing to often marginalized groups and for reconnecting digital creators with the processes and resources used to publish their work, but they are not without challenges, constraints, and inequities. Although static websites offer attractive lower financial and resource costs, they come

at the price of unevenly distributed labor, most often falling to the technical developer (Dombrowski 2022). Because of linguistic inequity in software, documentation, and the programming languages themselves, this labor is amplified for developers for whom English is not their default language.

This article will examine the extent to which web publishing with Hugo, a static site generator, challenges English-language dominance in digital publishing in the humanities and to what extent it perpetuates it through a case study on the making of the bilingual static website for newspaper recipes *Recetas de las Américas*, developed by the US Caribbean & Florida Digital Newspaper Project (USCFDNP). This article will introduce the USCFDNP and the original recipe material featured in *Recetas,* trace the development of the project step-by-step, and evaluate the methods and software used for their adherence to minimal computing principles and their applicability to other multilingual digital humanities (DH) web projects.

The US Caribbean & Florida Digital Newspaper Project

Launched in 2022, *Recetas de las Américas* (hereafter *Recetas*) is a bilingual static website that allows users to engage with recipes originally published in *Diario las Américas*, a Hispanic, Spanish-language newspaper that started in Miami, Florida, in 1953.[1] A decade's worth of this newspaper's archive was digitized through USCFDNP, a collaborative project between the University of Florida's George A. Smathers Libraries and the library systems at the University of Puerto Rico–Río Piedras and the University of the Virgin Islands (USCFDNP, n.d.).

USCFDNP is part of the National Digital Newspaper Program (NDNP), which is funded through the National Endowment for the Humanities (NEH). The NEH NDNP grant provides funding for the digitization of historic newspapers from across the United States and its territories for public access, for the varying tasks from title selection to digitization, and for efforts to raise awareness and promote the use of the content made available. The goal is to preserve and increase access to valuable, historical information from newspapers in a freely accessible online central platform (NDNP 2023).

USCFDNP has received continuous funding from the NEH for the past 10 years and recently acquired additional funds to run the project through September 2025. This funding allowed us to hire a project coordinator (who is 100% grant funded) who oversees all technical and day-to-day aspects of the project. The funds also provide support for outreach events and travel and, indirectly, to develop digital humanities

1. Sarah Tew and Melissa Jerome, *Recetas de las Américas*, https://recetas.domains.uflib.ufl.edu.

projects such as *Recetas*. Also, through NEH funding, we were granted the opportunity to digitize over 30 newspapers from Florida, Puerto Rico, and the Virgin Islands from 1836 to 1963, with more to come.

NDNP digitization involves scanning microfilm to create TIFF and JPEG 2000 that retain the image of the actual newspaper for viewing and using optical character recognition (OCR) to create TXT and PDF files for every page scanned (NDNP 2023). The scanned images and all digital outputs are made available for free public access via Chronicling America (a website developed and managed by the Library of Congress for the NDNP), as well as through the University of Florida Digital Collections (UFDC) and the Digital Library of the Caribbean (dLOC).[2] The digital outputs required by the grant allow the newspaper content to be freely used as data for DH and other projects by anyone.

Recetas de las Américas

Recetas was born out of a desire to entice users to utilize newspaper data from the historic newspapers digitized as part of the USCFDNP. To promote the newspaper content, we sought to develop a digital humanities project that would garner the attention of users while having a low technical barrier so it could be replicated for other newspaper data sets. The "Receta del Día" ("Recipe of the Day") section in *Diario las Américas* proved to be prime data with which to start; there was one recipe published in just about every issue of the newspaper, and most recipes followed a standard format of including the "Receta del Día" header, a recipe title, an ingredients list, and recipe directions.

Food also seemed to be an easy way to engage a large audience as it is essential to our human existence and serves as a connector across social and geographical borders. Although many of the other newspapers we have digitized through the USCFDNP include recipes, we chose to focus on those printed in *Diario las Américas* to acknowledge the past and present ties between communities in Florida, Latin America, and the Caribbean. Highlighting content of Hispanic origin was personally important to us as well, as we both work in the Latin American and Caribbean Collection at the University of Florida Libraries and Melissa Jerome is Latina.

Recetas currently consists of 53 recipes originally published in Spanish in *Diario las Américas* from 1954 to 1960. These can be explored in multiple ways (Jerome 2022).

2. Library of Congress, Chronicling America, accessed December 31, 2023, https://www.loc.gov/collections/chronicling-america; University of Florida George A. Smathers Libraries, University of Florida Digital Collections (UFDC), accessed December 31, 2023, https://ufdc.ufl.edu; and Digital Library of the Caribbean (dLOC), accessed December 31, 2023, https://dloc.com.

The Recetas tab provides a list (in alphabetical order) of the recipe titles for the recipes available. From this page, users can also opt to search for a recipe by typing in a recipe name in the search box (Figure 1).

Either option allows users to click on and view a recipe of choice. The Filter tab allows for more robust searching. With this search option, users have three categories of tags to choose from on the left-hand side: (1) dietary restrictions, (2) course, and (3) ingredients. Selecting any combination of these tags results in a list of recipes that incorporate the dietary restrictions, meal type, or ingredients selected. Additionally, users are presented with a search box with the ability to search for a recipe by ingredient (Figure 2).

Because of the small sample size and nature of static sites, the development of *Recetas* from ideation to creation was relatively quick. We jumped from coming up with a vision in July 2022, to harvesting, cleaning, and translating the data, to ultimately publicly launching the first iteration of the site three months later in October 2022. From its launch to December 2023, the site has received 3,082 visits (2,118 unique visitors) with 8,716 pages served. The site has received positive feedback from the public. For instance, a professor from another institution expressed interest in working with us to incorporate

Figure 1. Recipe list page in Spanish. Accessed April 15, 2024.
Página de lista de recetas en español. Consultada el 15 de abril de 2024.

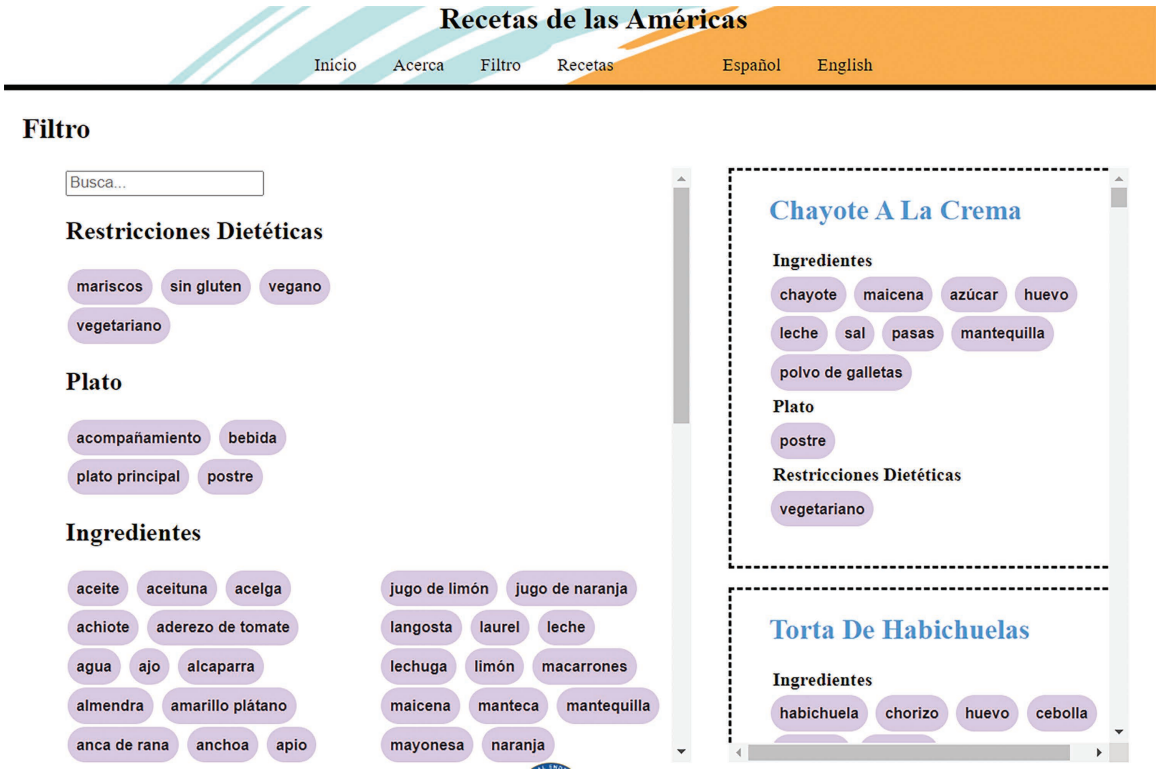

Figure 2. Filter page in Spanish. Accessed April 15, 2024.
Página de filtro en español. Consultada el 15 de abril de 2024.

this project into their Spanish course. Additionally, other individuals have tested the recipes and submitted photos, which are featured on the site's home page. This interest and involvement highlight the importance of accessibility and inclusivity in DH projects.

In academia and the field of DH, English has staked its claim as the default language. In the Global North, the odds of encountering non-English scholarship are low, often confined to only being published in special issues with designated languages, which further "reinforces the power imbalance between languages" (Nilsson-Fernàndez and Dombrowski 2022, 86). Since combating English hegemony is a personal and professional goal for the team, and the recipe content of our website was originally published in Spanish, *Recetas* was created as a bilingual site and uses Spanish as the default language. Publishing in Spanish better represents and connects users to the cultural origins of the recipes, their creators, and their intended audiences, while including English translations bridges cultural divides, reflects Florida's multicultural past and present, and supports broader accessibility and engagement with the materials.

In achieving bilingual functionality for the site, we leaned heavily on our own language proficiency. Being fluent in both Spanish and English proved advantageous, allowing us to discern nuances and contextual intricacies overlooked by automated

translation processes. Initially, we relied on Microsoft Excel's auto-translate feature to expedite the translation process. While this approach provided mostly accurate translations initially, it necessitated extensive editing for spelling, grammar, and contextual accuracy. Although professional translation services were considered, the associated high cost and limited language availability posed challenges (Nilsson-Fernàndez and Dombrowski 2022). Additionally, our budget constraints prohibited us from outsourcing this task, leaving us to shoulder the responsibility ourselves.

Minimal Computing

Minimal computing lowers barriers related to costs, technical expertise, and access to hardware and software. It is helpful in instances where innovation and creativity are essential, especially when there is little to no funding available to invest in proprietary programs that may not remain accessible or available long-term (Risam 2018). Minimal computing also facilitates the exploration of multilingual publishing, which is particularly significant for DH practitioners operating in languages other than English.

DH work in the Global North typically overshadows the valuable non-English-language work of others in the field (Gil and Ortega 2016). Providing content only in English results in lack of context and missing or misconstrued narratives that conflict with the aims of DH being intersectional and inclusive (Nilsson-Fernàndez and Dombrowski 2022). There is a need in the field of DH to "overcome the English-centric bias" to reshape for the multilingual needs of the growing global DH community (Tanasescu 2021, 38). It is crucial for DH practitioners to consider "tools, tutorials, and code that meet the needs of scholars working in languages other than English" (Tanasescu 2021, 30). One tool that DH practitioners can use to dismantle these barriers is minimal computing. Minimal computing opens doors for the "development of a digital cultural record that includes voices and stories from communities that have been elided in the cultural record" (Risam and Gil 2022, para. 21). This is what drives our work with *Recetas* and the reason why we made certain linguistic and technical decisions—to ensure that we acknowledge and represent communities whose voices have been historically and are continuously underrepresented and with the intent to create a model that can be shared and replicated by others.

Minimal computing is posed as a "bare minimum to achieve simple tasks" (Gil and Ortega 2016, 28), making it an appealing option for those who may be new or those with more expertise in the field alike. However, this approach is not "minimal"—there is labor which is often unevenly distributed, most often falling to the technical developer (Dombrowski 2022). Nor does "minimal" mean ease of use but rather "advocates for using *only* the technologies that are necessary and sufficient for developing digital

humanities scholarship" (Risam and Gil 2022). Staying focused on, as Alex Gil and Élika Ortega put it, "what we need" was the driving force for our decisions to keep *Recetas* "minimal" and publish as a multilingual static site.

Static Web Publishing

We made *Recetas* with Hugo, a static site generator, to help people better access historic newspaper recipes.[3] Static sites are just one of many tools for minimal computing and well suited to multilingual publishing. We opted for a static site to publish *Recetas* instead of more traditional resource-intensive approaches such as WordPress and Drupal because a static site requires less financial, computational, storage, preservation, and maintenance costs. The size of the finished site is currently 41 MB, and the size of the materials used by Hugo to create it is 108 MB. The website is generated in 1,030 ms (about 1 second).

Static sites are good choices for DH projects generally because of their lower costs. They use foundational web technologies and are especially well suited for projects with fixed content that do not require dynamic capabilities or frequent updates. Because of their simplicity, they also cost the user less time and require fewer computational resources to access. These features support our goal of creating a more accessible publishing model, particularly for those working outside of traditional institutions, with fewer financial and technological resources, and/or in languages other than English.

Although static site generators can be combined with headless content management systems (CMSs), we chose not to add a CMS to *Recetas* and to instead rely solely on Hugo's file directory structure for content management.[4] Eliminating all other software dependencies keeps *Recetas* smaller, more lightweight, and less technically complex while decreasing the site's vulnerabilities and, therefore, saving money and IT staff time.

Hugo is very similar to other static site generators, including Jekyll, which is behind many tools frequently used in DH such as GitHub Pages, Collection Builder, and Wax. Static sites for DH projects are well documented, and DH practitioners, including Kate Newson (2017) and Chris Diaz (2018) have published practical guides for understanding and using static site generators for scholarly publishing and digital projects.

While the lower financial and computational costs of static websites can potentially expand access to digital publishing to those outside of traditional models, some of these advantages come at the price of the developer's time and labor (Dombrowski 2022). In making *Recetas*, time and skills for templating came at the highest price.

3. Hugo, accessed December 31, 2023, https://gohugo.io.
4. For a brief overview of headless CMSs with static sites, see Newson (2017).

Static site generators require templates to compile the various components of a site. While the principles of templating are transferable, different generators use different templating languages, each with their own syntax, variables, functions, and methods. The templates are non-transferable. Templates and other assets can be grouped together in a theme. Hugo's website features 250 different free themes, mostly for blogs, digital portfolios, and documentation (Hugo, n.d.-a).

Writing templates and understanding Hugo's hierarchy of templates and themes was the most challenging aspect of the *Recetas* publication process. For *Recetas*, Sarah Tew wrote custom templates for the different types of pages required by the project and a bare-bones theme for the entire site. DH projects are often unique and require designs and capabilities beyond traditional blog or basic e-commerce that existing free templates often do not offer. The filter page in *Recetas* is one example of this issue which was ultimately resolved by writing a custom template with JavaScript.

Multilingual Hugo

We selected Hugo over other static site generators because of Hugo's better out-of-the-box multilingual handling. Hugo and many other static site generators can populate some basic text such as dates and times in multiple languages automatically but do not translate content. Hugo is a good choice for multilingual projects because it makes the multilingual content and its organization directly visible and equal, does not assume any language as the primary language of the site unless explicitly stated, and does not privilege monolingual publishing by requiring extra plugins or downloads for additional languages.

Unlike static site generators such as Jekyll and Eleventy, Hugo includes i18n multilingual support directly in the Hugo package (Hugo, n.d.-b). Once Hugo is downloaded and installed, it can handle a multilingual site as long as the developer knows where and how to declare the languages and organize the content. Eleventy, Gatsby, and most other static site generators rely on plugins to add multilingual support. This adds technical complexity and more opportunities for broken pipelines and frustration, especially for less experienced developers. There are ways to configure a multilingual Jekyll site without plugins, but these methods are more complex than Hugo's (Simple IT Rocks 2019). Eleventy also handles multilingual content well but with more complexity, requiring JavaScript to localize dates, for example.

Hugo also has faster build times than other static site generators. While not such a concern for our project right now, we hope to ingest hundreds if not thousands more recipes from other newspapers in the future. As the scale increases, the build speed becomes increasingly valuable, especially in trial development as the site is built and rebuilt locally as the developer makes changes. Even for small projects, build time can

become a frustrating limiting factor as large or deep zoom image files are added. Multilingual websites are larger than monolingual websites since the same page exists in each language, so the quicker build time becomes a greater advantage the larger and more files and more languages a site has, making Hugo a good choice for multilingual projects.

Assuming the developer has the technical skills to write their own templates, another advantage of static sites (made with Hugo or other generators) for multilingual projects is the ability to create unique templates for content in different languages. While *Recetas* does not exploit this capability—the Spanish and English templates are the same, only the content that fills them changes—this is useful for projects requiring different layouts for reading, accommodating left-to-right, right-to-left, and vertical languages all in the same website or even the same page more simply than other web publishing methods such as WordPress, Drupal, and Omeka. The templating language itself exhibits symptoms of English monolingualism, however. Go, the language behind Hugo, was created at Google by three computer programmers from Canada, Switzerland, and the United States with English as the apparent default language in the code. While Hugo has great potential to facilitate web publishing in languages other than English, it does not undermine English-language dominance in computer languages.

For developers, the lack of a graphical user interface (GUI) in Hugo is a double-edged sword. On the one hand, it means that novice developers need different technical skills than they may already possess if their previous experience has been with publishing platforms such as WordPress. On the other hand, it means that multilingual developers are not limited by a GUI in one language. With a stand-alone static site generator, the developer is better able to escape the technical and psychological confines of English being the default language for web publishing without jumping through any extra hoops to do so. What they cannot escape, however, is English as the default programming language. The templating language is still English-based, and, in Hugo's case, the official documentation for it is also only available in English.

While Hugo has sufficient documentation in English for multilingual implementation, it lacks any official documentation in any other language. Hugo's website also links to more detailed explanations and tutorials from third parties, including a two-part series by Régis Philibert (2018, 2019) on how to make a multilingual Hugo site and links to a YouTube series on Hugo produced by Giraffe Academy.[5] There are other similar how-to websites and YouTube channels producing high-quality Hugo tutorials in other languages, but these are not listed anywhere on the official Hugo website, undermining its utility as a tool to disrupt English-language supremacy in web publishing.

5. The Giraffe Academy playlist is available at https://youtube.com/playlist?list=PLLAZ4kZ9dFpOnyRlyS-liKL5ReHDcj4G3&feature=shared.

As is often the case, multilingual practitioners carry the burden of translation and education, creating tutorials and translating software documentation themselves. This work is extremely valuable and often underappreciated. Part of our motivation for publishing this article and more detailed walkthroughs in English and Spanish is to ease the burden on our multilingual colleagues and make this work more visible to everyone in the community. To help others create multilingual websites with Hugo we freely share the resources used to make it in the project's GitHub.[6] Sarah Tew (2023) also created step-by-step tutorials in English and Spanish which use a simplified version of the *Recetas* website as an example site and will detail the technical processing in the next section. Additionally, we are sharing this case study and our findings through conference talks, webinars, and publications in English and Spanish, including this article.[7]

From Spreadsheet to Static Website

Static websites can be made in any number of ways from diverse source material. In this section, we provide an overview of our process to transform spreadsheets into a website for the original 53 recipes. It describes what it was like to work on a bilingual project, analyzes how successful this method was, discusses changes that must be made to the pipeline as we increase recipe ingest, and critiques our use of proprietary software (Oxygen XML Editor and MS Excel) within the pipeline (Figure 3).

Recetas was created from a spreadsheet which was transformed through imports and XSLT transformations in Oxygen XML Editor into the Markdown (MD) files and HTML partials that Hugo used to compile the final website. We used a spreadsheet as the foundation of the project because (1) Melissa Jerome had already compiled 53 links to recipes in an Excel spreadsheet and (2) spreadsheets are ubiquitous in a variety of disciplines and familiar to people who otherwise feel intimidated by digital technology. Spreadsheets are so common for storing information, and developing a method for converting them into a website could open web publishing possibilities with colleagues at the UF Libraries and beyond.

The recipes currently published on the *Recetas* website were manually harvested using Melissa Jerome's original spreadsheet with the title of the recipe and a URL link to the *Diario las Américas* newspaper page in Chronicling America. We went to each URL, saved a JPEG snip of the recipe image, and copied the original Spanish OCR text into the Excel spreadsheet. As we scale up our recipe ingest, we are working on a Python

6. Sarah Tew, public-recetas, GitHub repository, https://github.com/SarahTew/public-recetas.

7. For a list of all the US Caribbean & Florida Digital Newspaper Project's presentations, including those about *Recetas de las Américas*, visit https://ufndnp.domains.uflib.ufl.edu/presentations/.

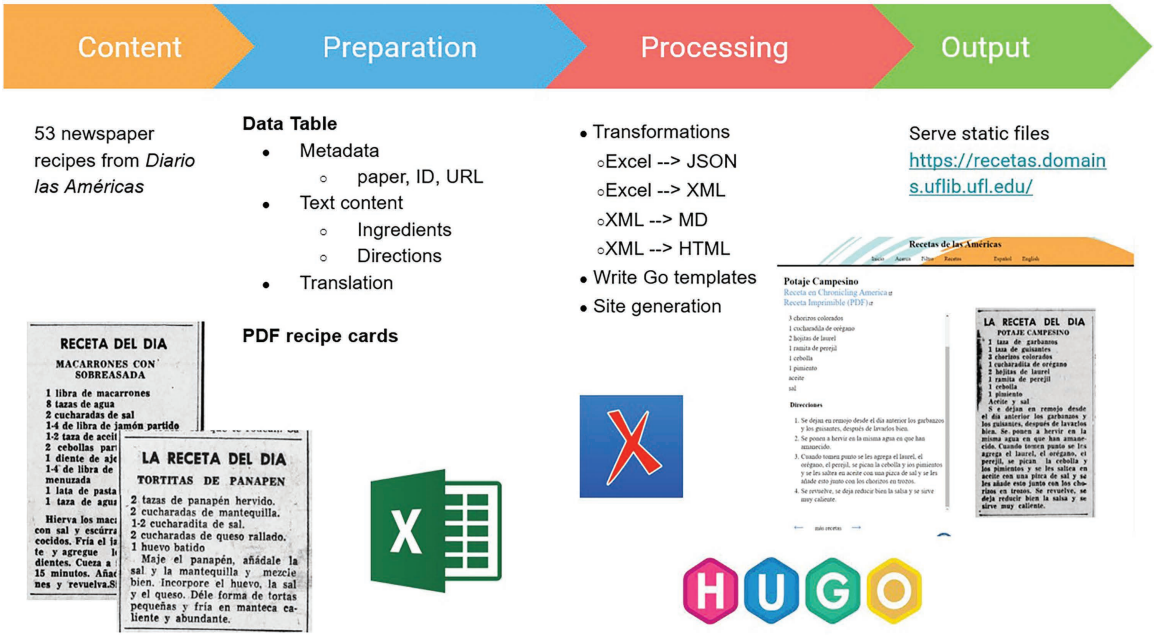

Figure 3. Diagram of the Recetas project pipeline.
Esquema del proceso para el proyecto Recetas.

program to automatically identify and harvest recipes through Chronicling America's application programming interface (API).

Once the original Spanish OCR text was in the spreadsheet, we read and manually cleaned the text. The OCR recipe text was generally in good condition. The papers themselves were mostly in very good to excellent condition with full pages free of smears, rips, and typos. The OCR text was created by our vendor during digitization using ABBYY FineReader Engine. The most frequent problem in the recipe OCR was fractions, which were often rendered as percent signs (%) rather than as digits. Weak type was also an issue, especially with accented characters. Currently, we are writing Python scripts to clean, restructure, and translate recipe data.[8]

After cleaning and editing the Spanish OCR text to match the printed text in the newspaper image, we semi-automatically translated the text into English. Using Microsoft Excel's built-in translation, we first translated the directions and ingredients cells into English, then skimmed the English text and corrected major errors. The automatic translation in Excel worked better than we expected, largely due to the grammatical simplicity and relative brevity of the *Diario las Américas* recipe content, which almost exclusively consists of lists of concrete nouns followed by a series of short, imperative sentences. That said, we relied heavily on our own bilingualism. Errors in the Excel

8. Sarah Tew, recetas2024.ipynb, Google Colab, 2024, https://colab.research.google.com/drive/1HJ036_cCPkk85XAi6_gKD0-8dIcFAWPt.

translations were most often due to errors in the Spanish OCR text, so as we reviewed the English translation, we returned to the source of the error in the original Spanish text and corrected it. As we broaden recipe ingest to other newspapers, we expect to confront more linguistic and grammatical diversity and are therefore developing more robust translation methods in Python. We are also curious to investigate the potential of translation as a tool for data cleaning and restructuring.

The final step in preparing the text within the spreadsheet was to manually extract simplified ingredient lists and add tags for dietary restrictions and course. This was the main source of introduced errors (e.g., "pan tostado" not being simplified to "pan"). We are currently working to semi-automate this process using Python to clean and restructure the text and to assign dietary restrictions based on ingredients.

Once the spreadsheet text was cleaned and structured correctly, XSLT style sheets were applied in Oxygen XML Editor to generate MD and HTML files. MD files (Figure 4) contain the metadata for each recipe, including a simplified ingredient list which feeds the filter page, as well as instructions to Hugo to pull the associated HTML partials. The HTML partials contain the full ingredients list and directions displayed on each recipe's web page (Figure 5). We chose to use HTML partials rather than include the content directly in the MD file because we wanted elements of the website separate and to use Hugo to compile them. The image files are also stored separately and called by an HTML layout file. Using HTML partials and Hugo layouts added technical complexity. This was useful for the *Recetas* project since we also wanted to further explore Hugo, practice writing XSLT, learn the templating language, and be better able to make more diverse websites in the future.

Since the original recipes were published in Spanish, Spanish is the default language of the published *Recetas* website. The Spanglish metadata fields illustrate some habits of a multilingual person working in two languages. Sarah Tew wrote the XSLT to transform the spreadsheet into the MD and HTML files used by Hugo to generate the site.

```
---
id: r39
title: gazpacho
description: Recipes de las Américas
etiquettas: ["vegetarian","vegan","gluten-free"]
ingredientes: ["garlic","peppercorn","salt","oil","water","vinegar","tomato","onion","pickles","pepper"]
plato: ["side"]
chronam: https://chroniclingamerica.loc.gov/lccn/sn82001257/1955-07-28/ed-1/seq-5/
jpg: r39.jpg
pdf: r39_en.pdf
---

{{< readFile file="./html/recipes/r39.en.html">}}
```

Figure 4. English Markdown file for Recipe 39, Gazpacho.
Archivo Markdown en inglés para la receta 39, Gazpacho.

```
<div class="ingredients">
   <h4 id="ingredhead">Ingredients</h4>
   <ul id="ingredients">
      <li>1 clove of garlic</li>
      <li>1/4 teaspoon peppercorns</li>
      <li>1/2 teaspoon salt</li>
      <li>3 tablespoons of oil</li>
      <li>4 cups of water</li>
      <li>2 tablespoons vinegar</li>
      <li>4 chopped tomatoes</li>
      <li>2 chopped onions</li>
      <li>2 chopped pickles</li>
      <li>2 peppers, chopped</li>
   </ul>
</div>


<div class="directions">
   <h4 id="dirhead">Directions</h4>
   <ol id="directions">
      <li>Grind the garlic along with the pepper, salt and add the oil, water and vinegar.</li>
      <li>Strain and add to chopped vegetables.</li>
      <li>Put in the fridge for several hours and add slices of bread before serving.</li>
   </ol>
</div>
```

Figure 5. English HTML partial for Recipe 39, Gazpacho.
HTML parcial en inglés para la Receta 39, Gazpacho.

The MD files contain metadata about each recipe, including title, paper, and ingredient lists. The website itself was developed bilingually, with Sarah switching between Spanish and English depending on the phase of the project and the material she was working with. The recipe content itself was in Spanish, but Oxygen XML Editor, Hugo documentation, and the Go templating language were all in English, which pulled her to and from Spanish and English throughout the development. The bilingual nature of the project is evidenced in the mixed languages within the MD files, with some fields, such as "Ingredientes" and "Etiquettas," in Spanish and others, such as "Title" or "Paper," in English. We will switch all fields to Spanish with the upcoming large-scale ingest and update to make the project more usable and replicable for others. Static site generators easily support metadata in mixed languages. While we feel this is ultimately undesirable for *Recetas,* it may be useful for other projects that would benefit from the flexibility to name and mix fields in multiple languages.

Once we created the MD files, HTML partials, and draft templates for the layouts written, we reviewed the site through Hugo's built-in local host command. The locally hosted site was a convenient tool for reviewing translations and catching errors in both the Spanish and English ingredient lists and tags. The filter page, which lists simplified ingredients alphabetically, was useful for finding typos and incorrectly simplified

ingredients (e.g., "sal" vs. "sal gruesa"). Besides finding typos, the flexibility of static sites to nimbly change how content is used and displayed makes them valuable tools for content management and exploration, even for projects with no intention of ever publishing a website. Templates for each page can be quickly edited to display different information; information in different formats, such as lists and paragraphs; and information sorted differently, such as by count or date.

Once satisfied with the draft website, we generated all the files required for the website from the Command Prompt with Hugo in a little over one second. We then uploaded these files to our institutionally provided server, LibDomains, using Core FTP and hosted the website.[9]

Proprietary Software

Because of institutional resources for server space, hosting, software licenses, and computers, the total cost to the grant project was $0. Although we achieved our goals at minimal direct financial cost, we were only partially successful in aligning *Recetas* with the principles of affordability and using open-source tools. Concerning the software used for *Recetas*, Hugo and Core FTP LE are free and open source, but Microsoft Excel and Oxygen XML Editor are proprietary and require purchase. Using open-source tools goes much further in supporting equal access to all DH practitioners. As we scale up recipe ingest, we are eliminating the proprietary Excel format from the pipeline in favor of comma-separated values (CSV). Instead, we plan to use Python to clean, structure, and translate the raw OCR text.

Unlike with Excel, we have no plans to remove Oxygen XML Editor from our pipeline. At this point, we are already familiar with Oxygen XML Editor, and using it frees up time and mental space to expand the collection, develop the website features, and explore the recipe data, all of which are higher priorities at this time. We considered switching to cheaper or open-source alternatives such as EditiX and Treeline, but did not seriously pursue these alternatives since neither was a simple enough substitution and it did not meaningfully support the growth of *Recetas*.[10] In this case, the goals and resources of our specific project outweighed our desire to use open-source software.

Oxygen XML Editor also bridges the gap between the static site generator and the developer efficiently, assuming they know one of the supported languages of the GUI.

9. University of Florida George A. Smathers Libraries, UFLib Domains, accessed April 14, 2024, https://domains.uflib.ufl.edu.

10. EditiX (https://www.editix.com) is open source but not free, yet it is much less expensive than Oxygen: one-time purchase of $19 for academic license binaries vs. Oxygen's $68 annual subscription. Treeline (https://treeline.bellz.org) is free and open source.

Oxygen XML Editor is a useful tool for managing directories and assembling project files in a more familiar way for inexpert developers. Hugo can also write new files, list directories, and move and copy files but relies on using the command line interface which often intimidates novice creators. In this regard, Oxygen's GUI is an asset. Directory navigation looks and functions similarly to Windows File Explorer, tools are available through standard drop-down menus, and a ribbon of icons helps users find top features quickly.

Oxygen's transformation scenarios were essential to *Recetas*. We created several XSLT style sheets and scenarios for generating the MD files and HTML partials in Spanish and English. These are available at the GitHub repository.[11] Oxygen includes an XSLT debugger, and the transformation scenarios were easier to use and understand than transforming in another program.

The GUI, however, is not equally accessible to all users. Oxygen's available out-of-the-box languages are limited to English (the default), German, French, Japanese, and Dutch. Spanish and other languages can be added, but doing so requires that users create an interface localization file (Oxygen XML Editor, n.d.). It is unfortunate this burden falls on users and is surprising that a world language such as Spanish is not automatically supported in the basic Oxygen package. The additional barrier of using software in another language or the time and energy required to adapt to another language should not be discounted. We hope that Oxygen adds Spanish as a built-in language in the future. Until then, it has lower value for practitioners in non-supported languages.

Conclusion

It is vital to invest time and energy into tools that can help make electronic publishing more accessible, especially to those publishing in languages other than English and outside of traditional or well-resourced institutions in the Global North. Static sites are a viable option for low-cost web publishing for DH projects along minimal computing principles. Unlike other popular static site generators, Hugo's built-in language handling decreases dependencies and complexity for multilingual websites. This helps lower barriers to multilingual scholarly publishing, potentially increasing participation by historically marginalized communities as publishers and readers.

Using Hugo, Excel, and Oxygen XML Editor to create the bilingual web project *Recetas de las Américas* was largely successful, but more work needs to be done to decrease reliance on proprietary software to maximize the potential for reuse, especially for under-resourced DH practitioners. Work is currently in progress to remove Excel

11. Sarah Tew, public-recetas, GitHub repository, https://github.com/SarahTew/public-recetas.

from the pipeline, but replacing Oxygen XML Editor's XSLT features and learning new software costs too much time, especially considering *Recetas* institutional support from the University of Florida Libraries.

Creating a simple static website with Hugo requires only a basic understanding of web technologies, but fully utilizing the generator to create a more complex web project requires intermediate skills. The most difficult part of using Hugo was the templating language; however, the flexibility of templates paired with Hugo's language support can potentially make multilingual websites easier to make than with other methods. Hugo's lack of documentation in languages other than English makes the software less accessible. As we continue to expand *Recetas*, we will continue to freely and openly share our methods, outcomes, and tutorials in Spanish and English.

References

Diaz, Chris. 2018. "Using Static Site Generators for Scholarly Publications and Open Educational Resources." *Code4Lib*, no. 42. https://journal.code4lib.org/articles/13861.

Dombrowski, Quinn. 2022. "Minimizing Computing Maximizes Labor." *Digital Humanities Quarterly* 16, no. 2. http://www.digitalhumanities.org/dhq/vol/16/2/000594/000594.html.

Gil, Alex, and Élika Ortega. 2016. "Global Outlooks in Digital Humanities: Multilingual Practices and Minimal Computing." In *Doing Digital Humanities: Practice, Training, Research*, edited by Constance Crompton, Richard Lane, and Ray Siemens. London: Routledge. https://doi.org/10.4324/9781315707860.

Hugo. n.d.-a. "Hugo Themes." Accessed April 15, 2024. https://themes.gohugo.io.

———. n.d.-b. "Multilingual Mode." Accessed April 15, 2024. https://gohugo.io/content-management/multilingual/.

Jerome, Melissa. 2022. "Recetas de las Américas/Recipes from the Americas." US Caribbean & Florida Digital Newspaper Project, October 5, 2022. https://ufndnp.domains.uflib.ufl.edu/recetas/.

Library of Congress. n.d. "Chronicling America." Accessed December 31, 2023. https://chroniclingamerica.loc.gov.

Lincoln, Matthew, Jennifer Isasi, Sarah Melton, and François Dominic Laramée. 2022. "Relocating Complexity: The Programming Historian and Multilingual Static Site Generation." *Digital Humanities Quarterly* 16, no. 2. http://www.digitalhumanities.org/dhq/vol/16/2/000585/000585.html.

NDNP (National Digital Newspaper Program). 2019. "About the Program." Library of Congress. Last modified March 14, 2019. https://www.loc.gov/ndnp/about.html.

———. 2023. "Technical Guidelines for Applicants." Library of Congress. Last modified July 25, 2023. https://www.loc.gov/ndnp/guidelines/NDNP_202426TechNotes.pdf.

Newson, Kaitlin. 2017. "Tools and Workflows for Collaborating on Static Website Projects." *Code4Lib,* no. 38. https://journal.code4lib.org/articles/12779.

Nilsson-Fernàndez, Pedro, and Quinn Dombrowski. 2022. "Multilingual Digital Humanities." In *The Bloomsbury Handbook to the Digital Humanities*, edited by James O'Sullivan, 83–92. London: Bloomsbury. https://www.bloomsbury.com/us/bloomsbury-handbook-to-the-digital-humanities-9781350232112/.

Oxygen XML Editor. n.d. "Localizing the User Interface." Accessed December 31, 2023. https://www.oxygenxml.com/doc/versions/26.0/ug-editor/topics/changing-user-interface-language.html.

Philibert, Régis. 2018. "Hugo Multilingual Part 2: Strings Localization." August 23, 2018. https://www.regisphilibert.com/blog/2018/08/hugo-multilingual-part-2-i18n-string-localization/.

———. 2019. "Hugo Multilingual Part 1: Content Translation." Last modified February 16, 2019. https://www.regisphilibert.com/blog/2018/08/hugo-multilingual-part-1-managing-content-translation/.

Risam, Roopika. 2018. *New Digital Worlds: Postcolonial Digital Humanities in Theory, Praxis, and Pedagogy*. Chicago: Northwestern University Press.

Risam, Roopika, and Alex Gil. 2022. "Introduction: The Questions of Minimal Computing." *Digital Humanities Quarterly* 16, no. 2. https://www.digitalhumanities.org/dhq/vol/16/2/000646/000646.html.

Simple IT Rocks. 2019. "Multilingual Jekyll without Plugins." Last modified March 27, 2019. https://simpleit.rocks/ruby/jekyll/tutorials/multilingual-jekyll-without-plugins/.

Tanasescu, Raluca. 2021. "Complexity and the Place of Translation in Digital Humanities." In *Exploring the Implications of Complexity Thinking for Translation Studies: Post-disciplinary Communities of Practice in the Translation Studies Network*, edited by Kobus Marais and Reine Meylaerts, 30–72. London: Routledge. https://doi.org/10.4324/9781003105114-3.

Tew, Sarah. 2023. "De una hoja de cálculo a un sitio web multilingüe utilizando el editor XML de Oxygen y el generador de sitios estáticos Hugo." https://sites.google.com/ufl.edu/dlf-2023-es/home. In English as "From Spreadsheet to Multilingual Website Using Oxygen XML Editor & Hugo Static Site Generator." https://sites.google.com/ufl.edu/dlf2023/home.

———. n.d. public-recetas. GitHub repository. https://github.com/SarahTew/public-recetas.

Tew, Sarah, and Melissa Jerome. n.d. *Recetas de las Américas*. https://recetas.domains.uflib.ufl.edu.

USCFDNP (US Caribbean & Florida Digital Newspaper Project). n.d. "About the Project." Accessed June 17, 2024. https://ufndnp.domains.uflib.ufl.edu/about/.

Varner, Stewart. "Minimal Computing in Libraries: Introduction." Minimal Computing, January 15, 2017. https://go-dh.github.io/mincomp/thoughts/2017/01/15/mincomp-libraries-intro/.