

Prototypes of Very High Fidelity, Accessibility, and Reusability with Laum

Francisco Oliveira¹, Rootedy Batista¹, Thiago Araujo¹, Breno Moraes¹,
Deivith Oliveira¹

¹FFIT – Serviços de Inovação e Tecnologia – Fortaleza – CE – Brazil

www.ffit.com.br

{Fran.oliveira,rootedy,Thiago.araujo,Breno.moraes}@ffit.com.br¹

Abstract. *This paper presents LAUM, a library of micro frontends (MFE) with very high usability and accessibility. LAUM's components are made available in design time via the Figma plugin. LAUM narrows the gap between designers and developers, making highly reusable components discoverable at design time. Accessibility is a given since it is built into all components. Early testing is assured by the plugin's HTML generator (accessible html + MFE calls).*

1. Context

Leading prototyping software companies help designers quickly build beautiful prototypes. However, the handover is no more than guidelines (usually poorly specified). Developers still have a lot of work ahead to implement what's been prototyped. When one looks closely into the design document (e.g., the internal structure of a Figma / Adobe prototype), one cannot recognize primitive widgets, like text fields, labels, etc. There are third-party code generator plugins. They scan the design document and generate HTML code. The quality of that code is debatable. Professional developers despise that code. More importantly, the prototype is not usable for early testing since users cannot appropriately interact with it.

We invested in a gallery of micro frontends (MFEs). The MFEs are live components, working software that can be reused in new applications. The approach has the advantage of offering discoverable and insertable components *into the prototype* during *participatory design sessions*. To make MFEs available in Figma, we developed a plugin and primitive components (buttons, text fields, etc.). The plugin generates high-fidelity HTML code to be used in early testing. Generated HTML makes calls to the MFEs the designer picked from the library. Coders and

designers collaborate to adjust our code generator to produce the best code possible. We named the ensemble (MFE gallery + Figma Plugin) *LAUM*.

All LAUM's components (MFEs and primitive widgets) are accessible for people with seeing and hearing disabilities. Generated HTML is also accessible, granted that only MFEs and primitive widgets are used in the prototype. MFEs have a life cycle so that future versions will accommodate other disabilities.

2. LAUM

2.1. LAUM – overview and impacts

All information in this section is available in detail and videos at <https://ffit.com.br/pt-br/blog/tag/laum>.

Reusability Figure 1 depicts a prototype (made in Figma) where a LAUM's MFE is included. In the leftmost panel, the "Consent" MFE, which is running on the URL: <https://mfe.ffit.com.br/mfe/consentimento>, is included in the prototype. For that, to configure the MFE call, the plugin receives the set of parameters, which are passed to the MFE during prototype execution. Such configuration makes reusing and inserting it in many different projects easier. LAUM's MFEs are responsive and can be used in mobile and web projects.



Figure 1- LAUM's MFEs -Running Code

Accessibility for people with hearing disability (HI). This particular MFE (Consent) requires a video (for sign language, the videoLibrasCode parameter. This is very handy for designers because, once they see that the MFE requires that parameter, accessibility for the deaf and hard of hearing (HI) has been taken care of. If another MFE does not require that parameter, it's because it is not necessary. One must remember that the HI are users accustomed to interacting with digital systems, so not all MFEs require special treatment.

Accessibility for people with seeing disability. All LAUM's MFEs score the highest grade on Google's lighthouse accessibility validator. They are readable by any screen reader and are implemented in the correct tab order. We make extended use of ARIA (Accessible Rich Internet Applications) labels, which are embedded into the MFEs. They provide additional information about the element's purpose and functionality to assistive technologies, such as screen readers.

Higher usability. More than two hundred usability items are checked before an MFE is made available in LAUM. Designers should get that right; forget about that micro-interaction. It's solved; focus on what's different and new in his/her project.

2.2. LAUM gallery itself

In Figure 2, one can see a particular LAUM gallery and one MFE (Consent) highlighted. When clicked, the buttons on the right-hand side display MFE's parameters, events, accessibility compliance, and an example of use. Designers and developers consult the gallery to insert an MFE into their design (in Figma) or write backend code.

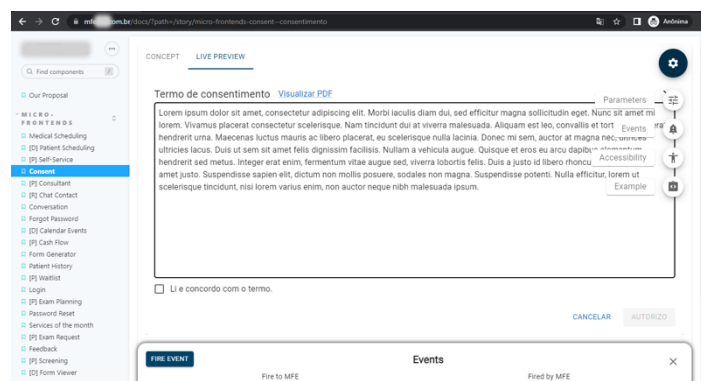


Figure 2 - The Gallery

2.3. Inserting an MFE into a Figma Project

Figure 3 shows how to insert a LAUM MFE into a Figma project. Once the designer has logged in, he/she clicks on the *Add MFE* button, and a panel will appear, requesting MFEs data for its configuration. In case of doubt, the designer can visit the gallery page for consultations. MFE configuration suffices for inserting it into the Figma project. The plugin can scan the current Figma to identify the other components (previously inserted by the designer). And that's of great help because it facilitates connecting the MFE with the rest of the design. Once fully configured, the designer has a preview of the MFE before finally inserting it into his/her Figma project.

Once inserted, a white box is displayed (similar to a text field) with the MFEs url written as text. Figma “play” button will not work for projects that use LAUM. The designer needs to export the whole project to see it running.

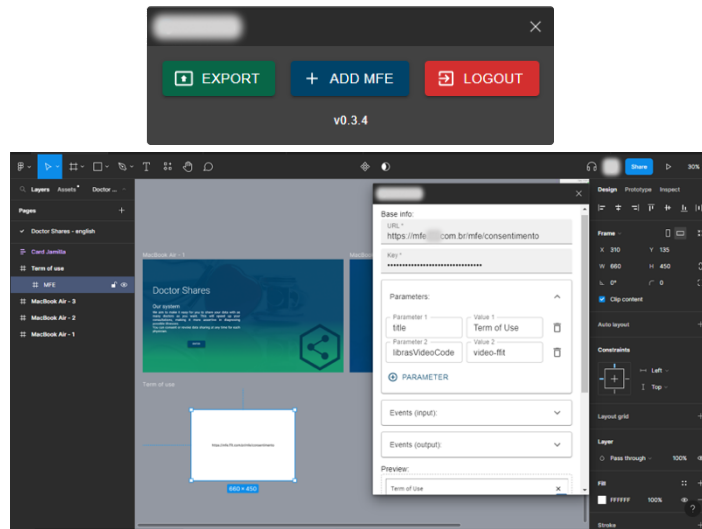


Figure 3 - Inserting an MFE into a Figma Project

2.4. LAUM's primitive Components

LAUM's primitive component sheet (Fig 4) is available for its users. When the designer uses a text input from the sheet, for example, the exported HTML will be a text input, not just a div that visually represents an input. That is very relevant because the final user (the one who will test the prototype) will have the actual interaction of being able to type in the input. There are several other components, such as buttons, checkbox radio buttons, and switches. It is worth mentioning that the primitive elements were produced based on the guidelines of the Material UI. Consequently, the exported components are from this design library.

2.5. Design System and HTML Generation

When the designer feels it is time to generate the html files for his/her project, he/she reruns the plugin and clicks the *Export* button.

Figma plugin generates. For each Figma frame, an HTML and a CSS file are created (Fig.5). Now, all the designers must upload the files to a server and share the URL (of the first frame) with the users who will test the prototype.

Every LAUM user can configure his/her design system. These definitions will accompany all prototypes and MFEs fired by that user.

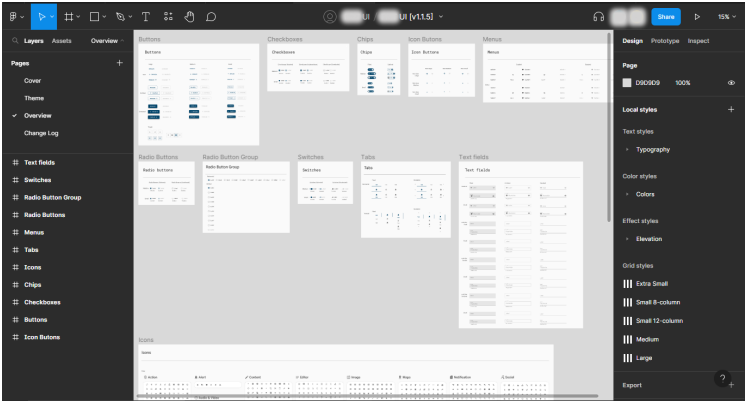


Figure 4 – LAUM’s primitive components

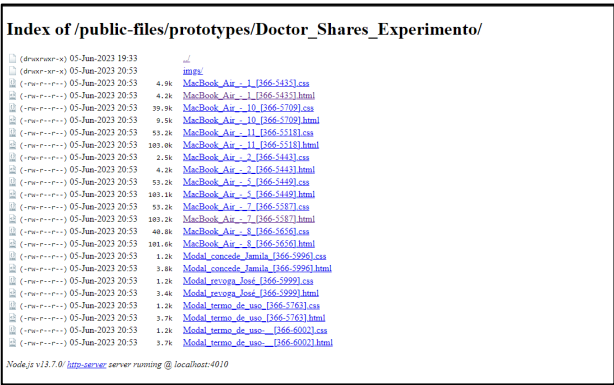


Figure 5 - The generated files