



Curriculum web

Diciembre de 2024

Christian Amauri Amador Ortega

ÍNDICE

1. Front-end	2
1.1. Sidebar	2
1.2. Content	3
1.2.1. Void container	3
1.2.2. Card container	4
1.2.3. Exp container	6
1.2.4. Form container	8
1.2.5. Perf container	10
1.3. Multimedia	13
1.3.1. Imágenes	13
1.3.2. Audios	13
2. Back-end	14
2.1. Base de datos	14
2.1.1. Tabla p	14
2.1.2. Tablas e, e_logs y e_resp	15
2.1.3. Tabla f	15
2.1.4. Tablas hab_con y comps	16
2.2. Conexión con la base de datos	16
3. Versiones anteriores	16
3.1. Primeros diseños	16
3.2. HTML puro	17
4. Hosting	18
4.1. GitHub Pages	18
5. Trabajo a futuro	20
6. Referencias	20

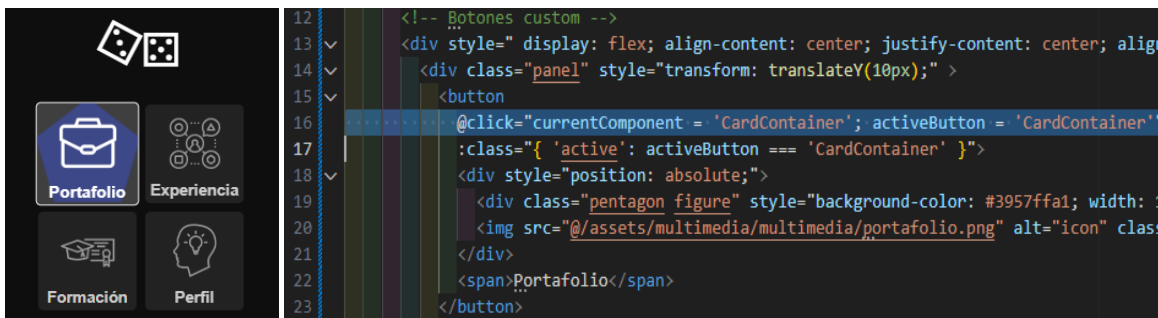
Para este Proyecto se utilizó Visual Studio Code 1.96.2.0

1. Front-end

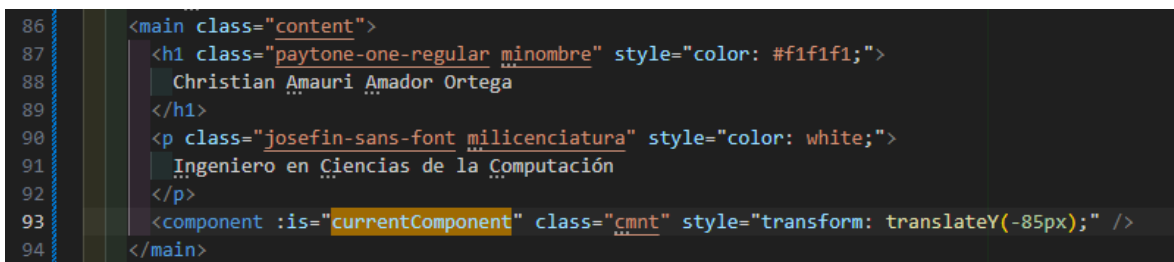
El front-end está dividido en dos componentes principales dentro del componente Vue principal (App.vue): **sidebar** y **content**.

1.1. Sidebar (donde están los botones y la sección de contacto, con vínculos en sus respectivos íconos hacia los contactos indicados, y el panel con los botones a seleccionar).

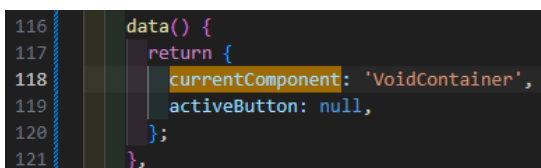
Para mantener el estado de “seleccionado” de cada contenedor tanto en los botones como en el contenedor como tal, se usó la siguiente lógica: Cada **button** dentro de **panel** cambia los valores de las variables “**currentComponent**” (para mostrar su contenedor correspondiente) y “**activeButton**” (para resaltar el botón seleccionado) mediante la directiva **@click**.



Así, en el cuerpo principal se renderiza el contenedor seleccionado (**currentComponent**):



Por default, estos valores son no corresponden a ningún elemento:



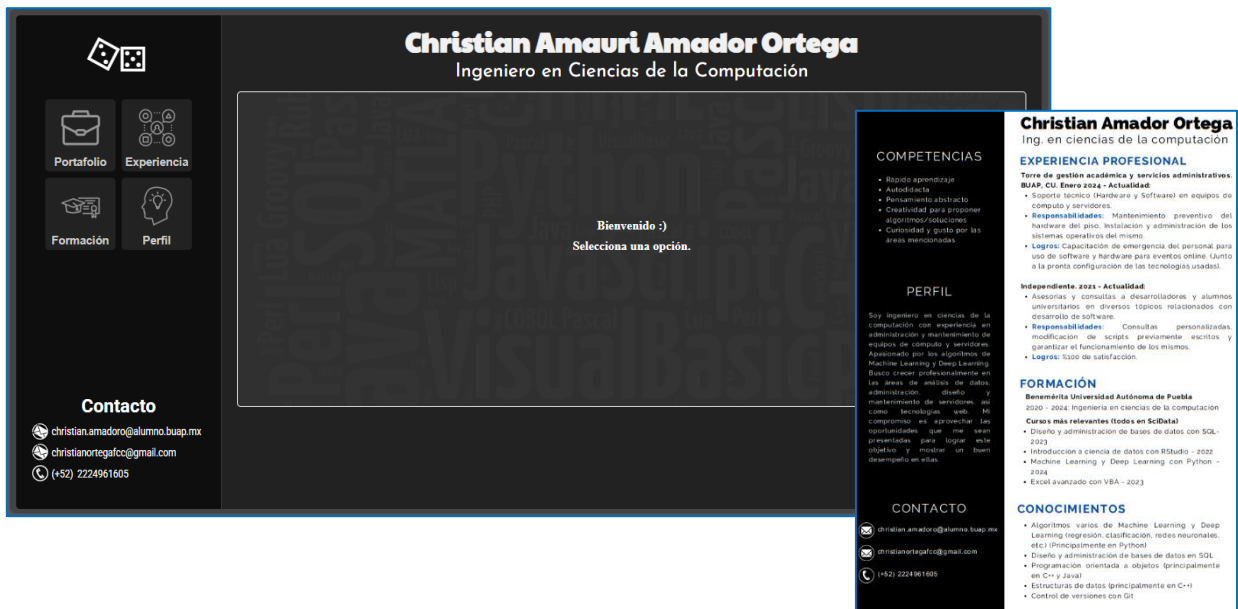
Para cambiar el valor del componente renderizado, basta con cambiar la variable **currentComponent** en la directiva indicada en **button**. Pero para mantener seleccionado un botón, tenemos que definir el siguiente método:

```
122     methods: {
123         setActive(button) {
124             this.activeButton = button;
125         },
126     }
```

1.2 Content. (donde está mi nombre y los cinco contenedores: Void, Portafolio, Experiencia, Formación y Perfil) tanto los contenedores como los elementos que usan están diseñados individualmente en su respectivo archivo con extensión **.vue**, los detalles son los siguientes:

1.2.1. Void container

El propósito de este contenedor es provocar que el usuario haga clic en un botón, y así activar la funcionalidad de audio sin errores, además de que también dé un vistazo a las cuatro opciones y al contacto. El front-end fue diseñado con el objetivo de que se pareciera a mi CV original (el de formato pdf) pero con categorías separadas.



El resto de los contenedores toman sus respectivos elementos de la base de datos.

1.2.2. Card container



La primera parte del **template** está dedicada a mostrar la animación de carga, mientras la consulta SQL no haya respondido.

```
1 <template>
2   <div class="card-container">
3     <!-- Mostrar gif de carga mientras isLoading sea true -->
4     <div v-if="isLoading" class="loading">
5       <div class="dice-container-loading">
6         <div class="spin-dice-loading">🎲</div>
7         <div style="transform: translateY(6px);">🎲</div>
8       </div>
9     </div>
```

la segunda imprime de golpe todas las **Card** (Card.vue) que obtenga como resultado de la consulta, usando la estructura de control **v-for**

```
11 <Card
12   v-for="item in data"
13   :key="item.id"
14   :nombre="item.nombre"
15   :descripcion="item.descripcion"
16   :imagen="item.img"
17   :url_gh="item.url_gh"
18 />
19 </div>
20 </template>
```

En la sección de **script**, necesitaremos importar nuestro elemento **Card**, **onMounted** (para solicitudes de datos o configuraciones, en este caso, lo primero), **ref** (para realizar cambios dinámicamente en el DOM) y **Supabase** (para realizar la consulta SQL).

```
22 <script>
23   import Card from "@/components/Card.vue";
24   import { onMounted, ref } from "vue";
25   import supabase from "C:/Users/amado/Desktop/Portafolios/front_3/welcome/src/supabase.js";
26
```

Para usar Supabase, tenemos que instalar el cliente de Supabase en nuestro proyecto vue, para ello introducimos el siguiente comando en la terminal de visual studio code:

```
npm install @supabase/supabase-js
```

Luego, escribimos la función asincrónica **fetchData**, en donde se hará la consulta SQL a nuestro servidor Supabase. Hacemos reactivas las variables '**data**' e '**isLoading**', mediante la función **ref** de Vue. En '**data**' se guardarán los datos de la consulta, en '**isLoading**' (true o false) se guarda el estado de carga de los datos de la consulta (true = cargando, mostrar animación || false = carga terminada, detener animación)

La parte más importante del script es la consulta:

```
let { data: rows, error } = await supabase.from("p").select("*").order("relevancia", { ascending: false });
```

Que en lenguaje SQL expresa: **SELECT * FROM p ORDER BY relevancia DESC;** (las de mayor relevancia se muestran primero). Se usa la palabra reservada "**await**" antes de la consulta, para que se suspenda la ejecución del resto del script hasta que Supabase responda. Finalmente, asignamos los datos de la consulta a la variable **rows** (desestructurando el objeto data que regresa la consulta), y dejamos el valor de **error** tal como está, para capturar cualquier posible error en la consulta.

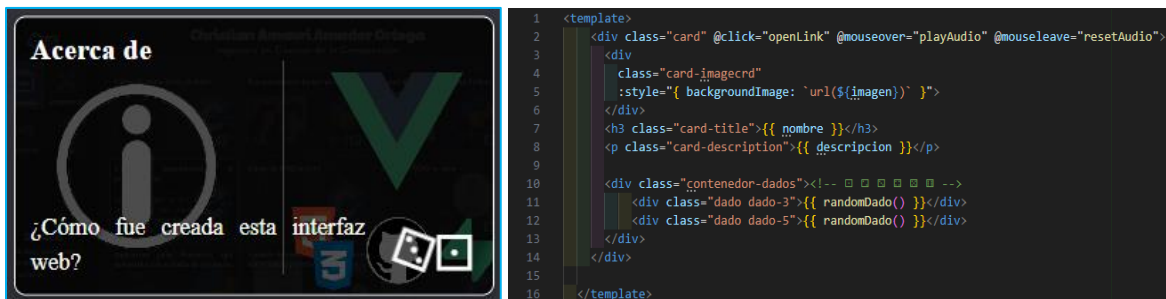
(y como parámetro de la función "**onMounted**", tenemos que mandar la función "**fetchData**" para ejecutar **fetchData** cuando el componente (**CardContainer**) sea montado

```
27 export default {
28   name: "CardContainer",
29   components: { Card },
30   setup() {
31     const data = ref([]);
32     const isLoading = ref(true);
33
34     const fetchData = async () => {
35       // ORDENADOS POR RELEVANCIA
36       let { data: rows, error } = await supabase.from("p").select("*").order("relevancia", { ascending: false });
37       if (error) {
38         console.error("Error fetching data:", error);
39       } else {
40         console.log("Data fetched from Supabase:", rows);
41         data.value = rows;
42       }
43       isLoading.value = false;
44     };
45
46     onMounted(fetchData);
47
48     return { data, isLoading };
49   },
50 };
51 </script>
```

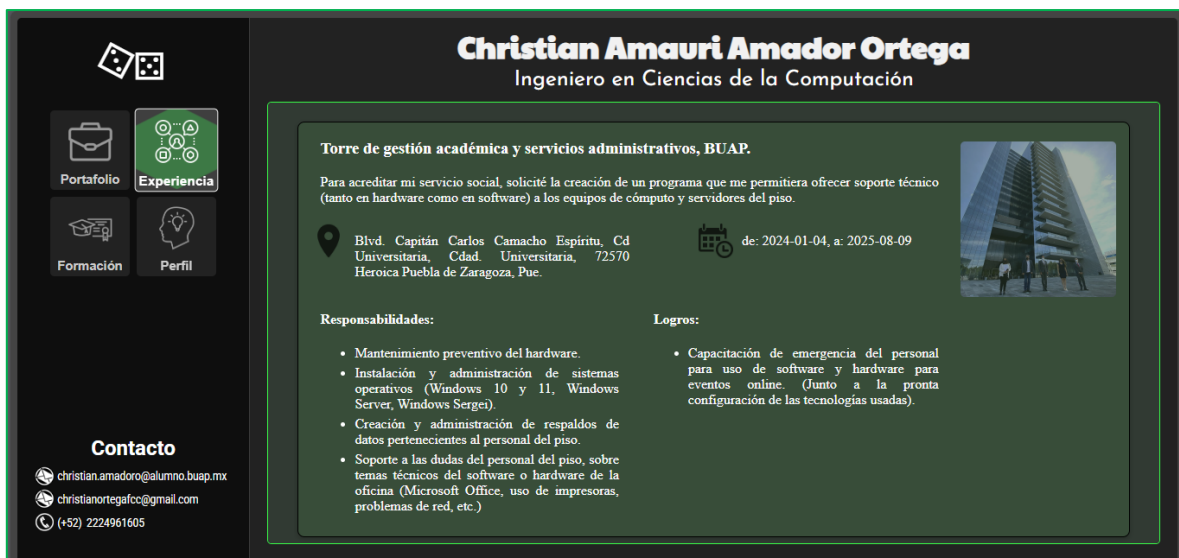
Entonces el flujo de consulta y renderizado de **CardContainer** es el siguiente:

1. **CardContainer** hace la consulta SQL a Supabase con `fetchData` (en el script).
2. **CardContainer** usa `v-for` (en el template) para crear las tarjetas (instancias de **Card**). (en este punto **CardContainer** ya tiene datos suficientes para saber cuántas **Card** va a haber, y cuáles son sus datos)
3. Las tarjetas (**Card**) se renderizan en la interfaz, dentro de **CardContainer** con los datos obtenidos.

En general, ese flujo funciona igual en los cuatro tipos de contenedores, aunque obviamente la estructura de los datos, y los datos como tal son diferentes en cada uno. Los elementos que ocupa **CardContainer** son las **Card**:



1.2.3. Exp container



Del mismo modo que CardContainer, la primera parte del template en **ExpContainer** está dedicada a la animación de carga, y la segunda a los elementos como tal.

```
1 <!-- CardContainer.vue -->
2 <template>
3   <div class="Exp-container">
4
5     <!-- Mostrar gif de carga mientras isLoading sea true -->
6     <div v-if="isLoading" class="loading">
7       <div class="dice-container-loading">
8         <div class="spin-dice-loading">❏</div>
9         <div style="transform: translateY(6px);">❏</div>
10      </div>
11    </div>
12
13    <ExpCard
14      v-for="item in data"
15      :key="item.id"
16      :id="item.id"
17      :nombre="item.nombre"
18      :descripcion="item.descripcion"
19      :ubicacion="item.ubicacion"
20      :url_ubicacion="item.url_ubicacion"
21      :fecha_inicio="item.fecha_inicio"
22      :fecha_fin="item.fecha_fin"
23      :relevancia="item.relevancia"
24      :url_web="item.url_web"
25      :img="item.img"
26      :resps="item.resps"
27      :logs="item.logs"
28    />
29  </div>
30 </template>
```

La consulta SQL funciona igual, solo que el criterio de ordenamiento de esta es por fecha (las más recientes primero)

```
46 // ORDENADOS POR EL MÁS RECIENTE
47 let { data: eData, error: eError } = await supabase.from("e").select("*").order("fecha_inicio", { ascending: false });
48
```


Además de hacer las consultas a sus tablas foráneas correspondientes: (**logs** para logros y **resp** para responsabilidades):

```
55 // Para cada elemento en 'e', traer sus relaciones de 'e_resp' y 'e_logs'
56 for (let item of eData) {
57   // Traer responsabilidades
58   let { data: resps } = await supabase
59     .from("e_resp")
60     .select("*")
61     .eq("id_exp", item.id);
62
63   // Traer logros
64   let { data: logs } = await supabase
65     .from("e_logs")
66     .select("*")
67     .eq("id_exp", item.id);
68
69   // Agregar las relaciones al elemento actual
70   item.resps = resps || [];
71   item.logs = logs || [];
72 }
```



Los elementos de **ExpContainer**, son los “**Exp**”:

Torre de gestión académica y servicios administrativos, BUAP.

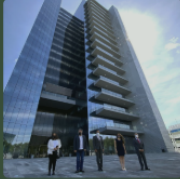
Para acreditar mi servicio social, solicité la creación de un programa que me permitiera ofrecer soporte técnico (tanto en hardware como en software) a los equipos de cómputo y servidores del piso.



Bldv. Capitán Carlos Camacho Espiritu, Cd Universitaria, Cdad. Universitaria, 72570 Heroica Puebla de Zaragoza, Pue.



de: 2024-01-04, a: 2025-08-09




Responsabilidades:

- Mantenimiento preventivo del hardware.
- Instalación y administración de sistemas operativos (Windows 10 y 11, Windows Server, Windows Sergei).
- Creación y administración de respaldos de datos pertenecientes al personal del piso.
- Soporte a las dudas del personal del piso, sobre temas técnicos del software o hardware de la oficina (Microsoft Office, uso de impresoras, problemas de red, etc.)

Logros:

- Capacitación de emergencia del personal para uso de software y hardware para eventos online. (Junto a la pronta configuración de las tecnologías usadas).

1.2.4. Form container



Portafolio

Experiencia

Formación


Perfil

Christian Amauri Amador Ortega


Ingeniero en Ciencias de la Computación

Ingeniería en Ciencias de la Computación - Benemérita Universidad Autónoma de Puebla

Ingeniería de cincuenta materias que incluyen desde las bases elementales de la computación, como metodología de la programación, circuitos eléctricos, ensamblador y diseño digital, hasta tecnologías prácticas como bases de datos, diseño web, web semántica, Deep Learning, entre otras. Tanto la BUAP como la Facultad de Ciencias de la Computación han sido reconocidas nacional e internacionalmente por su aprovechamiento académico y el talento de sus egresados.




de: 2020-01-06, a: 2024-12-05




Diseño y administración de bases de datos con SQL - SciData

Curso en línea, brindado por un administrador de bases de datos con experiencia, enfocado en los aspectos teóricos del diseño de bases de datos, llevados a la práctica con PostgreSQL.




de: 2023-07-03, a: 2023-08-04





Introducción a ciencia de datos con RStudio - SciData


Taller de una sesión que introduce al alumno a los principios de las ciencias de datos usando RStudio para los aspectos prácticos



Contacto

 christian.amador@alumno.buap.mx

 christianortegafcc@gmail.com

 (+52) 224961605

Misma lógica inicial de carga...

```
1 <template>
2   <div class="form-container">
3
4     <!-- Mostrar gif de carga mientras isLoading sea true -->
5     <div v-if="isLoading" class="loading">
6       <div class="dice-container-loading">
7         <div class="spin-dice-loading">G</div>
8         <div style="transform: translateV(6px);">B</div>
9       </div>
10    </div>
11
12    <FormCard
13      v-for="item in data"
14      :key="item.id"
15      :nombre="item.nombre"
16      :descripcion="item.descripcion"
17      :fecha_inicio="item.fecha_inicio"
18      :fecha_fin="item.fecha_fin"
19      :relevancia="item.relevancia"
20      :img="item.img"
21      :urlInfo="item.url_info"
22    />
23  </div>
24 </template>
```

Consulta con criterio de ordenamiento por relevancia (relevancia más elevada primero):

```
38 const fetchData = async () => {
39   // ORDENADOS POR RELEVANCIA
40   let { data: rows, error } = await supabase.from("f").select("*").order("relevancia", { ascending: false });
41 }
```

Los elementos de **FormContainer** son los “**Form**”:

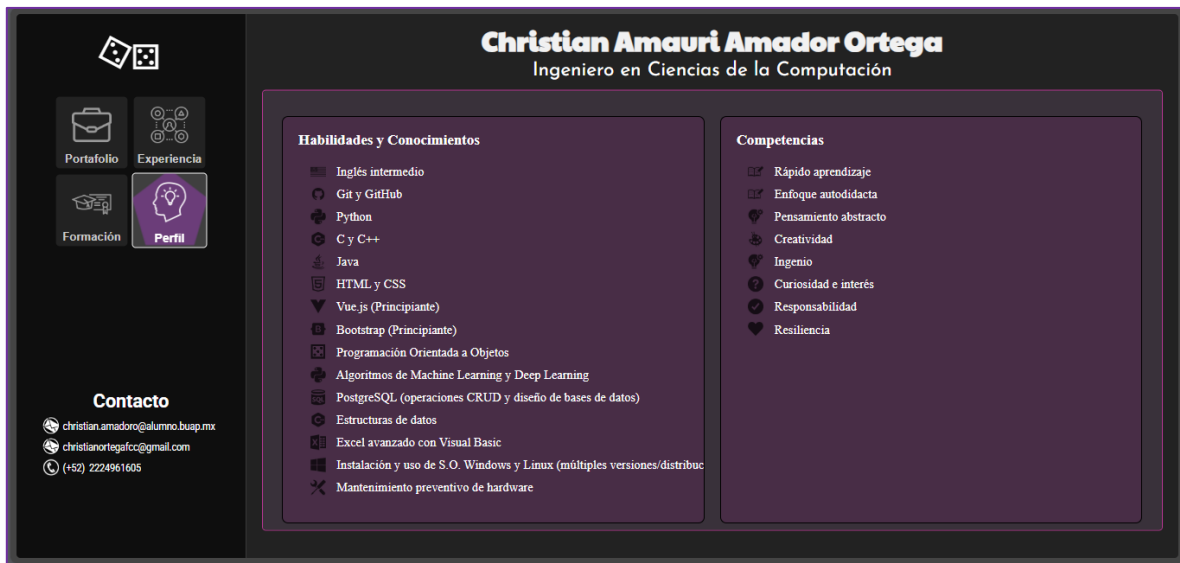
Ingeniería en Ciencias de la Computación - Benemérita Universidad Autónoma de Puebla

Ingeniería de cincuenta materias que incluyen desde las bases elementales de la computación, como metodología de la programación, circuitos eléctricos, ensamblador y diseño digital, hasta tecnologías prácticas como bases de datos, diseño web, web semántica, Deep Learning, entre otras. Tanto la BUAP como la Facultad de Ciencias de la Computación han sido reconocidas nacional e internacionalmente por su aprovechamiento académico y el talento de sus egresados.

 de: 2020-01-06, a: 2024-12-05



1.2.5. Perf container



En **PerfContainer**, tenemos dos bloques `<div>` que requerirán consulta SQL, “Habilidades y Conocimientos”, y “Competencias” Funcionan igual que el resto de los elementos de los otros contenedores.

```

3     <div class="perf-container">
4         <!-- Contenedor para "I: hab_con" y "II: comps" -->
5         <div class="perf-columns-container">
6             <!-- I -->
7             <div class="perf-column" @mouseover="playAudio" @mouseleave="resetAudio">
8                 <h3 style="margin-bottom: -15px;">Habilidades y Conocimientos</h3>
9                 <!-- Mostrar gif de carga mientras isLoading sea true -->
10                <div v-if="isLoading1" class="loading" style="transform: translateX(110px); border-radius: 5px; font-weight: 200px;">
11                    <div class="dice-container-loading">
12                        <div class="spin-dice-loading" style="color: white;">❏</div>
13                        <div style="transform: translateY(6px); color: white;">❏</div>
14                    </div>
15                </div>
16                <div style="height: 25px; white-space: nowrap;" v-for="item in habConData" :key="item.id">
17                    <PerfCard :descripcion="item.descripcion" :img="item.img" />
18                </div>
19            </div>
20        </div>
21
22        <!-- II -->
23        <div class="perf-column" @mouseover="playAudio" @mouseleave="resetAudio">
24            <h3 style="margin-bottom: -15px;">Competencias</h3>
25            <!-- Mostrar gif de carga mientras isLoading sea true -->
26            <div v-if="isLoading2" class="loading" style="transform: translateX(100px); border-radius: 5px; font-weight: 180px;">
27                <div class="dice-container-loading">
28                    <div class="spin-dice-loading" style="color: white;">❏</div>
29                    <div style="transform: translateY(6px); color: white;">❏</div>
30                </div>
31            </div>
32            <div style="height: 25px; white-space: nowrap;" v-for="item in compsData" :key="item.id">
33                <PerfCard :descripcion="item.descripcion" :img="item.img" />
34            </div>
35        </div>

```

divs de las columnas que harán consulta a las tablas hab_con y comps de la base de datos

Y finalmente tenemos otros bloques `<div>` que no requieren acceso a la base de datos, que tienen la información que necesitan en su mismo espacio.

```

37     <!-- Otros divs no vinculados a la base de datos -->
38     <div class="extra-content" style="background-color: #170020c0;" @mouseover="playAudio" @mouseleave="resetAudio">
39         <div class="flex-container-perf">
40             <h3 style="letter-spacing: 10px;">PERFIL</h3>
41             
42         </div>
43         <p>
44             Mis mayores fortalezas son la creatividad y el ingenio, mi principal valor es el compromiso.
45             En el ámbito profesional soy una persona cautelosa y metódica. Dedico tiempo a comprender a fondo los
46             problemas y proponer soluciones eficaces. Cuando realizo trabajo de investigación,
47             mi prioridad es simplificar cuestiones que puedan ser complicadas. Aunque prefiero tareas que ya manejo
48             con soltura y cuya ejecución es principalmente rutinaria, aunque sea un poco más
49             monótono, me considero más eficiente en esas tareas. Aprecio la comunicación efectiva y transparente, y
50             reconozco la importancia del trabajo en equipo.
51         </p>
52     </div>
53
54     <div class="extra-content extra-content-b" style="background-color: #ffffff13;" @mouseover="playAudio" @mouseleave="resetAudio">
55         <h3 style="align-self: center;">Valores y rasgos dominantes</h3>
56     </div>

```

Perfil (texto)

```

55 <div class="extra-content extra-content-b" style="background-color: #ffffff13;" @mouseover="playAudio" @m
56 <h3 style = "align-self: center;">Valores y rasgos dominantes</h3>
57 <p style="align-self: center;">
58 (Resultados de una prueba psicométrica obtenidos a través de una evaluación en Computrabajo).
59 </p>
60
61 <div class="perfimgs">
62 <div class="imgbg-div">
63 
66 </div>
67 
68 
69 
70 
71 
72 
73 </div>
74 </div>
75 </div>
76

```

Perfil (imágenes)

P E R F I L

Mis mayores fortalezas son la creatividad y el ingenio, mi principal valor es el compromiso. En el ámbito profesional soy una persona cautelosa y metódica. Dedico tiempo a comprender a fondo los problemas y proponer soluciones eficaces. Cuando realizo trabajo de investigación, mi prioridad es simplificar cuestiones que puedan ser complicadas. Aunque prefiero tareas que ya manejo con soltura y cuya ejecución es principalmente rutinaria, aunque sea un poco más monótono, me considero más eficiente en esas tareas. Aprecio la comunicación efectiva y transparente, y reconozco la importancia del trabajo en equipo.

Valores y rasgos dominantes

(Resultados de una prueba psicométrica obtenidos a través de una evaluación en Computrabajo).

Tus valores



Respeto

Suelen tener consideración hacia los demás, valorando sus opiniones y diferencias. Por lo general, actúan con cortesía y empatía, mostrando un trato amable y comprensivo en la mayoría de las interacciones. Aunque en ocasiones te cueste aceptar otras opiniones, sueles demostrar tolerancia y aceptación hacia las diversas perspectivas y culturas, promoviendo un ambiente inclusivo y armonioso.



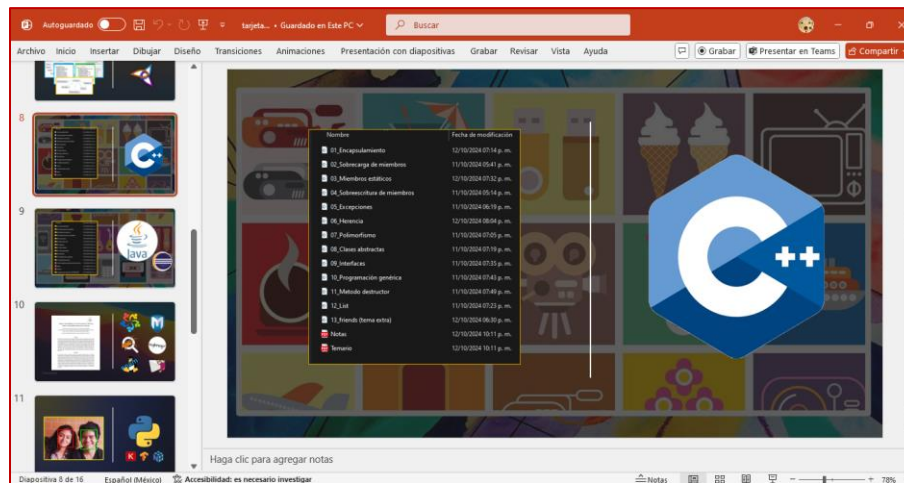
Responsabilidad

Demuestras una responsabilidad excepcional hacia tus obligaciones y tareas asignadas. Te caracterizas por tu confiabilidad y capacidad para cumplir con tus compromisos. Asumes responsabilidades con seriedad y muestras iniciativa para abordar desafíos. Tu ética laboral y personal refleja un compromiso constante con la excelencia y el cumplimiento de deberes de manera diligente.

1.3. Multimedia

1.3.1. Imágenes

Las imágenes de las tarjetas de portafolio son editadas desde Powerpoint (o cualquier sencillo programa de edición de imagen) luego son convertidas a pdf (también desde Powerpoint), y luego a png desde fuentes externas (por ejemplo) [\[1\] iloveimg.com](https://www.iloveimg.com):



Los íconos para diversos elementos como pines o botones son descargados desde páginas de íconos gratuitos, como por ejemplo [\[2\] icon-icons.com](https://www.icon-icons.com)

1.3.2. Audios

El efecto de audio usado en los elementos de los contenedores fue descargado en algún sitio de audios gratis sin copyright, por ejemplo [\[3\] pixabay.com](https://pixabay.com) o [\[4\] freesound.org](https://freesound.org). Y para usarlos en Vue, hay que añadir un método de **playAudio**, **stopAudio** y **resetAudio** en la sección de script, y referenciarlos en el elemento que deseamos que suene:

```
37 playAudio() {
38   if (!this.isPlaying) {
39     this.audio.play();
40     this.isPlaying = true;
41   }
42 },
43 stopAudio() {
44   this.audio.pause();
45   this.audio.currentTime = 0;
46 },
47 resetAudio() {
48   this.isPlaying = false;
49 },
```

```
1 <template>
2   <div class="card" @click="openLink" @mouseover="playAudio" @mouseleave="resetAudio">
3     <div
```

2. Back-end

2.1. Base de datos

La base de datos consta de siete tablas ('**p**' para los datos de las 'Card' del portafolio, '**e**' para los datos de las 'Exp' de la sección experiencia, '**e_logs**' para los logros en una 'Exp', '**e_resp**' para las responsabilidades en una 'Exp', '**f**' para los datos de las 'Form' en la sección de formación, '**hab_con**' para las habilidades/conocimientos y '**comps**' para las competencias en la sección de perfil) (no todos los atributos de las tablas son visibles en el front-end, algunos son solo para ordenar los datos)

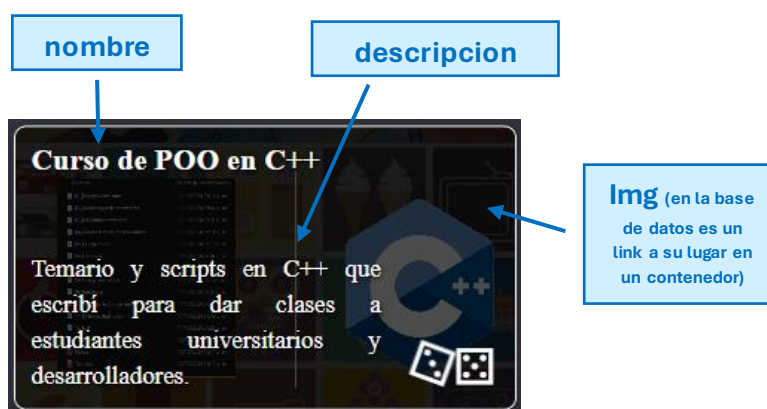
Para Alojarse la base de datos, usé **Supabase**, que tiene un plan gratuito de 2GB de almacenamiento, y transferencia de 5GB mensuales, que es apropiado para este proyecto. Además de ser compatible con Vue, tener una interfaz muy cómoda y ser muy fácil de usar.

Table	Columns	Data Types
p	id, nombre, descripcion, fecha, relevancia, img, url_gh	int8, varchar, text, date, float4, text
e_logs	id, id_exp, descripcion	int8, int8, text
e	id, nombre, descripcion, ubicacion, url_ubicacion, fecha_inicio, fecha_fin, relevancia, url_web, img	int8, varchar, text, text, text, date, date, float4, text, text
f	id, nombre, descripcion, fecha_inicio, fecha_fin, relevancia, img, url_info	int8, varchar, text, date, date, float4, text, text
hab_con	id, descripcion, img	int8, text, text
e_resp	id, id_exp, descripcion	int8, int8, text
comps	id, descripcion, img	int8, text, text

2.1.1. Tabla p

Column	Data Type	Visibility
id	int8	No visible
nombre	varchar	Visible
descripcion	text	Visible
fecha	date	No visible
relevancia	float4	No visible
img	text	Visible
url_gh	text	Visible

Al hacer clic sobre la Card



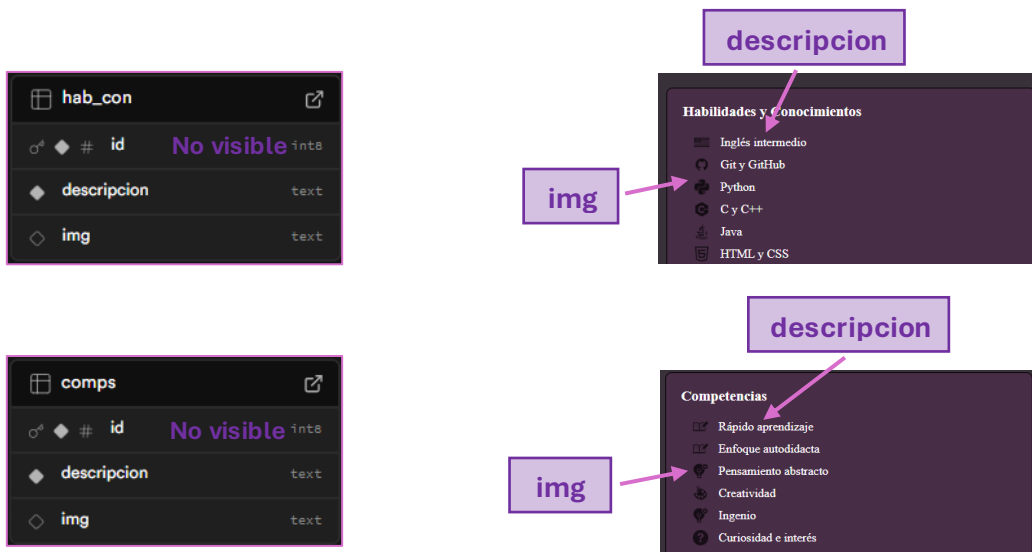
2.1.2. Tablas e, e_logs y e_resp



2.1.3. Tabla f



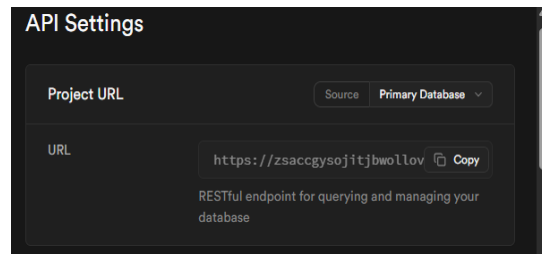
2.1.4. Tablas hab_con y comps



2.2. Conexión con la base de datos

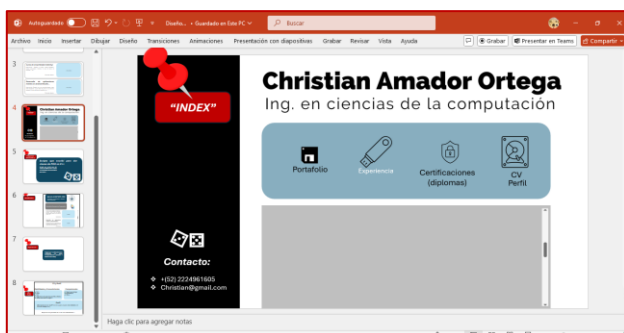
Para que nuestras consultas funcionen, tenemos que crear un archivo de JS en el directorio donde se encuentra nuestro App.vue, y usar el URL y el anon key que nos proporciona Supabase en nuestro lado de servidor (**Project settings** -> **API settings**), para crear un cliente de Supabase, de esta forma:

```
1 import { createClient } from '@supabase/supabase-js';
2
3 const supabaseUrl = 'https://zsaccgysojitjbwollov.supabase
4 const supabaseKey = 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
5
6 const supabase = createClient(supabaseUrl, supabaseKey);
7
8 export default supabase;
```

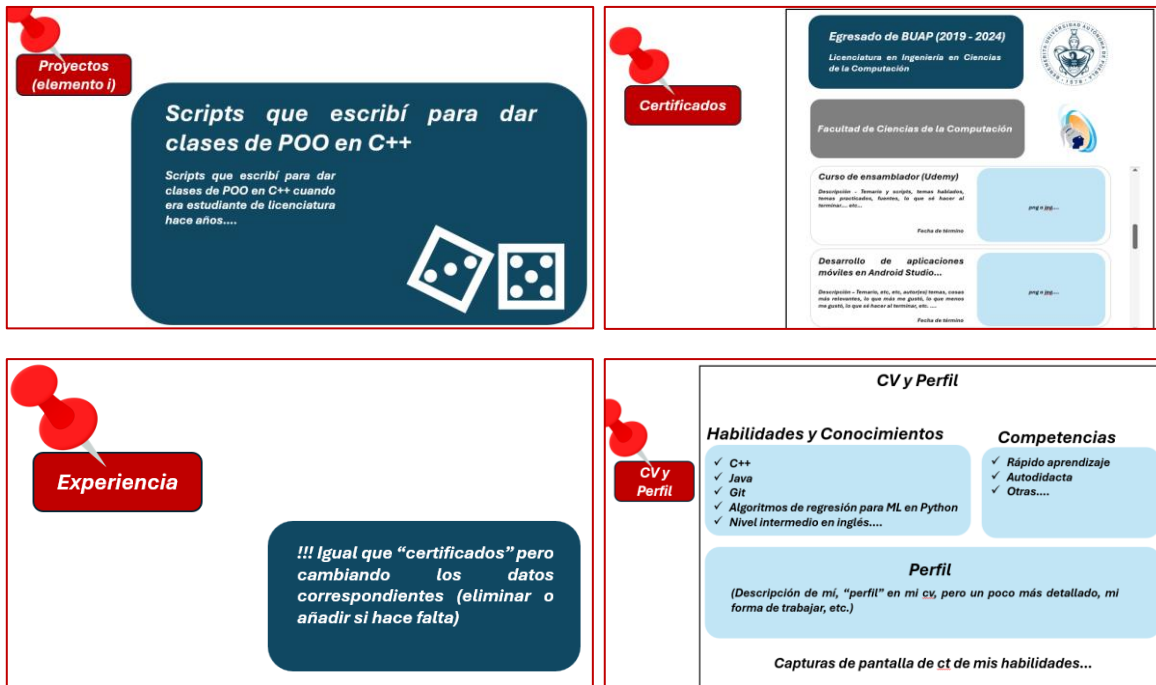


3. Versiones anteriores

3.1. Primeros diseños:



Los diseños conceptuales del front-end fueron hechos en diapositivas de Powerpoint:



3.2. HTML puro:

También consideré escribir todo el proyecto en HTML, CSS y JS puro, para demostrar una mayor comprensión del funcionamiento interno de estas tecnologías, pero estaba resultando muy tardado tanto en codificación como documentación, además de causar menor impacto visual:





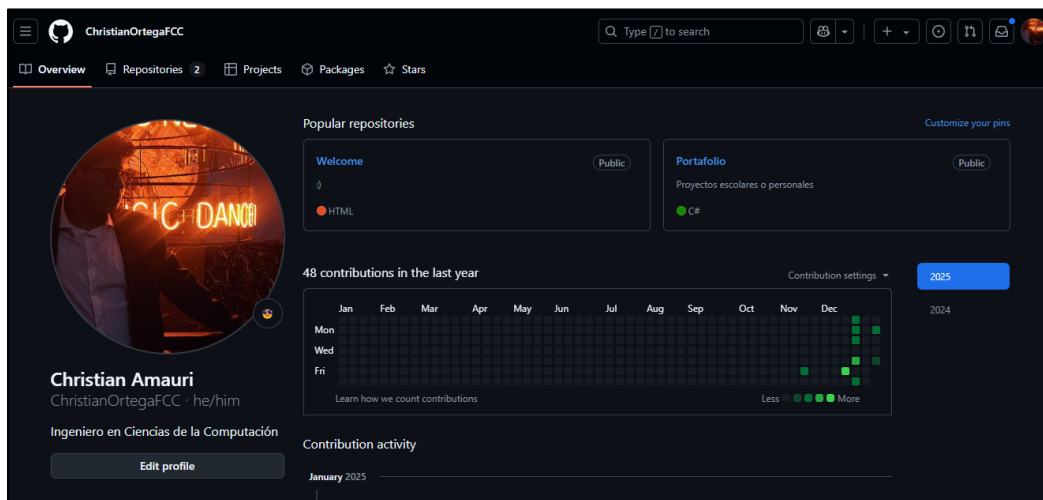
Bocetos HTML iniciales

4. Hosting

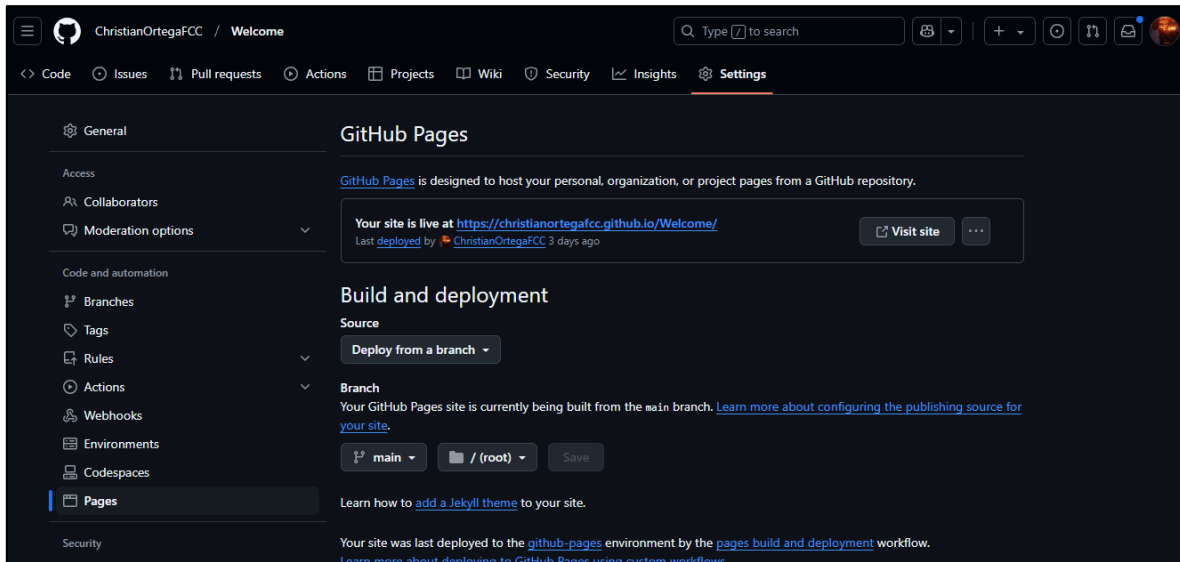
Para hacer disponible este CV por la web consideré usar un servidor personal, configurado manualmente, pero dedicar una máquina física a servir este proyecto era demasiado ambicioso, costoso y poco práctico, además de que no tengo dominio propio y no lo planeo tener por un tiempo. Así que usé GitHub Pages.

4.1. Github Pages

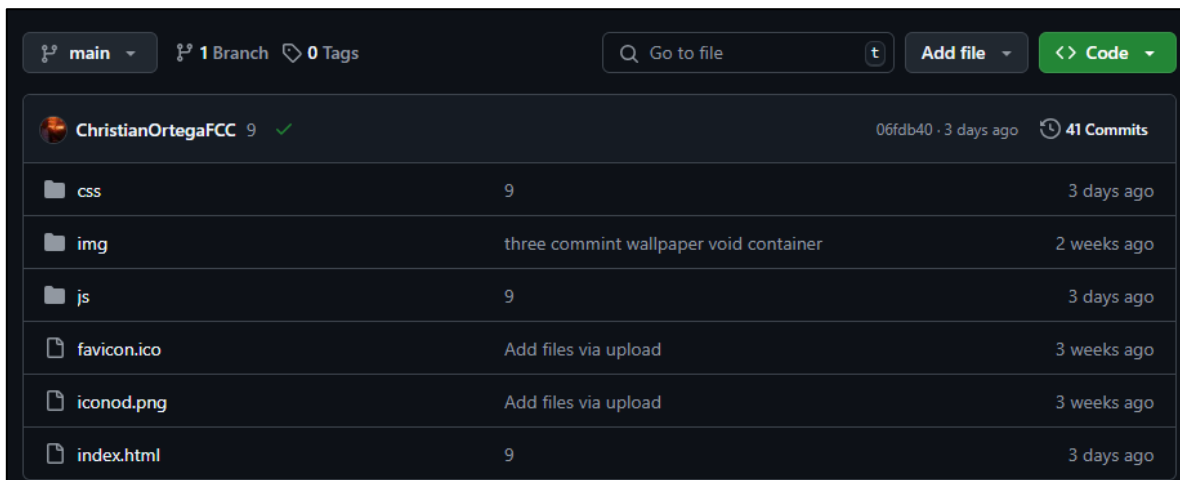
Hice dos repositorios, uno para depositar los proyectos del portafolio, y uno para alojar el front-end.



Para que un proyecto de Vue.js funcione en GitHub Pages, hay que ir a la configuración del repositorio en donde vamos a guardar los scripts, seleccionar la rama main, directorio root, y de ser necesario, las configuraciones adicionales que se requieran.



Con todo esto listo, simplemente hay que depositar en el repositorio de “host” todos los elementos que contiene la carpeta dist de nuestro proyecto (incluso las carpetas). (para obtener la carpeta dist de nuestro proyecto vue, tenemos que escribir en la terminal de visual studio code, el comando “npm run build”).



5. Trabajo a futuro

- I. Añadir las opciones de “ordenar por: [relevancia, fecha, nombre] x [ascendente, descendente]” para portafolio, experiencia y formación (perfil no la necesita) aunque eso podría hacer que el proyecto sea muy rebuscado.
- II. Resaltar de forma más visible los elementos más importantes, de algún modo. Quizá implementando un valor especial en las tablas de la base de datos, o quizás simplemente añadir manualmente los elementos usando HTML en sus documentos con extensión .vue correspondientes.
- III. Revisar si vale la pena implementar medidas de seguridad más robustas en la base de datos. ¿Alguien estaría dispuesto a vulnerarla? ¿vale la pena modificar algo?

6. Referencias

- [1] *iLovePDF. (s.f.). iLoveIMG. iLoveIMG | The Fastest Free Web App for Easy Image Modification. Recuperado en diciembre de 2024, de <http://www.iloveimg.com>*
- [2] *Previ, C. (s.f.). Free icons PNG, ICO, ICNS and SVG. Icon-icons.com. Recuperado en diciembre de 2024, de <https://icon-icons.com/>*
- [3] *(S.f.). Pixabay.com. Recuperado en diciembre de 2024, de <http://pixabay.com>*
- [4] *(S.f.). Freesound.org. Recuperado en diciembre de 2024, de <http://freesound.org>*
- [5] *Supabase. (s.f.). Supabase. Recuperado en diciembre de 2024, de <https://supabase.com/>*
- [6] *GitHub Pages (s.f.) Recuperado en diciembre de 2024, de <https://pages.github.com/>*
- [7] *Vue.js (s.f) Recuperado en diciembre de 2024, de <https://vuejs.org/>*