



Escuela Confederación Suiza

et26.edu.ar

MANUAL DE USUARIO PARA EL MANEJO DE LA WEB DEL COLEGIO

Proyecto realizado para Prácticas Profesionizantes

Brandon Porcel, Danilo Bautista

Profesor: Luna, Federico
Buenos Aires, 2022



Índice

1. Introducción
 - a. Acerca de la página
 - b. Acerca del manual
 - c. Acerca de las tecnologías usadas
 - d. Instalacion De Proyecto
2. Diseño
 - a. Logo
 - b. Colores
 - c. Fuentes
3. Archivos Multimedia (Imágenes)
 - a. Iconos y SVG
4. Home
 - a. Hero
 - b. Novedades/Noticia Principal
 - c. Sección Proyectos
5. Banner
6. Galeria
7. Institucional
 - a. Wikipedia
 - b. Autoridades
8. Programas Curriculares
9. Proyectos
 - a. Proyecto Descripción
10. Contacto
11. Footer
 - a. Iconos
 - b. Escuela
 - i. Información Útil
 - c. Más
 - i. Github
 - ii. Cecs
 - iii. Drive
 - iv. Biblioteca Virtual
12. Deploy



Introducción

Acerca de la página

El proyecto consta de realizar el sitio web institucional de la escuela Confederación Suiza. La página tendrá apartados donde se muestra información sobre las especialidades que ofrece el colegio, un apartado donde se expongan los proyectos realizados por alumnos y más.

El propósito del proyecto es acercar la información de una manera clara y fácil acceso a padres y estudiantes.

Para el desarrollo de la página tuvimos que realizar investigaciones sobre distintos temas como la historia del colegio, la cronología de los distintos logos, etc. Llegamos a una entrevista con el rector Pablo Conrado para profundizar en diferentes cuestiones.

Acerca del manual

El manual es para que las personas que administren el sitio web puedan hacerlo de la manera más simple y con las menores complicaciones sin importar que tengan grandes conocimientos de programación o no.

Recomendamos ver algún video corto de youtube sobre como funciona React.js

Acerca de las tecnologías utilizadas

Las tecnologías utilizadas para el desarrollo de la página son:

- React.js
Librería de JavaScript que se destaca por el manejo de componentes que son archivos jsx que se piensan reutilizar.
- Styled Components
Librería que permite escribir los estilos CSS en el mismo componente. Esta librería se basa en crear componentes que llevan consigo estilos.

```
//dentro de cualquier componente jsx
import styled from 'styled-components'; //importamos

const ButtonStyle = styled.button`
  cursor: pointer;
  ...
`;
export default function Button({ children }) {
  return <ButtonStyle
className="bg">{children}</ButtonStyle>
</>
}
```



- React Router
Librería que facilita el manejo de rutas /
Para crear una nueva ruta, ir a **/src/App.js** y

```
//importar componente/página
import Home from './components/Home/Home';

<ContactProvider>//provider
  <ProjectsProvider> //provider
    <Routes>
      <Route path="/" element={<Home />} />
    </Routes>
  </ProjectsProvider>//provider
</ContactProvider>//provider
```

Instalación

Ir al [repositorio](https://github.com/brandonporcel/confederacionsuiza.git) , clonarlo localmente mediante la consola (o descargarlo), instalar las dependencias, correr el proyecto.

El signo \$ significa que el comando deberá ser escrito en consola (recomiendo git bash. Podría ser cualquiera).

\$ git clone <https://github.com/brandonporcel/confederacionsuiza.git>

\$ cd confederacionsuiza

\$ npm install **instala las dependencias(librerías)

\$ code . **Abre el editor de código predeterminado. Si no tiene, abrir el proyecto manualmente.

\$ npm start **corre el proyecto y abre una pestaña en el navegador usando un puerto(3000)



Diseño

El figma donde se ven las ideas que se tuvo a lo largo del proceso:

[skooly team library – Figma](#)

Colores

Los colores están estipulados como variables en el archivo **/src/index.css** y escritos de forma hexadecimal. Podrían cambiarlos por el tipo rgb, hl o de forma empírica poniendo el nombre del color.

Están guardados en variables para que estén centralizadas y se puedan reutilizar en distintas partes y que en caso de querer cambiarlas solo haya que modificarlas allí.

En las variables se pueden guardar distintos tipos de datos. Implementación:

```
:root{
  --nombreVariable: #fafafa;
  --nombreVariable: Arial;
}
.nombreClase{
  font-family
  color: var(--nombreVariable);
}
```

Fuente

La fuente elegida fue Encode Sans. Gracias a su compatibilidad con fonts google solo hay que importar la cdn de fonts.google y luego importar la fuente siendo en index.html o en index.css.

En el código está en src/index.css. En caso de cambiar la fuente, cambiar la variable --font, importarla y cambiar la cdn de index.html y poner la que provee fonts Google (en caso de usar una que esté allí. Si no, descargar la fuente[ott,wpp] e importarla en index.css).



Archivos Multimedia (Imágenes)

Los videos, audios, imágenes están en la carpeta /public con sus respectivas carpetas para mantener el orden.

Chequear bien las el nombre y extensiones de las imágenes. En local no va a haber problema, en producción sí.

Iconos y SVG

Los iconos y archivos SVG están guardados en /src ya que serán aplicados con más continuidad en los componentes.

En src/svg están los svg que pueden ser aplicados sin muchas modificaciones y están más orientados a estar en el source de una imagen. ``

En src/componentes/svg está en formato de componente dispuesto a recibir propiedades que podrían cambiar su color por ejemplo.

Si la imagen es .../banner.jpg y en el código ponemos

```
<Banner img="./images/computacion/banner.JPG" />
```

Va a funcionar localmente. Pero en producción, con la página ya deployada, el servidor no va a poder encontrar la img.

Debería ser así

```
<Banner img="./images/computacion/banner.jpg" />
```



Home

Hero

Las imágenes seleccionadas del hero están determinadas por const herolImages.

Novedades/Noticia Principal

Como se ve en el título , se quiere que lo primero que vea el usuario luego del hero es la noticia/ítem más importante del último tiempo relacionado con la escuela.

Podría ser un pdf de autorización, de inscripción,; una imagen

Para editar, ir a **src/components/Novedades.jsx** e insertar el código que quieran. Pueden insertar el embed code de pdfs, videos de yt, imágenes

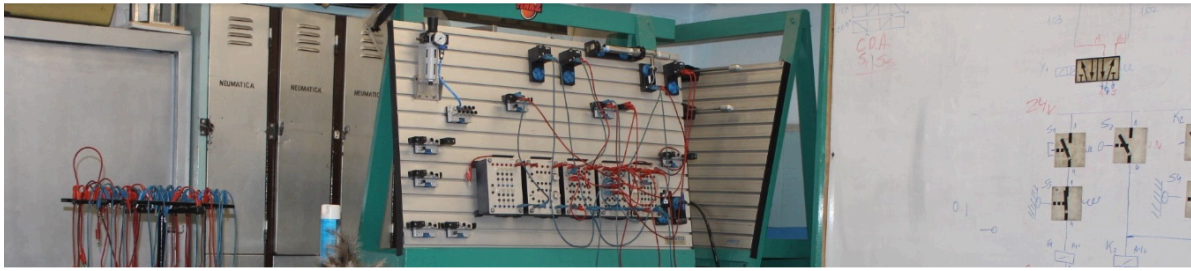
Sección Proyectos

Esta sección está en **src/components/ProyectosSection.jsx**. Muestran los 3 primeros que están enlistados en el que vendría a ser un estado global en

src/context/ProjectsContext.js. Son traídas por la función “firstThree” escrita en ese archivo y provista por las props.

Si quieren mostrar otros, simplemente tienen que cambiar el orden.

Banner



El código del Banner está en **src/components/Banner.jsx** y para poder implementarlo hay que entregarle las propiedades **img** y opcionalmente **height**.

```
<Banner img="./images/599.jpg" />
```


Galería



El source está en **src/components/ImgGallery.jsx** y para usarlo hay que entregarle la propiedad **galleryImages** que tiene que ser un array de objetos con la siguiente estructura:

```
{
  id: 0, // importante que los siguientes sean un número mayor
  al anterior.
  title: 'El robot del proyecto!', //descripción de la img.
  Será lo que esté en el alt de la <img/>
  url: './images/proyectos/coparobotica2019/error1.jpg',
}
```

Si se pasa un array de un solo objeto que no tenga id, se va a romper la app. Al menos un id tiene que tener.

El id puede ser cualquier número siempre y cuando el objeto anterior sea un número menor y el siguiente, un número mayor.

```
const arrayDeObjetos= [
  {
    id: 14,
    title: 'El robot del proyecto!',
    url: './images/proyectos/coparobotica2019/error1.jpg',
  },
  {
    id: 15,
    title: 'Estudiantes de la ET26 estudiando',
    url: './images/proyectos/coparobotica2019/error1.png',
  },
]
```



Se pueden poner videos pero tiene que ser especificado en el objeto con la propiedad “type” y el valor “video”.

```
{  
  id: 4,  
  title: 'Video Presentación Equipo',  
  type: 'video',  
  url: './videos/coparobotica2.mp4',  
},
```

Implementación:

```
<ImgGallery galleryImages={arrayDeObjetos} />
```



Institucional

El código está en **src/components/Institucional.jsx**

Wikipedia

Creando una cuenta en wikipedia ya puedes editar. Por favor, ser cuidadoso con la información que se pone.

Si no pueden modificar la info con su cuenta de wikipedia. Ingresar con la que se creó el artículo (no debería traer complicaciones):

user: Tecnica26

contra: et26wikipedia

mail :eSkueLy.26.et@gmail.com

contra: eSkueLy26et

Autoridades

La información de los mandos de la escuela están concentradas en un array de objetos en el archivo **src/components/Autoridades.jsx**.

```
const autoridades = [  
  {  
    nombre: 'pablo conrado',  
    cargo: 'rector',  
  },  
  ...  
]
```



Programas Curriculares

1 Año	Historia
	Ed. Física
	Inglés
	Educación Artística
	Matemática
	Lengua Y Literatura
	Biología
	Ed. Ciudadana
	Tecnología De La Representación

Los programas están provistos en un drive que tiene control el rector Pablo Conrado.

Los programas fueron implementados en /PrimerCiclo, /Computación y /Automotores.

Para la implementación hay que usar el componente <ProgramasTablas> que está en **src/Especialidades/ProgramasTablas.jsx** y enviarle como propiedad **materias** que tiene que ser un array de objetos donde se enumeren las materias justamente.

```
const computacionMaterias = [  
  {  
    ano: '3',  
    programas:  
    'https://drive.google.com/drive/folders/1eof7ydGeFfb9SY3fGV8Tuc',  
    //link especifico para del programa del año.  
    materias: [  
      'Lengua y Literatura',  
      'ed. física',  
      ...  
    ],  
    {  
      ano: '4', ...  
    }  
  ]
```



Implementación

```
<SubtitleAndDescription title="Programas Curriculares">  
  <ProgramasTablas materias={computacionMaterias} />  
</SubtitleAndDescription>
```

SubtitleAndDescription es un componente creado especialmente para ser usado en Institucional, pciclo, compu automotores en la que ya trae estilos definidos que nos ayuda para la implementación de ProgramasTablas.

Proyectos

Proyectos

Proyectos de alumnos coordinados por docentes o creados de forma independiente.
Participaciones en ferias tecnológicas, competencias, exposiciones y más.



Flecha Cósmica

2018

Alumnos De 13 A 15 Años De La Escuela
Se Quedaron Con El Primer Puesto Del
Torneo Nacional First Lego League (FL...



Copa Robótica

2019

La Copa Robótica 2019 Fue La Etapa De
Clasificación Local Organizada Por El
Ministerio De Educación E Innovación Y.



Desafio Eco Edición 2013

2013

El Desafio ECO Es Una Competencia En
El Que Los Alumnos De Escuelas
Técnicas De Todo El País.

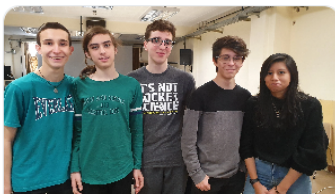


En /proyectos se pueden ver todos los proyectos destacados que se realizaron con relacion a la escuela.

Los proyectos son, en cuestión, los que fueron enumerados en
/src/context/ProjectsContext.js.

La estructura de cada proyecto-objeto tiene que respetar la siguiente sintaxis para que se en la card de previsualización de un proyecto se vea con la información correcta.

```
{
  title: 'Copa Robótica',
  link: '/coparobotica2019', //ruta del proyecto donde hay +
  info.***
  description: 'La Copa Robótica 2019 fue la etapa de clasificación
  local organizada por el Ministerio de Educación e Innovación y
  Educabot para elegir al equipo que participe en el mundial de
  First Global Challenge.',
  thumbnail:
    './images/proyectos/coparobotica2019/coparobotica2019.webp',
  year: '2019',
}, {...}
```



Copa Robótica

2019

La Copa Robótica 2019 Fue La Etapa De
Clasificación Local Organizada Por El
Ministerio De Educación E Innovación Y.

*** Esa ruta tiene que crearse en /App.js y tiene que tener la misma url

```
<Route path="/coparobotica2019"
  element={<CopaRobotica/>} />
```



Proyecto Descripción

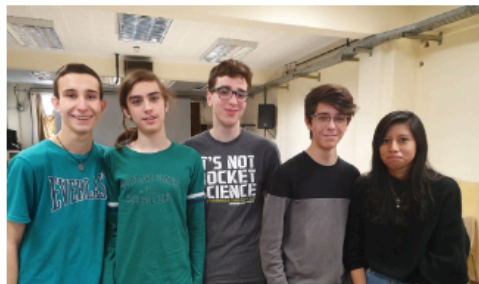
Hay un diseño pre armado en `/src/ProjectContainer.jsx` para el detalle de cada proyecto que con se implementa pasandole props. Dentro de `<ProjectContainer></ProjectContainer>` Se puede poner más cosas como `<SubTitleAndDescription>` que también tiene estilos ya definidos para sumar información.

Las props son `title`, `mainImg`, `when` y `specialty`.

```
<ProjectContainer
title="La escuela represento a CABA en la copa robótica nacional"
mainImg="./images/proyectos/coparobotica2019/coparobotica2019.webp"
when="2019" //fecha de proyecto
specialty="c" //c=computación - a=automotores - x=primer ciclo
>
  <SubTitleAndDescription
    title="descripción"
    w="100"
    specialClass="subtitleClassProject">
    <p>Lorem ipsum dolor sit amet icing elit.ea </p>
  </SubTitleAndDescription>
</ProjectContainer>
```

La Escuela Represento A CABA En La Copa Robotica Nacional

2019 • Computación





Contacto

Para empezar si van a realizar una modificación que altere el height del recuadro/contenedor blanco, volver a chequear el height fondo gris. No se ajusta automáticamente.

Los contactos están expuestos en **/src/context/ContactContext.js** y si hay que agregar otro contacto ya sea número o mail lo tienen que agregar a la variable `contactData`. Si quieren poner quien es la persona coordinadora, el valor no será un string sino que un objeto en el que podrán el nombre de tal persona.

```
const contactData = {
  telefono: '11 4931-1947',
  escuela: 'det_26_de6@bue.edu.ar',
  'Departamento de Orientación al Estudiante':
'doe@et26.edu.ar',
  computación: {
    contact: 'computacion@et26.edu.ar',
    coordinador: 'Pablo Fiscella',
  },
}
```

El código de que se hace para mostrar todos los contactos puede ser un poco difícil de leer así que si quieren pueden poner los contactos uno por uno llamando la prop de cada contacto:

```
<div className="contactWayItem">
  <span className="contact-item-title">Computación</span>
  <span>Coordinado por
{contactData['computación'].coordinador}</span>
  <a href={`mailto:${contactData['computación'].contact}`}>
{contactData['computación'].contact}
  </a>
</div>
<div className="contactWayItem">
...
</div>
```

Lo mismo para las redes sociales, modificar `socialLinks` en el estado global y se aplicaran los cambios donde se use.



Footer

Iconos

Los iconos del footer están en **src/svg** y son exportados a través del archivo `svgAsImage` para que puedan ser importados en otros componentes e incrustarlos en el `src` de ``.

Si quieren poner otro icono, borrar el código `svg` y poner el nuevo; reemplazar el archivo por otro; o hacer un componente al `svg` y utilizarlo en otros componentes.

Escuela

- **Informacion Util**

Vendría a ser las FAQ (preguntas frecuentes). El archivo es **src/components/InformacionUtil.jsx** y para modificar, eliminar o agregar preguntas, modificar la constante `faqData`.

Más

- **Github**

Organización de github oficial del colegio.

La idea de tener un github institucional es poder mostrar los trabajos que realizan los chicos en taller, clase o de forma independiente que sea bueno mostrar.

La organización sería administrada por los docentes de taller de computación de 3er a 6to año. Cada docente podrá destacar el proyecto de un alumno simplemente haciendo un fork (es como compartir su proyecto y se muestre en tu perfil/organización) al repositorio del alumno de su cuenta.

<https://github.com/tecnica26>

- **CECS**

Sitio web del centro.

Hablar con las correspondientes autoridades para modificar algo!

- **Programas Curriculares**

Hablar con Conrado si quieren modificar el drive.

https://drive.google.com/drive/folders/10X_ZSrh4DpRsJtE2NmuzIsxB24j3wFV?usp=sharing

- **Biblioteca Virtual**

Sitio web de la biblioteca donde se ven todos los libros existentes de literatura en la biblioteca del 4to piso.

Siendo administrador, se puede agregar, eliminar y modificar libros. Hablar con las bibliotecarias para pedir contraseñas.

Para hablar sobre código, **Brandon Porcel** o viendo el repositorio mismo.

user: admin — contra: la5u1z4borges



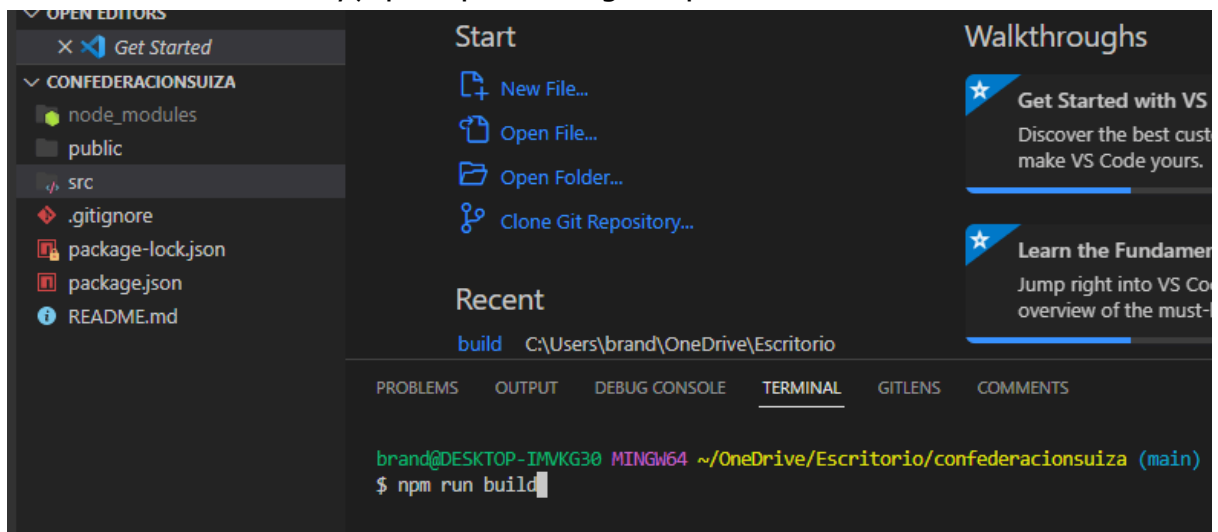
Deploy

El hosting así como el dominio es administrado por la escuela. Si quieren más detalles, hablar con Conrado.

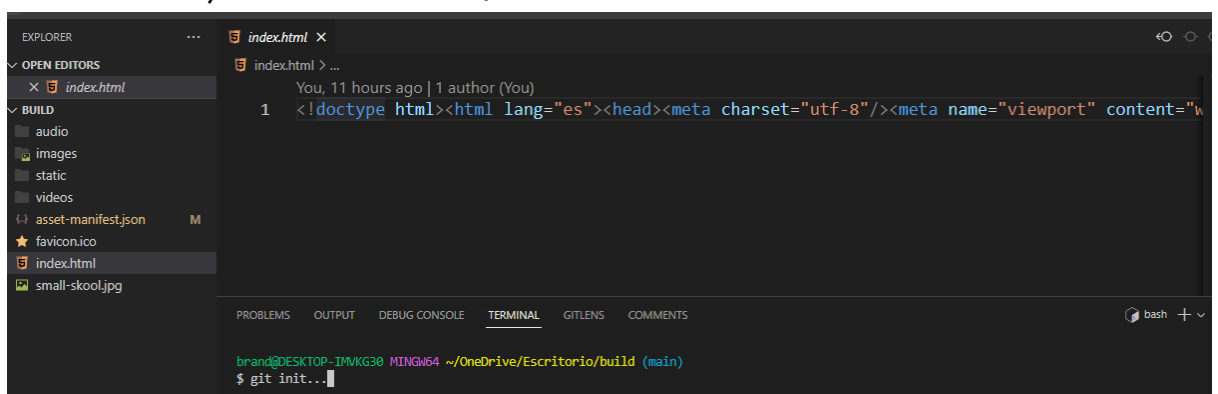
Una vez clonado el repositorio, haber hecho cambios y pusheado al repositorio, se cargaran los cambios en <https://et26.vercel.app/>, sin embargo, no importa mucho ya que los cambios tienen que ir a para et26.edu.ar.

Los pasos para deployar al hosting edu.ar son:

1. Dentro del proyecto, ejecutar el comando `npm run build`. Esto generará una carpeta con los archivos `html`, `css` y `js` para que el navegador pueda entenderlos.




2. Llevar la carpeta `build` al escritorio y pushear al repositorio especial para el build. No lo pusheamos al repositorio principal de la página porque pesa mucho.
3. Se puede corroborar que funcione instalando la extensión de vscode “live server” e iniciar el servidor. Si al iniciarlo los lleva a una ruta no deseada, puede modificar `asset-manifest.json` -> “`index.html`”: ‘/’



4. Hablar con Gabriel Teruel, que es el que maneja el hosting edu.ar para finalmente subir los cambios.








[Pull requests](#) [Issues](#) [Codespaces](#) [Marketplace](#) [Explore](#)

[brandonporcel / confederacionsuiza-build](#) Private Unwatch 1 Fork 0

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#) [Settings](#)

 **main**


 1 branch









 0 tags

[Go to file](#)


[Add file](#)


[Code](#)


 **brandonporcel initial commit** e05e053 11 hours ago 1 commit

 audio	initial commit	11 hours ago
 images	initial commit	11 hours ago
 static	initial commit	11 hours ago
 videos	initial commit	11 hours ago
 asset-manifest.json	initial commit	11 hours ago
 favicon.ico	initial commit	11 hours ago
 index.html	initial commit	11 hours ago
 small-skool.jpg	initial commit	11 hours ago

About
build de la web del co

 0 stars

 1 watching

 0 forks

Releases
No releases published
[Create a new release](#)

Packages
No packages published
[Publish your first package](#)