



COMICSMARVELSPOOLS

NOMBRE: ALEJANDRO HERNÁNDEZ PÉREZ

CURSO: DAW 2

Selección del framework e instalación de lo necesario

Framework: react que será utilizado en ionic.

- Nodejs Linux:

Pondremos el comando “sudo apt-get install nodejs” para instalar nodejs.

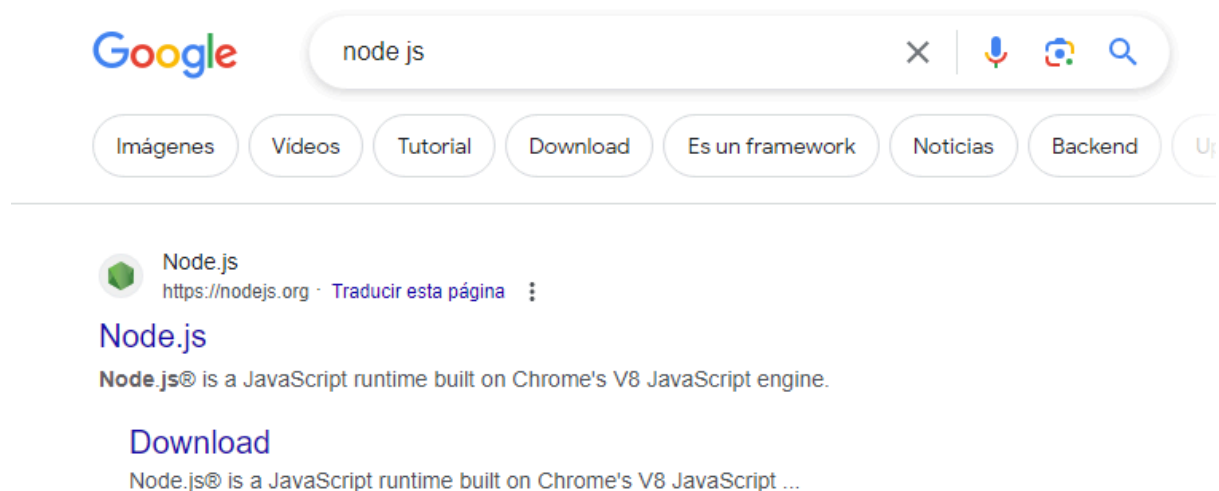
```
administrador@Xubuntu22:~$ sudo apt-get install nodejs
```

Pondremos el comando “sudo apt install npm” para instalar npm.

```
administrador@Xubuntu22:~$ sudo apt install npm
```

- Nodejs Windows:

Iremos a internet donde pondremos node js y buscaremos y daremos en download.



Seleccionamos en LTS y damos en 64-bit de Windows Installer (.msi).

Downloads

Latest LTS Version: 20.11.0 (includes npm 10.2.4)

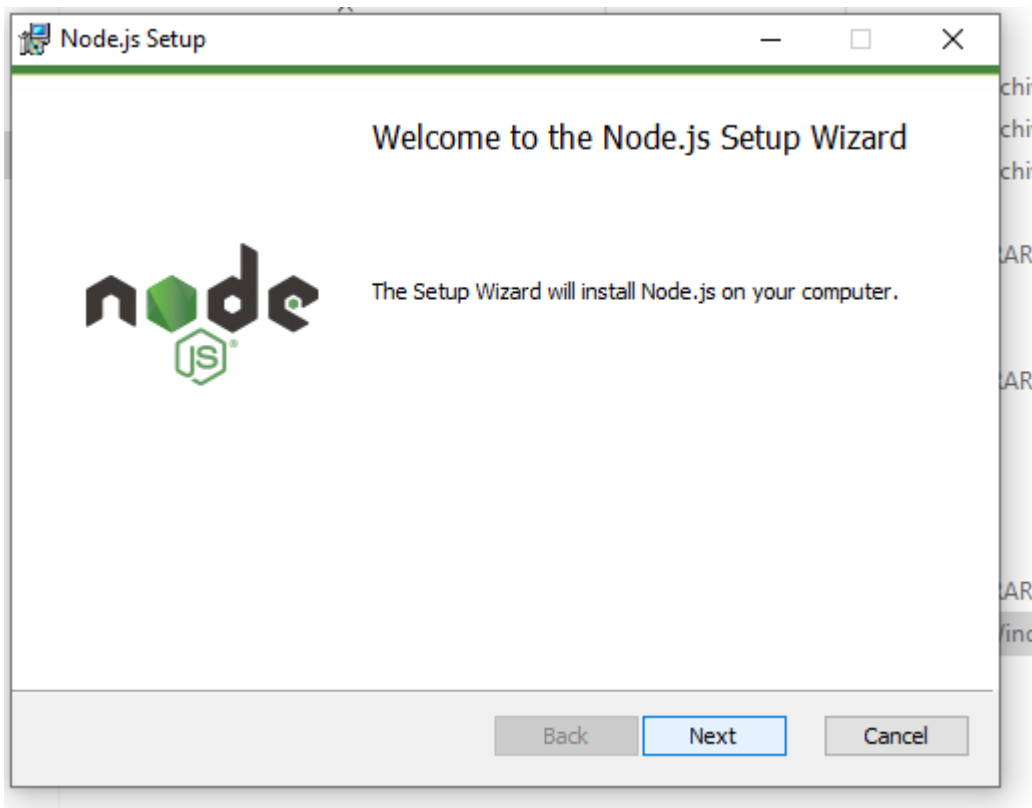
Download the Node.js source code or a pre-built installer for your platform, and start developing today.

LTS Recommended For Most Users	Current Latest Features	
 Windows Installer node-v20.11.0-x64.msi	 macOS Installer node-v20.11.0.pkg	 Source Code node-v20.11.0.tar.gz

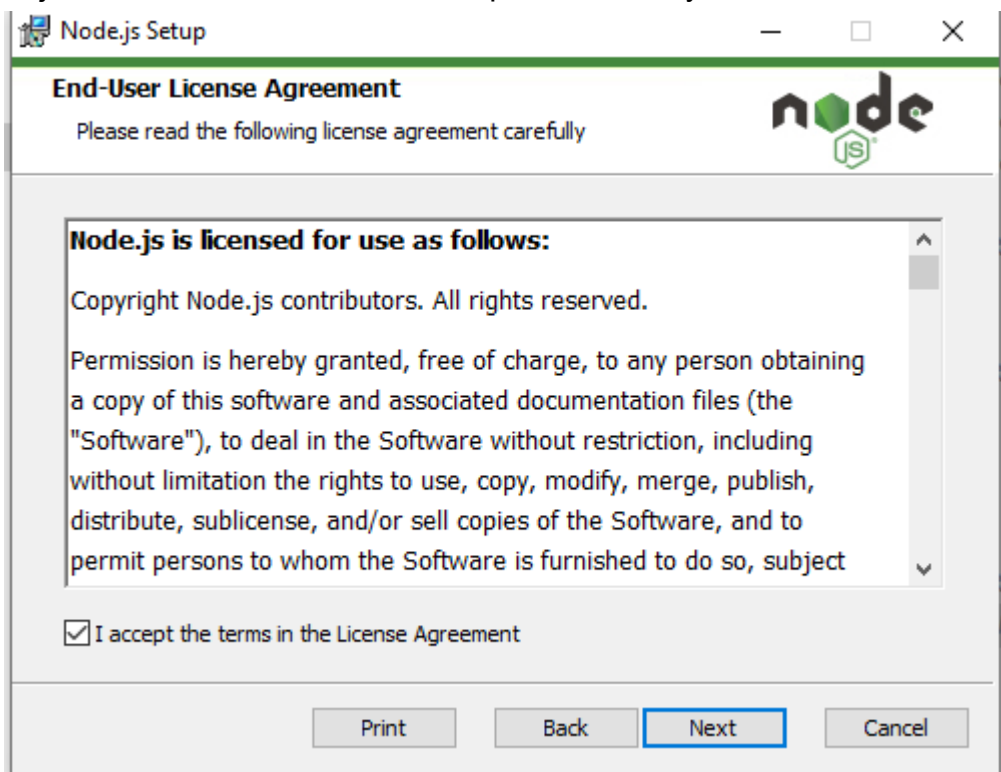
Windows Installer (.msi)	32-bit	64-bit	ARM64
Windows Binary (.zip)	32-bit	64-bit	ARM64
macOS Installer (.pkg)	64-bit / ARM64		
macOS Binary (.tar.gz)	64-bit	ARM64	
Linux Binaries (x64)	64-bit		
Linux Binaries (ARM)	ARMv7	ARMv8	

Ya descargado damos en node-v20.11.0-x64.

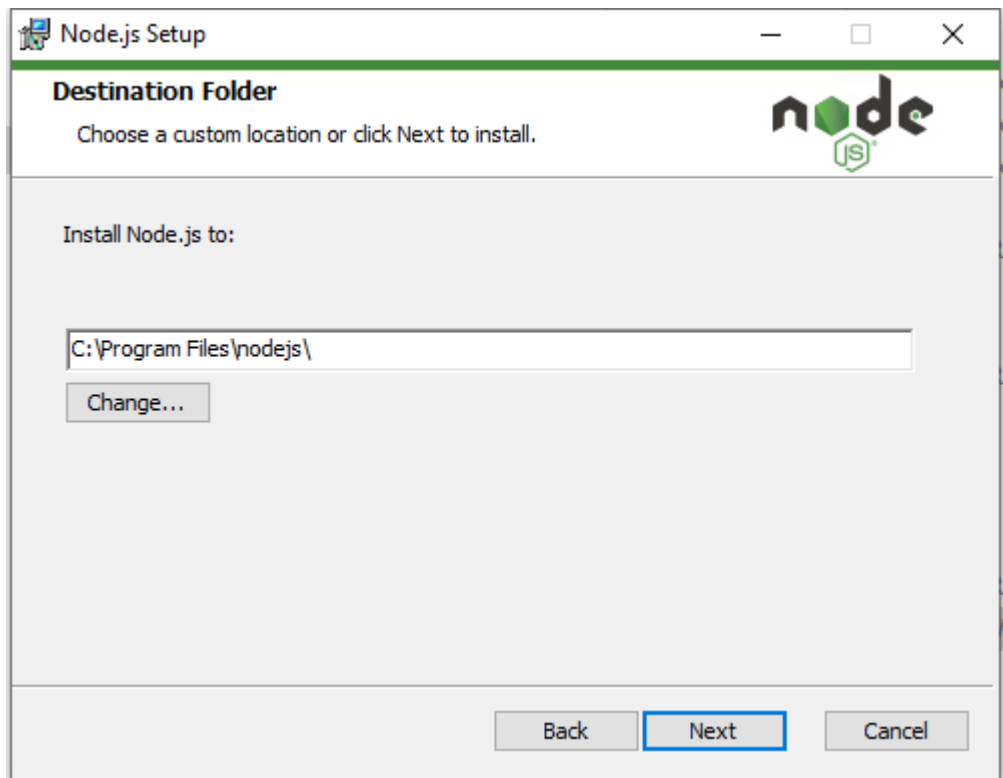
Daremos en Next.



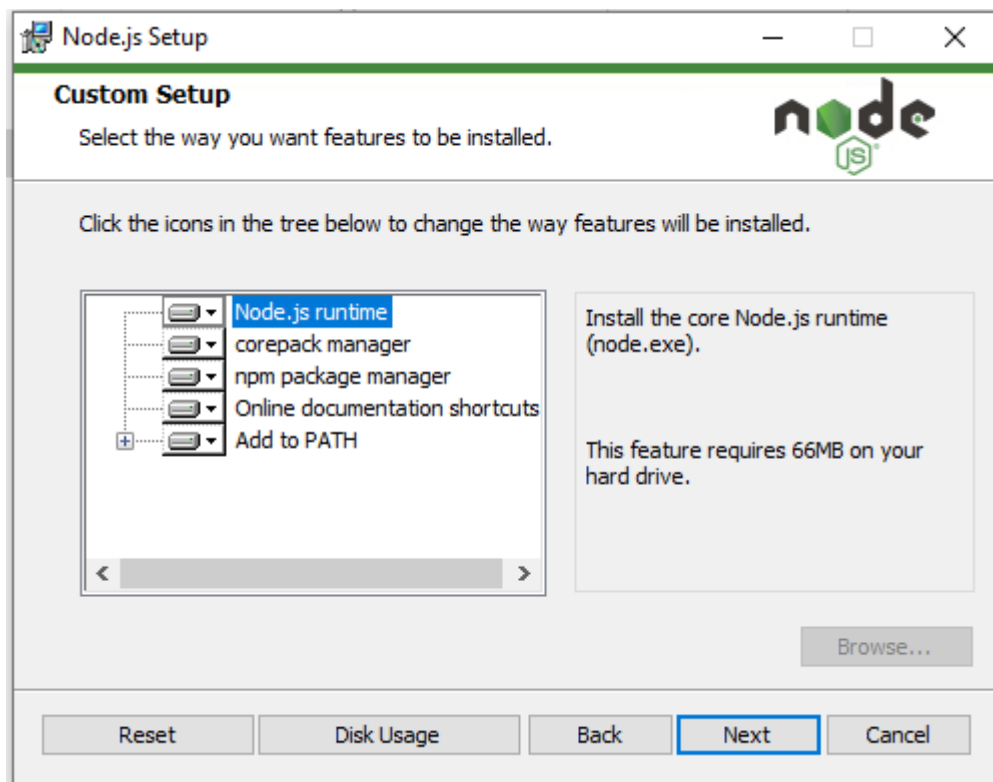
Dejaremos seleccionado en el aceptar términos y daremos Next.



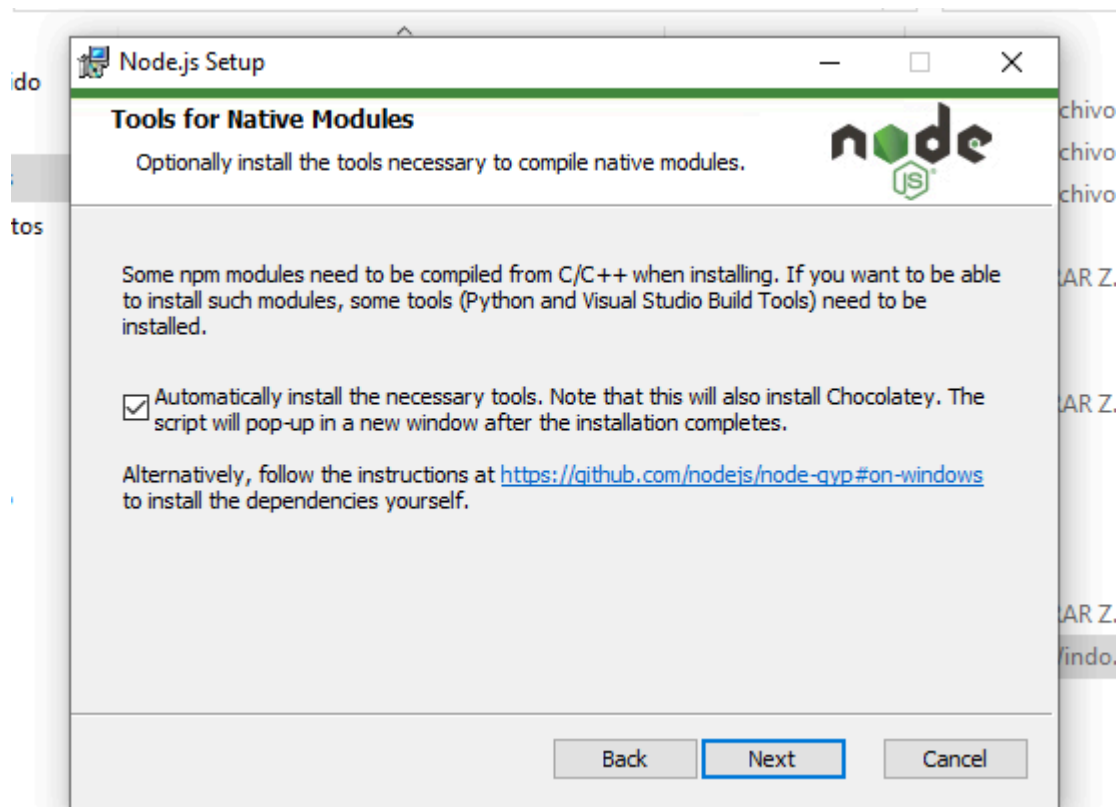
Daremos en Next.



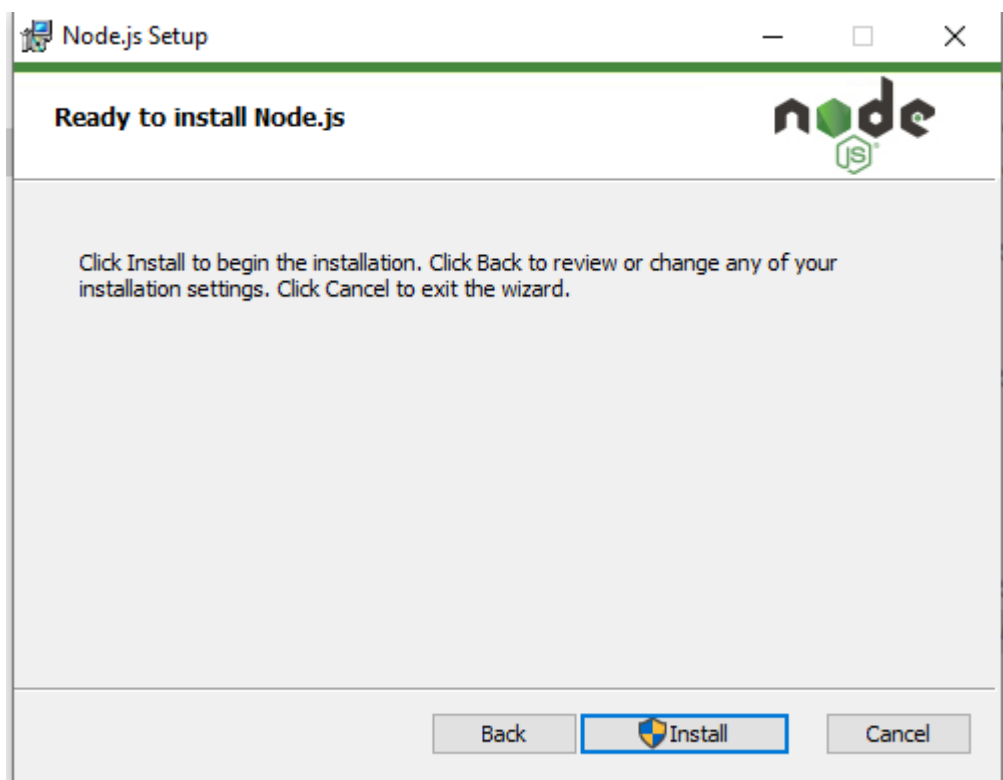
Daremos en Next.



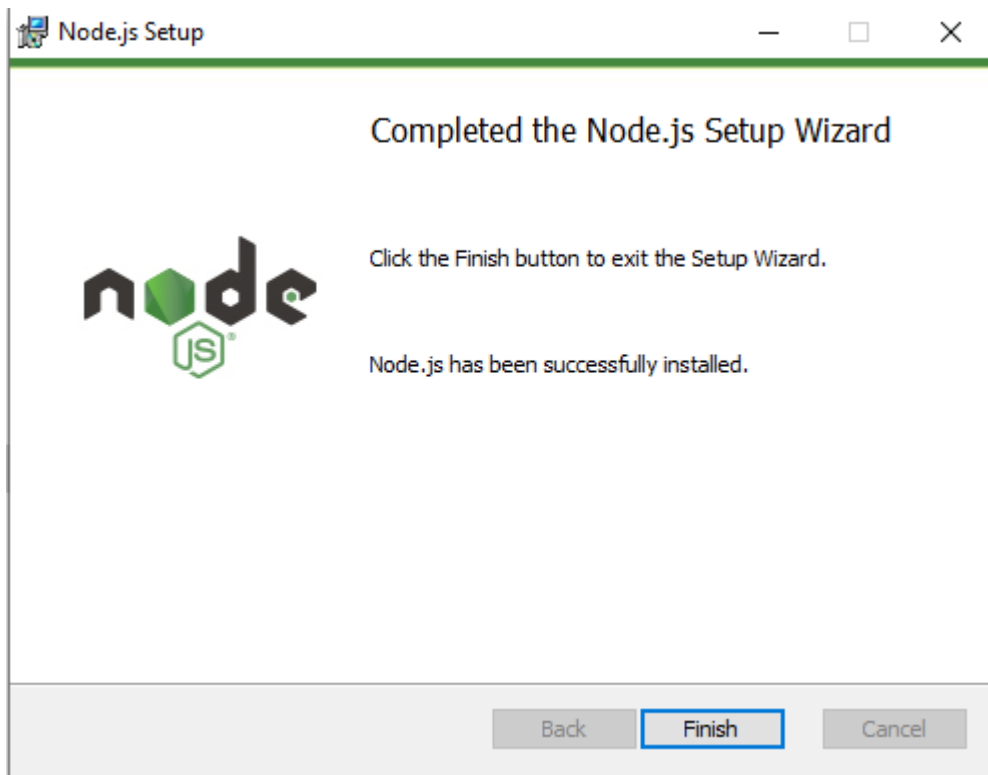
Seleccionamos el único que aparece y daremos en Next.



Damos en Install.



Y cuando damos a finish nos saldrá el cmd con un mensaje donde daremos al cualquier tecla unas cuantas veces y nos instalará el paquete que seleccionamos.



Pondremos el comando “node -v npm -v” para comprobar la versión de node.js.

```
C:\Users\admin>node -v npm -v  
v20.11.0
```

Instalación Ionic:

Pondremos el comando “npm install -g @ionic/cli” para instalar ionic.

```
C:\Users\admin>npm install -g @ionic/cli
```

Pondremos el comando “npm install -g create-react-app” para instalar react.

```
C:\Users\admin>npm install -g create-react-app
```

Planteamiento de la aplicación final

- **IonReactRouter:** Es un paquete de enrutamiento diseñado específicamente para aplicaciones Ionic desarrolladas con React. Proporciona integración directa con los componentes de enrutamiento de Ionic, lo que facilita la navegación entre las páginas de tu aplicación.
- **Redirect:** Es un componente proporcionado por la biblioteca react-router-dom que se utiliza para redirigir a los usuarios a otra ubicación en la aplicación. Cuando se renderiza un componente Redirect, automáticamente cambia la URL en la barra de direcciones del navegador y navega a la nueva ubicación especificada.

- **useParams:** Es un hook proporcionado por React Router que te permite acceder a los parámetros de la URL en los componentes de tu aplicación. Este hook es útil cuando necesitas acceder a valores dinámicos pasados como parte de la ruta.
- **.map():** en JavaScript se utiliza para iterar sobre todos los elementos de un array y ejecutar una función para cada elemento. Esta función puede realizar una operación en cada elemento del array y devolver un nuevo array con los resultados de estas operaciones.
- **Link:** se utiliza en React Router para crear enlaces entre diferentes rutas de tu aplicación. Cuando un usuario hace clic en un enlace creado con Link, React Router se encarga de actualizar la URL y renderizar el componente correspondiente asociado a la nueva ruta, sin necesidad de recargar la página completa.
- **useState:** Es un hook proporcionado por React que te permite añadir estado a componentes funcionales en React. Con useState, puedes declarar variables de estado en tus componentes y React se encargará de mantener su estado y volver a renderizar el componente cuando el estado cambie.
- **IonRouterOutlet:** Es parte de Ionic Framework y se utiliza en combinación con IonReactRouter para gestionar la renderización de las rutas en una aplicación Ionic React. Funciona de manera similar al componente Switch en react-router-dom, pero está diseñado específicamente para funcionar con los componentes de Ionic y proporciona algunas características adicionales.
- **Route:** Es una parte fundamental de React Router y se utiliza para definir cómo deben coincidir las URL del navegador con los componentes que se deben renderizar en una aplicación React.
- **IonTabbutton:** Es un componente proporcionado por Ionic que se utiliza para representar un botón dentro de una barra de pestañas (IonTabBar). Este componente se utiliza para crear cada una de las pestañas en una aplicación Ionic React.

Capturas de los distintas secciones de la aplicación

Usaremos el ionic start para crear un nuevo proyecto, nos preguntará algo app wizard diremos no y cuando salga la selección de frameworks seleccionamos en React y damos enter.

```

o administrador@Xubuntu22:~/Escritorio/Proyecto react final$ ionic start
? Use the app creation wizard? No

Pick a framework! 🤖

Please select the JavaScript framework to use for your new app. To bypass this prompt next time, supply a value
for the --type option.

? Framework:
  Angular | https://angular.io
  > React  | https://reactjs.org
  Vue     | https://vuejs.org
  
```

Ponemos el nombre ComicsMarvelsPools y damos enter

```
supply name, the first argument to ionic start.
```

```
? Project name: ComicsMarvelsPools
```

En template seleccionamos en tabs.

```
Let's pick the perfect starter template! 🚀
```

Starter templates are ready-to-go Ionic apps that come packed with everything you need to build your app. To bypass this prompt next time, supply `template`, the second argument to `ionic start`.

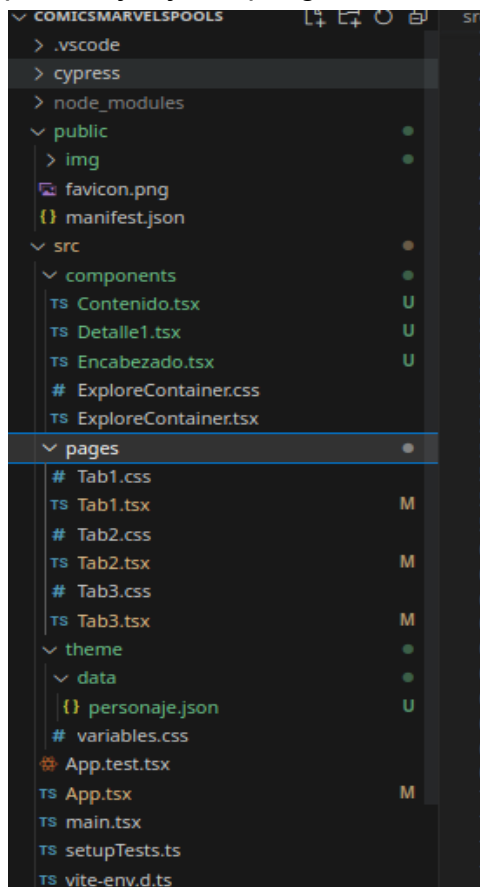
```
? Starter template:
```

blank	A blank starter project
list	A starting project with a list
my-first-app	A template for the "Build Your First App" tutorial
sidemenu	A starting project with a side menu with navigation in the content area
tabs	A starting project with a simple tabbed interface

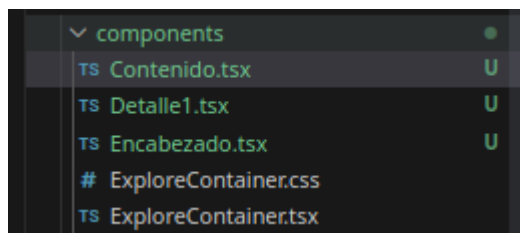
Aquí en donde App.tsx si nos vamos nos abra montado todos los tabs con sus links de forma automática, lo que haremos es cambiar los iconos como aparece en la captura en la parte del IonIcon y ponemos color dark en IonTabBar.

```
40 setupIonicReact();
41
42 const App: React.FC = () => (
43   <IonApp>
44     <IonReactRouter>
45       <IonTabs>
46         <IonRouterOutlet>
47           <Route exact path="/tab1">
48             <Tab1 />
49           </Route>
50           <Route exact path="/tab2">
51             <Tab2 />
52           </Route>
53           <Route path="/tab2/:id">
54             <Detalle1 />
55           </Route>
56           <Route path="/tab3">
57             <Tab3 />
58           </Route>
59           <Route exact path="/">
60             <Redirect to="/tab1" />
61           </Route>
62         </IonRouterOutlet>
63         <IonTabBar color={'dark'} slot="bottom">
64           <IonTabButton tab="tab1" href="/tab1">
65             <IonIcon aria-hidden="true" icon={apps} />
66             <IonLabel>Inicio</IonLabel>
67           </IonTabButton>
68           <IonTabButton tab="tab2" href="/tab2">
69             <IonIcon aria-hidden="true" icon={book} />
70             <IonLabel>Datos</IonLabel>
71           </IonTabButton>
72           <IonTabButton tab="tab3" href="/tab3">
73             <IonIcon aria-hidden="true" icon={mail}></IonIcon>
74             <IonLabel>Contacto</IonLabel>
75           </IonTabButton>
76         </IonTabBar>
77       </IonTabs>
78     </IonReactRouter>
79   </IonApp>
80 );
```

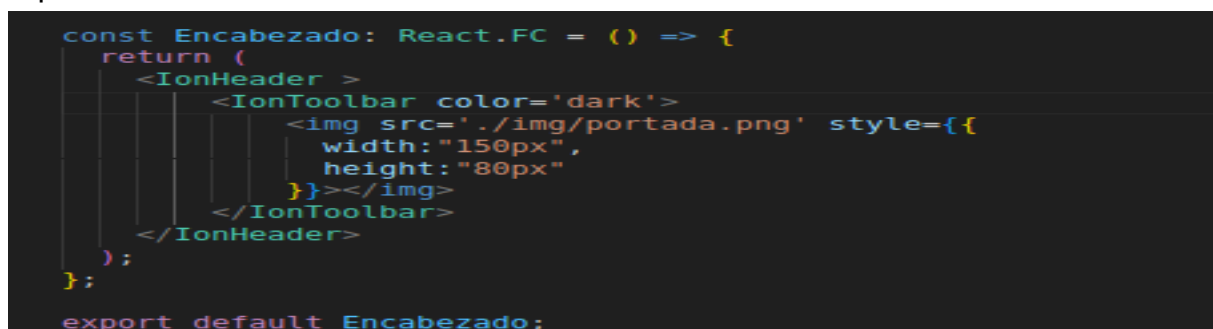

Aquí si nos vamos donde se ve los directorios pondremos una carpeta img en public donde meteremos las imágenes, en pages donde theme meteremos personajes.json que guardará los datos de nuestros superhéroes.



Iremos a la carpeta components y crearemos todos los componentes Contenido.tsx, Detalle1.tsx y Encabezado.tsx.



En Encabezado.tsx crearemos pondremos IonHeader, despues el Iontoolbar para poner color dark para que aparezca el encabezado de color oscuro en nuestro encabezado, y pondremos un img y los estilos indicados como en la imagen y lo exportamos



Ahora iremos a tab1.tsx, Y pondremos una etiqueta encabezado para que aparezca el encabezado automáticamente pero primero lo importamos, ahora lo que haremos es poner un IonContent con color danger para que aparezca de color rojo y los estilos en la imagen.

Y ponemos un img con el estilo que pone, la imagen será un saludo que aparece para saludar al nuevo cliente.

```

1 import { IonContent, IonHeader, IonPage, IonTitle, IonToolbar } from '@ionic/react';
2 import ExploreContainer from '../components/ExploreContainer';
3 import './Tab1.css';
4 import Encabezado from '../components/Encabezado';
5
6 const Tab1: React.FC = () => {
7   return (
8     <IonPage>
9       <Encabezado></Encabezado>
10      <IonContent color='danger' style={{textAlign:"center"}} fullscreen>
11        <img src='../img/mini_deadpool.png' style={{
12          width:"99%",
13          height:"95%"
14        }}></img>
15      </IonContent>
16    </IonPage>
17  );
18 };
19
20 export default Tab1;

```

En nuestro personajes.json almacenaremos lo que sería lo siguiente:

- La id para cuando tengamos que buscarlo.
- Nombre: nombre del personaje.
- Imagen: la dirección de la url en donde se localiza la imagen.
- Descripción: la información del personaje.
- Asociación: Información de grupos o personajes relacionados con el, se guarda como array.
- Habilidades: Su arsenal de poderes que tiene el personaje, se guardará como un array.

```

1 [{
2   "id": "1",
3   "nombre": "Iron man",
4   "imagen": "../img/Iron man.jpg",
5   "descripcion": "Tony Stark es un genio, multimillonario, playboy y filántropo. Él no solo inventó la armadura de Iron Man, sino que también la usa para combatir el crimen y salvar al mundo.",
6   "asociacion": ["Los 4 Fantásticos", "Guardianes de la Galaxia", "X-Men", "The Avengers"],
7   "habilidades": ["Superfuerza", "durabilidad y resistencia por armadura", "Vuelo supersónico", "Repulsores de energía y misiles"],
8 },
9 {
10  "id": "2",
11  "nombre": "Thor",
12  "imagen": "../img/Thor.jpg",
13  "descripcion": "Thor, es un príncipe-guerrero de Asgard, dios del trueno, protector de la tierra, y de los 9 reinos. Él emplea el poder del Mjolnir para combatir el mal.",
14  "asociacion": ["The Avengers", "Guardians of the Galaxy", "Los tres guerreros"],
15  "habilidades": ["Fuerza sobrehumana", "resistencia", "durabilidad", "velocidad", "agilidad", "reflejos", "curación", "longevidad y regeneración"],
16 },
17 {
18  "id": "3",
19  "nombre": "Hulk",
20  "imagen": "../img/Hulk.jpg",
21  "descripcion": "Después de una exposición accidental a los rayos gamma durante la detonación de una bomba experimental, Bruce Banner se transforma en el poderoso y destructivo Hulk.",
22  "asociacion": ["The Avengers", "Wolverine"],
23  "habilidades": ["resistencia", "invulnerabilidad", "super fuerza", "regeneración y curación acelerada"],
24 },
25 {
26  "id": "4",
27  "nombre": "Capitán América",
28  "imagen": "../img/Capitan América.jpg",
29  "descripcion": "El personaje es generalmente representado como el alter ego de Steve Rogers, un joven frágil mejorado a la fuerza sobrehumana por el Siero del Soldado Inmortal.",
30  "asociacion": ["Wolverine", "Los 4 Fantásticos", "The Avengers", "Namor"],
31  "habilidades": ["Sentidos", "agilidad", "velocidad y fuerza sobrehumanas", "Gran habilidad con su escudo", "Instinto de liderazgo"],
32 },
33 {
34  "id": "5",
35  "nombre": "Doctor Strange",
36  "imagen": "../img/Doctor Strange.jpg",
37  "descripcion": "Es un neurocirujano que sufrió daños en los nervios de sus manos en un accidente de coche. Buscó a Ancestral Magic y aprendió el arte del hechizo.",
38  "asociacion": ["Los Vengadores", "Guardianes de la Galaxia", "Eternos", "Ned Leeds", "El Caballero Luna", "América Chávez", "Barón"],
39  "habilidades": ["Magia", "teletransportación", "creación de ilusiones", "Manipulación de la realidad", "Protección de espaldas", "Poderes Mágicos"],
40 }
41 ]

```

```

{
  "id": "5",
  "nombre": "Doctor Strange",
  "imagen": "./img/Doctor Strange.jpg",
  "descripcion": "Es un neurocirujano que sufrió daños en los nervios de sus manos en un accidente de coche. Buscó a Ancestral con la esperanza de",
  "asociacion": ["Los Vengadores", "Guardianes de la Galaxia", "Eternos", "Ned Leeds", "El Caballero Luna", "América Chávez", "Barón Mordo", "Barón Zemo"],
  "habilidades": ["teletransportación", "generación de ilusiones", "proyección de energía", "Poderes Místicos"]
},
{
  "id": "6",
  "nombre": "Spiderman",
  "imagen": "./img/Spiderman.jpg",
  "descripcion": "Se trata de un superhéroe que emplea sus habilidades sobrehumanas, reminiscientes de una araña, para combatir a otros supervillan",
  "asociacion": ["Antorcha Humana", "Daredevil", "Gata Negra", "Araña Escarlata", "Deadpool", "Wolverine", "Iron Man", "Capitán América"],
  "habilidades": ["fuerza", "combate", "inteligencia", "producir y lanzar telarañas sintéticas", "trepar", "adherirse y desplazarse a través de muros y"]
},
{
  "id": "7",
  "nombre": "Ojo de halcón",
  "imagen": "./img/Ojo de halcón.jpg",
  "descripcion": "Su nombre real es Clint Barton. Lo que hace especial a Ojo de Halcón es su asombrosa pericia y su ojo desarrollado con el arco y",
  "asociacion": ["Los Vengadores", "Kate Bishop", "Hombre Lobo", "Hombre Anfibio", "Makkari", "Barney Barton", "Barón Mordo"],
  "habilidades": ["Maestro en arqueria", "Habilidad de combate", "Acróbata experto", "Reflejos, velocidad y agilidad al máximo", "Experto en diseño de"]
},
{
  "id": "8",
  "nombre": "Viuda negra",
  "imagen": "./img/Viuda negra.jpg",
  "descripcion": "Natasha nació en Stalingrado (ahora Volgogrado), Rusia. La primera y más conocida Viuda Negra, es una agente rusa entrenada como",
  "asociacion": ["Los Vengadores", "Barón Mordo", "Nick Fury"],
  "habilidades": ["Experta táctica", "combatiente cuerpo a cuerpo y agente secreta", "Envejecimiento lento y sistema inmunológico mejorado", "Tirador"]
},
{
  "id": "8",
  "nombre": "Viuda negra",
  "imagen": "./img/Viuda negra.jpg",
  "descripcion": "Natasha nació en Stalingrado (ahora Volgogrado), Rusia. La primera y más conocida Viuda Negra, es una agente rusa entrenada como",
  "asociacion": ["Los Vengadores", "Barón Mordo", "Nick Fury"],
  "habilidades": ["Experta táctica", "combatiente cuerpo a cuerpo y agente secreta", "Envejecimiento lento y sistema inmunológico mejorado", "Tirador"]
},
{
  "id": "9",
  "nombre": "Black panther",
  "imagen": "./img/Black panther.jpg",
  "descripcion": "Es el primer superhéroe de tez negra de la historia de Marvel y se debe destacar que el personaje fue creado con anterioridad al",
  "asociacion": ["Los Vengadores", "Guardianes de la Galaxia", "Okoye", "Everett K. Ross", "W'Kabi"],
  "habilidades": ["Fuerza sobrehumana", "velocidad", "reflejos", "agilidad", "resistencia", "durabilidad", "sentidos"]
},
{
  "id": "10",
  "nombre": "Star Lord",
  "imagen": "./img/Star Lord.jpg",
  "descripcion": "Peter Jason Quill es un híbrido Celestial-humano y el antiguo líder de los Guardianes de la Galaxia. Tras la muerte de su madre,",
  "asociacion": ["Guardianes de la Galaxia", "Los Vengadores", "Los Defensores", "Guardia del Infinito", "Ravagers", "Guardia Imperial Shi'ar", "Unidad"],
  "habilidades": ["uniforme que le otorga fuerza y durabilidad aumentadas", "habilidad de viajar a través del espacio", "Arma Elemental", "pistola es"]
},
{
  "id": "11",
  "nombre": "Bruja escarlata",
  "imagen": "./img/Bruja escarlata.jpg",
  "descripcion": "El personaje, presentado en sus inicios en una mutante, posee habilidades para alterar la realidad de formas no especificadas y",
  "asociacion": ["Los Vengadores", "X-Men", "Magnet", "Doctor Strange", "Agatha Harkness"],
  "habilidades": ["Alteración y distorsión de la realidad", "absorción de energía", "magia del caos", "hechicería", "proyección de energía", "fragoquir"]
},
{
  "id": "12",
  "nombre": "Deadpool",
  "imagen": "./img/Deadpool.jpg",
  "descripcion": "Un antiguo mercenario mentalmente inestable y desfigurado famoso por gozar de un sentido del humor poco convencional,",
  "asociacion": ["Cable", "Spider-Man", "X-Men Coloso", "Domino", "Daredevil", "Wolverine", "Bob"],
  "habilidades": ["Fuerza", "resistencia", "agilidad y reflejos sobrehumanos", "Francotirador", "espada", "experto en artes marciales y e"]
}

```

Nos vamos a Tab2.tsx donde pondremos un IonPage, después etiqueta Encabezado y IonContent con el color danger y pondremos la etiqueta Contenido.

```
const Tab2: React.FC = () => {
  return (
    <IonPage>
      <Encabezado></Encabezado>
      <IonContent color='danger' fullscreen>
        <Contenido></Contenido>
      </IonContent>
    </IonPage>
  );
};

export default Tab2;
```

Iremos a contenido.tsx donde tenemos que importar nuestra data de personajes.json como para en la imagen, meteremos la etiqueta IonGrid, IonRow para crear columnas ahora pondremos corchetes{} dentro pondremos personajes.map para que nos saque como un foreach para que pase saque todos los datos de nuestros personajes y lo pondremos todo como en la captura como aparece.

Ahora pondremos link sobre el Ioncard donde pondremos el key pondremos element.id para señalar la id y en el to la url que buscará la pagina pero meteremos en comillas francesas`` para poder meter la id para que linke la información de ese personaje en concreto.

Lo pondremos de esta manera `/tab2/\${element.id}`.

```
import Personajes from '../theme/data/personaje.json'
import { Link } from 'react-router-dom';

const Contenido: React.FC = () => {
  return (
    <div>
      <IonGrid>
        <IonRow>
          {
            Personajes.map((element)=>{
              return(
                <Link key={element.id} to={`/${tab2}/${element.id}`} >
                  <IonCard style={{"height": "75%"}}>
                    <img style={{
                      "width": "100%",
                      "height": "70%"
                    }} src={element.imagen}/>
                    <IonCardHeader>
                      <IonCardTitle>{element.nombre}</IonCardTitle>
                    </IonCardHeader>
                  </IonCard>
                </Link>
              </IonCol>
            )
          }
        </IonRow>
      </IonGrid>
    </div>
  );
};

export default Contenido;
```

Nos volvemos a App.tsx y añadiremos un Route donde pondremos en path la ruta '/tab2/:id' para que solo nos busque ese id en concreto y pondremos la etiqueta de la página Detalle1.

```
<Route path="/tab2/:id">
  <Detalle1/>
</Route>
```

Nos iremos a Detalle1 y pondremos un const entre corchetes id para usar el useParams para que coja la id que pedimos, tenemos ponerlo así useParams<{id:string}>() para que no nos de problemas y pondremos personajeselect para buscar al personaje con esa id para eso utilizaremos el find() en Personajes para que nos lo busque con esa id.

Ya iremos a return ponemos lo mismo en la captura, pero en la parte asociación y habilidades se tiene que usarse el params porque la información metida está en un array, ponemos eso tendremos que poner element para señalar como saldrá los datos y un index para indicar la posición y pondremos un IonChip donde tenemos que poner un key donde pondremos el index para señalar la posición y así poner element y lo tendremos señalado para que muestre la información.

```
1  import { IonCard, IonCardContent, IonCardHeader, IonCardTitle, IonChip, IonContent, IonHeader, IonPage
2  import Personajes from '../theme/data/personaje.json'
3  import { useEffect, useState } from 'react';
4  import { RouteComponentProps, useParams } from 'react-router';
5  import Encabezado from './Encabezado';
6
7
8  const Detalle1: React.FC = () => {
9    const { id } = useParams<{id:string}>();
10    const personajeselect=Personajes.find(e=>e.id===id);
11    return (
12      <IonPage>
13        <Encabezado></Encabezado>
14        <IonContent style={{textAlign:"center"}} color={'danger'} fullscreen>
15          <IonTitle style={{fontSize:"30px"}}>{personajeselect?.nombre}</IonTitle>
16          <img style={{
17            width:"50%",
18            height:"50%",
19            borderRadius:"5px",
20          }} src={personajeselect?.imagen}/>
21          <p>Asociacion:{personajeselect?.asociacion?.map((element,index)=>(
22            <IonChip color={'warning'} key={index}>{element}</IonChip>)
23          )}</p>
24          <p>Habilidades:{personajeselect?.habilidades?.map((e,index)=>(
25            <IonChip color={'dark'} key={index}>{e}</IonChip>
26          )}</p>
27          <p>Descripcion:{personajeselect?.descripcion}</p>
28        </IonContent>
29      </IonPage>
30    );
31  };
32
33  export default Detalle1;
```

Iremos a Tab3.tsx donde meteremos un const[formulario,setformulario] donde pondremos un useState y almacenaremos de información como un nombre, email y un mensaje en formulario y el setformulario como una función para almacenar, pondremos el const handleChange como en la imagen y pondremos como esta para poder almacenar todos los datos de nuestro formulario.

En el const handleSubmit pondremos lo mismo que en la imagen y esta función nos servirá para enseñar una alerta y en consola que la información ha sido almacenada.

```
const [formulario, setFormulario] = useState({
  nombre: '',
  email: '',
  mensaje: ''
});

const handleChange=(e:any)=> {
  const { name, value } = e.target;
  setFormulario({ ...formulario, [name]: value });
}

const handleSubmit=(e:any)=> {
  e.preventDefault();
  console.log('Formulario enviado:', formulario);
  alert('Formulario enviado:'+formulario.nombre);
  alert(formulario.email);
  alert(formulario.mensaje);
}
```

Aquí pondremos todas las etiquetas igual que en la captura, en form pondremos onSubmit={handleSubmit} para cuando le de al botón enviar nos enseñe la alerta, en los inputs meteremos los valores del formulario punto con el dato de cada input para que se almacene el value y pondremos onChange={handleChange} que escribamos se almacene con su repostrectivo formulario dato.

```
return (
  <IonPage>
    <Encabezado></Encabezado>
    <IonContent color='danger' fullscreen>
      <form onSubmit={handleSubmit}>
        <IonItem>
          <IonLabel position="stacked">Nombre</IonLabel>
          <IonInput type="text" name="nombre" value={formulario.nombre} onChange={handleChange} />
        </IonItem>
        <IonItem>
          <IonLabel position="stacked">Email</IonLabel>
          <IonInput type="email" name="email" value={formulario.email} onChange={handleChange} />
        </IonItem>
        <IonItem>
          <IonLabel position="stacked">Mensaje</IonLabel>
          <IonTextarea name="mensaje" value={formulario.mensaje} onChange={handleChange} />
        </IonItem>
        <IonButton type="submit">Enviar</IonButton>
      </form>
    </IonContent>
  </IonPage>
);

export default Tab3;
```

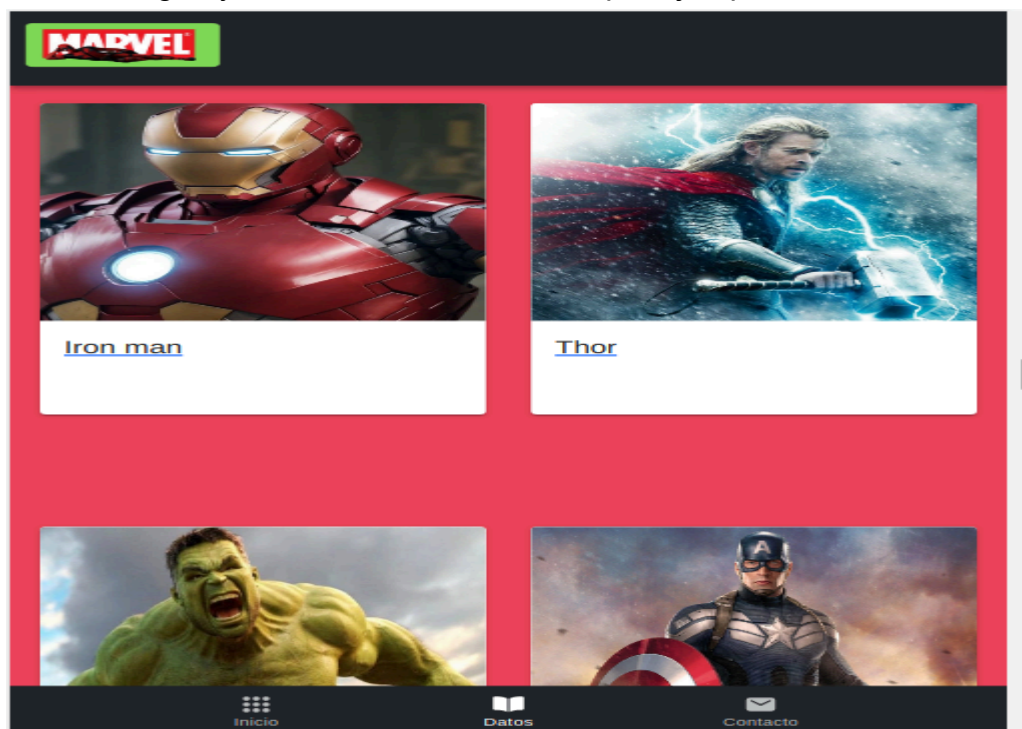
Pondremos el comando ionic serve para comprobar que funciona.

```
administrador@Xubuntu22:~/Escritorio/Proyecto react final/ComicsMarvelsPools$ ionic serve
```

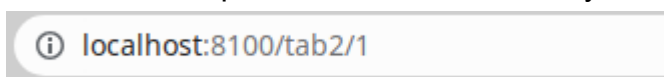

Ya iniciado nos mandará al tab1 que es el inicio y aparecerá un mini deadpool saludando.



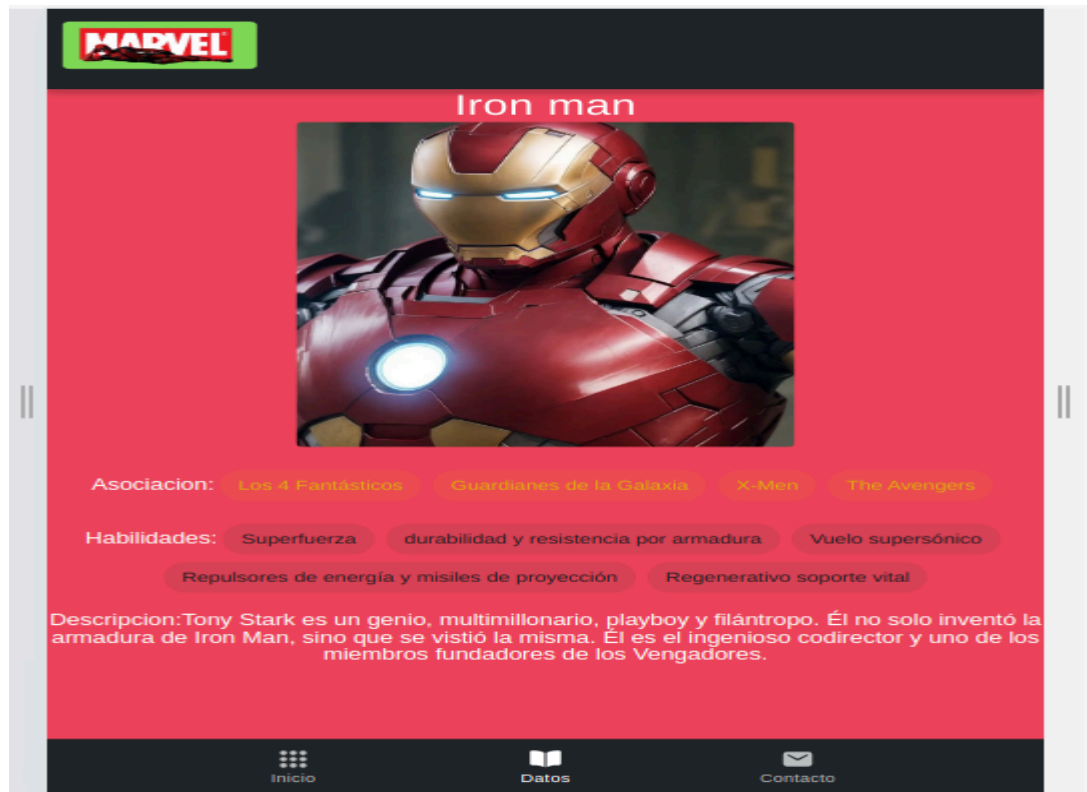
Si nos vamos a datos se nos mostrarán las cartas de cada personajes en columnas con su imagen y nombre si damos en uno por ejemplo en iron man.



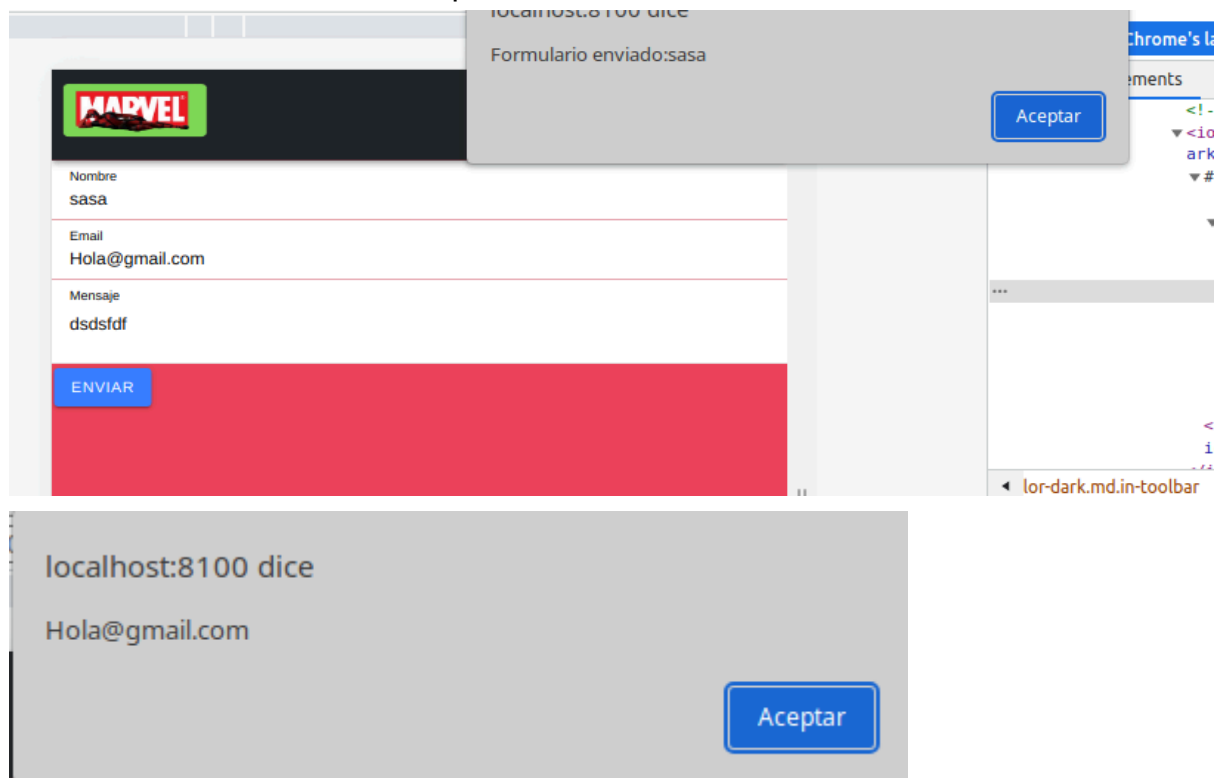
Si vamos nos aparecerá en la url el tab2 y la id del personaje.



Nos llevará a la página Detalle1 nos mostrará toda la información del personaje que buscamos y seleccionamos.



Si nos vamos a la página Contacto aparecerá un formulario que si escribimos algo y damos en el botón enviar nos aparecerá un alerta de formulario enviado.



localhost:8100 dice

dsdsfdf

Aceptar

Y en consola nos aparecerán los datos que hemos puesto en el formulario.

Formulario enviado: Tab3.tsx:24
 ▶ {nombre: 'sasa', email: 'Hola@gmail.com', mensaje: 'dsdsfdf'}

Si da fallo ionic serve y da The Ionic CLI will exit. Please check any output above for error details.

```
C:\Users\Linter409\Downloads\ComicsMarvelsPools\ComicsMarvelsPools>ionic serve
> vite --host=localhost --port=8100
[vite] (node:19536) Warning: To load an ES module, set "type": "module" in the package.json or use the .mjs extension.
[vite] (Use `node --trace-warnings ...` to show where the warning was created)
[vite] C:\Users\Linter409\Downloads\ComicsMarvelsPools\ComicsMarvelsPools\node_modules\.bin\vite:2
[vite] import { performance } from 'node:perf_hooks'
[vite] ^^^^^^
[vite] SyntaxError: Cannot use import statement outside a module
[vite]     at internalCompileFunction (node:internal/vm:77:18)
[vite]     at wrapSafe (node:internal/modules/cjs/loader:1288:20)
[vite]     at Module._compile (node:internal/modules/cjs/loader:1340:27)
[vite]     at Module._extensions..js (node:internal/modules/cjs/loader:1435:10)
[vite]     at Module.load (node:internal/modules/cjs/loader:1207:32)
[vite]     at Module._load (node:internal/modules/cjs/loader:1023:12)
[vite]     at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:135:12)
[vite]     at node:internal/main/run_main_module:28:49
[vite] Node.js v20.10.0

[ERROR] vite has unexpectedly closed (exit code 1).

The Ionic CLI will exit. Please check any output above for error details.
```

Pondremos el comando “npm install vite@latest --save-dev” y damos enter .

```
administrador@Xubuntu22:~/Escritorio/Proyecto react final/ComicsMarvelsPools$ npm install vite@latest --save-dev
```

pondra el comando “npm cache clean --force” y damos enter

```
administrador@Xubuntu22:~/Escritorio/Proyecto react final/ComicsMarvelsPools$ npm cache clean --force
```

Si volvemos a poner ionic serve y damos enter, si sale el este sin el fallo significa que funciona.

```
administrador@Xubuntu22:~/Escritorio/Proyecto react final/ComicsMarvelsPools$ ionic serve
> vite --host=localhost --port=8100
[vite] + Local: http://localhost:8100/
[vite] + press h + enter to show help

[INFO] Development server running!

Local: http://localhost:8100

Use Ctrl+C to quit this process
```

BIBLIOGRAFÍA

JSON:

<https://www.youtube.com/watch?v=aJgAwjP20RY>

Imágenes por json:

<https://www.youtube.com/watch?v=ZwSwp8iRk2E&t=11s>

Routers, Navlinks y navegación:

https://www.youtube.com/watch?v=WATv_Md3y_o&list=WL&index=2&t=2s

<https://www.youtube.com/watch?v=QKyxDtIVGPY>

useState y useEffect hooks:

<https://www.youtube.com/watch?v=C7CZzM7ux0E&list=PL8VCLfCNXjSZfAhqZT7RfWv5Anb8l9PfY&index=4>

Para portada: herramienta canva

Quitar fondo:

<https://www.remove.bg/es>