

## 2º Práctica evaluable - DWEC

Nombre:

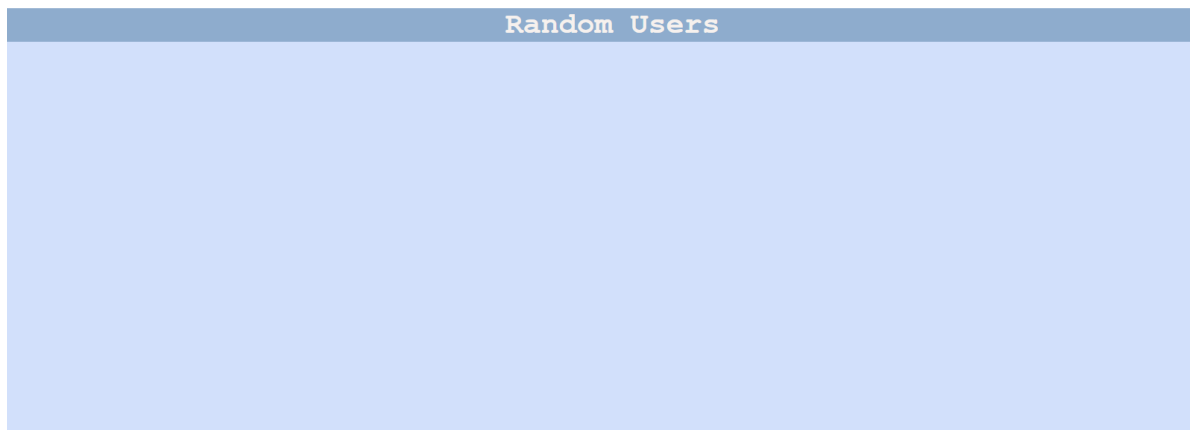
### EJERCICIO 1:

En esta tarea deberás crear un proyecto nuevo de vanilla usando **Vite** js.

- Elimina y limpia los archivos que creas necesario
- Añade los estilos facilitados en la tarea
- Copia el siguiente contenido en el **index.html**

```
<body>
  <h1>Random Users</h1>
  <div class="container"></div>
  <script type="module" src="/main.js"></script>
</body>
```

- Si toda va bien, se debe ver así:



# JUNTA DE ANDALUCÍA

## CONSEJERÍA DE EDUCACIÓN

### Curso 24/25

- Con la siguiente estructura html, crea todo mediante **js** y añade los estilos necesarios para que se aplicación se vea como sigue:
- Nota: la imagen facilitada se debe poner en la carpeta **public**

```
<body>
  <h1>Random Users</h1>
  <div class="container">
    <div class="card">
      
      <span>
        <strong>Name:</strong>
        " name surname"
      </span>
      <span>
        <strong>Mail:</strong>
        " mail"
      </span>
      <span>
        <strong>Phone:</strong>
        " phone"
      </span>
      <span>
        <strong>Location:</strong>
        " city"
      </span>
      <span>
        <strong>Current Time:</strong>
        " time"
      </span>
    </div>
    <button>GENERATE USER</button>
  </div>
```



# JUNTA DE ANDALUCÍA

## CONSEJERÍA DE EDUCACIÓN

### Curso 24/25

- Cada vez que se pulse, **GENERATE USER**, se deberá mostrar otro usuario diferente.
- Usando la API ofrecida en <https://randomuser.me/> debes obtener los siguientes datos de la respuesta obtenida:
  - name.first, name.last, email, phone, picture.large, location.city
- Se deberá crear una clase **Person**, con todos los atributos obtenidos en la respuesta anterior. El constructor debe tener como parámetros el **nombre y el apellido**. Se deben crear todos los getter y setter computados.
- Debes parsear los datos obtenidos con la API en un objeto **Person**, para esto deberás crear una **promesa**. Se de implementar la situación de rechazo.
  - Tras pulsar el botón



- El campo **Current Time** se debe obtener usando la siguiente API <https://rapidapi.com/BasedAPI/api/world-time-by-based-api> y debe mostrar la hora actual dependiendo de la localización del usuario que se esté mostrando.

#### Método de entrega:

- Se deben añadir todos los archivos necesarios en un zip, excluir la carpeta dist y node\_module.
- Cuando yo lo descargue y lo descomprima ejecutaré npm install y después npm run dev, si el proyecto no despliega la práctica no se evaluará

**JUNTA DE ANDALUCÍA**  
**CONSEJERÍA DE EDUCACIÓN**  
*Curso 24/25*

***Rúbrica de evaluación***

<b><i>Tarea</i></b>	<b><i>Puntuación</i></b>
El proyecto se ha creado con vite y se ha limpiado/eliminado todos los archivos innecesarios, al ejecutar el proyecto con npm todo funciona bien	0.5
Se crean todos los elementos html con su propiedades y estilos de forma correcta y la apariencia es igual a la imagen facilitada	1.5
Se recupera la información del usuario usando la API dada, mediante promesas	1.5
Se parsea la información del usuario a una clase, usando promesas, se debe implementar también la situación de rechazo	2
Cada vez que se pulsa el botón un nuevo usuario se carga	2
Se muestra la información de la hora actual dependiendo del usuario que se está mostrando	2
Se tendrá en cuenta el control de errores mediante try/catch	0.5
Implementación	

***Nota: El uso de buenas prácticas y la buena implementación se tendrán en cuenta***