



Pontificia Universidad Católica de Chile  
Escuela de ingeniería  
Departamento de ciencias de la computación

## Trabajo Individual 4

### IIC2513 Tecnologías y aplicaciones WEB

Primer semestre 2023  
Fecha límite de entrega: 7 de julio 2023, 20:00 hrs.

#### Indicaciones

##### Objetivo General:

Usted deberá crear una SPA con Vue, Svelte u otro framework JS distinto de React, siguiendo la estructura que se ve a continuación:

Diagrama de una SPA (Single Page Application) que muestra un formulario con los siguientes elementos:

- Título:** Un campo de texto con el valor "magnam facilis autem".
- Texto:** Un campo de texto con el valor "dolore placeat quibusdam ea quo vitae magni quis enim qui quis quo nemo aut saepe quidem repellat excepturi ut qui sunt ut sequi eos ea sed quas".
- id:** Un campo de texto con el valor "7".
- Cargar:** Un botón de acción.
- contador:** Un campo de texto con el valor "11".

##### Descripción del componente:

1. Hay un campo *título* y otro campo *texto* que NO son editables y son de tipo texto

2. Hay un campo *id*, editable, de tipo numérico
3. Hay un campo *contador*, No editable, de tipo numérico
4. Hay un botón "cargar"
5. Hay un botón de avance ">"
6. Hay un botón de retroceso "<"

### Comportamiento:

1. Al inicio de la aplicación, todos los campos de texto estarán vacíos y los numéricos tendrán valor 0.
2. Al presionar cargar, se debe traer TODOS los "posts" (son 100) desde la API externa que provee jsonplaceholder ubicada en: <https://jsonplaceholder.typicode.com/>
  - a. Para el uso de la API remitirse al sitio de ésta, en el link anterior y leer la documentación de ésta
3. Usted debe cargar la respuesta de la API en una estructura que le permita realizar las operaciones descritas en este enunciado.
4. El campo *contador* se incrementará en 1 cada vez que este es presionado y obtiene con éxito los datos provenientes desde la API
5. Presionado el botón cargar:
  - a. El campo *id* se deja en "1"
  - b. El campo *título* cargará el contenido dado por el atributo "title" para el elemento 1
  - c. El campo *texto* cargará el contenido del por el atributo "text" para el elemento 1
6. Al presionar el botón ">" (avance), se deberá cambiar "avanzar" en el siguiente elemento dentro de la estructura almacenada (del n, pasa al n+1), modificando valores de título, texto y id acorde al elemento "n+1"
7. Al presionar el botón "<" (retroceso), se deberá cambiar "retroceder" en el elemento anterior dentro de la estructura almacenada (del n, pasa al n-1), modificando valores de título, texto y id acorde al elemento "n-1"
8. También puede ingresar directamente un id en el campo *id* (valor de 1 a 100) y desplegar el elemento n-ésimo solicitado.
9. Cada vez que se presione cargar se DEBE invocar a la API y obtener la totalidad de los elementos
10. Usted debe controlar que no se puede retroceder más allá del elemento 1 ni avanzar más allá del elemento 100
11. Usted debe controlar que no se puedan ingresar elementos menores a 1 ni mayores a 100 en el campo *id*

### Manejo de errores, su aplicación en **front** debe al menos:

1. Evitar que se ingrese cualquier otro caracter que no sea número en el campo *id*.
2. Deben manejar eventuales errores de conexión con la API

### Estilos:

Los colores, fuentes, estilo de botones, campos de ingreso, etc. son a su elección, la imagen presentada en este enunciado es referencial y NO refleja el estilo de colores.

Se debe respetar los campos, botones y sus funcionalidades pero usted puede querer presentar etiquetas, botones, etc, de otra forma y desplegarlos con otra distribución en la pantalla

NOTA: debe considerar la usabilidad de su pantalla, debe ser de uso intuitivo, claro y orientado a la funcionalidad. (Si quiere puede replicar la pantalla que se entrega en el enunciado, no es problema)

### **Entrega del trabajo:**

Usted deberá entregar los archivos fuentes, documentación de instalación y uso y recursos que haya utilizado (imágenes, por ejemplo) en una estructura de directorio adecuada (se puntuará que sea una entrega ordenada) con nombres de archivo y directorios coherentes.

### **Sobre la corrección:**

Si utiliza chatgpt o copilot u otro, indíquenos (no está prohibido) si no lo indican y se detecta su uso, pueden ser evaluados con un 1.

Si usan chatgpt o copilot (u otro similar) y lo declaran, la corrección se fijará en otros detalles de su implementación al momento de evaluar. Por ejemplo, se considerarán aspecto de estilo, documentación, usabilidad, control de errores, orden en la entrega (carpetas, nombres de archivos), etc.

### **Detalles de la entrega:**

La entrega de todos los archivos se hará en un buzón de tareas en canvas. Se debe entregar un zip que contenga todos los archivos. En el nombre del zip se debe especificar el número de alumno y nombre y apellido, de la siguiente manera

12345678\_Parraguez\_Ricardo.zip

### **NO se aceptarán:**

- Entregas por mail (ya sea al profesor o ayudantes)
- Entrega en otro sistema que no sea el que se ha provisto para estos efectos

## **Recomendaciones**

---

1. Pregunten y consulten, usen foro, colaboren entre ustedes (NO COPIEN<sup>1</sup>). Los ayudantes están para apoyarlos

---

<sup>1</sup> La copia se sanciona con un 1 en el trabajo y la reprobación inmediata del curso

2. TODA referencia debe ser documentada. Por ejemplo si usaron stackoverflow o chatGPT, indíquelos. Si no lo indican será considerado plagio y la evaluación serpa de un 1.0
3. Trabajen con tiempo, no esperen a último momento para comenzar con la tarea o despejar dudas

## Dudas

---

Para que todo el curso se vea beneficiado, hagan sus preguntas sobre el material del curso, sobre tecnologías web, y sobre el proyecto a través de los foros del curso dispuestos para estos efectos. No se responderá ninguna duda de tareas por e-mail.