

# Conversando con máquinas, parte II

*“La pregunta de si las máquinas pueden pensar [...] es tan relevante como la pregunta de si los submarinos pueden nadar.”*

Edsger Dijkstra – The Threats to Computing Science.

## ¿Qué te gustaría comer hoy?

¡Te damos la bienvenida a la toolbox de formularios! Pero, un momento... ya se había tocado este tema anteriormente, ¿verdad?

En el primer sprint te presentamos la toolbox llamada “Conversando con máquinas”, en la cual aprendiste sobre formularios y las distintas etiquetas que se utilizan para crearlos. ¿Puedes recordarlas? Te dejamos [aquí](#) una lista de las más importantes.

En esta toolbox te estaremos mostrando cómo trabajar con etiquetas de formulario, pero manejando sus valores con React. Los formularios y sus distintos controles (inputs de texto, select, radio buttons) están presentes en todas las aplicaciones webs, ya que permiten establecer un diálogo entre la persona y el sistema. El formulario, como pieza de frontend, ofrece una forma de solicitar una respuesta, opinión, criterio de búsqueda o decisión sobre algún tema. Por ejemplo, ¿qué te gustaría comer hoy?

☐ Pizza 🍕

☐ Hamburguesa 🍔

☐ ¿Por que tengo que escoger solo una? 😞



## Formularios en React

En React, el valor de los formularios y las etiquetas debe ser controlado a través del concepto de [estado](#). En la toolbox anterior te dimos un ejemplo de esto, cuando manejamos el estado del checkbox con React. Al tener la información de los inputs en el estado, vamos a poder programar decisiones que consideren distintos tipos de criterio, por ejemplo, validar que un input solo acepte letras en vez de números.

Un formulario de texto simple con un placeholder que dice "Ingresa tu nombre". El formulario está encerrado en un recuadro gris claro.

Cada tipo de input tendrá que ser manejado acorde a su tipo de valor: el checkbox será manejado con un valor booleano, mientras que el input de texto, tendrá que ser manejado con valor de string.

### Inputs de texto

Para manejar el estado en inputs de texto tendremos que valernos del objeto evento para acceder al valor escrito en la caja de texto creada por la etiqueta `<input type=text />`. Todo lo que se escriba en un input de texto podrá ser leído desde el objeto evento en `event.target.value` (asumiendo que se le ha dado el nombre `event` al objeto evento).

En el siguiente video te mostramos cómo manejar el estado para un input de texto.

[El estado como string](#)

### Inputs de fechas

Manejar el estado de un input `type=date` consiste en un proceso similar al anterior, con la diferencia de que el dato proveniente de `event.target.value` (un string representando una fecha) lo tendremos que transformar a una `Date` con el constructor de `new Date()`, el cual viste en la meeting de composiciones.



Este manejo de eventos te será muy útil en el proyecto de reserva de hoteles, donde tendrás que filtrar resultados según un rango de fechas. En el siguiente video te contamos cómo trabajar con inputs de fecha en React.

[El estado como fecha](#)

## Selects

Como recordarás del Sprint 1, la etiqueta `<select>` es la que te permite presentar un listado de opciones a través de una vista desplegable. Si bien su implementación con HTML es distinta a la etiqueta `input`: Vamos a poder acceder al valor seleccionado de la misma forma que en el ejemplo del `input` de texto.

...

```
<select>

  <option value="manzana">Manzana</option>

  <option value="naranja">Naranja</option>

  <option value="uva">Uva</option>

</select>
```

...

Dale un vistazo al siguiente video donde te contamos cómo hacerlo:

[El estado en la etiqueta <select>](#)

## Extra: producción gráfica funcional

Al poder usar Javascript dentro de nuestros componentes de React, vamos a poder mezclar la potencia del lenguaje para parametrizar elementos y poder abstraer en funciones su producción gráfica.



La técnica de la producción gráfica funcional consiste en crear funciones internas a un componente, que retornen JSX, para luego invocarlas en el return del mismo componente. Presta atención al siguiente ejemplo:

[Producción gráfica funcional](#)

## Extra: renderizado condicional en inputs

Esta técnica consiste en aprovechar la lógica condicional de Javascript para mostrar y ocultar componentes, o partes gráficas de nuestra interfaz según una condición. Observa la sintaxis en el siguiente video:

[Renderizado condicional](#)

## En resumen

En la toolbox de hoy te dimos una introducción al manejo de formularios con React. Identificamos que en React, toda la data que pertenece a los distintos tipos de inputs, son estados y también profundizamos en las diferentes formas que toma el estado, más allá del valor booleano.

Como extra, te presentamos dos técnicas para mostrar y ocultar elementos dentro de los componentes: la producción gráfica funcional, que te permite crear funciones internas en los componentes para generar porciones de la interfaz gráfica, y el renderizado condicional, que te permite definir qué mostrar según una condición booleana.


## ¡Prepárate para la próxima meeting!

### Profundiza

 [Documentación oficial de formularios en React](#)



### Challenge

 En el proyecto **Taskineitor**, construye y maneja con React los estados para los siguientes inputs:

- Inputs de fecha (desde / hasta)
- Select de categorías
- Select de estados de tarea (completada / no completada)

Asegúrate de manejar los valores de estos inputs con React. Trae tu proyecto a la próxima meeting.

### Potencia tu Talento - Portales de empleo



#### Ofertas laborales

Si quieres comenzar a trabajar dentro de tu rama de estudio es importante que investigues las ofertas de puestos disponibles. Ya hemos recorrido tus opciones de desarrollo, los referentes de la industria y tienes un primer proyecto para demostrar lo que sabes.

Revisar las ofertas de empleo disponibles te permitirá identificar qué aspectos es mejor resaltar en tu perfil, te dará información muy valiosa para pensar tu perfil de LinkedIn, CV y Portfolio de manera más estratégica.



#### Portales de empleo

Aquí te proponemos que indagues portales de empleo, no todas las ofertas están en LinkedIn y te puedes estar perdiendo de oportunidades muy interesantes si solo te quedas allí.

Además recuerda que en nuestro Slack de estudiantes tenemos el canal #oportunidades-laborales abierto a toda la comunidad, para compartir búsquedas de las distintas áreas de estudio de Acámica.

También puedes investigar plataformas que ofrecen oportunidades no tradicionales como Topcoder.



[Portales de empleo](#)



[Trabajo remoto: construyendo una carrera con TopCoder](#)

