UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

React 03

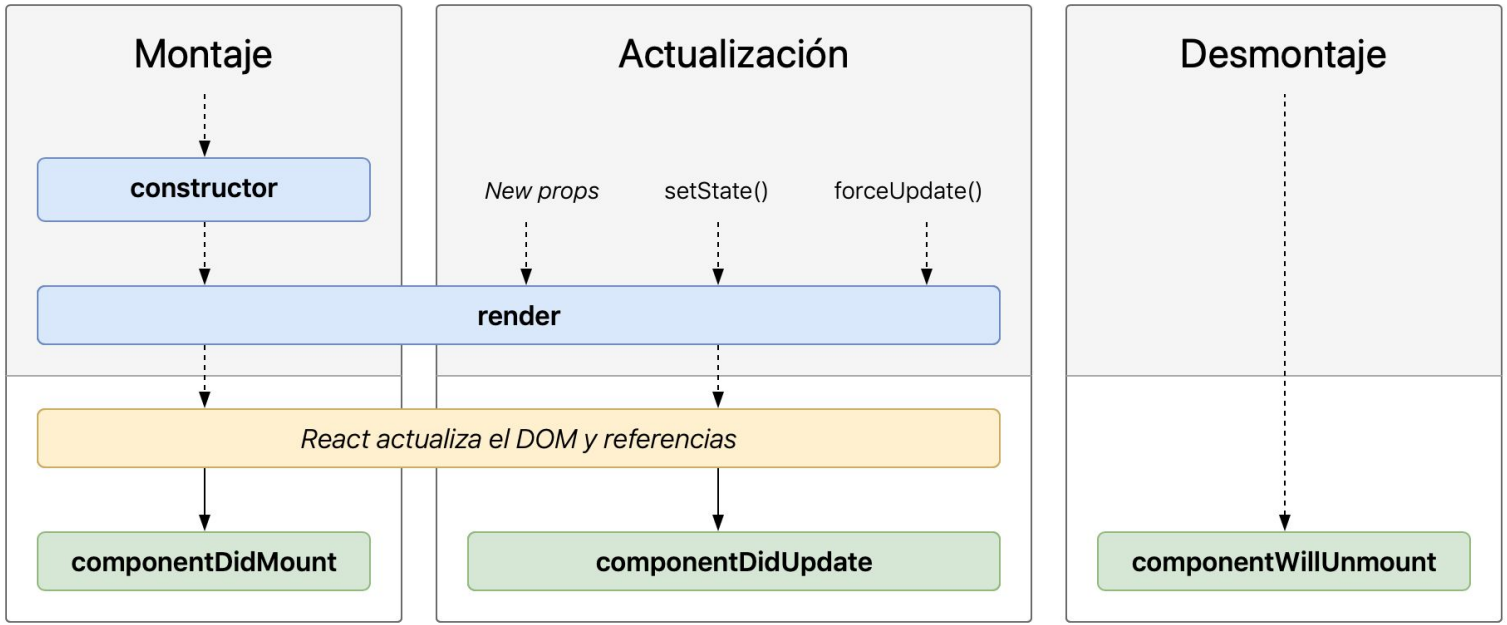
Ciclo de vida de los componentes

"Fase Render"

Pura y sin efectos co-laterales, puede ser pausada, abortada o reiniciada por React.

"Fase Commit"

Puede operar sobre el DOM, ejecutar side-effects, agendar actualizaciones.



Montaje

Es el proceso que involucra la instanciación del componente e incorporar dicho componente dentro del DOM (estructura del documento)

Montaje

Cuando el componente se inicia por primera vez sigue el siguiente proceso:

1. Se llama al constructor()
2. Se llama al método render()
3. Se llama al método componentDidMount()

constructor

Solo creamos un constructor si

- Deseamos inicializar el estado del componente (usando this.state = {};)
- Deseamos enlazar (bind) los métodos que se llaman desde eventos al componente (para utilizar this)

render

El método render() es obligatorio en los componentes

Consideraciones:

- NO se debe modificar el estado
- Devuelve siempre el mismo resultado cada vez que se lo invoca
- No interactúa directamente con el navegador
- Retorna JSX
- Puede hacer uso de this.props y this.state

componentDidMount

Este método es utilizado para obtener registros de un servidor remoto o necesitemos inicializar alguna propiedad usando como referencia el espacio que ocupa el componente o algún elemento visual.

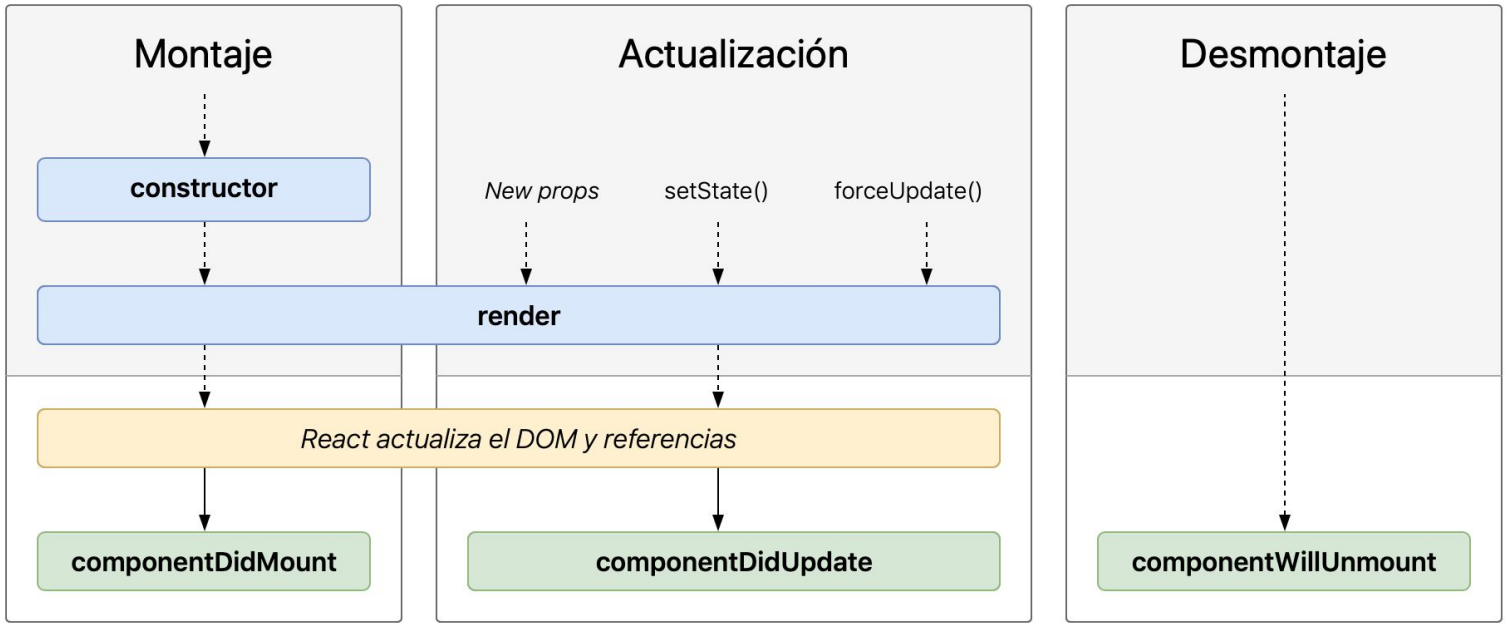
En caso de realizar un cambio en el estado, debemos llamar a this.setState();. Esto provocará que se ejecute el método render nuevamente.

"Fase Render"

Pura y sin efectos co-laterales, puede ser pausada, abortada o reiniciada por React.

"Fase Commit"

Puede operar sobre el DOM, ejecutar side-effects, agendar actualizaciones.



Actualización

En los siguientes casos se produce una actualización

- Llamamos a setState
- Se pasa una nueva propiedad o se modifica una propiedad
- Llamamos a forceUpdate() para forzar la actualización

componentDidUpdate

Este método es llamado cuando se produjo una actualización del estado (no es llamado la primera vez, en el montaje)

Puede ser utilizado para hacer peticiones de red adicionales (Ej: si se seleccionó un nuevo cliente, las direcciones del nuevo cliente)

En caso de realizar una petición y actualización del estado, es importante incluir una condición (para no crear un bucle infinito)

componentDidUpdate

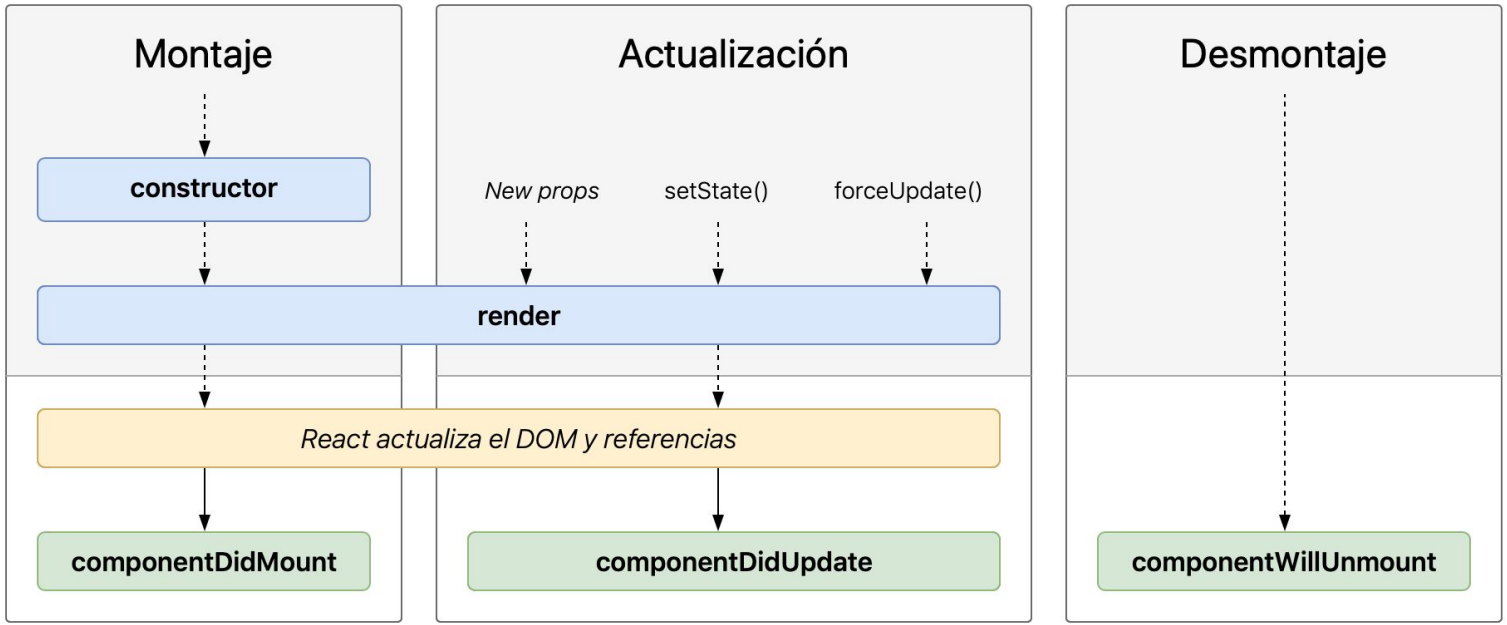
```
componentDidUpdate(prevProps) {
  // Uso tipico (no olvides de comparar los props):
  if (this.props.userID !== prevProps.userID) {
    this.fetchData(this.props.userID);
  }
}
```

"Fase Render"

Pura y sin efectos co-laterales, puede ser pausada, abortada o reiniciada por React.

"Fase Commit"

Puede operar sobre el DOM, ejecutar side-effects, agendar actualizaciones.



Desmontaje

Es el proceso por el cual se elimina el componente del DOM y se destruye el componente

Antes de destruir el componente se llama al método componentWillUnmount()

componentWillUnmount

Es el método llamado antes de destruir el componente

En caso que hayamos realizado alguna suscripción o abierto una conexión persistente, aquí debemos cancelar todos los procesos que hayamos abierto

Práctica

Agregar al proyecto el método componentDidMount() con un console.log(‘Paso componentDidMount’); para ver su ejecución

Conexión con servidores remotos

Conexión HTTP

Para realizar conexiones HTTP con un servidor, necesitamos instalar algún componente para dicha función

```
npm install --save axios
```

Uso de axios

```
import axios from 'axios';

...

export default class MiComponente extends React.Component {
    async componentDidMount() {
        var respuesta = await axios.get('https://<server endpoint>');

    }

}
```

Verificar el código de respuesta del servidor

```
var respuesta = await axios.get('');
...
respuesta.status
```

Acceder a los datos

```
var respuesta = await axios.get('');
...
respuesta.data
```

Manejo de la respuesta

```
state = {
    listado: []
}
async componentDidMount() {
    var respuesta = await axios.get('https://jsonplaceholder.typicode.com/comments');
    if (respuesta.status===200) {
        console.log('ok');
    }
    this.setState({listado: respuesta.data});
}
```

Práctica

Implementar en el componentDidMount la llamada al servidor remoto (y que la misma actualice la interfaz)

Estructura sugerida

Estructura sugerida

Crear por cada componente una carpeta, y que cada carpeta tenga un componente adentro (siguiendo la estructura de anidación)

Práctica

Práctica 01

Crear uno o varios componentes que se conecten a tu servidor express de Tareas y que muestre el listado de todas las tareas cargadas en el servidor.