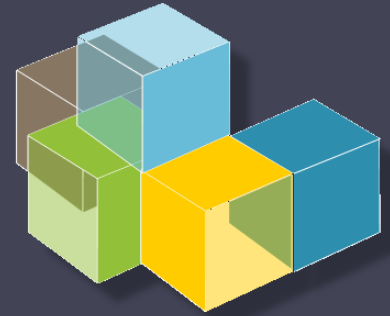


Universidad de Oviedo



Escuela de
Ingeniería
Informática



ARQUITECTURA
DEL SOFTWARE

Arquitectura del Software

Lab. 12

Monitorización y evaluación de rendimiento (profiling)

2019-20

Jose Emilio Labra Gayo
Pablo González
Irene Cid
Hugo Lebrede

Monitorización y profiling

Monitorizar: Observar comportamiento de un software

Cuadros de mando

Habitualmente, después del despliegue

Profiling (caracterizar): Medir rendimiento de un software mientras se ejecuta

Identificar partes que contribuyen a un problema

Mostrar dónde centrar los esfuerzos para mejorar rendimiento

Suele hacerse antes del despliegue

Monitorización & profiling

Monitorizar una aplicación mientras se ejecuta
Registrar uso de CPU, memoria, hilos, etc.

JavaScript:

Chrome (Timeline), Firefox Developer Edition
(Performance tool), Microsoft(Ajax View)

Herramientas de servidor:

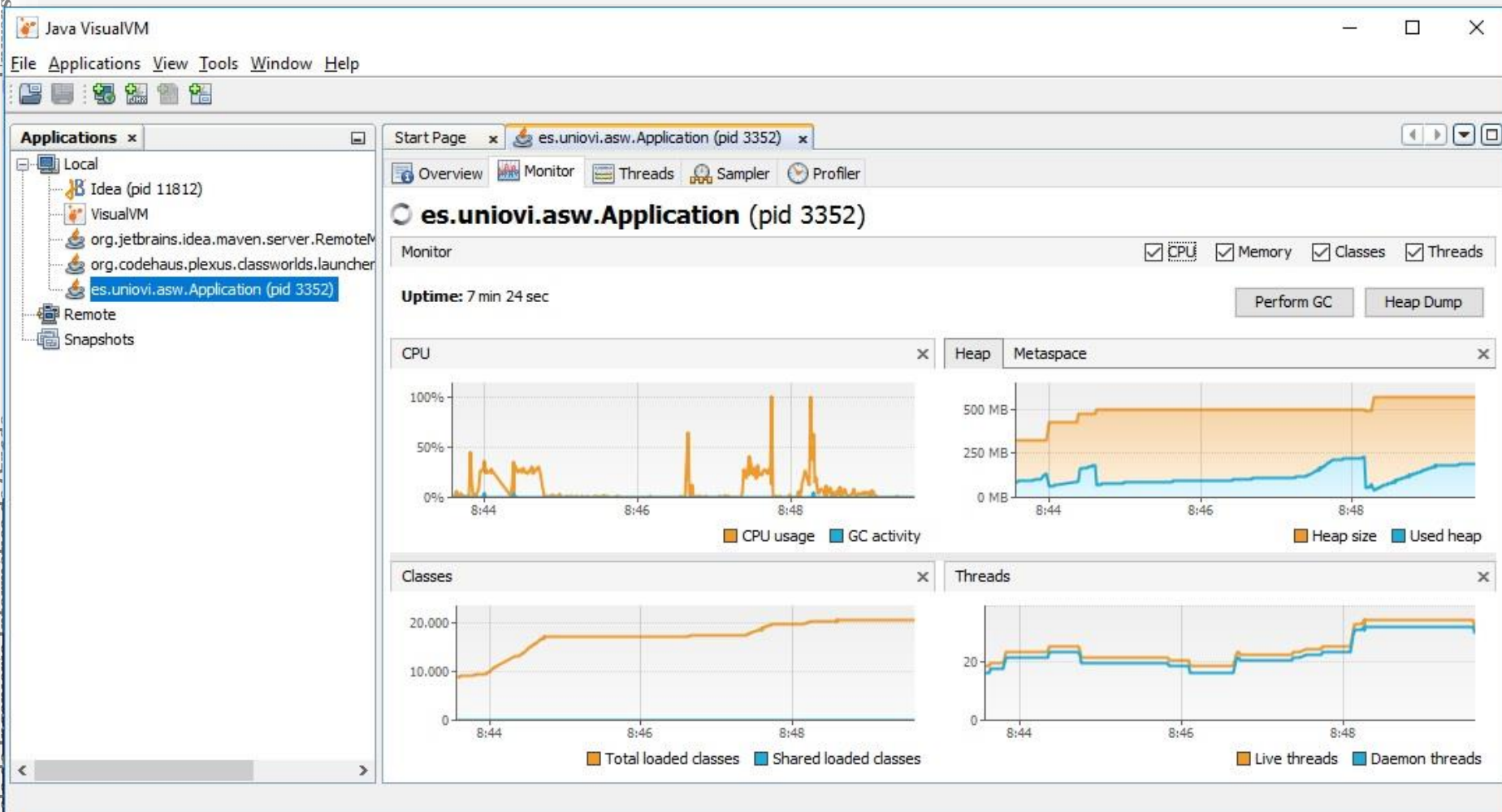
VisualVM, JProfiler, YourKit, etc.

VisualVM

<https://visualvm.github.io/>

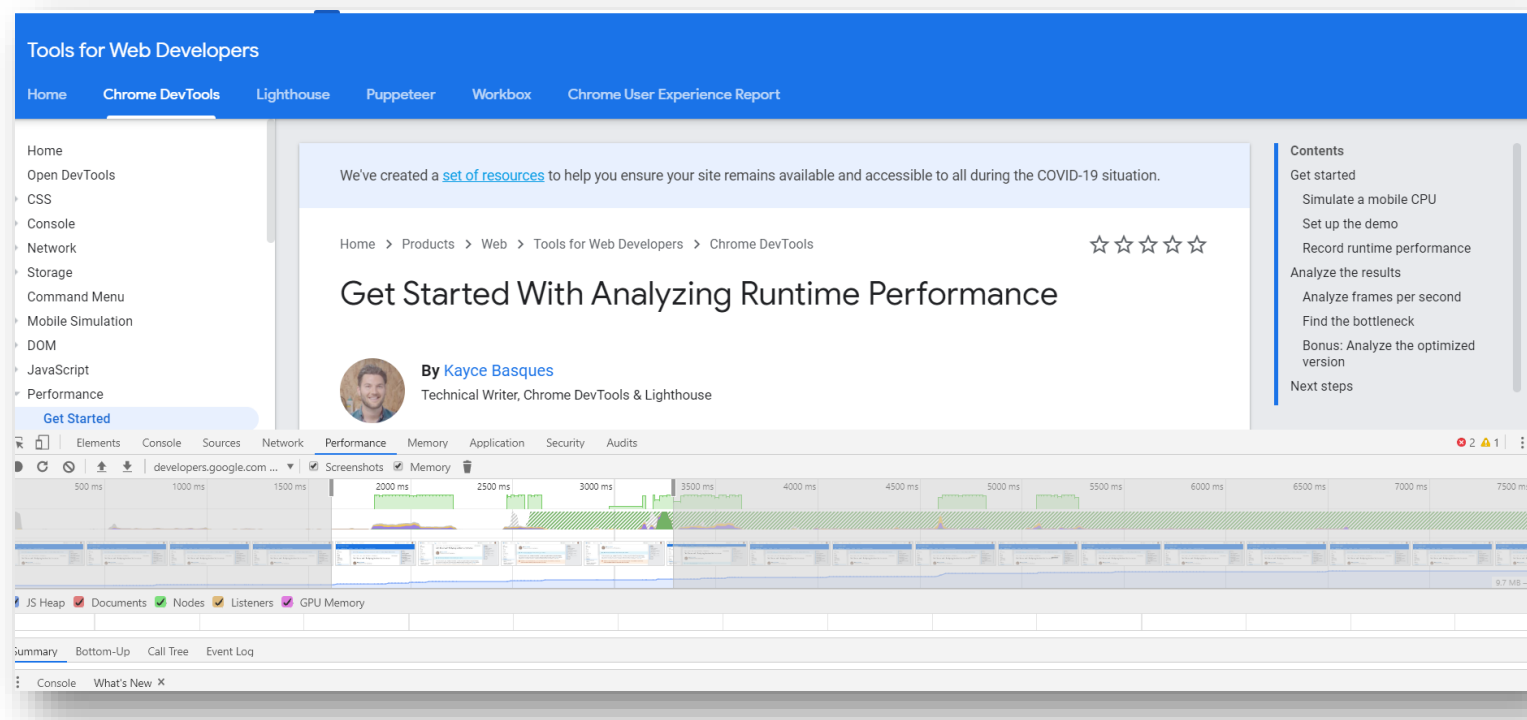
Ya está instalada con el JDK: `jvisualvm`

Server/Java: VisualVM



Browser: developer tools

- Monitorizar/chequear rendimiento



<https://developers.google.com/web/tools/chrome-devtools/evaluate-performance>

Ejemplo con Google Chrome

Modo incógnito

En la esquina superior derecha, click en los tres puntos y nueva ventana incógnito

Windows, Linux, or Chrome OS: Ctrl + Shift + n.

Mac: ⌘ + Shift + n.

Chrome DevTools

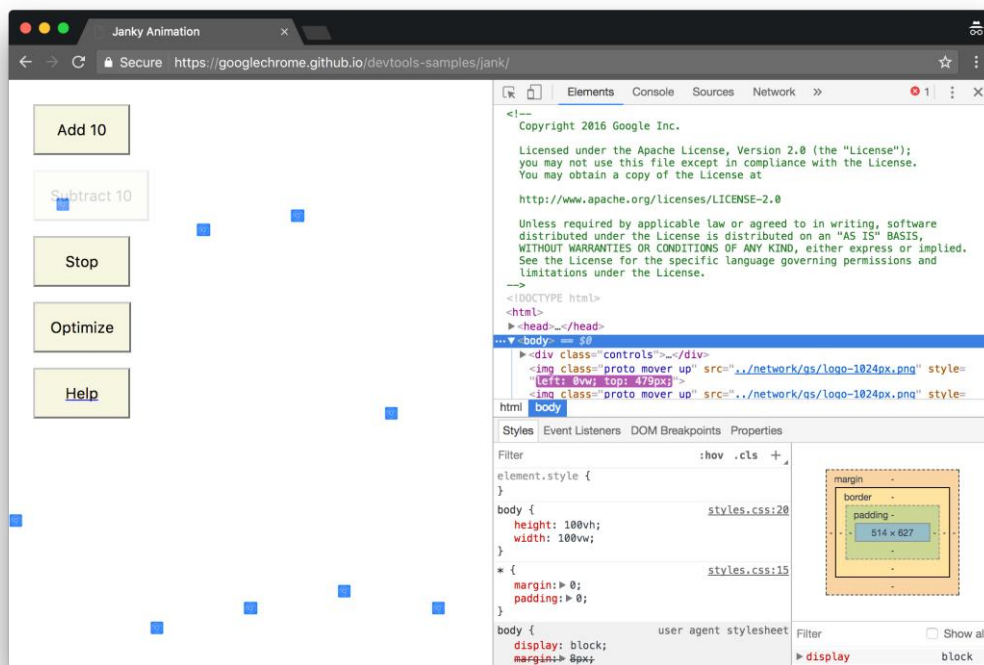
Windows, Linux: Control+Shift+I

Mac: Command+Option+I

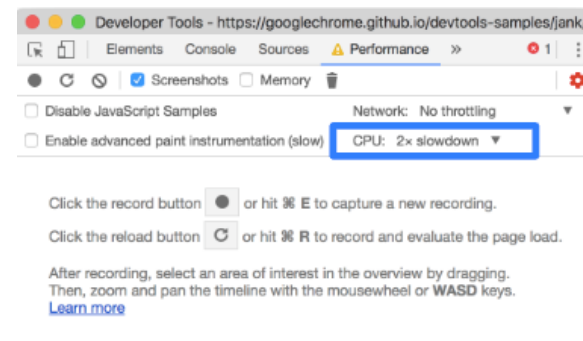


Ejemplo con Google Chrome

<https://googlechrome.github.io/devtools-samples/jank/>



Performance > CPU > 2 x Slowdown

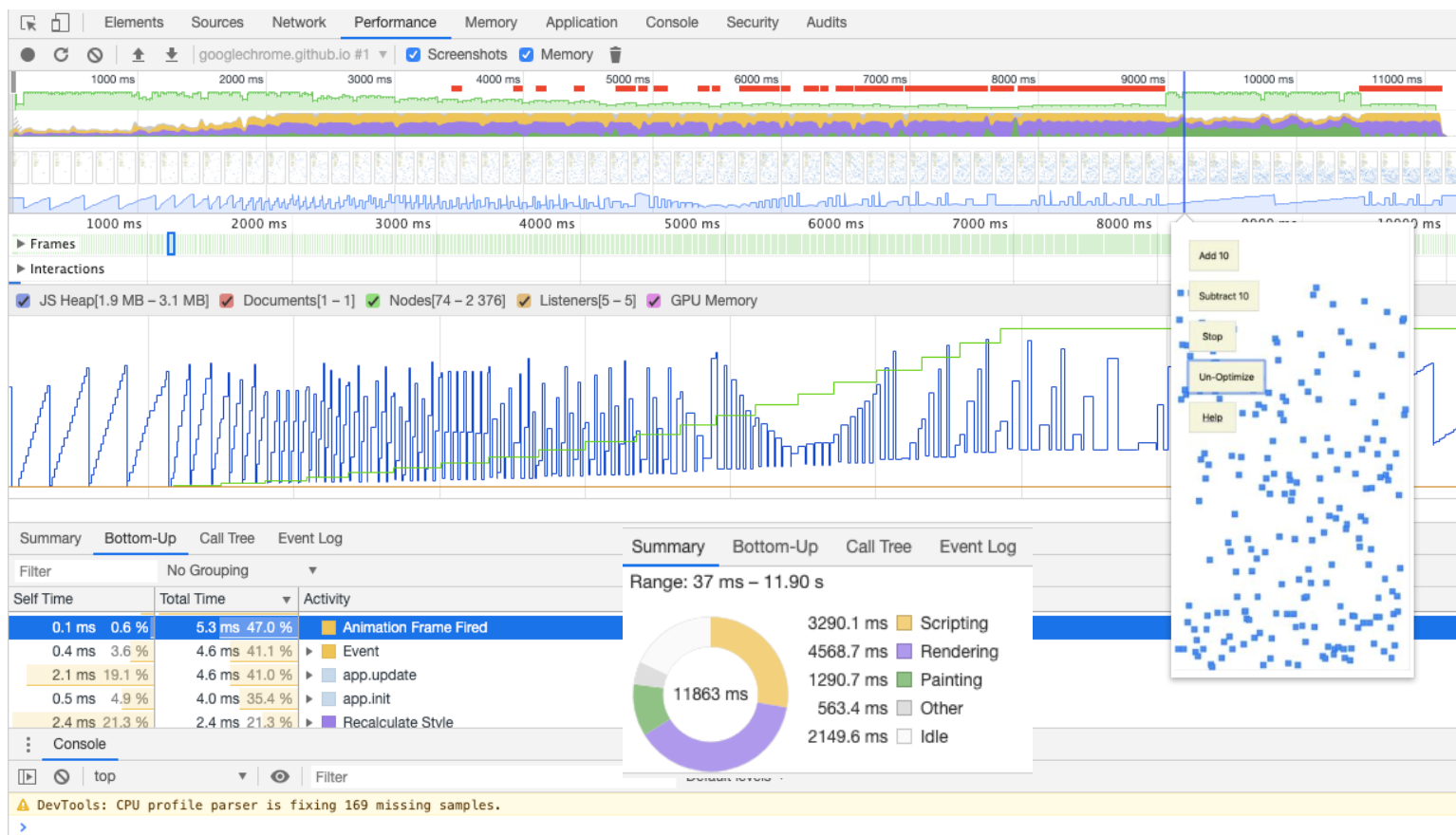


Performance > Record
 click Add 10 (20 veces)
 Optimize / Un-optimize
 Stop

Ejemplo con Google Chrome

Profile:

Frames per Second → □
CPU → □



Cuello de botella → □

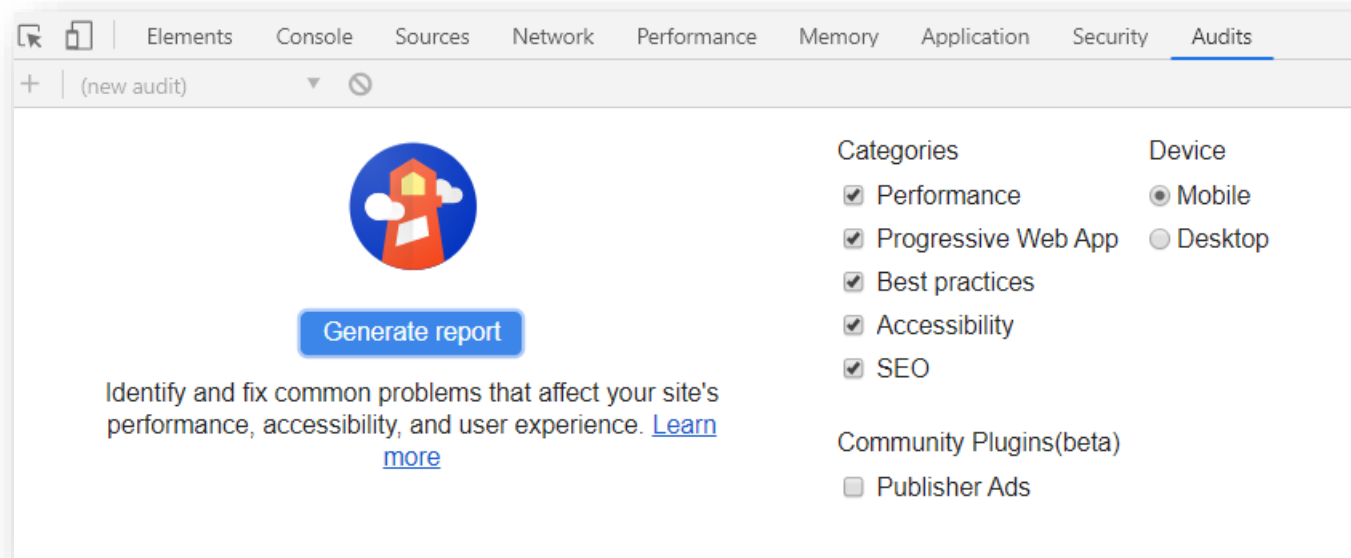
Otras herramientas browser

RAIL model (Response, Animation, Idle, Load)

<https://developers.google.com/web/fundamentals/performance/rail>

<https://webpagetest.org/easy>

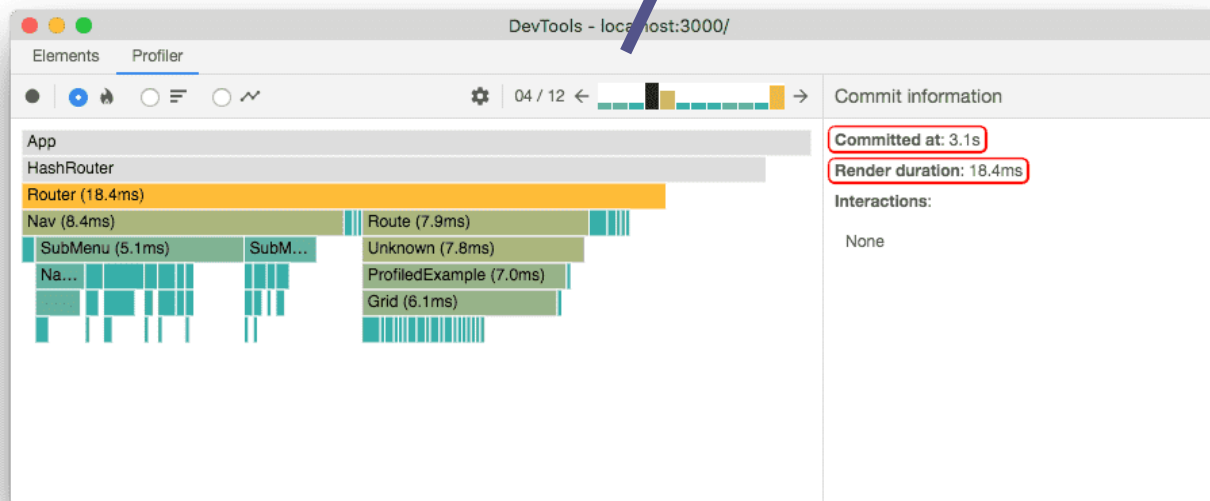
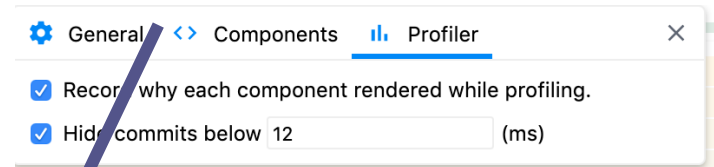
Lighthouse (viene con Chrome)



React Developer Tools

React trabaja en dos fases:

- Render
- Commit



React Developer Tools

The screenshot shows the React Developer Tools Profiler interface. The top part displays a form with the following fields:

- Dirección: avda galicia
- Locality: Oviedo
- Postal Code: 33005
- Region: Asturias

The Profiler panel is open, showing a list of renders. The first render is highlighted in orange:

- Xd key="subject:._:userprofile_shex_UserProfileAddress__parts_4" (0.2ms)

The right-hand pane shows the details for this render:

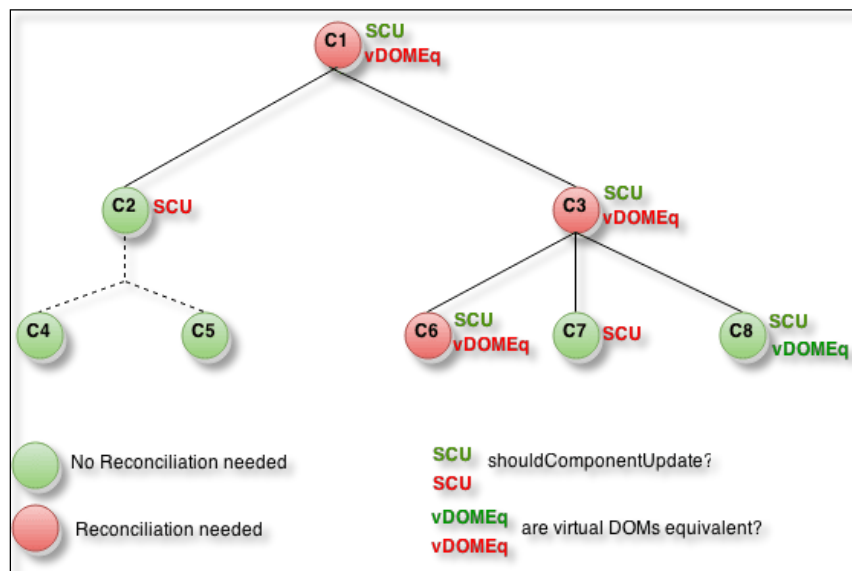
Why did this render?

- Props changed: (fieldData, modifyFormObject, formObject, onSave)

Rendered at:

- 4.1s for 17.3ms
- 4.5s for 83.8ms
- 4.6s for 19.9ms
- 4.7s for 19.9ms
- 5s for 16.1ms** (highlighted in blue)
- 6.6s for 21.4ms

React DOM - Virtual DOM



```

class CounterButton extends React.PureComponent {
  constructor(props) {
    super(props);
    this.state = {count: 1};
  }

  render() {
    return (
      <button
        color={this.props.color}
        onClick={() => this.setState(state => ({count: state.count
+ 1}))}>
        Count: {this.state.count}
      </button>
    );
  }
}
  
```

```

shouldComponentUpdate(nextProps, nextState) {
  if (this.props.color !== nextProps.color) {
    return true;
  }
  if (this.state.count !== nextState.count) {
    return true;
  }
  return false;
}
  
```

Alternativas monitorización Web

- Spring-boot proporciona “Actuator” para características en producción
- Algunos systems:
 - Prometheus, Graphite, Grafana, Datadog, Nagios, Sensu, ...



Presentando la arquitectura

Ideas para la presentación

Presentación de la arquitectura

Para este curso

Presentación: 15-20'

Preguntas: 5-15'

Los profesores seleccionan al profesor

¿Qué presentar?

Enfocarse en aspectos claves

- Presentar arquitectura y sistema
- Atributos de calidad/estrategias para resolverlos
- Demo
- Pruebas (pruebas de aceptación, carga,...)

Presentando arquitecturas

DeveloperToArchitect.com

Software Architecture Monday with Mark Richards Lesson 31 - Presenting Architecture



Mark Richards

Independent Consultant

Hands-on Software Architect / Published Author / Conference Speaker

Founder, DeveloperToArchitect.com

www.wmrichards.com

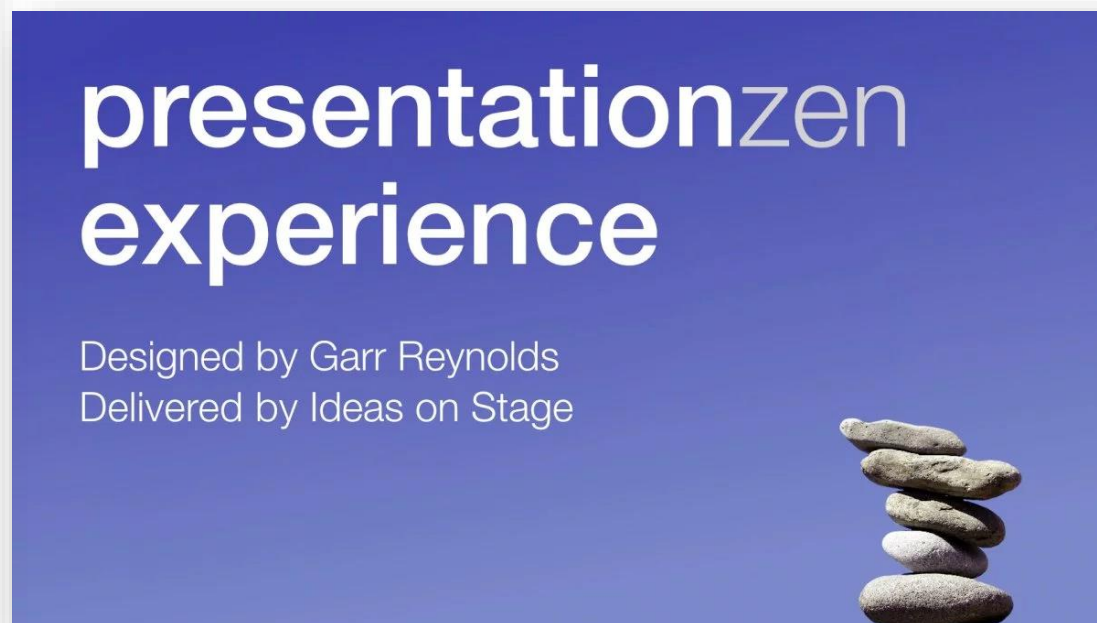
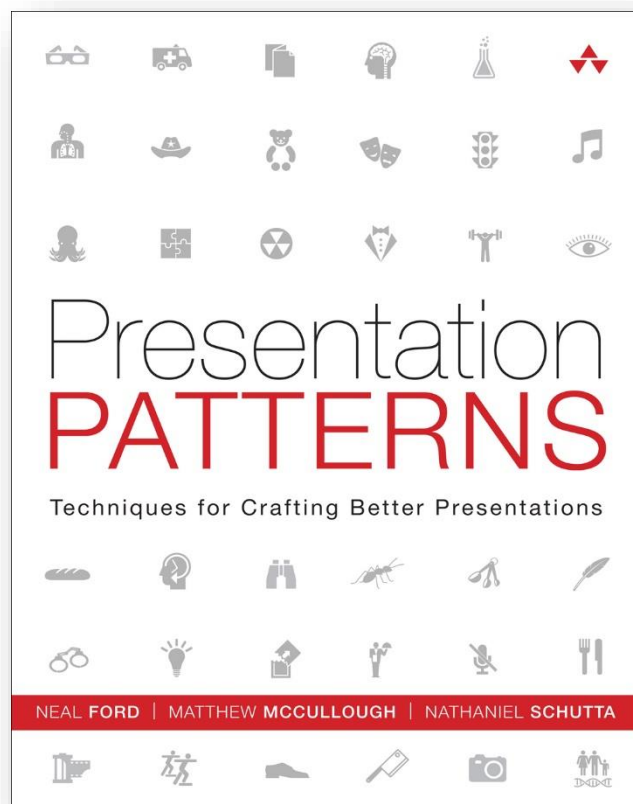
<https://www.youtube.com/watch?v=pJcol2DASpo&t=299s>

Mejorar las presentaciones

<https://presentationpatterns.com/resources/>

<https://www.oreilly.com/library/view/presentation-patterns/9781491954980/>

<https://www.presentationzen.com/>



Bibliografía y enlaces

- Monitorización y profiling
 - [Get Started With Analyzing Runtime Performance](https://developers.google.com/web/tools/chrome-devtools/evaluate-performance/)
<https://developers.google.com/web/tools/chrome-devtools/evaluate-performance/>
 - [How to Use the Timeline Tool](https://developers.google.com/web/tools/chrome-devtools/evaluate-performance/timeline-tool#profile-js)
- <https://developers.google.com/web/tools/chrome-devtools/evaluate-performance/timeline-tool#profile-js>
- Presentaciones
 - [Presentation Zen](https://www.presentationzen.com/) Garr Reynolds
<https://www.presentationzen.com/>
 - <https://www.amazon.com/gp/product/0321811984>