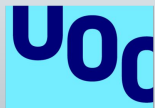


Aplicación web de katas de código



UOC

Universitat
Oberta
de Catalunya

David Marchena Blanco
Grado de Ingeniería Informática
Desarrollo web

Gregorio Robles Martínez
Santi Caballe Llobet



12/06/2020



Because JavaScript can be used without understanding, the understanding of the language is often never attained.

Kyle Simpson

“You Don't Know JavaScript: Up and Going”



1

Objetivo



ARIKATA

Una aplicación para
adquirir y afianzar
conocimiento





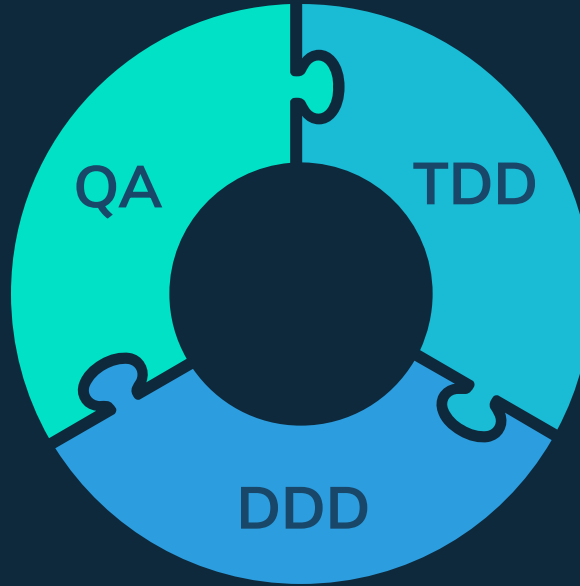
2

Otros objetivos

- UX, en especial en el editor de código
- Rendimiento web
- Desarrollar sobre una infraestructura similar a producción
- Desplegar en producción
- Poner a prueba conocimientos del Grado
- Aprendizaje y experiencia

Quality Assurance

- Delegar responsabilidades del programador a automatizaciones
- Consistencia de código
- Bajar tasa de errores



Test-driven Development

- Robustez del código
- Facilitar refactorizaciones
- Experiencia más real del ratio beneficios / coste

Domain-driven Design

- Desacoplar lógica de aplicación de detalles técnicos
- Aprendizaje aprovechando un desarrollo desde cero

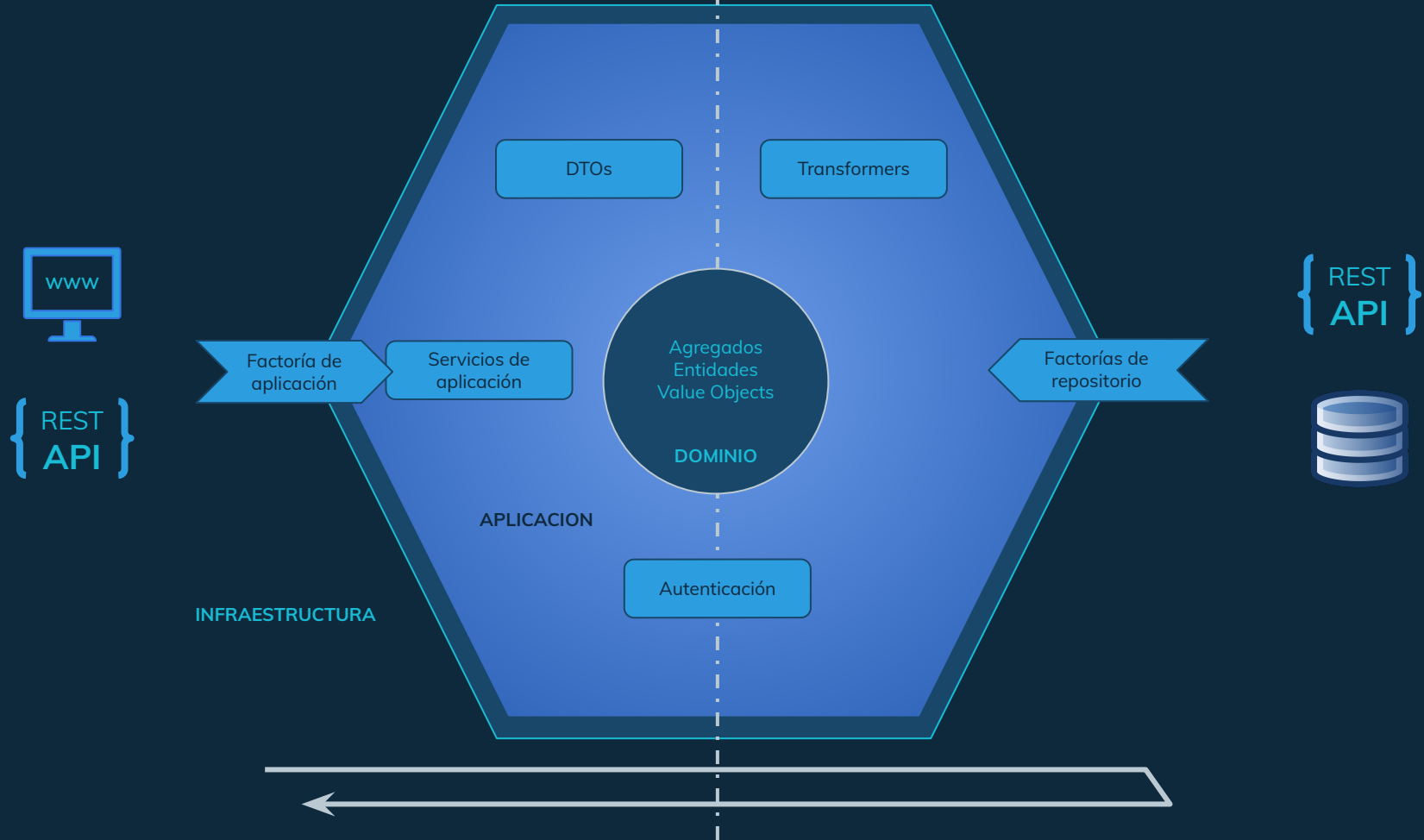
3

Arquitectura



Interfaz de usuario

Origen de datos





{ REST API }

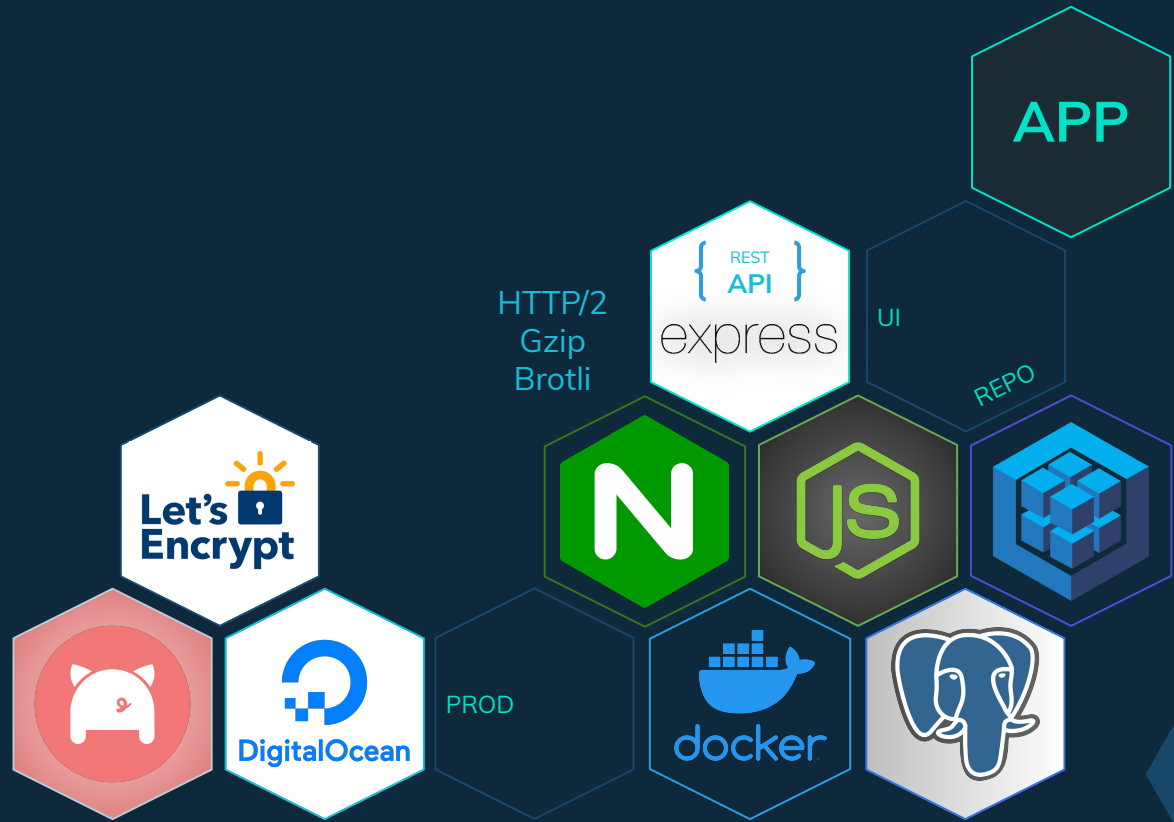


Front end

ES2020

CSS 2020/CR






Back end



4

Desarrollo

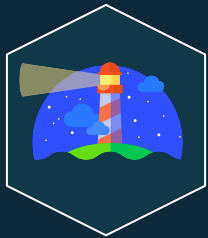
Entorno de desarrollo
Quality Assurance
Test-driven Development



```
dmarchena@macbook ~/dev/arikata master
```



Quality Assurance + Test-driven Development



Objetivo: 90-100

Conseguido

The image shows a browser window with two panes. The left pane displays a dark-themed website with the text "IMPROVE YOUR JS SKILLS" and "Practice and repetition are the way of the JS developer". A yellow square highlights the "JS" in "JS SKILLS". Below the text is a button that says "Show me the katas!". The right pane shows the Lighthouse performance tool interface, which includes a "Generate report" button, a description of the tool's purpose, and several configuration sections: "Categories" (Performance, Progressive Web App, Best practices, Accessibility, SEO), "Community Plugins(beta)" (Publisher Ads), and "Device" (Mobile, Desktop).

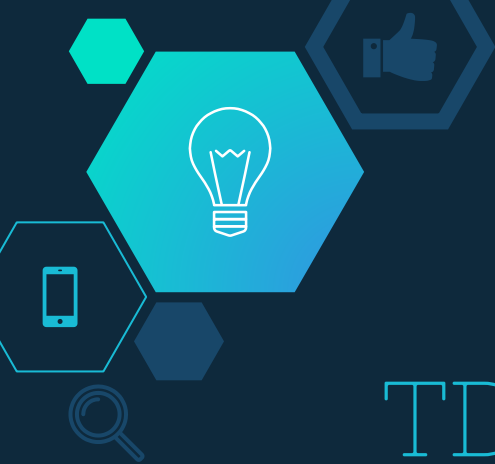
Optimización de rendimiento web



5

Conclusiones

Objetivo cumplido, a pesar de ser un MVP



TDD



Test-driven Design

Mejor definición de datos de entrada y respuesta.



Detección rápida de errores

Favorece refactorizaciones al perder el miedo a causar errores.



Impacto en la planificación

El coste a veces supera el de la funcionalidad.



Objetivo de cobertura 80%

Obtenido: 62-66%



DDD



Separación de intereses



Independencia de infraestructura

Facilita las migraciones técnicas y el cambio de modelo cliente-servidor a otro.



Añade complejidad

A veces la solución es más simple y mejor.



Mejoras

- ◇ Gestión de requisitos. Mejorar su visibilidad.
- ◇ Utilizar sprints para seguimiento de planificación.

Futuro

- ◇ Soporte Markdown
- ◇ Solución propuesta
- ◇ Estadísticas





¡Gracias!

En especial a Arrate y Lur por su paciencia y apoyo.

Podéis escribirme a dmarchenab@uoc.edu

<https://arikata.dev>

<https://github.com/dmarchena/arikata>

