

TDDiendo la Arquitectura

nicopaez@computer.org
@inicopaez



Versioning Strategy for DevOps Implementations

Construcción de software: una mirada ágil

Nicolás Paez • Diego Fontdevila • Pablo Suárez
Carlos Fontela • Marcio Degiovannini • Alejandro Molinari



EDUNTREF



UNTREF UNIVERSIDAD NACIONAL
DE TRES DE FEBRERO

Nicolás Paez
Department of Science and Technology
Universidad Nacional de Tres de Febrero
Caseros, Argentina
nicopaez@computer.org

Flipped Classroom Experience Teaching Software Engineering

Technical and Organizational Agile Practices: A Latin-American Survey

Nicolás Paez^(✉), Diego Fontdevila, Fernando Gainey,
and Alejandro Oliveros

Nicolás Martín Paez
Department of Science and Technology
Universidad Nacional de Tres de Febrero
Saenz Peña, Buenos Aires, Argentina
nicopaez@computer.org

Universidad Nacional de Tres de Febrero, Caseros, Argentina
nicopaez@computer.org,
{dfontdevila, fgainey, aoliveros}@untref.edu.ar

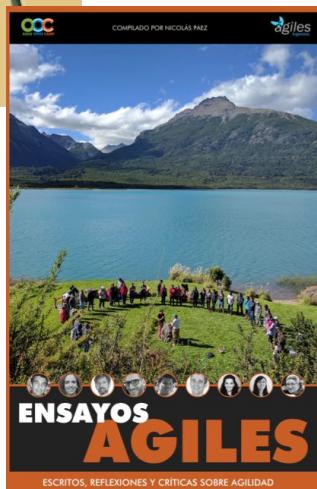
Characterizing Technical and Organizational Practices in the Agile Community

Nicolás Paez, Diego Fontdevila, Alejandro Oliveros

Departamento de Ciencia y Tecnología

Universidad Nacional de Tres de Febrero

nicopaez@computer.org, dfontdevila@untref.edu.ar, aoliveros@untref.edu.ar



HELENA Study: Initial Observations of Software

Development Practices in Argentina

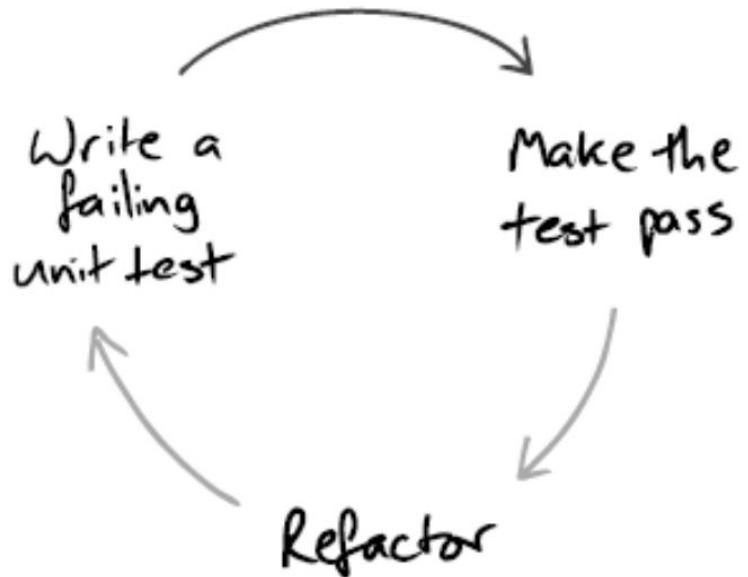
Nicolás Paez, Diego Fontdevila and Alejandro Oliveros

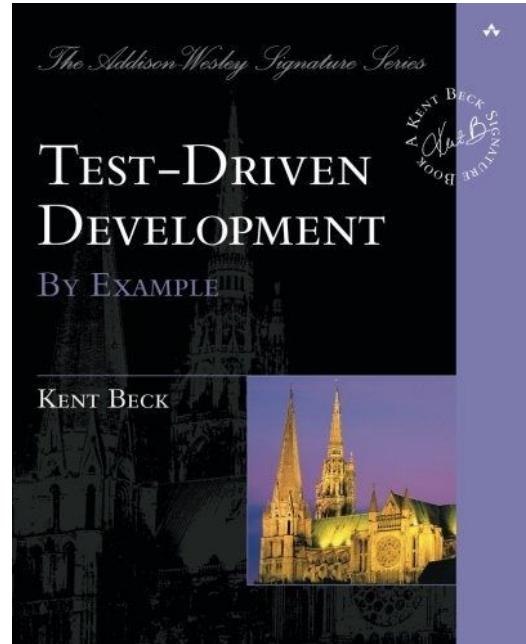
Universidad Nacional de Tres de Febrero, Caseros, Argentina
nicopaez@computer.org, {dfontdevila, aoliveros}@untref.edu.ar

¿qué sabemos de TDD?

<https://strawpoll.com/h8x47234>

Test-Driven Development







Software Engineering Radio

The Podcast for Professional Software Developers

[Home](#)[About SE-Radio](#)[Team](#)[Advertise](#)[Recognition](#)[Contact](#)[Legal](#)

Episode 167: The History of JUnit and the Future of Testing with Kent Beck

Filed in [Episodes](#) by [SE-Radio](#) on September 26, 2010 • 28 Comments

Recording Venue: Skype

Guest: Kent Beck

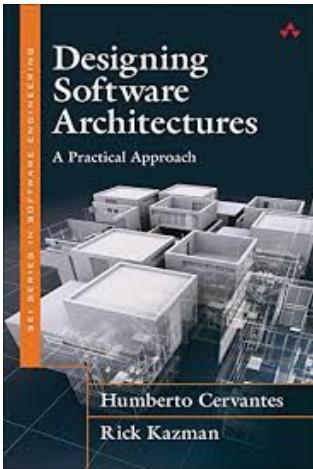




Arquitectura

Arquitectura

Decisiones relevantes de diseño



Design Purpose

Quality Attributes

Primary Functionality

Architectural Concerns

Constraints

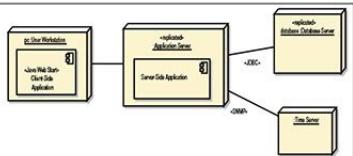
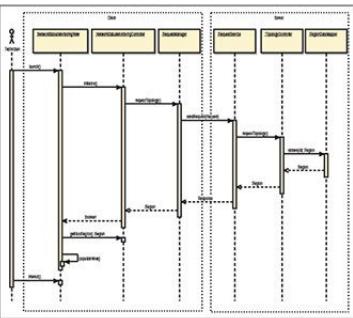
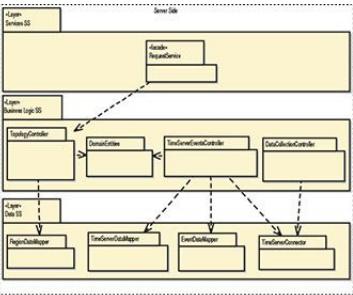
Architectural Drivers

Design Concepts

<<selects and instantiates>>

<<uses>>

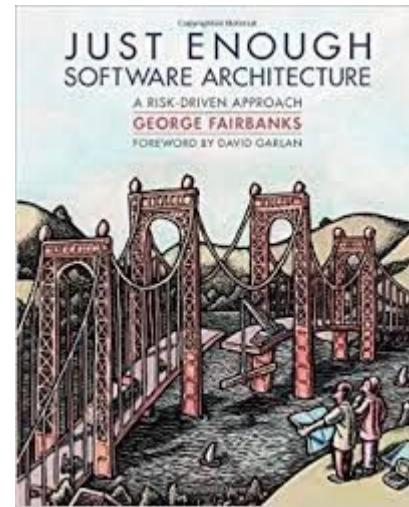
<<produces>>



(Documented) Structures resulting from design decisions

The Architect

Del Big Design Up Front....



....a la Emergencia

El Proceso de Construcción

Walking Skeleton, Arquitectura Hexagonal, BDD / TDD, CI/CD

Premisa #0:

El resultado es consecuencia del proceso

Premisa #1:

El desarrollo es guiado por pruebas

Premisa #2:

El proceso es iterativo y los incrementos son
pequeños

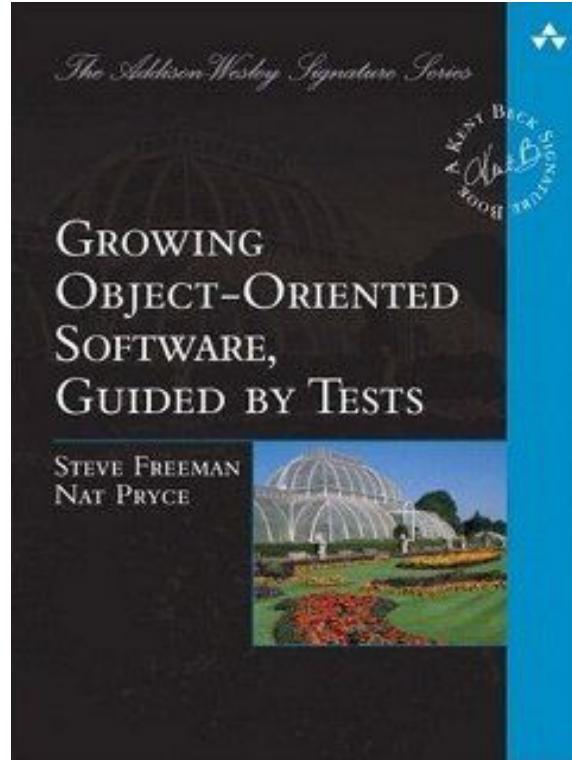
Premisa #3:

Integración, verificación y entrega continua

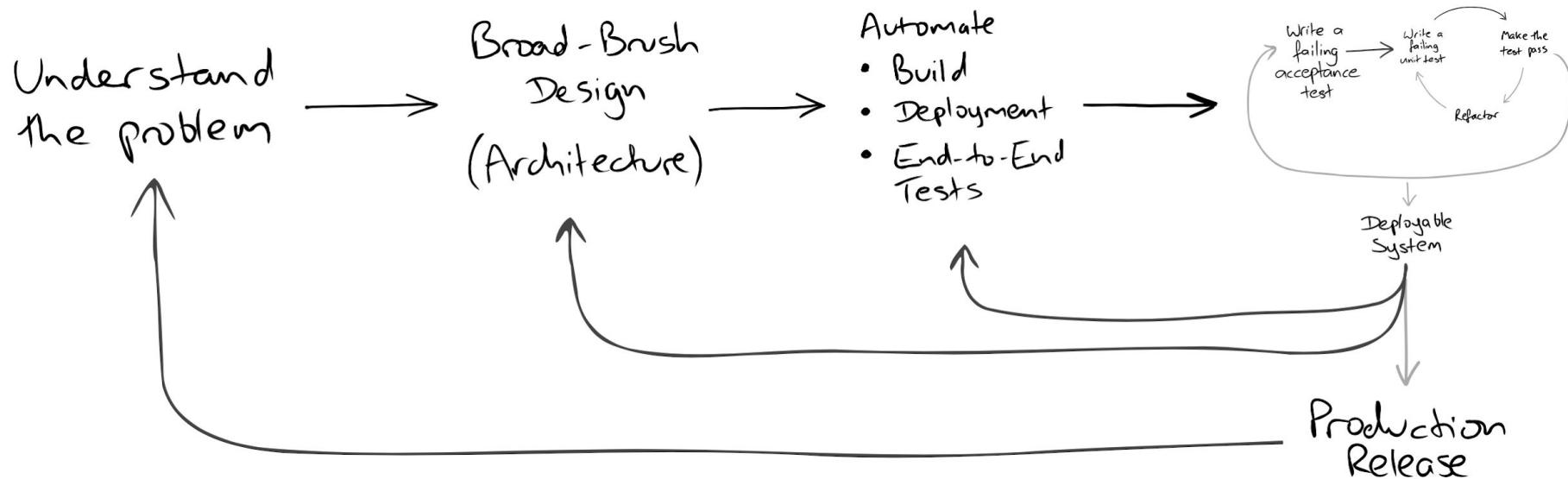
Premisa #4:

Mantenerlo Estúpidamente Simple (KISS)

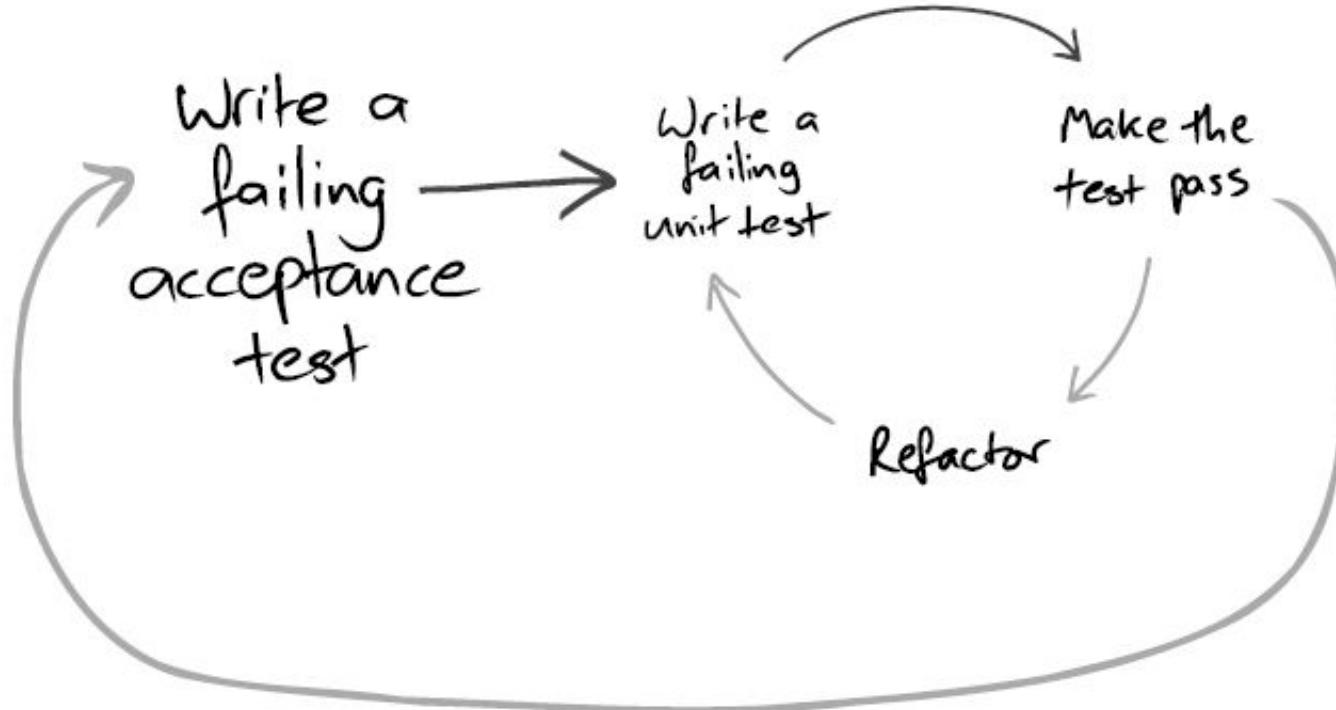
- BDD / TDD
- Integración continua / Despliegue continuo
- Walking Skeleton
- Arquitectura Hexagonal



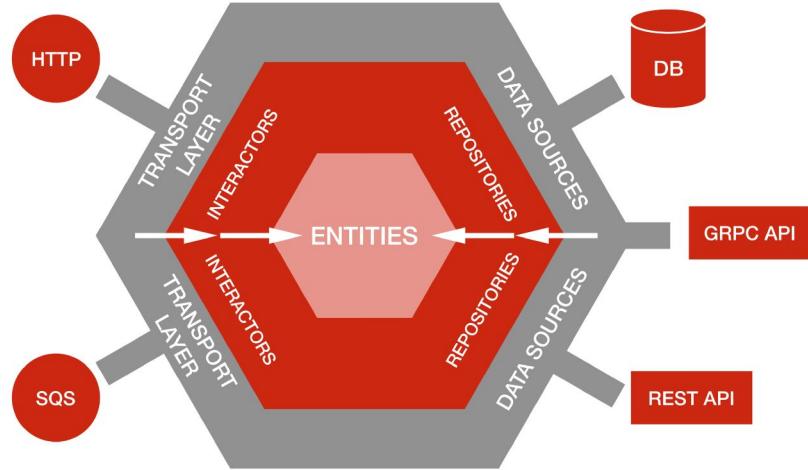
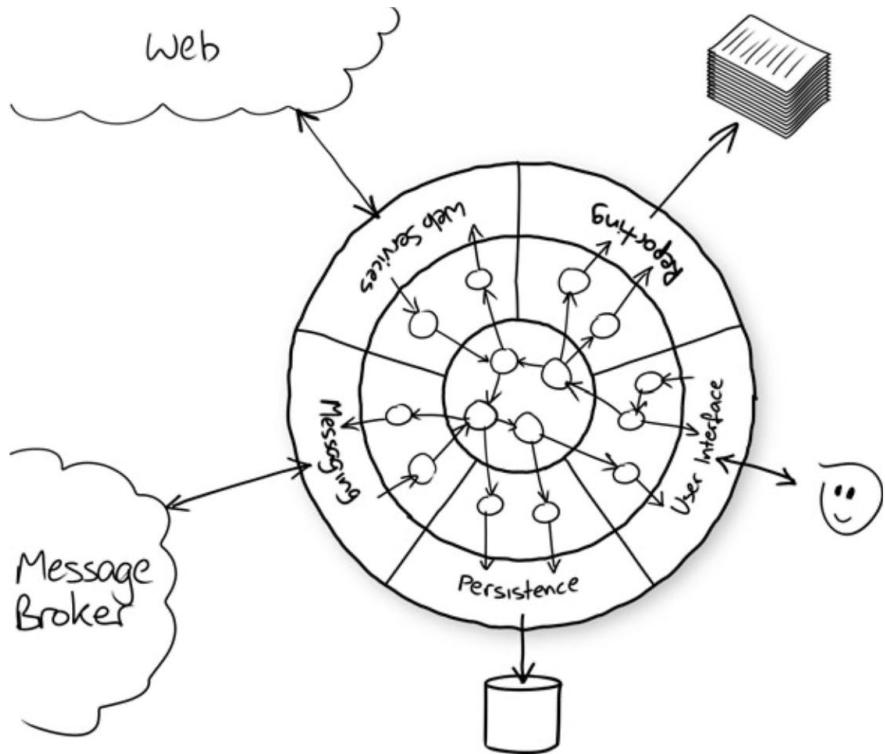
Proceso de alto nivel



BDD / TDD Cycle



Hexagonal Architecture



Walking Skeleton / Tracer Bullet

Funcionalidad mínima, end-to-end*,

integrada, probada y desplegada

Objetivo:

Mitigar riesgos, Despejar incertidumbre, Sentar las bases

Recomendador de ropa

En base a la temperatura actual de Buenos Aires y las preferencias de vestido, la aplicación (http-api) debe recomendar una prenda.

Reglas:

- * Preferencias: liviana | pesada
- * pref:liviana & temp > 20, bermuda
- * pref:liviana & temp <= 20, remera
- * pref:liviana & temp <= 10, buzo
- * pref:pesada & temp > 20, jean
- * pref:pesada & temp <= 20, camisa
- * pref:pesada & temp <= 10, campera

Desarrollo Guiado por Pruebas & Incremental

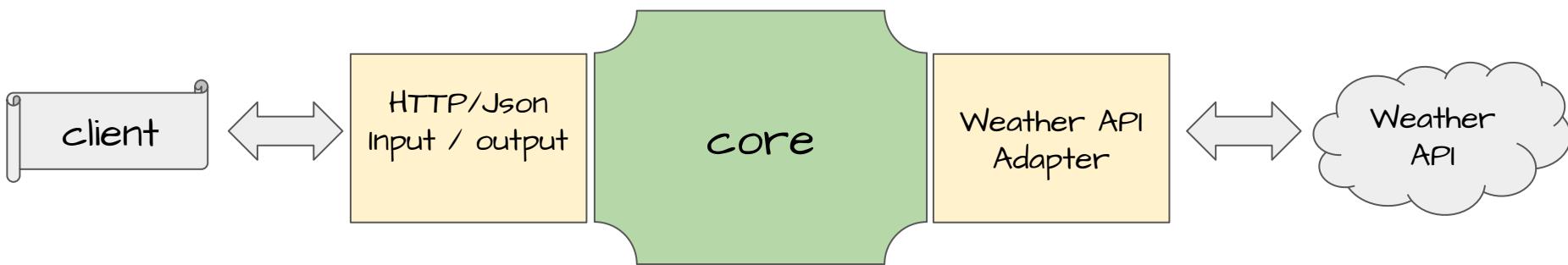
Si la temperatura es mayor a 20:

=> Recomendar remera

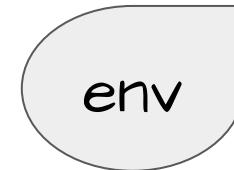
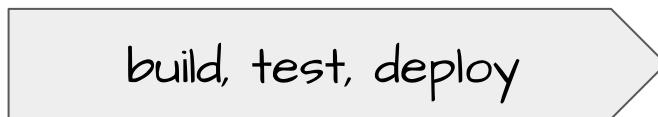
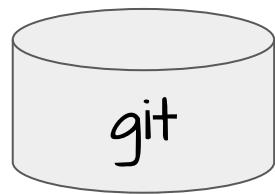
=> Si no camisa



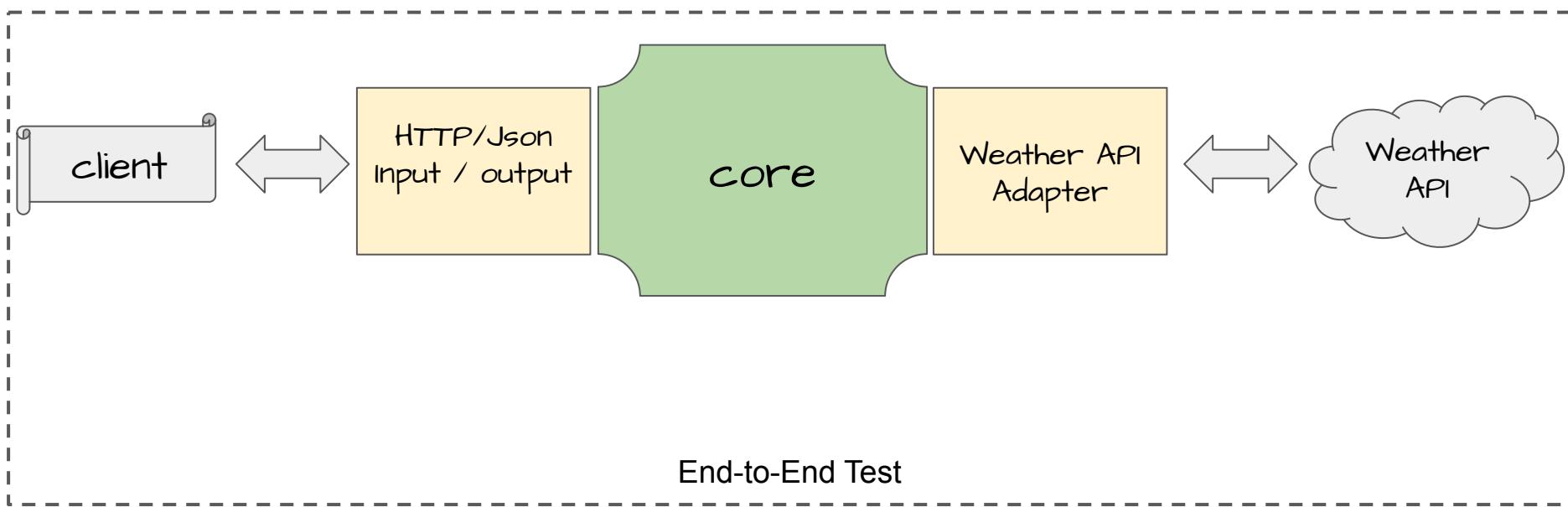
Arquitectura Hexagonal



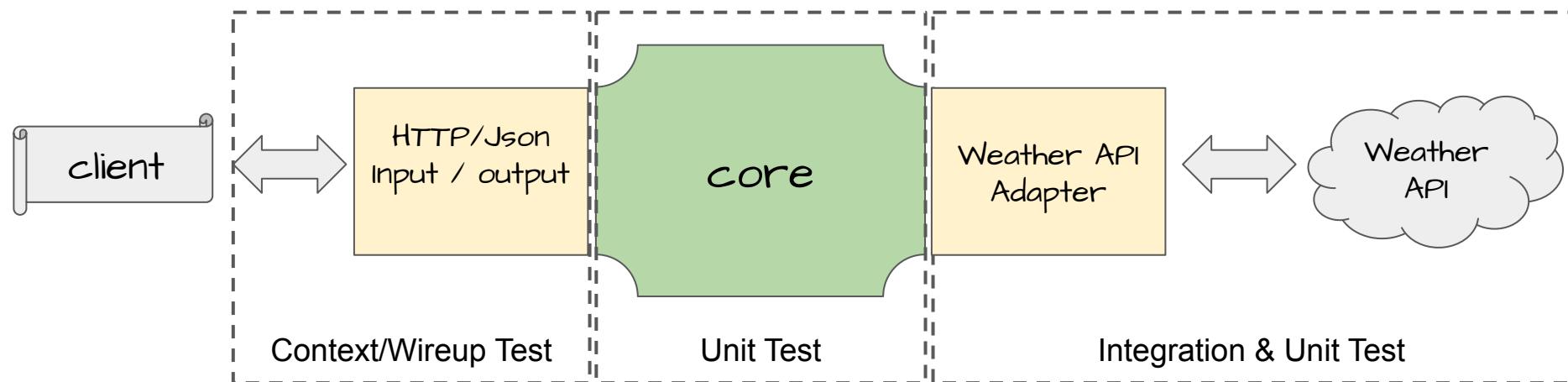
Integración, verificación y entrega continua



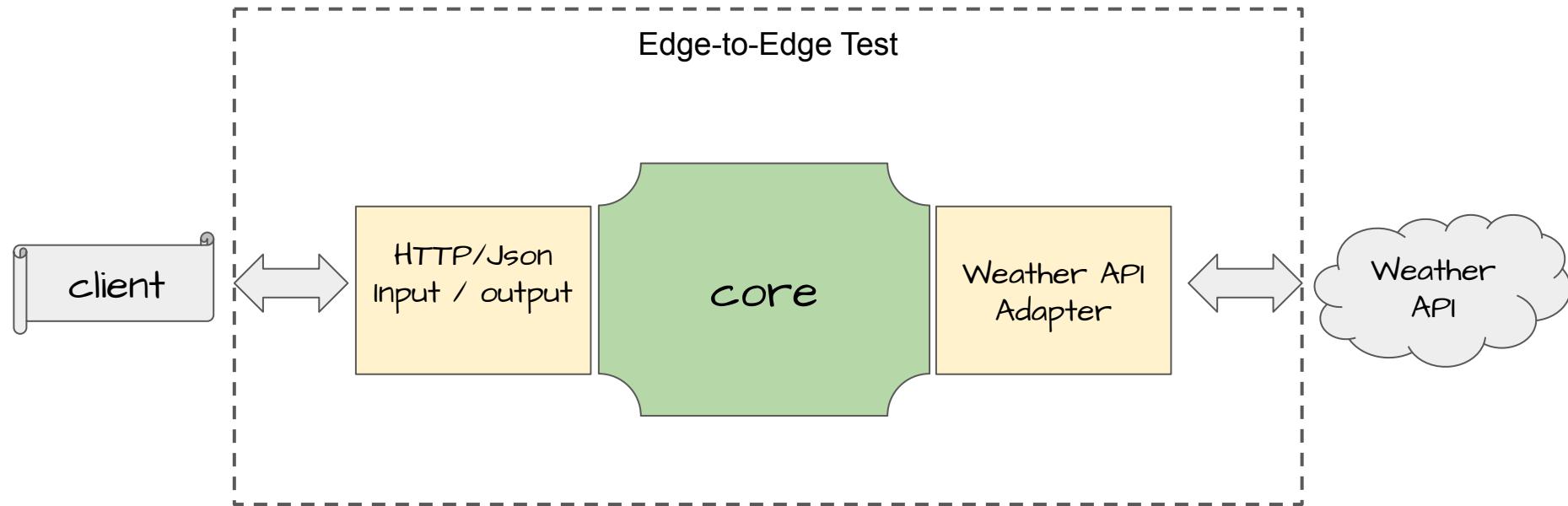
Arquitectura Hexagonal



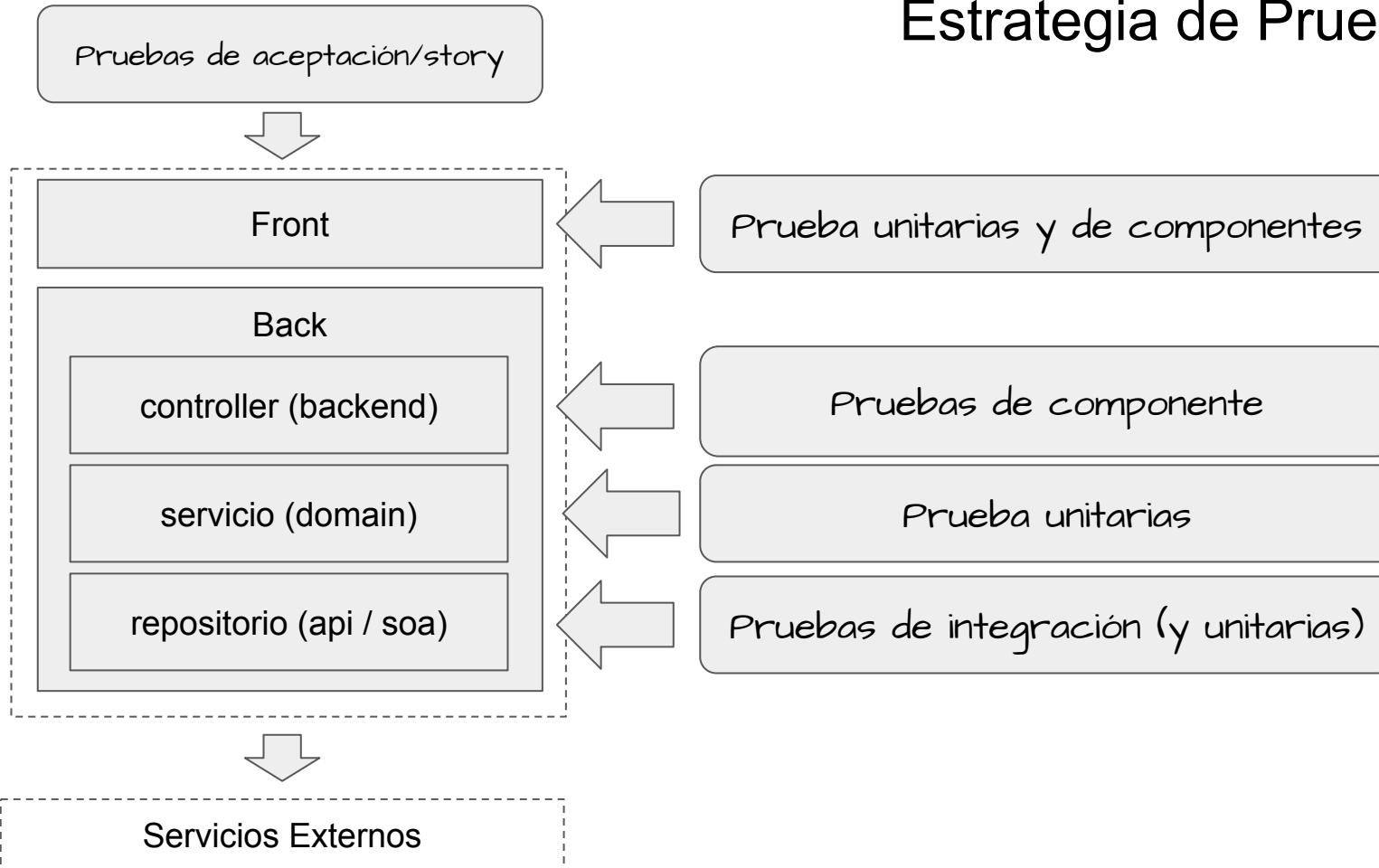
Arquitectura Hexagonal



Arquitectura Hexagonal

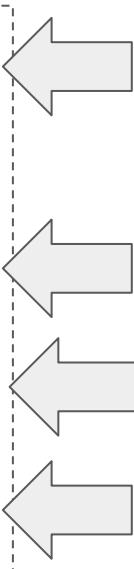
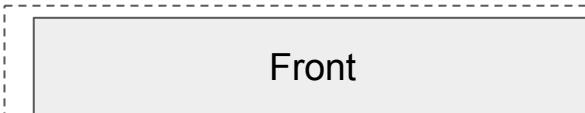


Estrategia de Pruebas



Estrategia de Pruebas - Visión

Pruebas de aceptación/story son edge-2-edge mookeando las api/servicios externos



Prueba unitarias y de componentes sin consumir back

Pruebas de componente se prueba el contexto web mookeando repositorios. Apunta a probar todo el wire up de la aplicación web: inyección de dependencias, configuración, rutas http, validación de token, etc, etc

Prueba unitarias mookeando dependencias. Apunta a probar lógica de negocio pura.

Pruebas de integración consumiendo el servicio para probar correcta integración con apis/servicios de terceros.
Pruebas unitarias para verificar lógica de mapping.



Servicios Externos

Estrategia de Pruebas - Visión

1

Pruebas de aceptación/story son edge-2-edge mockeando las api/servicios externos



Front

2

Prueba unitarias y de componentes sin consumir back

Back

controller (backend)

3

Pruebas de componente se prueba el contexto web mockeando repositorios. Apunta a probar todo el wire up de la aplicación web: inyección de dependencias, configuración, rutas http, validación de token, etc, etc

servicio (domain)

4

Prueba unitarias mockeando dependencias. Apunta a probar lógica de negocio pura.

repositorio (api / soa)

5

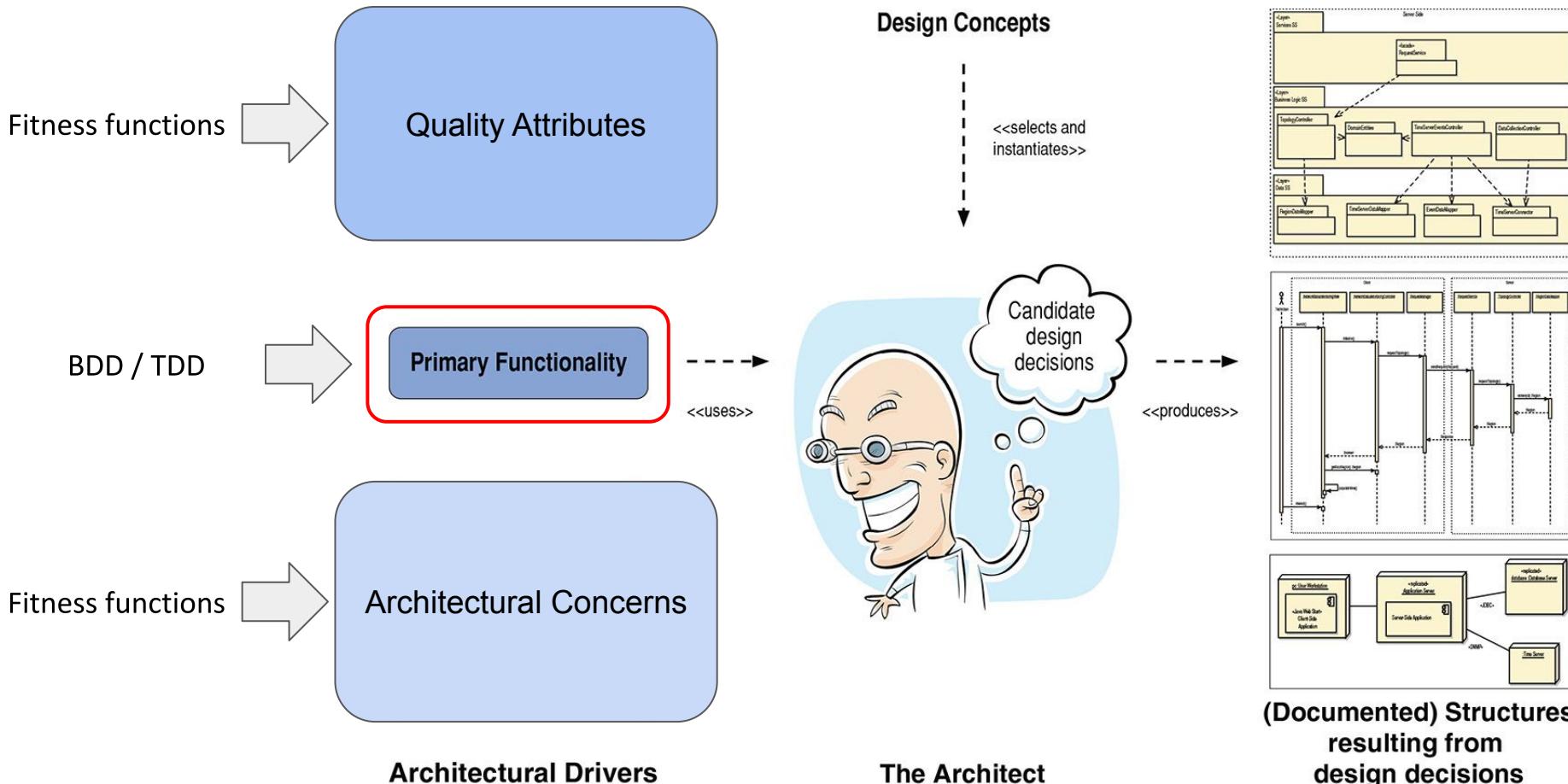
Pruebas de integración consumiendo el servicio para probar correcta integración con apis/servicios de terceros.
Pruebas unitarias para verificar lógica de mapping.



Servicios Externos

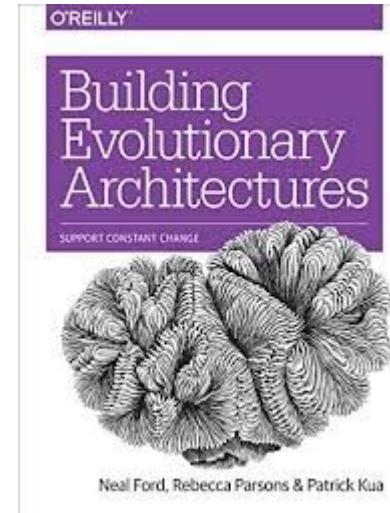
Story1:

1	2	2	3	3	4	5	5
1	2	2	3	3	4		
1	2	3	4	4	5		



Architectural Fitness Functions

Asegurar características
de la arquitectura más allá
de las funcionalidades



Fitness Function-Driven Development

<https://www.thoughtworks.com/insights/articles/fitness-function-driven-development>

Fin

nicopaez@computer.org

@nicopaez

blog.nicopaez.com

