



**GUSTAVO CORONEL**  
DESARROLLA SOFTWARE



# ENTERPRISE JAVA DEVELOPER WITH SPRING BOOT

## TRABAJAMDO CON POSTMAN

**Eric Gustavo Coronel Castillo**

**I N S T R U C T O R**

[youtube.com/DesarrollaSoftware](https://youtube.com/DesarrollaSoftware)

[gcoronelc@gmail.com](mailto:gcoronelc@gmail.com)

# LOGRO ESPERADO

Al finalizar esta lección el participante estará en capacidad de hacer pruebas de APIs REST con POSTMAN.



# **INTRODUCCIÓN A LAS PRUEBAS DE SOFTWARE**

# Definiciones preliminares

- **Defecto o bug**

Es una imperfección en el código que hace que el componente o sistema falle.

- **Error o mistake**

Es una acción que introduce un defecto en el código. Los errores los producen las personas.

- **Falla o failure**

Es una desviación en el comportamiento de un sistema con respecto a lo que se esperaba de él.



# Definiciones preliminares

- **Testing**

Permite comprobar si un componente se comporta como se esperaba, permite encontrar fallas.

- **Depuración**

Es la investigación de una falla. Permite encontrar el motivo de la falla.

- **Calidad**

El grado en que un sistema o componente cumple con los requerimientos especificados, y las necesidades y/o expectativas del cliente o usuario.



# Objetivos del testing

- Encontrar defectos
- Ganar confianza sobre el nivel de calidad
- Proveer información para la toma de decisiones
- Prevenir defectos

Es mucho mas barato para el desarrollo de un proyecto encontrar un defecto lo antes posible o si es posible evitarlo.



# Niveles de pruebas

- Pruebas de componente (unitarias)
- Pruebas de integración
  - ✓ Integración de componentes
  - ✓ Integración de sistemas
- Pruebas de sistema
- Pruebas de aceptación
  - ✓ Pruebas de aceptación de usuario
  - ✓ Pruebas operacionales
  - ✓ Pruebas de regulación y contrato
  - ✓ Alpha and beta testing



# Tipos de pruebas

- Pruebas estructurales

Pruebas de caja blanca, se tiene acceso al código, se comprueba la calidad del código.

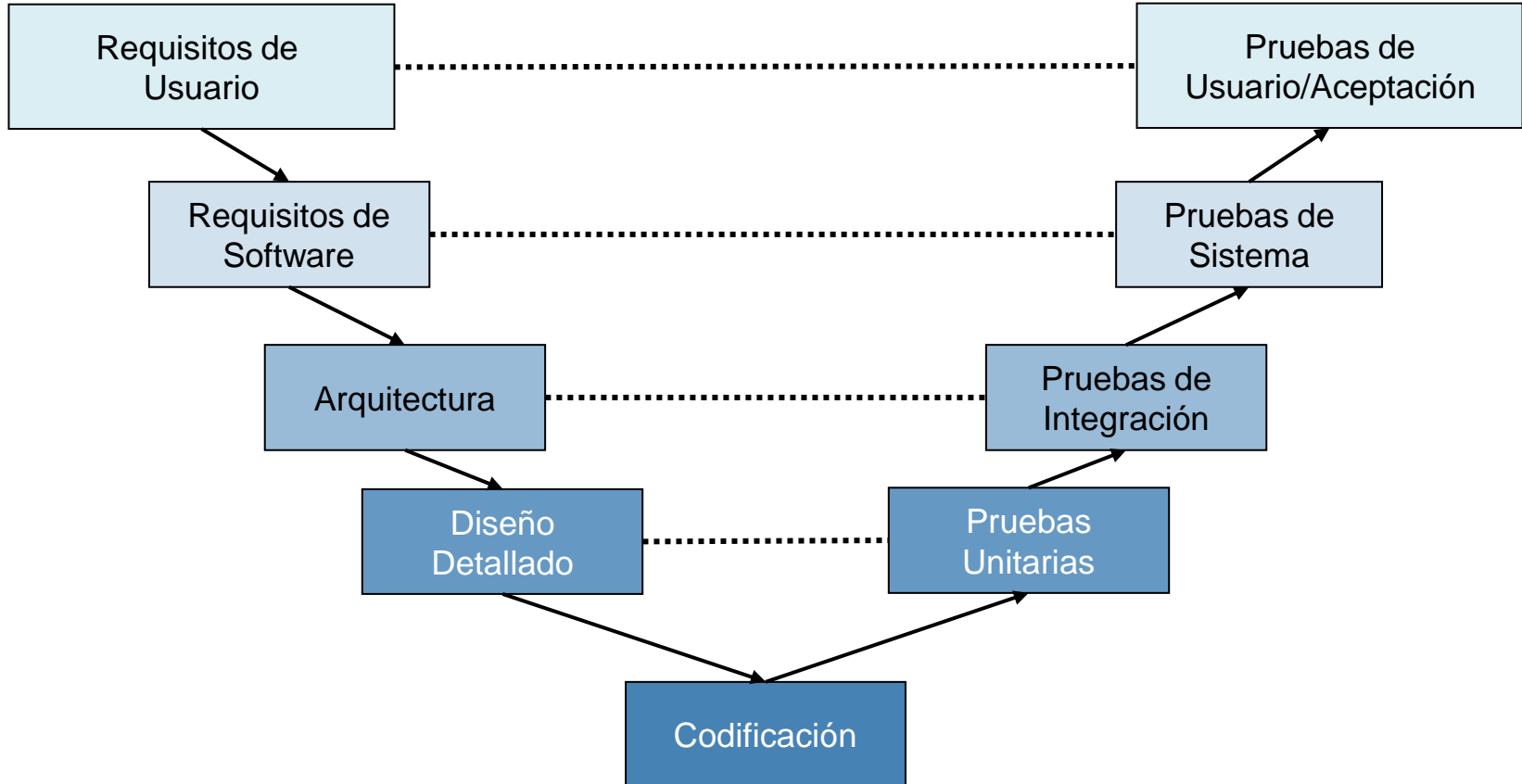
- Pruebas relacionadas con cambios

- ✓ Retesting
- ✓ Pruebas de regresión (Automatización)





# Modelo de ciclo de vida en V



# Pruebas unitarias

- Propósito: Encontrar defectos en un módulo o función.
- Normalmente las pruebas unitarias son hechas por el programador del módulo.



# Diseño de pruebas unitarias

- Las pruebas de cada función/método se especifican junto a la especificación de cada función.
- Debemos asegurarnos de incluir pruebas para:
  - Casos habituales y casos extremos de los parámetros.
  - Todas las excepciones.
  - Todos los tipos de efectos laterales.
- Un mismo caso de prueba puede mostrar una excepción y un efecto lateral.



unit testing

# INTRODUCCIÓN A LAS API REST

# REST

- REST: Representational State Transfer
- Separación entre cliente y servidor
- Visibilidad y escalabilidad
- Independiente de la plataforma y lenguaje
- Intercambio de información basado principalmente en XML y JSON



# Ejemplo XML

```
<equipo>
  <jugadores>
    <jugador>
      <nombre>Martín Martínez</nombre>
      <edad>31</edad>
      <posicion>Defensa</posicion>
    </jugador>
    <jugador>
      <nombre>Álvaro Álvarez</nombre>
      <edad>29</edad>
      <posicion>Centrocampista</posicion>
    </jugador>
    <jugador>
      <nombre>Rodrigo Rodríguez</nombre>
      <edad>24</edad>
      <posicion>Delantero</posicion>
    </jugador>
  </jugadores>
</equipo>
```

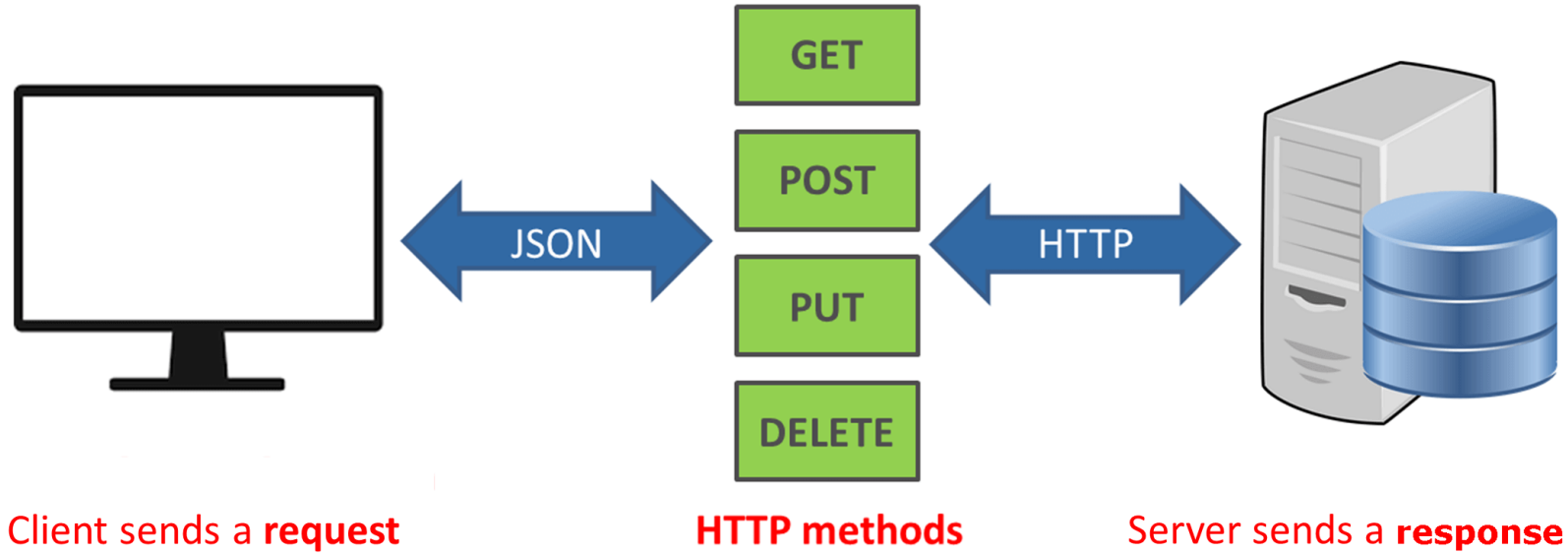


# Ejemplo JSON

```
{
  "equipo": {
    "jugadores": [{
      "nombre": "Martín Martínez",
      "edad": "31",
      "posicion": "Defensa"
    },
    {
      "nombre": "Álvaro Álvarez",
      "edad": "29",
      "posicion": "Centrocampista"
    },
    {
      "nombre": "Rodrigo Rodríguez",
      "edad": "24",
      "posicion": "Delantero"
    }
  ]
}
```



# Métodos HTTP





## Elementos de una petición HTTP

```
GET /endpoint?QueryParam=ValueExample HTTP/1.1  
Host: example.com  
Content-Type: application/json
```

En caso de métodos **PUT** y **POST** => **Body** (JSON, XML...)

# Elementos de una respuesta HTTP

- Status code
- Body
- Headers

```
Response status code: 403
Response body
{
    code: 5,
    message: "Error message"
}
```

# Algunos códigos de estado

## ■ 2xx Respuesta exitosa

- ✓ 200 OK
- ✓ 201 Creado
- ✓ 202 Aceptado
- ✓ 204 Sin contenido
- ✓ etc.

## ■ 4xx Error en el cliente

- ✓ 400 Petición Incorrecta
- ✓ 401 No autorizado
- ✓ 403 prohibido
- ✓ 404 No encontrado
- ✓ 410 desaparecido
- ✓ etc.

**Postman**

# ¿Qué es Postman?

- Postman es una plataforma de colaboración para el desarrollo de APIs que permite a los desarrolladores diseñar, probar, documentar y compartir APIs.
- Es una herramienta de software que facilita el proceso de desarrollo de APIs al permitir a los desarrolladores realizar solicitudes HTTP y ver las respuestas en tiempo real.
- Postman es ampliamente utilizado por desarrolladores y equipos de desarrollo en todo el mundo para probar y validar API antes de su implementación en aplicaciones o sistemas.
- La plataforma ofrece una amplia variedad de características, como la creación y gestión de colecciones de API, pruebas automatizadas, documentación de API, monitoreo y análisis de API, integraciones con herramientas de colaboración y muchos más.



# Instalación

The image shows a browser window at [postman.com/downloads/](https://postman.com/downloads/) and a screenshot of the Postman application interface.

**Website Content:**

- Navigation: Product, Pricing, Enterprise, Resources and support, Explore. Buttons: Sign In, Sign Up for Free.
- Section: **Download Postman**
- Text: "Download the app to get started using the Postman API Platform today. Or, if you prefer a browser experience, you can try the web version of Postman."
- Section: **The Postman app**
- Text: "Download the app to get started with the Postman API Platform."
- Buttons: Windows 64-bit, Mac 64-bit, Linux 64-bit.
- Text: "By downloading and using Postman, I agree to the [Privacy Policy](#) and [Terms](#)."
- Text: "Release Notes · Product Roadmap"
- Text: "Not your OS? Download for Mac (Intel Chip, Apple Chip) or Linux (x64, arm64)"

**Postman Application Interface:**

- Home Workspaces API Network Explore
- Search Postman
- Twitter's Public Workspace New Import Overview
- GET Single Tweet
- ... / Tweet Lookup / Single Tweet
- Save
- GET https://api.twitter.com/2/tweets/:id Send
- Params Authorization Headers Body Pre-request Scripts Tests Settings Cookies
- Query params
- Table with 4 columns: KEY, VALUE, DESCRIPTION, Bulk edit
- Path Variables

# Primera prueba de Postman

Para estos ejemplos utilizaremos la API Postman ECHO: [postman-echo.com](https://postman-echo.com)

## Postman Echo

Postman Echo is service you can use to test your REST clients and make sample API calls. It provides endpoints for `GET` , `POST` , `PUT` , various auth mechanisms and other utility endpoints.

The documentation for the endpoints as well as example responses can be found at <https://postman-echo.com>

### Request Methods









HTTP has multiple request "verbs", such as `GET` , `PUT` , `POST` , `DELETE` , `PATCH` , `HEAD` , etc.

An HTTP Method (verb) defines how a request should be interpreted by a server. The endpoints in this section demonstrate various HTTP Verbs. Postman supports all the HTTP Verbs, including some rarely used ones, such as `PROPFIND` , `UNLINK` , etc.

For details about HTTP Verbs, refer to [RFC 2616](#)

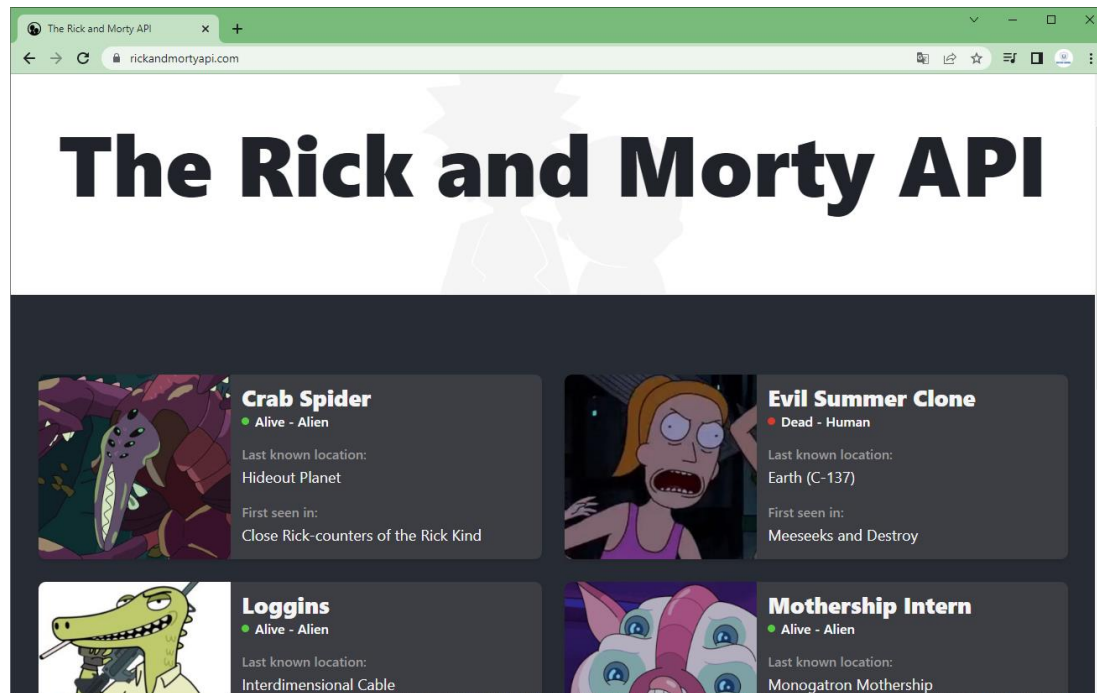
#### JUMP TO

Introduction

- >  Request Methods
- >  Headers
- >  Authentication Methods
- >  Cookie Manipulation
- >  Utilities
- >  Utilities / Date and Time
- >  Auth: Digest
- >  Server Events

# Colecciones

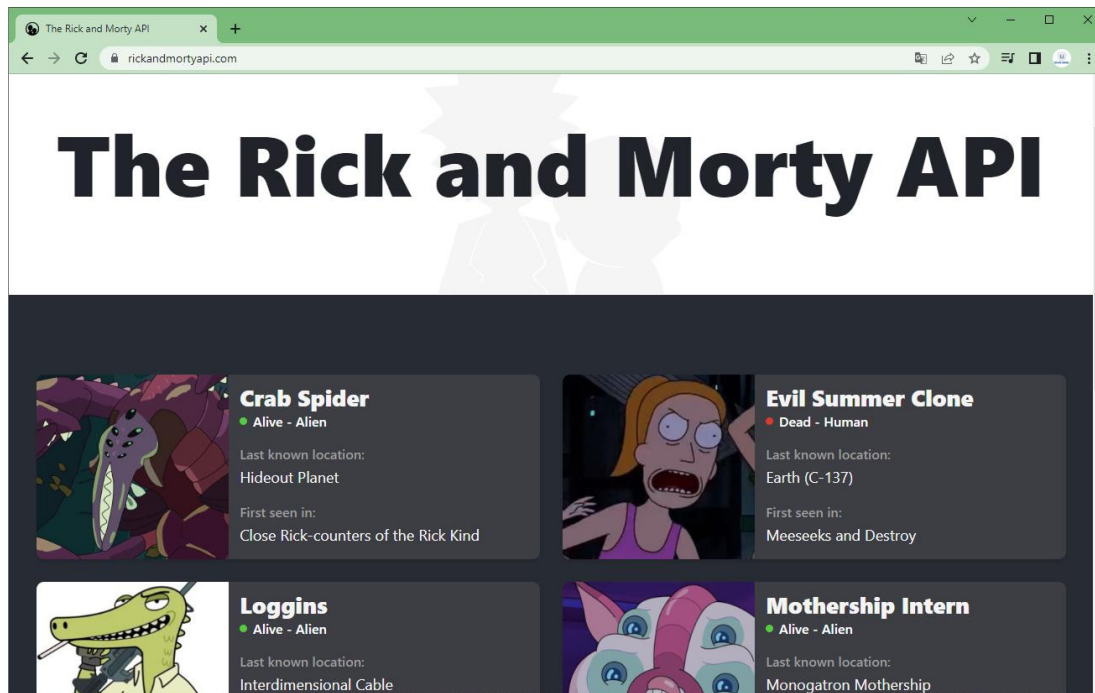
Para estos ejemplos utilizaremos la API: [rickandmortyapi.com](http://rickandmortyapi.com)





# Variables y Entornos

Para este tema continuaremos con la API: [rickandmortyapi.com](http://rickandmortyapi.com)



# Automatizando pruebas

Para este tema continuaremos con la API: [rickandmortyapi.com](https://rickandmortyapi.com)

The screenshot displays a REST client interface with the following components:

- Request Bar:** Method `GET`, URL `https://rickandmortyapi.com/api/character/2`, and a `Send` button.
- Tabs:** Params, Authorization, Headers (7), Body, Pre-request Script, **Tests** (active), and Settings.
- Test Scripts:** A JavaScript script with three tests:

```
1 pm.test("Status code is 200", function () {  
2   pm.response.to.have.status(200);  
3 });  
4  
5 pm.test("Validate character name", function () {  
6   var jsonData = pm.response.json();  
7   pm.expect(jsonData.name).to.eql("Morty Smith");  
8 });  
9  
10 pm.test("Response time is less than 1000ms", function () {  
11   pm.expect(pm.response.responseTime).to.be.below(1000);  
12 });
```
- Test Results:** A sidebar on the right showing:
  - Test scripts are written in JavaScript, and are run after the response is received. Learn more about [tests scripts](#).
  - Snippets:
    - Status code: Code is 200
    - Response body: Contains string
    - Response body: JSON value check
    - Response body: Is equal to a string
- Response Bar:** Status: `200 OK`, Time: `502 ms`, Size: `3.06 KB`, and a `Save Response` button.
- Test Results List:** A table with filters (All, Passed, Skipped, Failed) and three rows, all marked `PASS`:
  - Status code is 200
  - Validate character name
  - Response time is less than 1000ms

GRACIAS  
**TOTALES**



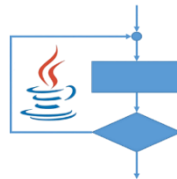
**Gustavo Coronel**  
gcoronelc.github.io



[gcoronelc.github.io](https://gcoronelc.github.io)

[youtube.com/DesarrollaSoftware](https://youtube.com/DesarrollaSoftware)

[facebook.com/groups/desarrollasoftware](https://facebook.com/groups/desarrollasoftware)



## FUNDAMENTOS DE PROGRAMACIÓN CON JAVA

Inicia tu aprendizaje, utilizando las mejores prácticas de programación



## CURSO PROFESIONAL DE JAVA ORIENTADA A OBJETOS

Aprende programación en capas, patrones y buenas prácticas



## PROGRAMACIÓN DE BASE DE DATOS ORACLE CON PL/SQL

Aprende a obtener el mejor rendimiento de tú base de datos



## PROGRAMACIÓN DE BASE DE DATOS ORACLE CON JDBC

Aprende a programar correctamente con JDBC