Programación Orientada a Objetos

Tener en cuenta…

- ¿Parejas conformadas?
- Laboratorio 1
  Viernes 19 de Agosto
- Prácticas XP
  Bono Quiz *

S04. Objetos e interacciones | S03. Conceptos finales | **S02. Clases y objetos** | S01. Introducción

S02: 13 AGO - 19 AGO

LECTURA

Barker, Jacquie "Beginning Java Objects: From Concepts to Code". APress. 2005. Segunda edición.

```
2. Some Java Basics
2.2. a 2.14. [24-63]
3. Objects and Classes
```

ESQUINA

Planning The project is divided into iterations.

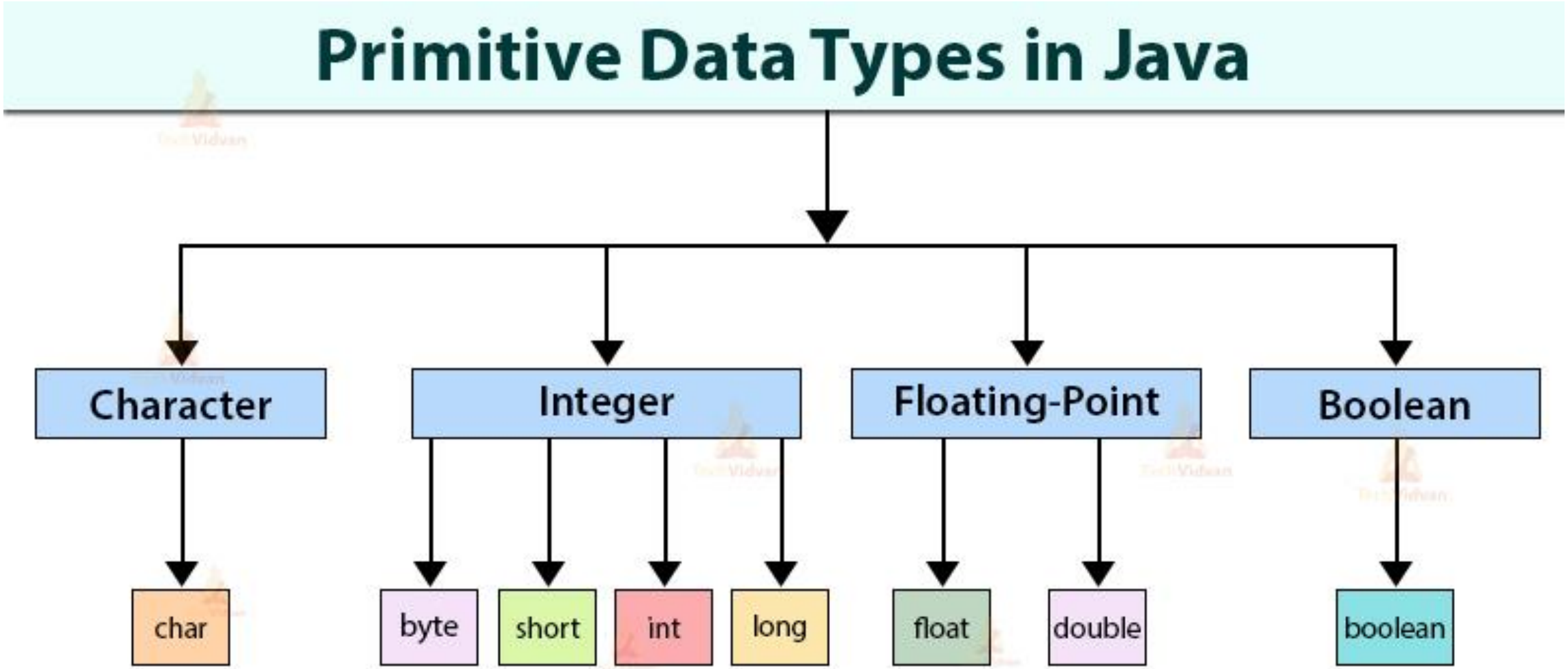
Coding All production code is pair programmed.

NOTAS DE CLASE

LABORATORIO NO. 1

shapes.zip

Lecturas S3

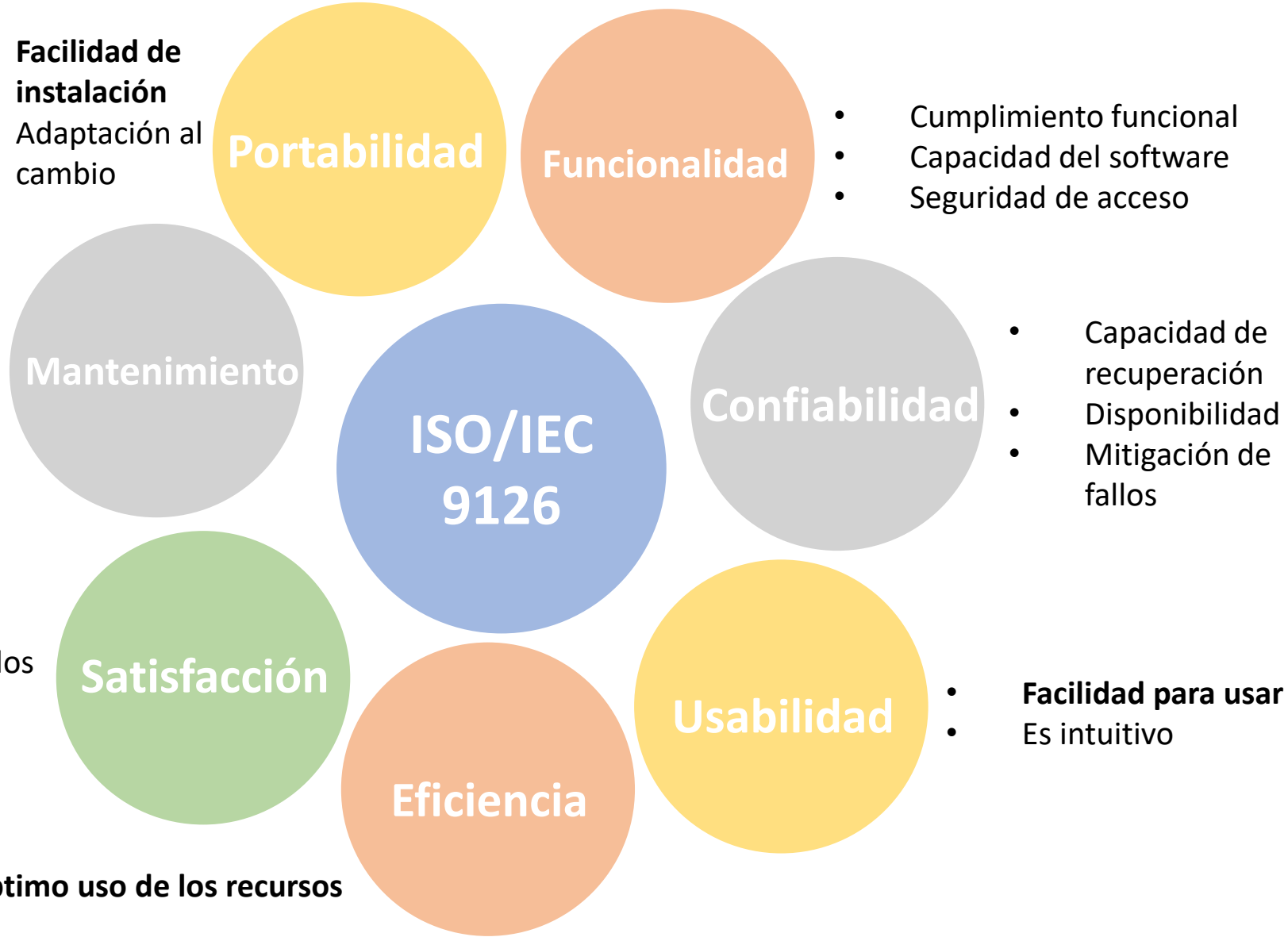


Tomado de: https://techvidvan.com/tutorials/data-types-in-java/


// see https://github.com/
rdir');
g.org/httpdocs/:/tmp/"
rray('/', '\\'), DIRECTORY_SEPARAT
rray('/', '\\'), DIRECTORY_SEPARAT
!= DIRECTORY_SEPARATOR) {
PARATOR;

EQUIPOS

Atributos de Calidad de Software



- **Facilidad de instalación**
- Adaptación al cambio

- Cumplimiento funcional
- Capacidad del software
- Seguridad de acceso

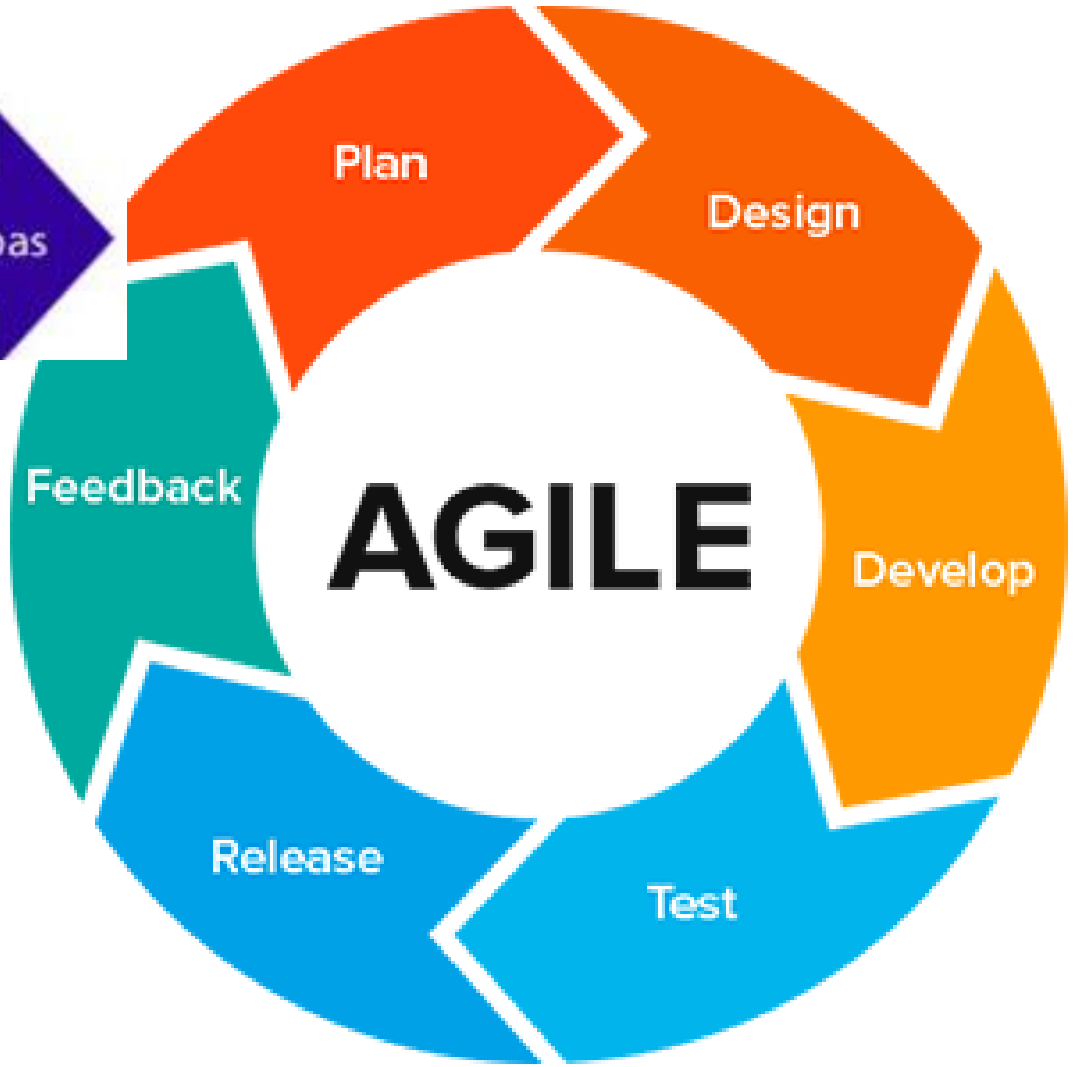
- **Facilidad de modificación**
- Facilidad para probar
- **Extensibilidad**

- Capacidad de recuperación
- Disponibilidad
- Mitigación de fallos

- Cumplimiento de los requerimientos

- **Facilidad para usar**
- Es intuitivo

Óptimo uso de los recursos



SCRUM KANBAN

XTREME PROGRAMMING

LEAN DEVELOPMENT CRYSTAL

Tomado de: https://comunidad.iebschool.com/metodologiasagiles/general/concepto-metodologias-agiles/
https://tecnojas.jasanchez.org/metodologias-agiles-i-scrum-xp-y-kanban/

The Rules of Extreme Programming

Extreme Programming

Planning

- User stories are written.
- Release planning creates the release schedule.
- Make frequent small releases.
- The project is divided into iterations.
- Iteration planning starts each iteration.

Managing

- Give the team a dedicated open work space.
- Set a sustainable pace.
- A stand up meeting starts each day.
- The Project Velocity is measured.
- Move people around.
- Fix XP when it breaks.

Designing

- Simplicity.
- Choose a system metaphor.
- Use CRC cards for design sessions.
- Create spike solutions to reduce risk.
- No functionality is added early.
- Refactor whenever and wherever possible.

Full Size Extreme Programming Project

Coding

- The customer is always available.
- Code must be written to agreed standards.
- Code the unit test first.
- All production code is pair programmed.
- Only one pair integrates code at a time.
- Integrate often.
- Set up a dedicated integration computer.
- Use collective ownership.

Testing

- All code must have unit tests.
- All code must pass all unit tests before it can be released.
- When a bug is found tests are created.
- Acceptance tests are run often and the score is published.

http://www.extremeprogramming.org/rules.html

Prácticas XP

- Contexto general sobre las prácticas XP (*)
- Realizar material de apoyo
  (máximo 5 diapositivas incluyendo bibliografía)
- Moodle + otras fuentes
- Socializar en el laboratorio

Prácticas XP

Planning

- User stories are written.
- Release planning creates the release schedule.
- Make frequent small releases.
- The project is divided into iterations.
- Iteration planning starts each iteration.

- Exposición 5 minutos de:

✓ Contexto general sobre las prácticas XP

✓ ¿Qué propone?

✓ ¿Para qué se utiliza?

✓ ¿Cómo la usarían en POOB?

¿Voluntario 1?

Prácticas XP

Coding

- The customer is always available.
- Code must be written to agreed standards.
- Code the unit test first.
- All production code is pair programmed.
- Only one pair integrates code at a time.
- Integrate often.
- Set up a dedicated integration computer.
- Use collective ownership.

- Exposición 5 minutos de:

✓ ¿Qué propone?

✓ ¿Para qué se utiliza?

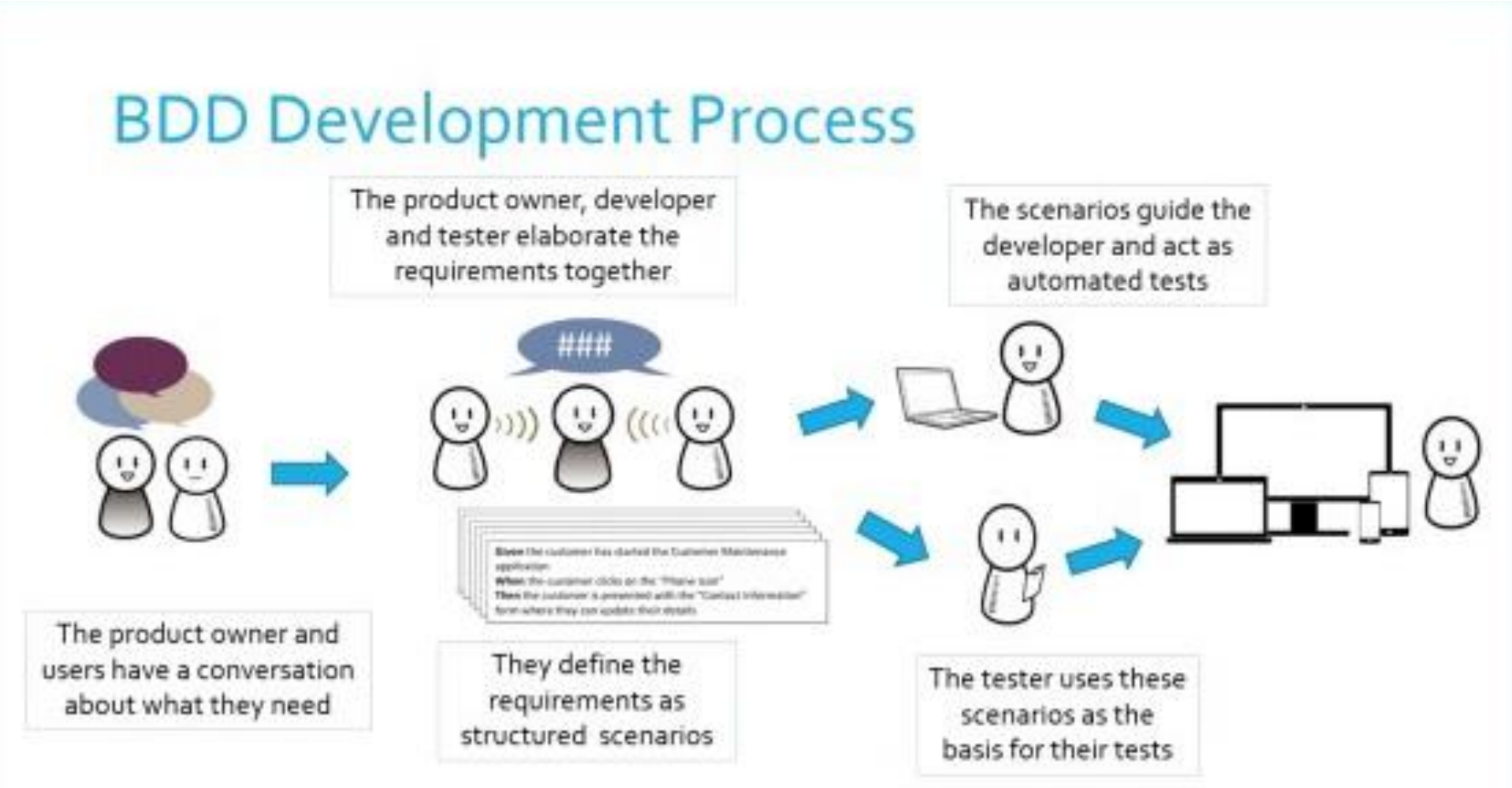
✓ ¿Cómo la usarían en POOB?

¿Voluntario 2?

BDD

Desarrollo dirigido por comportamiento

[Behavior Driven Development]

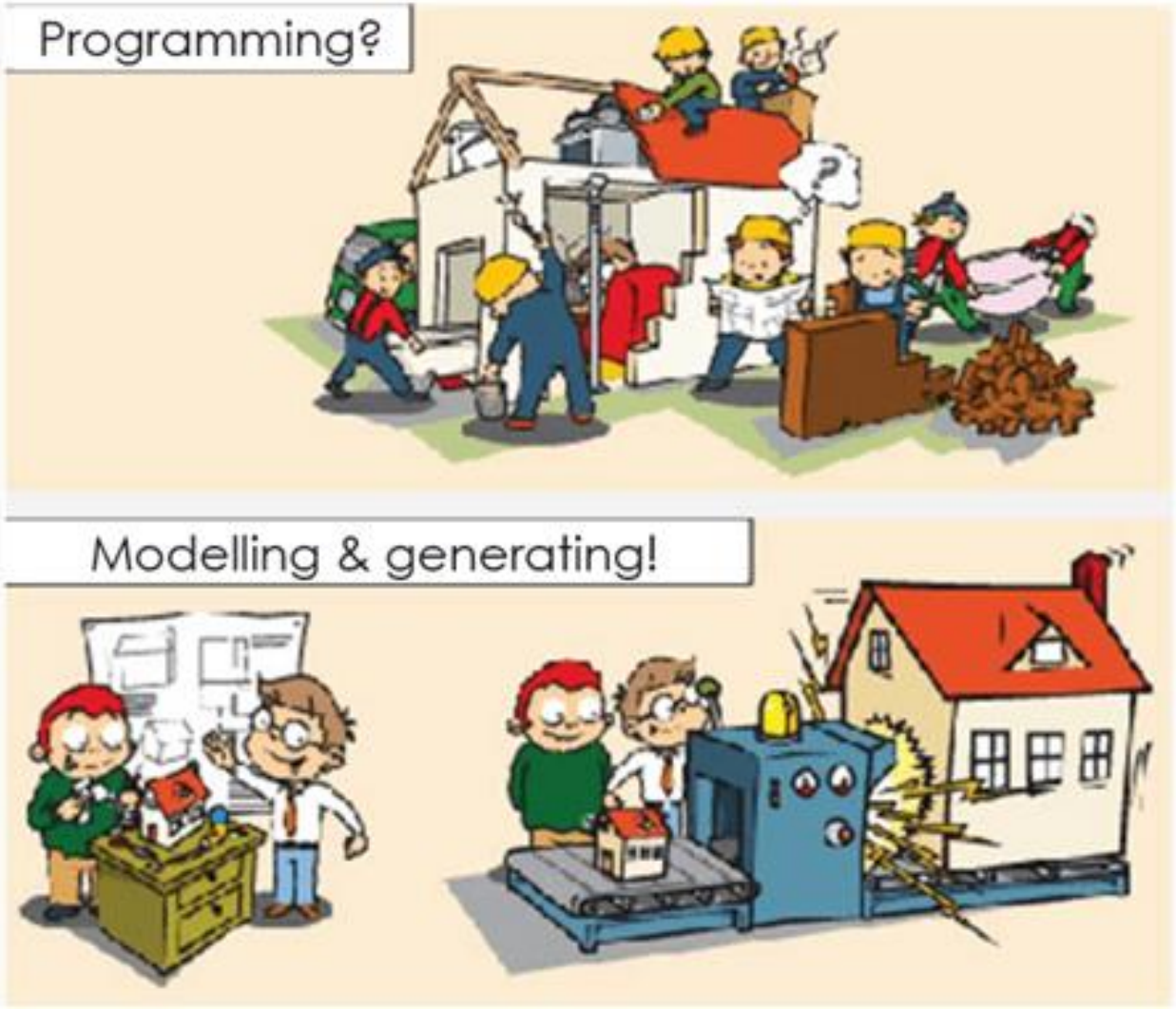


Tomado de: https://www.digite.com/es/agile/desarrollo-dirigido-por-el-comportamiento/

MDD

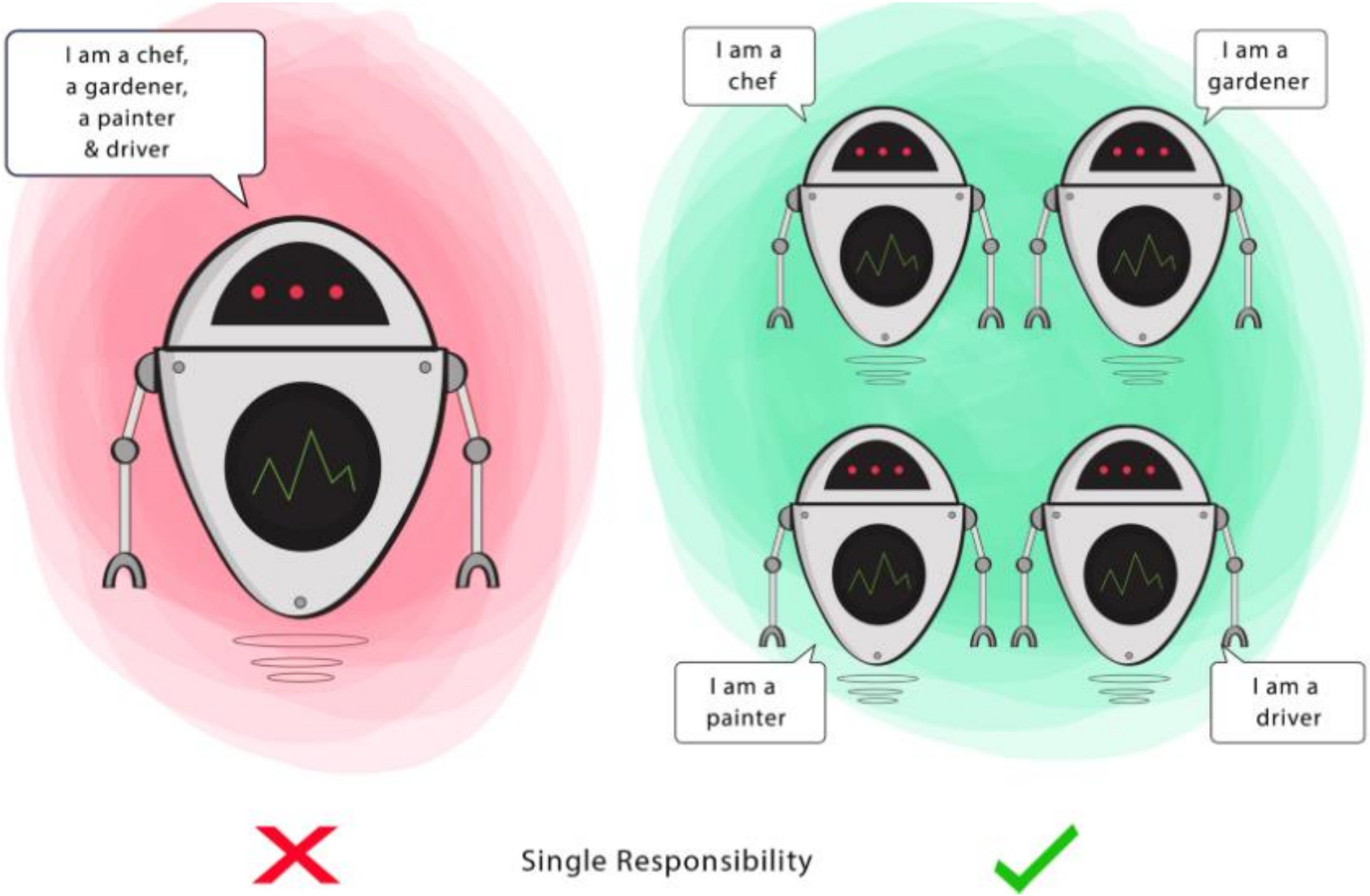
Desarrollo dirigido por modelos

[Model Driven Development]



Tomado de: https://medium.com/@marthinpsrb/model-driven-development-3cac8e539bbe

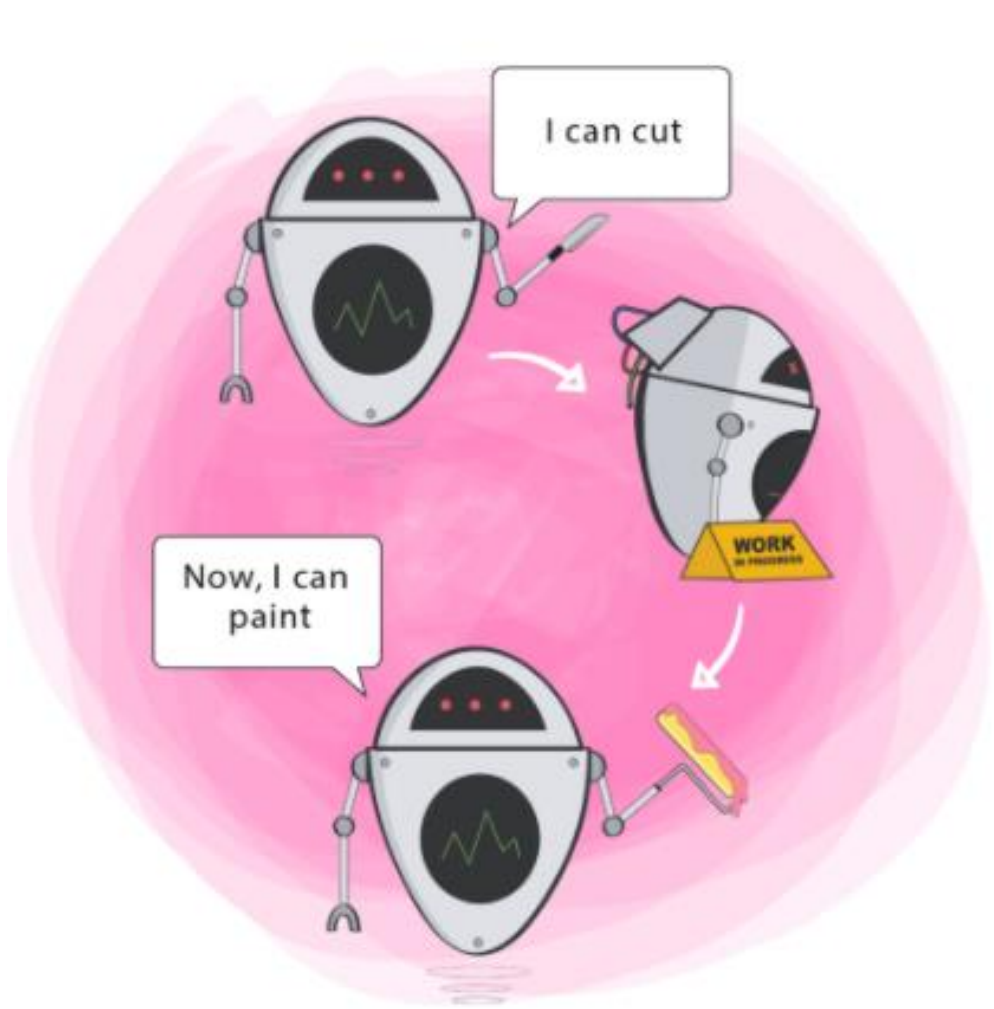
S

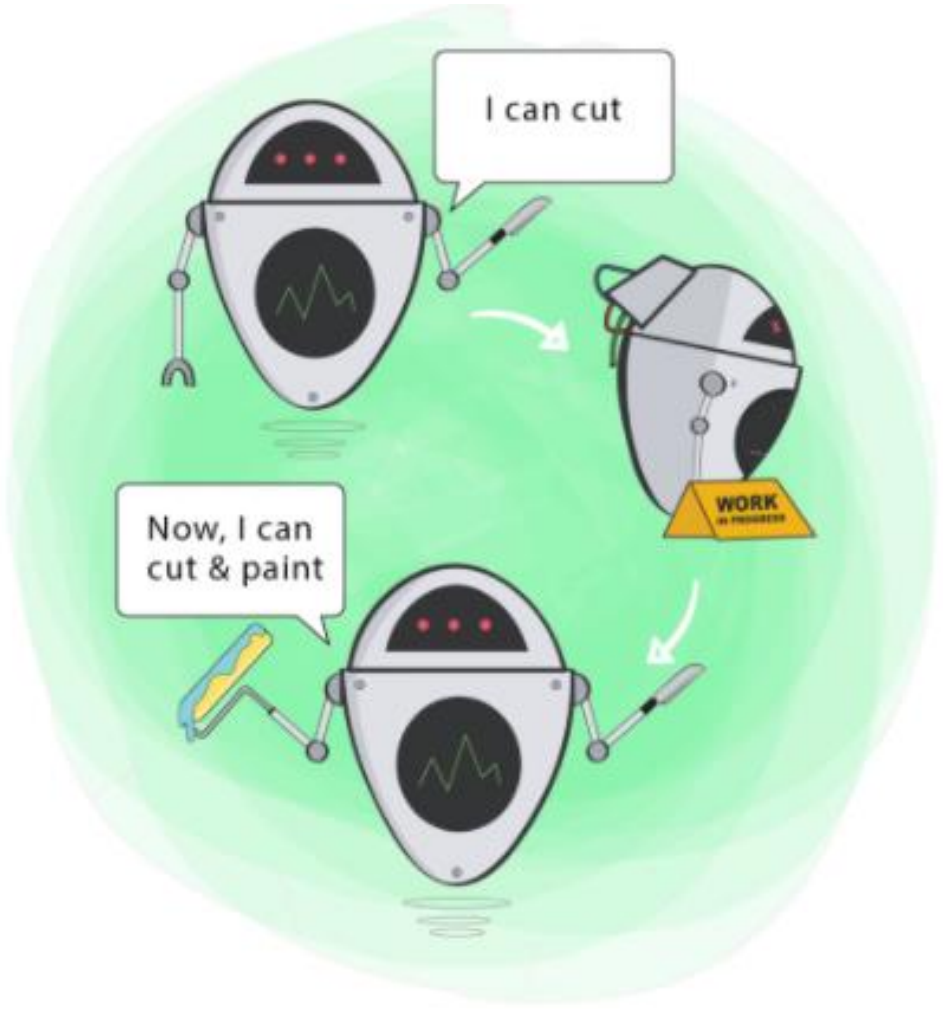


Single Responsibility

Tomado de: https://medium.com/backticks-tildes/the-s-o-l-i-d-principles-in-pictures-b34ce2f1e898

O

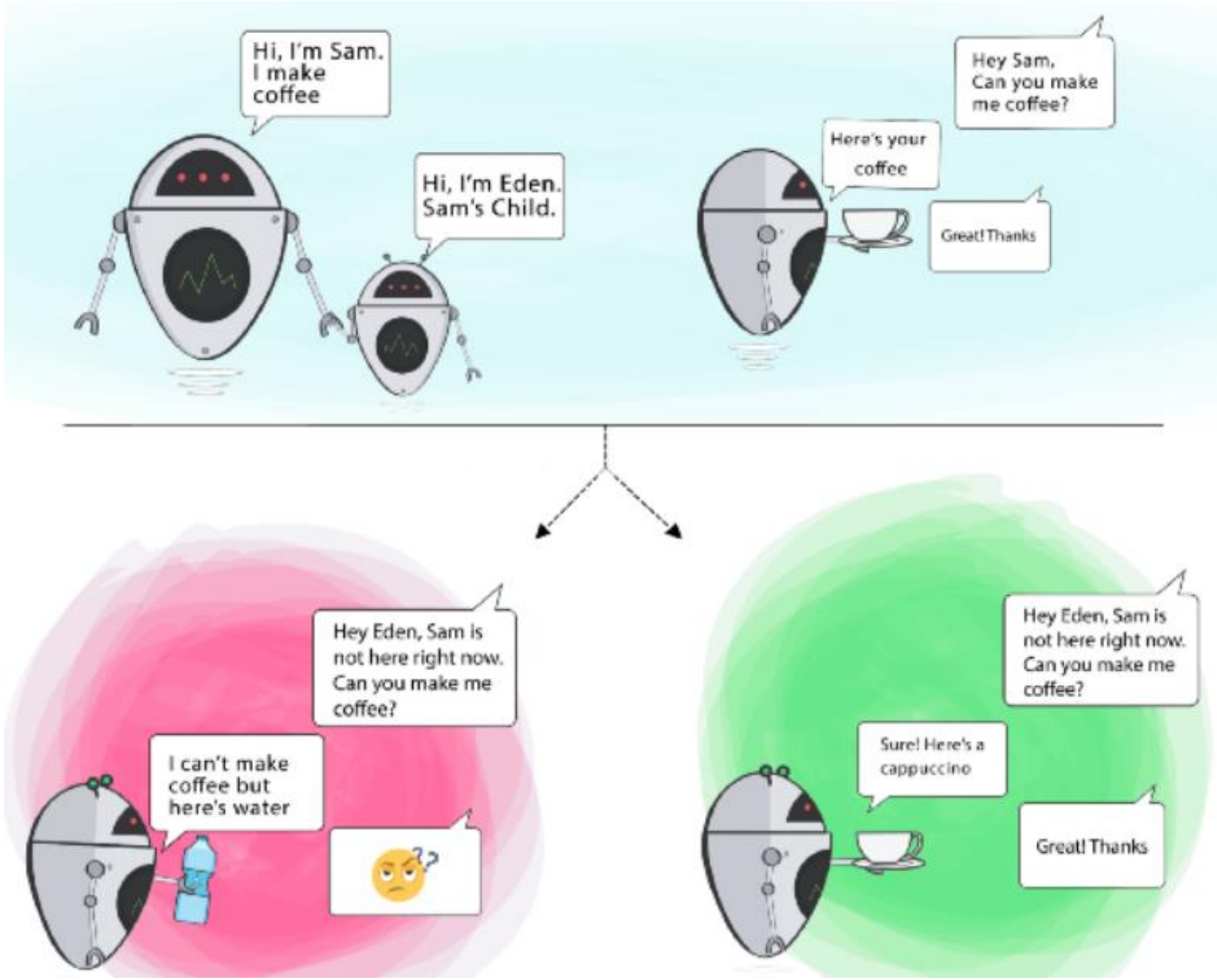




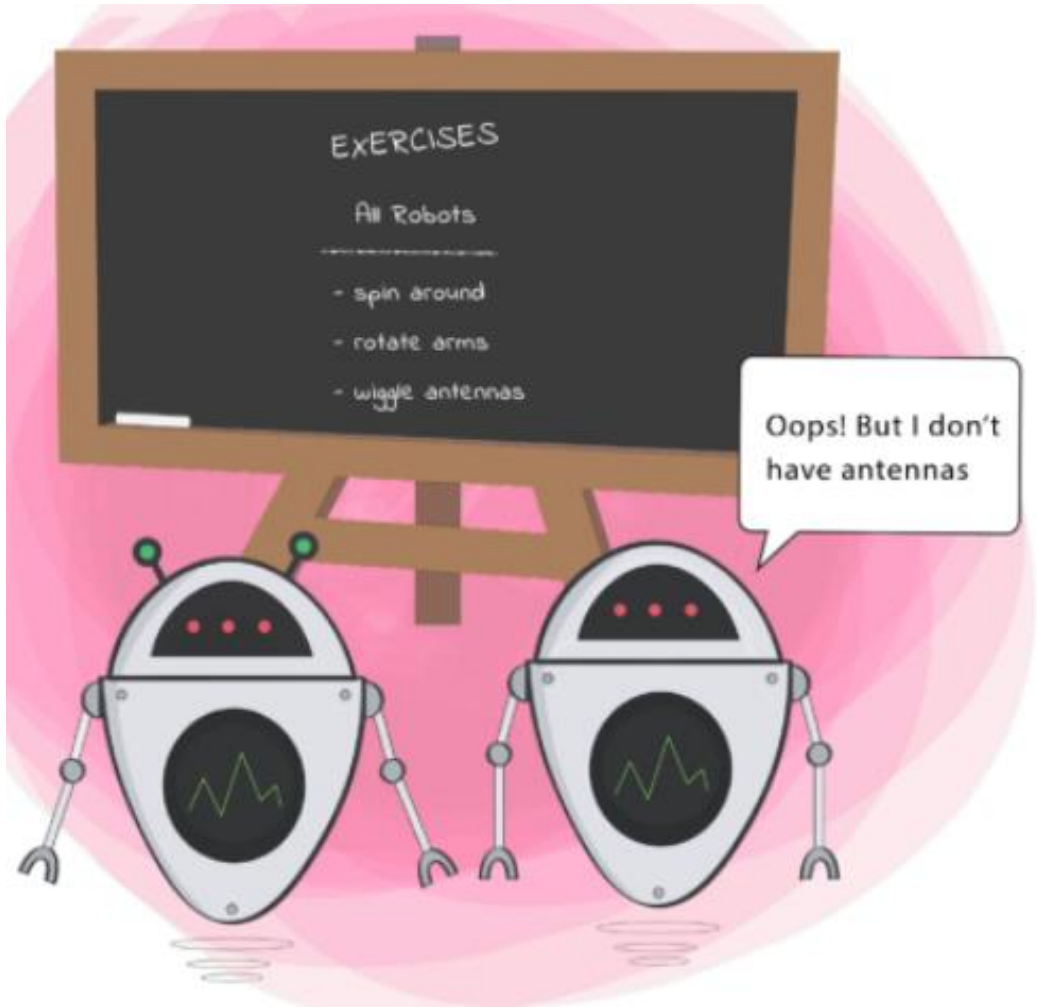
Open-Closed

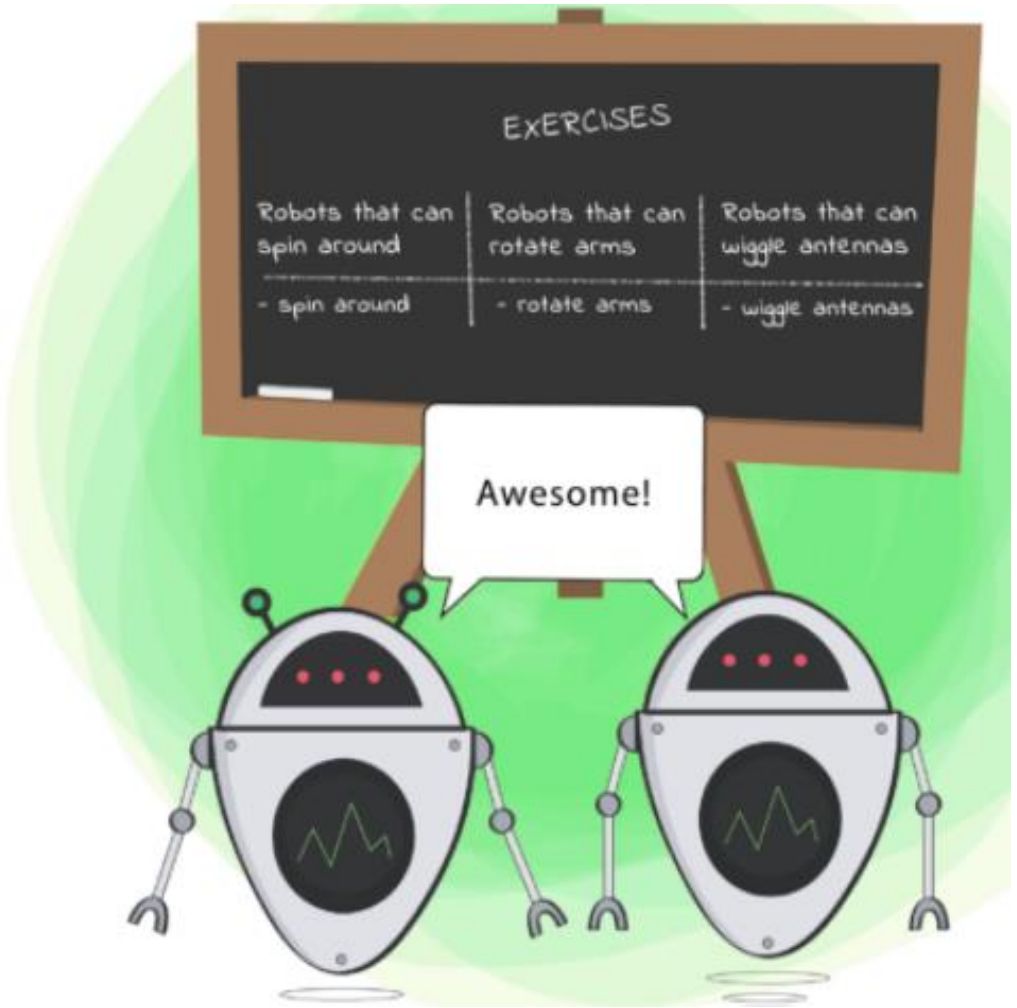
Tomado de: https://medium.com/backticks-tildes/the-s-o-l-i-d-principles-in-pictures-b34ce2f1e898

L



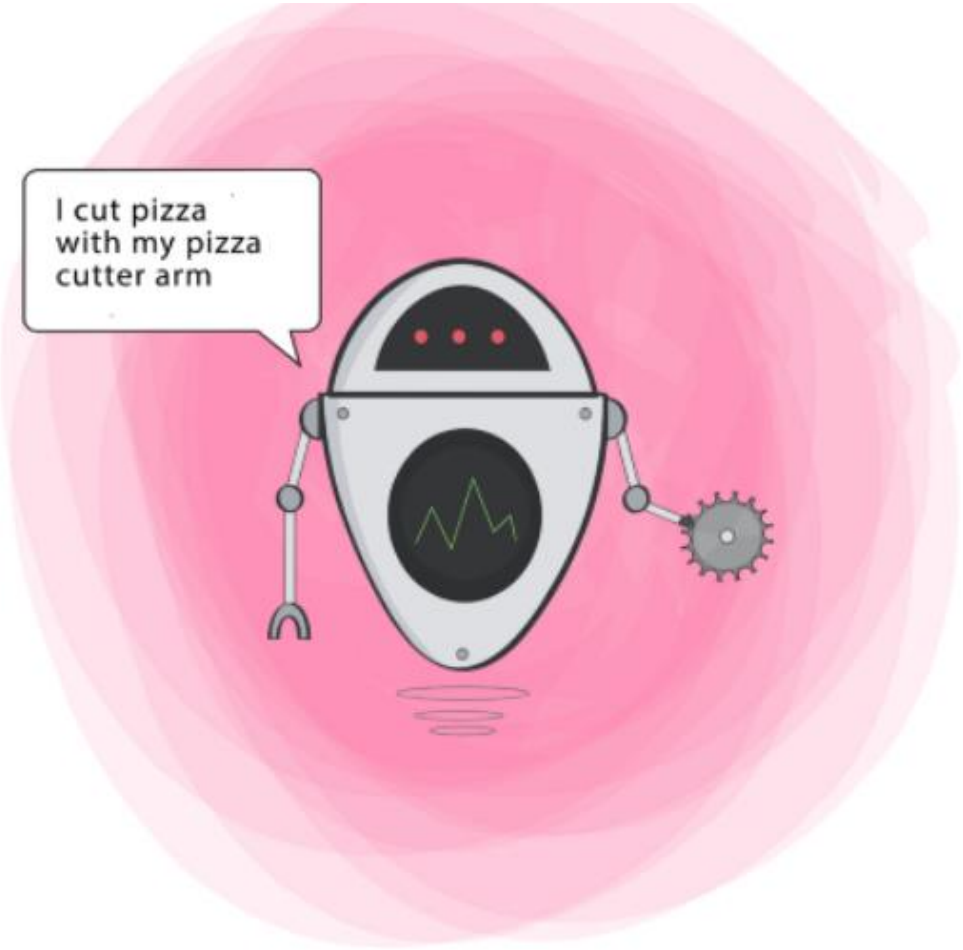
Tomado de: https://medium.com/backticks-tildes/the-s-o-l-i-d-principles-in-pictures-b34ce2f1e898

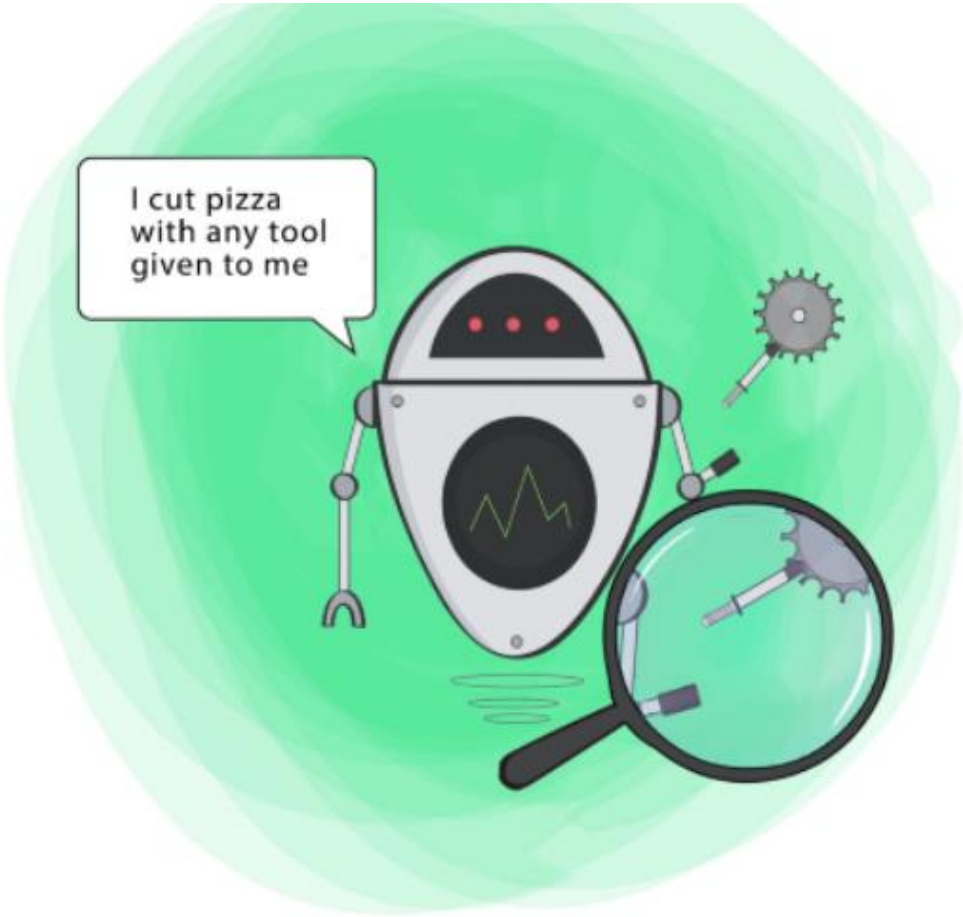




Tomado de: https://medium.com/backticks-tildes/the-s-o-l-i-d-principles-in-pictures-b34ce2f1e898

D





Dependency Inversion

Tomado de: https://medium.com/backticks-tildes/the-s-o-l-i-d-principles-in-pictures-b34ce2f1e898