## SOLID

Uma introdução ao conjunto de princípios fundamentais para

O desenvolvimento de software saudável

#### Princípios solidos

#### Princípios sólidos

#### Organização

#### **Objetivos do**

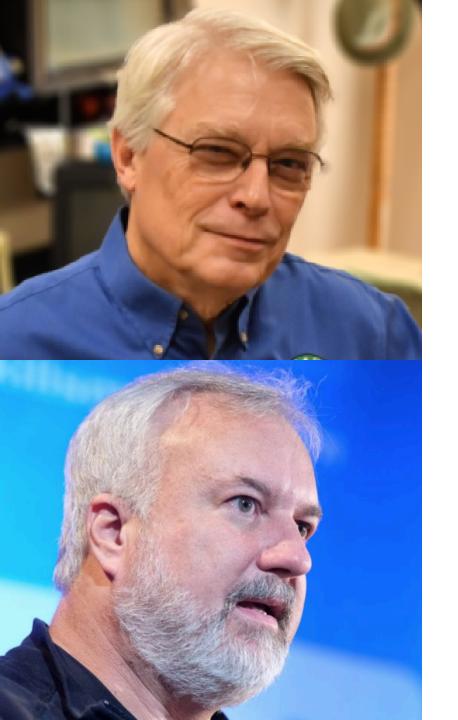
### SOLID

- Tolerar mudanças
- Sejam faceis de entender
- Sejam a base de código que possa ser usada em muitos sistemas de software

Breve

# SOLID

história



Robert C Martin (signatário do Manifesto Agil, autor do da série de livros codigo limpo e desenvolvedor desde a década de 70), vem escrevendo sobre qualidade de software a muito tempo e em 2000 já havia estabelecido um conjunto de princípios e práticas em seus trabalhos e publicações.

Foi ai que em 2004, Michael Feathers (um importante e antigo desenvolvedor da comunidade C++) percebeu que se reorganizasse os princípios, as primeiras letras de cada principio poderiam formar a palavra SOLID.

Assim nasceu os princípios SOLID.

### 

# 

# Single

Responsability

## SRP

#### **Principio**

#### Responsabilidade

Unica

## SRP

```
public class Robot {
   void cook() { ... }
   void text() { ... }
   void state() { ... }
   void image() { ... }
   void video() { ... }
   void youtube() { ... }
}
```

## SRP

```
public class RobotChef {
 void cook() { ... }
public class RobotGarderner {
 void clean() { ... }
public class RobotPainter {
 void paint() { ... }
public class RobotDriver {
 void drive() { ... }
```

## 

#### Open-Closed

#### Princípio

#### (1141) Aberto-Fechado

## OCP

```
public class Robot {
 void exec() {
   this.skill.cut();
public class Robot {
 void exec() {
   this.skill.paint();
```

## OCP

```
public class Robot {
  Robot(Array<Skil>1 skills){ this.skills = skills;}
  void exec() {
   for (Skill skill: this.skills){
      skill.exec();
public class CookSkill implements Skill {
  void exec() { ... }
public class PaintSkill implements Skill {
  void exec() { ... }
// ... more skills
// using
Robot robot = new Robot([
  new CookSkill(),
 new PaintSkill()
]);
```

Liskov

#### Substitution

#### Princípio

# LSP

## Substituição

Liskov

## LSP

## LSP

Interface

#### Segregation

#### Princípio

# ISP Segregação

**Interface** 

# 

#### **Dependency**

#### Inversion

#### Princípio

# nn lersão

Dependência