

# **Variability book**

José Miguel Horcas

2026-01-16

# Table of contents

<b>Preface</b>	<b>3</b>
<b>I I: Fundamentos</b>	<b>4</b>
<b>2 El Efecto Safety Rail</b>	<b>6</b>
2.1 Los Tres Pilares de la Barandilla . . . . .	6
2.1.1 1. Prevención de Inconsistencias Semánticas . . . . .	6
<b>II II: Modelado</b>	<b>7</b>
<b>4 Introduction</b>	<b>9</b>
<b>5 Summary</b>	<b>10</b>
<b>References</b>	<b>11</b>

# Preface

This is a Quarto book.

To learn more about Quarto books visit <https://quarto.org/docs/books>.

## **Part I**

# **I: Fundamentos**

**1**

## 2 El Efecto Safety Rail

El **Safety Rail Effect** (Efecto de Barandilla de Seguridad) es una metáfora en SPL que describe cómo las restricciones del sistema impiden que el usuario cometa errores de configuración, guiándolo siempre por un camino válido.

En frameworks extensibles como **jMetal**, la infraestructura de la SPL actúa como una “guía física” que evita que el desarrollador “caiga” en errores de implementación.

### 2.1 Los Tres Pilares de la Barandilla

#### 2.1.1 1. Prevención de Inconsistencias Semánticas

Las reglas del Modelo de Características (Cross-Tree Constraints) impiden combinar elementos incompatibles.

```
graph TD
  A[Algoritmo] --> B{Codificación}
  B --> C[Binaria]
  B --> D[Real]
  A --> E{Operador Crossover}
  E --> F[Single Point]
  E --> G[SBX]
```

```
G -.-> |Requiere| D
```

```
F -.-> |Excluye| D
```

```
style G fill:#f96,stroke:#333
```

```
style C fill:#dfd,stroke:#333
```

## **Part II**

# **II: Modelado**

**3**



## 4 Introduction

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

## 5 Summary

In summary, this book has no content whatsoever.

## References

Knuth, Donald E. 1984. “Literate Programming.” *Comput. J.* 27 (2): 97–111. <https://doi.org/10.1093/comjnl/27.2.97>.