The Solid Principles

SOLID is an acronym:

- **S: Single Responsibility Principle**
  "An object must have a single responsibility"

- **O: Open / Closed Principle**
  "software entities should be open to extension and closed to modification"

- **L: Liskov Substitution Principle**
  "objects in a program must be replaceable by instances of their subtypes without altering the correctness of the program"

- **I: Interface Segregation Principle**
  "many client-specific interfaces are preferable to one general-purpose interface"

- **D: Dependency Inversion Principle**
  "we must depend on abstractions and not on concretions"

Modularity, abstraction and encapsulation

These mentioned methods help to improve the functionality, development and understanding of a code, among them it can be highlighted that:

- **Modularity**
  Is the ability to ignore the details of the parts in order to focus attention on a higher level of a problem.

- **Abstraction**
  Happens when something is wrapped in a protective layer or shielded from anything that might harm it.

- **Encapsulation**
  Is the act of packing or protecting data or attributes with methods.