Tu socio tecnológico

nunsys®

COMUNICACIONES · SISTEMAS · SOFTWARE · MARKETING · FORMACIÓN

Introducción modelo de desarrollo

28 de septiembre 2016

Introducción

0 Ciclo de vida de una aplicación

1 Factores clave

2 Medir la calidad de nuestro código

4 Ejercicio práctico “My IMDB”

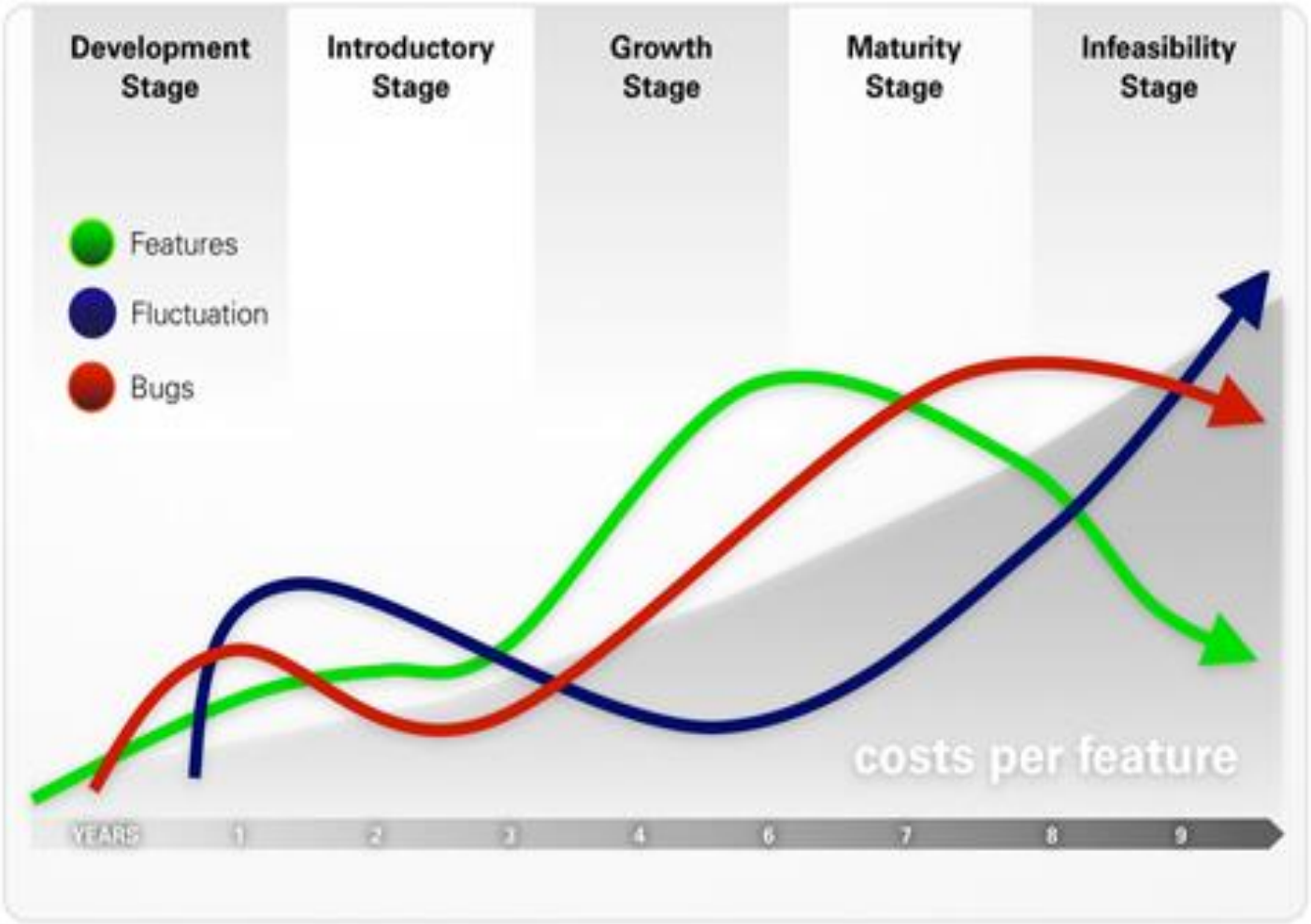
Tu socio tecnológico

nunsys®

COMUNICACIONES · SISTEMAS · SOFTWARE · MARKETING · FORMACIÓN

Cliclo de vida de una aplicación

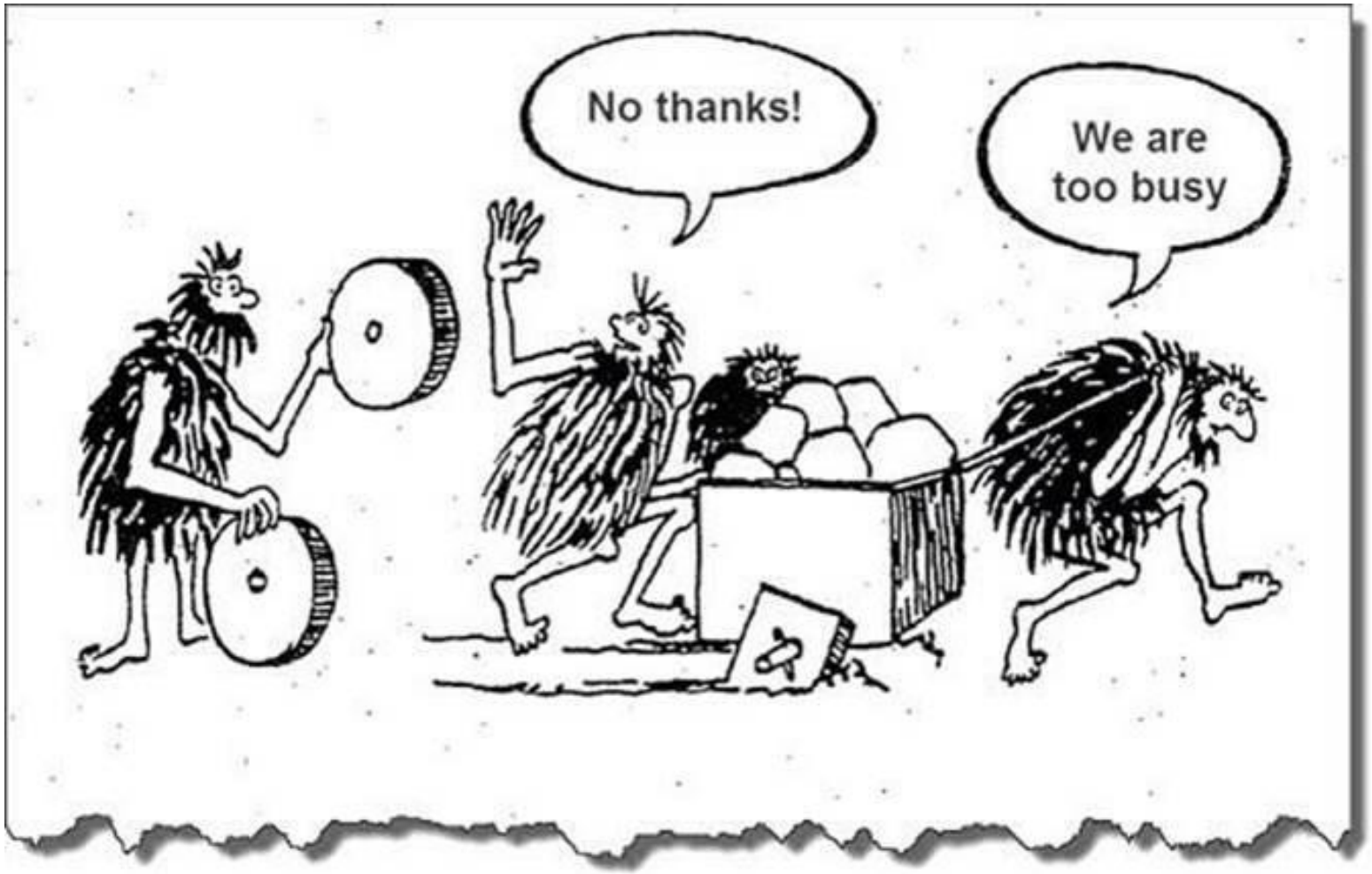
Ciclo de vida de una aplicación



Sistema inviable…

….trabajo para héroes

¿Por qué se llega a estas situaciones?



¿Por qué se llega a estas situaciones?

- Errores en la toma de requisitos
- Gestión de la configuración inexistente o deficiente
- Código de mala calidad
  - Lógica duplicada
  - Clases muy grandes
  - Métodos muy largos
  - Nombres de variables, métodos y clases poco explicativos
  - Muchos parámetros en las funciones
  - Complejidad ciclomática alta
  - La aplicación depende de librerías o frameworks obsoletos
  - Existe una alta complejidad ciclomática
  - Existe un alto acoplamiento entre clases
  - Código inalcanzable (Dead code)
  - Código especulativo. Intentar solucionar problemas que aún no se han presentado
- Código no testeable

Código de mala calidad – Ejemplo 1

```
public class Customer {

    private CustomerDAO customerDAO;
    private int customerID;
    private String name;
    private String email;

    public Customer(CustomerDAO customerDAO) {
        this.customerDAO = customerDAO;
    }

    public String GetName() {
        name = name.toUpperCase();
        return name;
    }

    public String GetEmail() {
        email = email.toLowerCase();
        return email;
    }

    public void SetName(String name) {
        this.name = name;
    }

    public void SetEmail(String email) {
        this.email = email;
    }

    public void save() {
        customerDAO.save(this);
    }
}


public interface CustomerDAO {
    Customer findById(int customerID);
    List<Customer> findAll();
    void save(Customer customer);
}
```

Código de mala calidad – Ejemplo 2

```java
import org.apache.log4j.LogManager;
import org.apache.log4j.Logger;

public class Main {
    private static final Logger logger = LogManager.getLogger(Main.class);

    public static void main(String[] args) {
        Main main = new Main();
        main.execute(args);
    }

    private void execute(String[] args) {
        try {
            validateArgs(args);
            doWork(args[0]);
        } catch (Exception e) {
            logger.error("Invalid parameters", e);
        }
    }

    private void validateArgs(String[] args) throws
                    NumberOfParametersException, ParameterFormatException {

        if (args.length == 1) {
            if (!isValidArgument(args[0]))
            {
                throw new ParameterFormatException();
            }
        } else {
            throw new NumberOfParametersException();
        }
    }

    private boolean isValidArgument(String arg) {
        return arg.length() == 5;
    }

    private void doWork(String idContract) {
        System.out.println(idContract);
    }

}
```

Código de mala calidad – Ejemplo 3

```
public class OrderProcessor {

    public void process(Order order) {
        if (order.isValid() && this.save(order))
        {
            this.sendConfirmationMessage(order);
        }
    }

    private boolean save(Order order) {
        boolean result = false;

        // TODO Save order in database

        return result;
    }

    private void sendConfirmationMessage(Order order) {
        String name = order.getCustomer().getName();
        String email = order.getCustomer().getEmail();

        // TODO send email

    }

}
```

Código de mala calidad – Ejemplo 4

```
public class OtherOrderProcessor {

    public void process(Order order) {
        OrderRepository repository = new OrderRepository();

        if (order.isValid() && repository.save(order))
        {
            EmailNotifier notifier = new EmailNotifier();
            notifier.sendConfirmationMessage(order);
        }
    }
}
```

Código de mala calidad – Ejemplo 5

```
public class CustomerRepository {
    private static CustomerRepository INSTANCE = null;

    private CustomerRepository(){}

    private static void createInstance() {
        if (INSTANCE == null) {
            INSTANCE = new CustomerRepository();
        }
    }

    public static CustomerRepository getInstance() {
        if (INSTANCE == null) createInstance();
        return INSTANCE;
    }
}
```

Código de mala calidad – Ejemplo 6

```
import junit.framework.TestCase;
import org.apache.log4j.LogManager;
import org.apache.log4j.Logger;

class OrderProcessorTest extends TestCase {
    private static final Logger logger =
LogManager.getLogger(OrderProcessorTest.class);

    void testProcess() {

        OrderProcessor orderProcessor = new OrderProcessor();

        Customer customer = new Customer();
        customer.SetEmail("customer@email.com");
        customer.SetName("Peter Smith");

        Order order = new Order();
        order.setOrderId(1);
        order.setValid(true);
        order.setCustomer(customer);


        try {
            orderProcessor.process(order);
        } catch (Exception) {
            logger.debug("Order not processed");
        }

    }

}
```

Código de mala calidad – Ejemplo 7

```
public class Payroll {

    public Money calculatePay(Employee e) throws
InvalidEmployeeTypeException {
        // Enter in fuction CalculatePay
        switch (e.type) {
            case COMMISSIONED:
                // Calculate commissioned pay
                return calculateCommissionedPay(e);
            case HOURLY:
                // Calculate hourly pay
                return calculateHourlyPay(e);
//            case MONTHLY:
//                // Calculate monthly pay
//                return calculateMonthlyPay(e);
            case SALARIED:
                // Calculate salaried pay
                return calculateSalariedPay(e);
            default:
                // throw InvalidEmployeeTypeException
                throw new InvalidEmployeeTypeException(e.type);
        }
    }

    private Money calculateSalariedPay(Employee e) {
        // TODO - Calculate salaried pay
        return null;
    }

    private Money calculateHourlyPay(Employee e) {
        // TODO - Calculate hourly pay
        return null;
    }

    private Money calculateCommissionedPay(Employee e) {
        // TODO - Calculate commissioned pay
        return null;
    }

}
```

Código de mala calidad – Ejemplo 8

```
public class Rectangle {

    private int x1;
    private int y1;
    private int x2;
    private int y2;

    public Rectangle(int x1, int y1, int x2, int y2) {
        this.x1 = x1;
        this.y1 = y1;
        this.x2 = x2;
        this.y2 = y2;
    }

    public boolean contains(int x1, int y1, int x2, int y2){
        return (this.x1 < x1 && y1 < y1 &&
                this.x1 + x2 - x1 > this.x2  &&
                y1 + y1 - y2 > this.y2);
    }

}
```

Código de mala calidad – Ejemplo 9

```
public enum EnumResponseType {
    RESPONSE_OK,
    RESPONSE_READING_ERROR,
    RESPONSE_DEVICE_ERROR
}

public interface QRDevice {

    EnumResponseType open();
    EnumResponseType readData();
    String getData();
    EnumResponseType close();

}
```

Código de mala calidad – Ejemplo 10

```java
import java.util.List;

public class Project {

    private List<ProjectFile> projectFileList;

    public List<ProjectFile> getProjectFileList() {
        return projectFileList;
    }

    public void setProjectFileList(List<ProjectFile> projectFileList)
        this.projectFileList = projectFileList;
    }

    public void loadAllFiles() {
        for (ProjectFile file: projectFileList) {
            file.loadFileData();
        }
    }

    public void saveAllFiles() throws InvalidOperationException {
        for (ProjectFile file: projectFileList) {
            if (!(file instanceof ReadOnlyFile)) {
                file.saveFileData();
            }
        }

    }

}
```

```java
public class ProjectFile {

    private String filePath;
    private byte[] fileData;

    public String getFilePath() {
        return filePath;
    }

    public void setFilePath(String filePath) {
        this.filePath = filePath;
    }

    public byte[] getFileData() {
        return fileData;
    }

    public void setFileData(byte[] fileData) {
        this.fileData = fileData;
    }

    public void loadFileData() {
        // TODO Retrieve FileData from disk
    }

    public void saveFileData() throws InvalidOperationException {
        // TODO Write FileData to disk
    }
}


public class ReadOnlyFile extends ProjectFile {

    public void saveFileData() throws InvalidOperationException {
        throw new InvalidOperationException();
    }

}
```

¿Cómo podemos evitar estos problemas?

- Entendiendo bien el problema que tenemos que resolver -> Comunicación
- Desarrollo iterativo e incremental
- Gestionando nuestro código fuente con un sistema de control de versiones
- Aplicando buenas prácticas de programación
  - Principios SOLID
  - Patrones de diseño
  - Código limpio
  - Diseño basado en múltiples capas
  - Automatización de pruebas
- Medir la calidad del código
- Implantar un sistema de integración continua

Tu socio tecnológico

nunsys®

COMUNICACIONES · SISTEMAS · SOFTWARE · MARKETING · FORMACIÓN

Factores clave

Factores clave

Tu socio tecnológico

nunsys®

COMUNICACIONES · SISTEMAS · SOFTWARE · MARKETING · FORMACIÓN

Medir la calidad de nuestro código

Medir la calidad de nuestro código

http://docs.sonarqube.org/display/SONAR/Get+Started+in+Two+Minutes

Get Started in Two Minutes

1. Download and unzip the SonarQube distribution (let's say in "C:\sonarqube")
2. Start the SonarQube server:
On Windows, execute:
C:\sonarqube\bin\windows-x86-xx\StartSonar.bat

3. Download and unzip the SonarQube Scanner (let's say in "C:\sonar-scanner")
4. Download and unzip some project samples (let's say in "C:\sonar-examples")
5. Analyze a project:
On Windows:
cd C:\sonar-examples\projects\languages\java\sonar-runner\java-sonar-runner-simple
C:\sonar-scanner\bin\sonar-scanner.bat

cd C:\sonar-examples\projects\languages\javascript\javascript-sonar-runner
C:\sonar-scanner\bin\sonar-scanner.bat

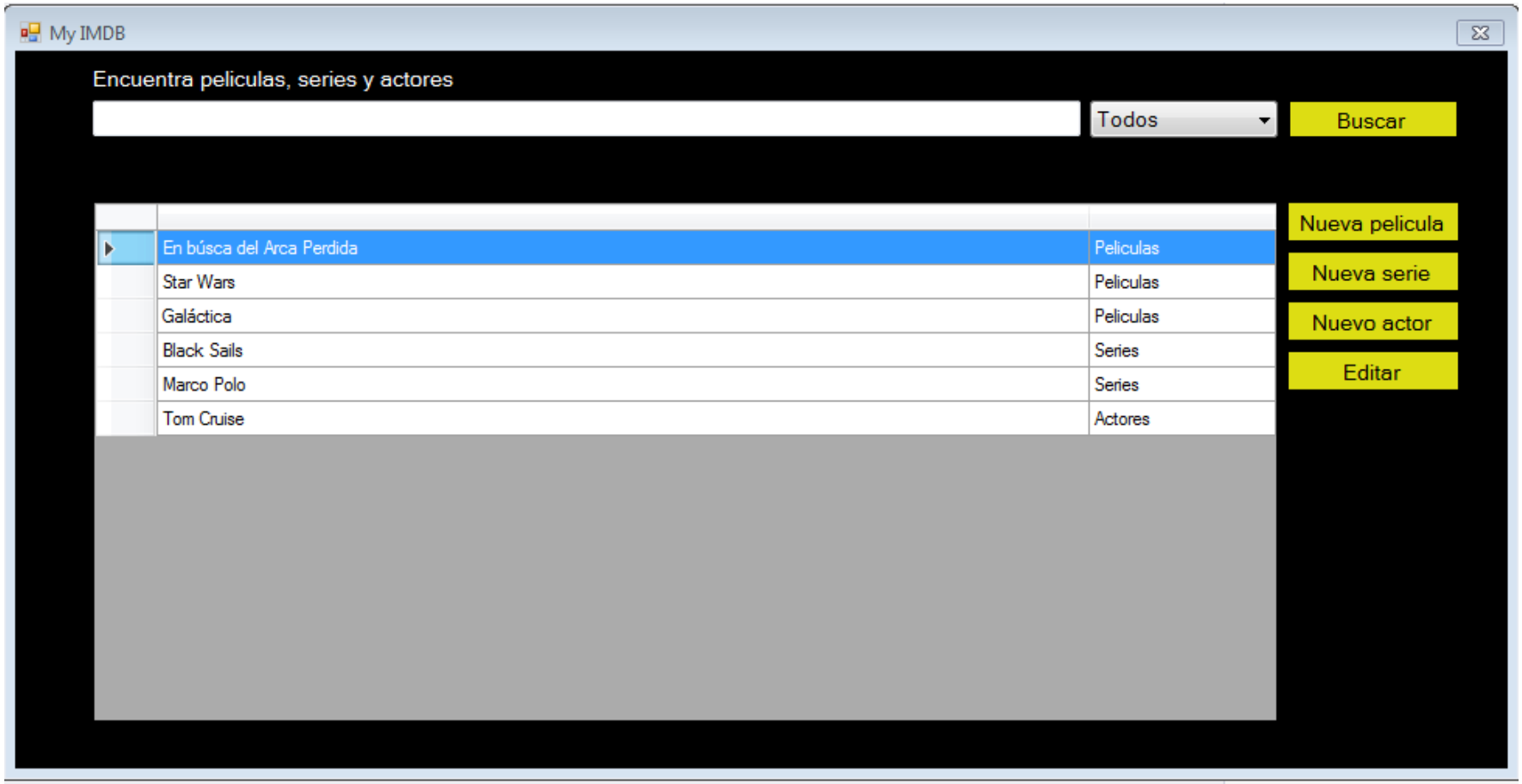
Tu socio tecnológico

nunsys®

COMUNICACIONES · SISTEMAS · SOFTWARE · MARKETING · FORMACIÓN

Ejercicio práctico “My IMDB”

Ejercicio práctico – My IMDB



Principios SOLID

Inicial	Acrónimo	Concepto
S	SRP	Principio de responsabilidad única (*Single responsibility principle*)la noción de que un objeto solo debería tener una única responsabilidad.
O	OCP	Principio de abierto/cerrado (*Open/closed principle*)la noción de que las "entidades de software … deben estar abiertas para su extensión, pero cerradas para su modificación".
L	LSP	Principio de sustitución de Liskov (*Liskov substitution principle*)la noción de que los "objetos de un programa deberían ser reemplazables por instancias de sus subtipos sin alterar el correcto funcionamiento del programa". Ver también diseño por contrato.
I	ISP	Principio de segregación de la interfaz (*Interface segregation principle*)la noción de que "muchas interfaces cliente específicas son mejores que una interfaz de propósito general".[5]
D	DIP	Principio de inversión de la dependencia (*Dependency inversion principle*)la noción de que se debe "depender de abstracciones, no depender de implementaciones".[5] La Inyección de Dependencias es uno de los métodos que siguen este principio.