SINGLETON PATTERN

EJEMPLOS

- Objetos que necesitamos se creen una sola vez:
  - registros
  - preferencias globales o configuración
  - logging
  - caches
  - pools
  - ...

“

Ensure a class only has one instance, and provide a global point of access to it.

— *GoF*

”

```java
package org.formacion.singleton;

public class Unico implements Serializable {

  private static Unico INSTANCE;

  public static Unico getUnico() {
    if (INSTANCE == null) {
      INSTANCE = new Unico();
    }
    return INSTANCE;
  }

}
```

¿problemas?

```java
package org.formacion.singleton;

public class Unico implements Serializable {

  private static Unico INSTANCE;

  private Unico() {}

  public static Unico getUnico() {
    if (INSTANCE == null) {
      INSTANCE = new Unico();
    }
    return INSTANCE;
  }

}
```

¿Más problemas?

```
package org.formacion.singleton;

public class Unico implements Serializable {

  private static Unico INSTANCE;

  private Unico() {}

  public synchronized static Unico getUnico() {
    if (INSTANCE == null) {
      INSTANCE = new Unico();
    }
    return INSTANCE;
  }

}
```

¿Más Problemas?

```java
package org.formacion.singleton;

public class Unico implements Serializable {

  private static Unico INSTANCE;

  private Unico() {}

  public synchronized static Unico getUnico() {
    if (INSTANCE == null) {
      INSTANCE = new Unico();
    }
    return INSTANCE;
  }

  private Object readResolve() {
      return INSTANCE;
  }
}
```

Controla la des-serialización de un objeto

A PARTIR DE JAVA 1.5 : USAR ENUMS

```
public enum Unico {

  INSTANCE;

  // metodos

}
```

A partir de Java 1.5 se recomienda usar Enums con una sola instancia

```
Unico unico = Unico.INSTANCE;
```