Universidad Nacional de Quilmes

CONSTRUCCIÓN DE INTERFACES DE USUARIO

1er Cuatrimestre de 2019

Validaciones

- Las validaciones son técnicas que permiten asegurar que los valores con los que se vaya a operar estén dentro de determinado dominio.

Validaciones

- Validaciones a nivel de Vista (View)
  - Aquellas que deben ser realizadas sin la necesidad de intervención del modelo
  - La validación se realiza en la misma capa.
  - Menos costosa
- Validaciones a nivel de Modelo
  - Aquellas que deben ser realizadas por el modelo dado que implican reglas de negocio.
  - Deben se informadas a la capa View.
  - Mas costosa

Validaciones a nivel Modelo

Regla importante:

Mensajes descriptivos que permitan al usuario identificar claramente cuál regla de negocio no se está cumpliendo y le permita continuar con el proceso.

Validaciones a nivel Vista

- “Los km solo pueden ser valores numéricos”
- “El formato de fecha es inválido; debe ingresar el formato dd/mm/aaaa”
- Debe ingresar un máximo de 20 dígitos ”

```
val numericField = TextBox(mainPanel)
numericField.bindValueToProperty<Int,ControlBuilder>( modelProperty: "input")
numericField.setWidth(12)
numericField.withFilter { event -> event.potentialTextResult.matches(Regex( pattern: "[0-9]*")) }
```

Validaciones a nivel Modelo

- “Debe ingresar un número de cliente válido”
- “Ya existe un capítulo con igual identificador para la serie seleccionada”

```
override fun extract(saldo: Int): Int {
    if (available >= discovered + saldo) {
        available -= saldo
    } else {
        throw Exception("No puedes superar el descubierto")
    }
    return available
}
```

En Arena las ventanas del tipo SimpleWindows manejan UserException mostrando el mensaje de error

Excepciones
Manejo de errores

Son errores propios de la programación, que hace que no sean modelados, simplemente ocurren y que identificarlas y tratarlas

Ejemplos de errores:

- NullPointerException
- ArrayIndexOutOfBoundsException
- ArithmeticException

Validaciones a nivel Modelo
Elevar Excepciones

Elevar excepciones, ya sea las excepciones existentes en Java/Kotlin o custom.

```
fun fail(message: String): Nothing {
    throw IllegalArgumentException(message)
}
```

Validaciones a nivel Vista en Arena

Tratar lasa excepciones elevadas por el modelo y elevar la excepción de Arena UserException

```
try {
   eliminarUsuariu(usuario)
}
catch (e: Exception) {
  UserException("No se puede eliminar un usuario asociado a una cuenta")
}
finally {
   // optional finally block
}
```

¿Preguntas?
¿Demo?