

## **DISEÑO DE PRUEBAS UNITARIAS**

**Diana Lorena Balanta Solano**

**Danna Alexandra Espinosa Arenas**

**Carlos Javier Bolaños Riascos**

**Computación y Estructuras Discretas III**

**Universidad Icesi**

**Profesor: Andrés Alberto Aristizabal Pinzon**

## Test Case Design

### Configuración de escenarios: Gramática.

Nombre	Clase	Escenario
setUp	GramaticaTest	Creación de un objeto Gramática, estos ya tienen definidas las variables, terminales y producciones necesarias.

### Diseño de Casos de Prueba

#### Prueba No. 1: Caso estándar.

<b>Objetivo de la Prueba:</b> Verificar que el método “ <b>validar_cadena</b> ” funcione correctamente al ingresar una cadena que cumple con la gramática “Yo creo que el final puede mejorar con”.				
Clase	Método	Escenario	Valores de Entrada	Resultado
Gramatica	validar_cadena / test_estandar	setUp	“Yo creo que el final puede mejorar con la propuesta de agregar mascotas”.	El método debe retornar “True”, ya que, esta cadena sigue el formato de la gramática.

#### Prueba No. 2: Casos límite I.

<b>Objetivo de la Prueba:</b> Verificar que el método “ <b>validar_cadena</b> ” funcione correctamente al ingresar una cadena vacía.				
Clase	Método	Escenario	Valores de Entrada	Resultado
Gramatica	validar_cadena / test_limite1	setUp	“”.	El método debe retornar “False”, ya que, esta cadena no sigue el formato de la gramática.

#### Prueba No. 3: Casos límite II.

<b>Objetivo de la Prueba:</b> Verificar que el método “ <b>validar_cadena</b> ” funcione correctamente al ingresar un solo terminal generado por la gramática.				
Clase	Método	Escenario	Valores de Entrada	Resultado
Gramatica	validar_cadena / test_limite2	setUp	“Yo”.	El método debe retornar “False”, ya que, esta “Yo” es una terminal, pero no es toda la gramática generada.

**Prueba No. 4: Caso interesante.**

**Objetivo de la Prueba:** Verificar que el método “**validar\_cadena**” funcione correctamente al ingresar varios elementos de la gramática.

Clase	Método	Escenario	Valores de Entrada	Resultado
Gramatica	validar_cadena / test_interesante	setUp	“Yo puede mejorar”.	El método debe retornar “False”, aunque los elementos sean parte de la gramática, no siguen el patrón de las producciones.

**Configuración de escenarios: Expresión.**

Nombre	Clase	Escenario
setUp	ExpresionTest	Creación de un objeto Expresión, estos ya tienen definidas las expresiones regulares a utilizar.

**Diseño de Casos de Prueba**

**Prueba No. 1: Caso estándar.**

**Objetivo de la Prueba:** Verificar que los métodos “**validateName**” y “**validateNewFinal**” funcionen correctamente al ingresar una cadena.

Clase	Método	Escenario	Valores de Entrada	Resultado
Expresion	validateName /test_estandar1	setUp	“JohnDue”.	El método debe retornar “True”, ya que, esta cadena cumple con la expresión regular.
Expresion	validateNewFinal /test_estandar4	setUp	“Yo”.	El método debe retornar “True”, ya que, esta cadena cumple con la expresión regular.

**Prueba No. 2: Casos límite I.**

**Objetivo de la Prueba:** Verificar que los métodos “**validateName**” y “**validateNewFinal**” funcionen correctamente al ingresar una cadena de valores límite.

Clase	Método	Escenario	Valores de Entrada	Resultado
Expresion	validateName /test_limite1	setUp	“A”.	El método debe retornar “False”, ya que, esta cadena no cumple con la expresión regular.
Expresion	validateNewFinal /test_limite2	setUp	“”.	El método debe retornar “False”, ya que, esta cadena no cumple con la expresión regular.

**Prueba No. 3: Caso interesante.**

<b>Objetivo de la Prueba:</b> Verificar que los métodos “ <b>validateName</b> ” y “ <b>validateNewFinal</b> ” funcionen correctamente al ingresar una cadena.				
Clase	Método	Escenario	Valores de Entrada	Resultado
Expresion	validateName /test_interesante1	setUp	“John-Due”.	El método debe retornar “False”, ya que, esta cadena no cumple con la expresión regular.
Expresion	validateNewFinal /test_interesante2	setUp	“CREO”.	El método debe retornar “True”, ya que, esta cadena cumple con la expresión regular.

**Configuración de escenarios: Transductor.**

Nombre	Clase	Escenario
setUp	TransductorTest	Creación de un objeto Transductor, estos ya tienen definidas las transiciones a utilizar para la traducción.

**Diseño de Casos de Prueba**

**Prueba No. 1: Caso estándar.**

<b>Objetivo de la Prueba:</b> Verificar que los métodos “ <b>translateDialog</b> ” y “ <b>init_transductor</b> ” funcionen correctamente al ingresar una cadena.				
Clase	Método	Escenario	Valores de Entrada	Resultado
TransductorTest	translateDialog /test_estandar1	setUp	“Nvk od oev”.	El método debería retornar “Soy el eco”, con esto, la prueba debe retornar “True”, ya que, cumple la traducción.

**Prueba No. 2: Casos límite I.**

<b>Objetivo de la Prueba:</b>				
Clase	Método	Escenario	Valores de Entrada	Resultado
TransductorTest	translateDialog /test_limite1	setUp	“ “	El método debería retornar una lista vacía. Con esto, la prueba debe retornar “True”, ya que cumple la traducción.

**Prueba No. 3: Caso interesante.**

Objetivo de la Prueba:				
Clase	Método	Escenario	Valores de Entrada	Resultado
TransductorTest	translateDialog/ test_interesante1	setUp	"Nk o oev o"	El método debería retornar una cadena diferente a la traducción definida "Soy el eco". Por lo que la prueba debe retornar "False".

#### Configuración de escenarios: Autómata.

Nombre	Clase	Escenario
setUp	AutomataTest	Creación de un objeto Automata, estos ya tienen definidas las transiciones y estados de aceptación.

#### Diseño de Casos de Prueba

**Nota:** La definición del autómata y el cambio de sus estado se hizo todo en el método `__init__`, por lo que nos decidimos por probar la definición del objeto autómata y como interactúa con las opciones ingresadas por el usuario.

#### Prueba No. 1: Caso estándar.

Objetivo de la Prueba: Validar que la definición del autómata sea acepte las elecciones ingresadas por el usuario.				
Clase	Método	Escenario	Valores de Entrada	Resultado
Automata	<code>__init__</code> / test_estandar	setUp	"ab"	El resultado de esta prueba debe ser True, ya que, la definición de nuestro autómata acepta esta cadena de opciones.

#### Prueba No. 2: Casos límite I.

Objetivo de la Prueba: Validar que la definición del autómata sea acepte las elecciones ingresadas por el usuario.				
Clase	Método	Escenario	Valores de Entrada	Resultado
Automata	<code>__init__</code> / test_limite1	setUp	"bc"	El resultado de esta prueba debe ser False, ya que, la definición del autómata no acepta la cadena de opciones ingresada por el usuario

				porque no existe una transición para la opción "c".
Automata	<code>__init__/ test_limite2</code>	setUp	"b"	El resultado de esta prueba debe ser False, ya que, la definición del autómata no acepta la cadena de opciones ingresada porque no llega a un estado de aceptación.

**Prueba No. 3: Caso interesante.**

**Objetivo de la Prueba:** Validar que la definición del autómata sea acepte las elecciones ingresadas por el usuario.

Clase	Método	Escenario	Valores de Entrada	Resultado
Automata	<code>__init__/ test_interesante</code>	setUp	"caac"	El resultado de esta prueba debe ser False, ya que, la definición del autómata no acepta cadenas de opciones con vayan a un estado de aceptación con opciones inválidas.