*Logic Specification Template*

|  |  |  |  |
| --- | --- | --- | --- |
| **Student** | Gerardo Aldair Ponce Gomez | **Program #** | 4 |

|  |  |
| --- | --- |
| **Class Name** | Calcular |

|  |  |
| --- | --- |
| **Method Name** | calculaGamma |

|  |  |
| --- | --- |
| **Parameters** | x: double |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
| If x ==1 |
| Return 1 |
| Else x == 0.5 |
| Return sqrt(pi) |
| Else |
| Return (x-1) gamma(x-1) |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

|  |  |
| --- | --- |
| **Class Name** | Calcular |

|  |  |
| --- | --- |
| **Method Name** | calculaValor |

|  |  |
| --- | --- |
| **Parameters** | x: double |
|  | dof: int |
|  |  |
|  |  |
|  |  |

|  |
| --- |
| xi = 0 |
| W = x/numseg |
| step1, step2, aux |
| fx = 0 |
| Vector myvector |
| num\_seg= 10 |
| Gamma = calculaGamma(x) |
| distT = Se calcula totalmente gamma |
| Ciclo for hasta numero de segmentos{ |
| Xi = W \* i |
| Step1 = Se calcula primera parte de t |
| Aux = Guarda el calculo de la potencia |
| Step2 = pow(step1,aux) |
| Fx = gamma \* step2 |
| Se meten datos al vector |
| } |
| Double acum4 = 0 |
| Ciclo for de 2 en 2 { |
| Aux = Agarra los valores del vector y los multiplica por 4 |
| Acum4 += aux |
| } |
| Double acum2 = 0 |
| Ciclo for de 2 en 2{ |
| Aux = Agarra los valores del vector y los multiplica por 2 |
| Acum2 += aux |
| } |
| Double P = 0 |
| Aux = W/3 |
| P= Calcula la ecuación final |
| Return p |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

|  |  |
| --- | --- |
| **Class Name** | Imprimir |

|  |  |
| --- | --- |
| **Method Name** | vacioDatos |

|  |  |
| --- | --- |
| **Parameters** | - |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
| Imprime mensaje de error |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

|  |  |
| --- | --- |
| **Class Name** | Imprimir |

|  |  |
| --- | --- |
| **Method Name** | errorDatos |

|  |  |
| --- | --- |
| **Parameters** | - |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
| Imprime mensajes de datos invalidos |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

|  |  |
| --- | --- |
| **Class Name** | Imprimir |

|  |  |
| --- | --- |
| **Method Name** | imprimeResultados |

|  |  |
| --- | --- |
| **Parameters** | x: double |
|  | dof: int |
|  | p: double |
|  |  |
|  |  |

|  |
| --- |
| Imprime X |
| Imprimei dof |
| IMprimie P |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |