**Especificaciones Operacionales**

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiantes** | Sandra V. Londoño, Oscar C. Lopera, Yeifer A. Herrera | **Fecha** | 09/09/2017 |
| **Programa** | Programa 3 | **Programa #** | 3 |
| **Instructor** | Juan Carlos Marin | **Lenguaje** | Java |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Número del escenario** | 1 | **Objetivo de usuario** | Ingresar los conjuntos de datos a procesar. | |
| **Objetivo del escenario** | | Validar los datos ingresados. | | |
| **Fuente** | **Paso** | **Acción** | | **Comentarios** |
| Sistema | 1 | El sistema solicita al usuario ingresar la ruta del archivo | |  |
| Usuario | 2 | El usuario ingresa la ruta del archivo que contiene el primer conjunto de datos muestrales para el cálculo de la regresión. | | La ruta ingresada corresponde a un archivo existente, con la extensión correcta y correcta construcción de los datos. |
| Sistema | 3 | El sistema valida el correcto ingreso de la ruta del archivo. | |  |
| Sistema | 4 | El sistema verifica que la extensión del archivo sea correcta. | |  |
| Sistema | 5 | El sistema verifica que el archivo exista. | |  |
| Sistema | 6 | El sistema valida el contenido del archivo. | |  |
| Sistema | 7 | El sistema confirma el correcto ingreso del archivo y continúa si ejecución. | |  |
|  |  |  | |  |
| Sistema | 1 | El sistema solicita al usuario ingresar la ruta del archivo | |  |
| Usuario | 2 | El usuario ingresa la ruta del archivo que contiene el primer conjunto de datos muestrales para el cálculo de la regresión. | | La ruta ingresada corresponde a un archivo existente, con la extensión incorrecta. |
| Sistema | 3 | El sistema valida el correcto ingreso de la ruta del archivo. | |  |
| Sistema | 4 | El sistema identifica que la extensión ingresada no es correcta. | |  |
| Sistema | 5 | El sistema muestra al usuario mensaje informando que la extensión ingresada no es correcta. | |  |
| Sistema | 6 | El sistema finaliza la ejecución del programa. | |  |
|  |  |  | |  |
| Sistema | 1 | El sistema solicita al usuario ingresar la ruta del archivo | |  |
| Usuario | 2 | El usuario ingresa la ruta del archivo que contiene el primer conjunto de datos muestrales para el cálculo de la regresión. | | La ruta ingresada corresponde a un archivo que no existe. |
| Sistema | 3 | El sistema valida el correcto ingreso de la ruta del archivo. | |  |
| Sistema | 4 | El sistema identifica que el archivo no existe. | |  |
| Sistema | 5 | El sistema muestra mensaje informando al usuario que el archivo no existe. | |  |
| Sistema | 6 | El sistema finaliza la ejecución del programa. | |  |
|  |  |  | |  |
| Sistema | 1 | El sistema solicita al usuario ingresar la ruta del archivo | |  |
| Usuario | 2 | El usuario ingresa la ruta del archivo que contiene el primer conjunto de datos muestrales para el cálculo de la regresión. | | La ruta ingresada corresponde a un archivo existente, con la extensión correcta, pero el archivo se encuentra vacío. |
| Sistema | 3 | El sistema valida el correcto ingreso de la ruta del archivo. | |  |
| Sistema | 4 | El sistema verifica que el archivo existe. | |  |
| Sistema | 5 | El sistema valida el contenido del archivo. | |  |
| Sistema | 6 | El sistema identifica que el archivo se encuentra vacío. | |  |
| Sistema | 7 | El sistema muestra mensaje al usuario indicando que el archivo está vacío. | |  |
| Sistema | 8 | El sistema finaliza la ejecución del programa. | |  |
|  |  |  | |  |
| Sistema | 1 | El sistema solicita al usuario ingresar la ruta del archivo | |  |
| Usuario | 2 | El usuario ingresa la ruta del archivo que contiene el primer conjunto de datos muestrales para el cálculo de la regresión. | | La ruta ingresada corresponde a un archivo existente, con la extensión correcta, pero con el contenido del archivo incorrecto. |
| Sistema | 3 | El sistema valida el correcto ingreso de la ruta del archivo. | |  |
| Sistema | 4 | El sistema verifica que el archivo existe. | |  |
| Sistema | 5 | El sistema valida el contenido del archivo. | |  |
| Sistema | 6 | El sistema identifica que el contenido del archivo no es correcto. | |  |
| Sistema | 7 | El sistema muestra mensaje al usuario indicando que el contenido del archivo no es correcto. | | El archivo correctamente construido |
| Sistema | 8 | El sistema finaliza la ejecución del programa. | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiantes** | Sandra V. Londoño, Oscar C. Lopera, Yeifer A. Herrera | **Fecha** | 09/09/2017 |
| **Programa** | Programa 3 | **Programa #** | 3 |
| **Instructor** | Juan Carlos Marin | **Lenguaje** | Java |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Número del escenario** | 2 | **Objetivo de usuario** | Ingresar el valor estimado (). | |
| **Objetivo del escenario** | | Verificar el ingreso del valor estimado. | | |
| **Fuente** | **Paso** | **Acción** | | **Comentarios** |
| Sistema | 1 | El sistema solicita al usuario ingresar el valor estimado. | |  |
| Usuario | 2 | El usuario ingresa el valor estimado. | | Se ingresa correctamente el valor estimado a procesar. |
| Sistema | 3 | El sistema valida que el dato ingresado es un número | |  |
| Sistema | 4 | El sistema procesa la información y continúa con la ejecución | |  |
|  |  |  | |  |
| Sistema | 1 | El sistema solicita al usuario ingresar el valor estimado. | |  |
| Usuario | 2 | El usuario ingresa el valor estimado. | | Se ingresa incorrectamente un valor estimado a procesar. no numerico |
| Sistema | 3 | El sistema valida que el dato ingresado es no es un número | |  |
| Sistema | 4 | El sistema retorna un mensaje de error indicando al usuario que el dato es inválido y termina la ejecución | |  |
|  |  |  | |  |
| Sistema | 1 | El sistema solicita al usuario ingresar el valor estimado. | |  |
| Usuario | 2 | El usuario no ingresa el valor estimado. | |  |
| Sistema | 3 | El sistema valida que el dato ingresado es no es válido | |  |
| Sistema | 4 | El sistema retorna un mensaje de error indicando al usuario que el dato es inválido y termina la ejecución | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiantes** | Sandra V. Londoño, Oscar C. Lopera, Yeifer A. Herrera | **Fecha** | 09/09/2017 |
| **Programa** | Programa 3 | **Programa #** | 3 |
| **Instructor** | Juan Carlos Marin | **Lenguaje** | Java |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Número del escenario** | 3 | **Objetivo de usuario** | Conocer el resultado. | |
| **Objetivo del escenario** | | Procesar los datos y mostrar los resultados.. | | |
| **Fuente** | **Paso** | **Acción** | | **Comentarios** |
| Sistema | 1 | El sistema procesa los datos ingresados previamente por el usuario. | | Los datos han sido previamente ingresados de manera correcta, y el sistema ha verificado que sean correctos. |
| Sistema | 2 | El sistema retorna el resultado de los cálculos realizados mostrando en pantalla. | |  |

**Especificaciones Funcionales**

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiantes** | Sandra V. Londoño, Oscar C. Lopera, Yeifer A. Herrera | **Fecha** | 09/09/2017 |
| **Programa** | Programa 3 | **Programa #** | 3 |
| **Instructor** | Juan Carlos Marin | **Lenguaje** | Java |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Clase Nodo** | |  | | | | |
|  | | La clase Nodo crea objetivos de tipo nodo los cuales se utilizan para realizar la construcción de listas ligas con otra clase. | | | | |
|  | | En esta clase contiene los elementos de un nodo y como acceder a ellos o modificarlos. | | | | |
|  | |  | | | | |
|  | | | | | | |
| **Métodos** | | | | | | |
|  | **Declaración** | | | **Descripción** | | |
|  | public void setSiguiente(Nodo X) | | | Modifica el valor que direcciona el nodo siguiente del nodo actual.  Atributos:  Nodo X: (Nodo) Valor del nodo siguiente al cual quedara apuntando el nodo actual, como nodo siguiente. | | |
|  | public void SetDato(double dato) | | | Modifica el valor que contiene el el nodo  Atributos:  double dato: (Double) Valor que va a ser asignado al nodo que se desea modificar. | | |
|  | public double getDato() | | | Retorna el valor que contiene el nodo | | |
|  | public Nodo getSiguiente() | | | Retorna el valor que direcciona al nodo siguiente del nodo actual | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Clase Lista Ligada** | |  | | | | |
|  | | La clase la clase lista ligada crea un objetivo de tipo lista ligada con el objetivo de administrar los ingresos, modificaciones, búsquedas o borrados de nodos en ella. Esta lista crea y utiliza objetos de la clase nodo ayudando a crear la lista de información de manera continua. | | | | |
|  | |  | | | | |
|  | | | | | | |
| **Métodos** | | | | | | |
|  | **Declaración** | | | **Descripción** | | |
|  | public boolean vacia() | | | El método retorna la verifica si la lista ligada está vacía. | | |
|  | public Nodo getPrimerNodo() | | | El método retorna el valor del primer nodo en la lista | | |
|  | public Nodo getUltimoNodo() | | | El método retorna el valor del último nodo en la lista | | |
|  | public Nodo Back(Nodo X) | | | El método retorna el valor del nodo anterior al cual se envía como parámetro.  Atributos:  Nodo X: (Nodo) Nodo del cual se desea conocer el nodo anterior. | | |
|  | public void insertarNodo(double dato) | | | El método realiza la creación de un nodo con el valor ingresado como parámetro y lo agrega al final de la lista ligada creda.  Atributos:  double dato: (Double) Valor que debe contener el nuevo nodo que se va a agregar a la lista | | |
|  | public Nodo buscarNodo(double dato) | | | El método realiza la búsqueda de un nodo que contenga el valor ingresado como parámetro. Si encuentra un nodo con el valor igual realiza el retorno de este nodo.  Atributos:  double dato: (Double) Valor que debe contener el nodo que va a ser buscando en la lista ligada | | |
|  | public void Delete(double dato) | | | El método realiza la búsqueda de un nodo que contenga el valor ingresado como parámetro. Si encuentra un nodo con ese parámetro realiza el borrado de este nodo a atreves de apuntar el nodo anterior al nodo siguiente del encontrado.  Atributos:  double dato: (Double) Valor que se desea eliminar de una lista ligada, si se encuentra en ella. | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| Clase Programa3 | |  | | | | |
|  | | Esta es la clase principal del programa. En esta clase se utilizan objetos de la clase Nodo y de la clase ListaLigada con el objetivo de realizar los procesos administrativos de información y cálculo de datos para entregar al usuario la respuesta esperada si se ingresan los datos de manera adecuada. En esta clase se encuentra la interacción con el usuario para el ingreso de la información a ser leída. La utilización y los métodos que se requieren para encontrar los valores que se deben calcular y la forma como el sistema muestra los resultados obtenidos. | | | | |
|  | |  | | | | |
|  | | | | | | |
| Metodo | | | | | | |
|  | public static ListaLigada LeerArchivo (String archivo) | | | Este método se encarga de leer el archivo que se encuentra en la ruta ingresada por el usuario y llenar una lista ligada con la información contenido en el archivo si tiene la información adecuada para el proceso.  Atributos:  String archivo: En esta cadena de caracteres se encuentra la ruta del archivo que va a ser leído por el sistema | | |
|  | public static void main() | | | Este es el programa principal de ejecución en el cual se interactúa con el usuario y además se realizar las invocaciones de métodos para realizar la actividad que la aplicación cumple. | | |
|  | public static boolean verificarExtension(String ruta) | | | Este metodo verifica que la extensión del archivo ingresado por el usuario sea la adecuada para el proceso de ejecución.  Atributos:  String ruta: En esta cadena de caracteres se encuentra la ruta del archivo del cual se debe evaluar la extensión que tiene. | | |
|  | public static double desviacionEstandar(ListaLigada lista) | | | Este método realiza el cálculo de la desviación estándar con la información contenida en una lista ligada.  Atributos:  ListaLigada lista: En esta lista ligada se encuentran los datos con los cuales se va a calcular la desviación estándar. | | |
|  | public static double suma(ListaLigada lista) | | | Este método realiza la suma de todos los datos encontrados en una lista ligada.  Atributos:  ListaLigada lista:En esta lista ligada se encuentran los datos que se van a sumar para retornar el total. | | |
|  | public static ListaLigada multiplicar\_Listas(ListaLigada x,ListaLigada y) | | | Este método realiza la multiplicación posicion a posicion entre 2 listas ligadas.  Atributos:  ListaLigada x: Primera lista con los datos que se desean multiplicar con la segunda lista ingresada.  ListaLigada y: Segunda lista con los datos que se desean multiplicar con la primera lista ingresada. | | |
|  | public static double obtener\_B1(double xy,double media\_x, double media\_y, double xx, int cant\_datos) | | | Este método realiza el cálculo del valor B1 solicitado por el usuario en la entrega de resultados.  Atributos:  double xy: Suma del producto de los listas ligadas x y y.  double media\_x: Media calculada de la lista ligada x.  double media\_y: Media calculada de la lista ligada y.  double xx: Suma del producto de los listas ligadas x y x.  int cant\_datos: Cantidad de datos ingresados en una de las listas del programa. | | |
|  | public static double obtener\_B0(double media\_x, double media\_y, double b1) | | | Este método realiza el cálculo del valor B0 solicitado por el usuario en la entrega de resultados.  Atributos:  double media\_x: Media calculada de la lista ligada x.  double media\_y: Media calculada de la lista ligada y.  double b1: Valor calculado de B1. | | |
|  | public static double obtener\_r\_xy(int cant\_datos, double xy ,double x, double y,double xx, double yy) | | | Este método realiza el cálculo del valor R solicitado por el usuario en la entrega de resultados.  Atributos:  int cant\_datos: Cantidad de datos ingresados en una de las listas del programa.  double xy: Suma del producto de los listas ligadas x y y.  double x: Suma de los elementos contenidos en la lista ligada x.  double y: Suma de los elementos contenidos en la lista ligada y.  double xx: Suma del producto de los listas ligadas x y x.  double yy: Suma del producto de los listas ligadas y y y. | | |
|  | public static void mostrar\_datos(double b0 ,double b1, double r\_xy ,double r2, double yk) | | | En este método se realiza la impresión en pantalla para el usuario de la información que calculo el sistema y es la esperada como respuesta.  Atributos:  double b0: Valor calculado de B0.  double b1: Valor calculado de B1.  double r\_xy: Valor calculado de R de XY.  double r2: Valor calculado de R cuadrado.  double yk: Valor calculado de Y sub k. | | |

**Espicificación de Esados**

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiantes** | Sandra V. Londoño, Oscar C. Lopera, Yeifer A. Herrera | **Fecha** | 09/09/2017 |
| **Programa** | Programa 3 | **Programa #** | 3 |
| **Instructor** | Juan Carlos Marin | **Lenguaje** | Java |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  | | | | |
| **Nombre del estado** | | | | **Descripción** | | | |
|  | | | |  | | | |
| Nuevo nodo | | | | Se crea un nuevo objeto de la clase nodo con dato en cero (0) y siguiente en nulo. | | | |
| Dato modificado | | | | Se modifica el valor del dato, el valor de siguiente continúa nulo. | | | |
| Siguiente modificado | | | | Se modifica el valor de siguiente. | | | |
|  | | | |  | | | |
| **Función/Parámetros** | | | | **Descripción** | | | |
|  | | | |  | | | |
| setSiguiente(Nodo X) | | | | Modifica el valor del siguiente en un objeto tipo Nodo. | | | |
| SetDato(double dato) | | | | Modifica el valor del dato en un objeto tipo Nodo. | | | |
|  | | | |  | | | |
|  | | | |  | | | |
|  | | | |  | | | |
| **Estados/Próximos estados** | | | | **Condición de transisción** | | **Acción** | |
| Nuevo nodo/Dato modificado | | | | Se llama la función setDato y se le envía como parámetro el nuevo dato. | | El dato es modificado. | |
| Nuevo nodo/Siguiente modificado | | | | Se llama la función setSiguiente y se le envía como parámetro el nodo al que hará referencia. | | El siguiente es modificado. | |
| Dato modificado/Siguiente modificado | | | | Se llama la función setSiguiente y se le envía como parámetro el nodo al que hará referencia. | | El siguiente es modificado. | |
| Siguiente modificado/Dato modificado | | | | Se llama la función setDato y se le envía como parámetro el nuevo dato. | | El dato es modificado. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiantes** | Sandra V. Londoño, Oscar C. Lopera, Yeifer A. Herrera | **Fecha** | 09/09/2017 |
| **Programa** | Programa 3 | **Programa #** | 3 |
| **Instructor** | Juan Carlos Marin | **Lenguaje** | Java |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  | | | | |
| **Nombre de Estados** | | | | **Descripción** | | | |
|  | | | |  | | | |
| Lista vacia | | | | Se crea un nuevo objeto de la clase lista con cabeza[0] en nulo (null) y cabeza[1] en nulo (null). | | | |
| Lista modificada | | | | Se modifica la clase lista con cabeza[0] en nodo y cabeza[1] en nodo. | | | |
|  | | | |  | | | |
| **Función/Parámetros** | | | | **Descripción** | | | |
|  | | | |  | | | |
| insertarNodo(double dato) | | | | Modifica el valor cabeza[0] en nodo de valor dato y cabeza[1] en nodo de valor dato.  . | | | |
| Delete(double dato) | | | | Modifica el valor cabeza[0] en nulo (null) y cabeza[1] en nulo (null).. | | | |
|  | | | |  | | | |
|  | | | |  | | | |
|  | | | |  | | | |
| **Estados/Próximos estados** | | | | **Estados/Próximos estados** | | **Acción** | |
| Lista vacia/Lista modificada | | | | Se llama la función insertarNodo(double dato) y se le envía como parámetro el nuevo dato. | | la lista pasa a estado cabeza[0] = nodo de valor dato y cabeza[1] = nodo de valor dato | |
| Lista modificada  /Lista vacia/ | | | | Se llama la funciónDelete(double dato) y se le envía como parámetro el dato a eliminar | | la lista con un solo dato pasa a estado cabeza[0] =nulo (null) y cabeza[1] = nulo (null) | |

|  |  |  |
| --- | --- | --- |
|  | | |

**Especificación Lógica**

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiantes** | Sandra V. Londoño, Oscar C. Lopera, Yeifer A. Herrera | **Fecha** | 09/09/2017 |
| **Programa** | Programa 3 | **Programa #** | 3 |
| **Instructor** | Juan Carlos Marin | **Lenguaje** | Java |

|  |  |
| --- | --- |
| **Referencias de**  **Diseño** | Metodo Suma: |
|  | En este método se recibe una lista ligada con los datos que se sean sumar y se retorna un |
|  | valor de tipo double con la suma de los datos en la lista ligada. |

|  |  |
| --- | --- |
| **Parámetros** | Lista Ligada lista |

Metodo double Suma(listaLigada lista)

{

entero tamaño = lista.tamaño();

double suma = 0;

para (entero i = 1 hasta i <= tamaño con incremento 1) haga

{

suma = suma + lista [i];

}

retorne (suma) ;

}

|  |  |
| --- | --- |
| **Referencias de**  **Diseño** | Metodo Multiplicar Lista Ligada: |
|  | En este método se reciben 2 listas ligadas, las cuales se multiplican termino a término y |
|  | se regresa una lista ligada. |
|  |  |

|  |  |
| --- | --- |
| **Parámetros** | Lista Ligada X y Lista ligada Y |
|  |  |

Método listaLigada Multiplicar\_Listas( listaLigada lista1, listaLigada lista2)

{

listaLigada multiplicado = nueva listaLigada(lista1.tamaño());

Si (lista1.tamaño() == lista2.tamaño()) entonces

{

Para (entero i = 1 hasta i <= lista1.tamaño() con incremento 1) haga

{

multiplicado.agregar [i] = lista1 [i] \* lista2 [i];

}

retorne (nultiplicado);

}

Si No

{

Muestre: “Las listas NO son iguales”;

retorne (multiplicado);

}

}

|  |  |
| --- | --- |
| **Referencias de**  **Diseño** | Metodo Obtener valor B1: |
|  | En este método recibe el producto de 2 listas ligadas, la media de la lista ligada x, la |
|  | se regresa una lista ligada.media de la lista ligada y, el producto de la lista x con ella |
|  | misma y la cantidad de datos que se ingresaron al sistemas. |
|  | Con esto se realiza el cálculo del valor B1 con la fórmula dada en el documento. |

|  |  |
| --- | --- |
| **Parámetros** | Doble Producto XY, Doble media de X, Doble Media de Y, Doble Producto XX |
|  | y Entero Cantidad de Datos en el sistema. |
|  |  |

Método double Obtener\_B1(doble xy, doble media\_x, doble media\_y, doble xx, entero cant\_datos)

{

doble resultado\_B1 = 0;

resultado\_B1= xy - (cant\_datos \* (media\_x \* media\_y))

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ;

(xx - (cant\_datos \* (media\_x \* media\_x)))

retorne (resultado\_B1);

}

|  |  |
| --- | --- |
| **Referencias de**  **Diseño** | Método Obtener valor B1: |
|  | En este método recibe el producto de 2 listas ligadas, la media de la lista ligada x, la |
|  | se regresa una lista ligada.media de la lista ligada y, el producto de la lista x con ella |
|  | misma y la cantidad de datos que se ingresaron al sistemas. |
|  | Con esto se realiza el cálculo del valor B1 con la fórmula dada en el documento. |

|  |  |
| --- | --- |
| **Parámetros** | Doble Producto XY, Doble media de X, Doble Media de Y, Doble Producto XX |
|  | y Entero Cantidad de Datos en el sistema. |
|  |  |

Método double Obtener\_B1(doble xy, doble media\_x, doble media\_y, doble xx, entero cant\_datos)

{

doble resultado\_B1 = 0;

resultado\_B1= xy - (cant\_datos \* (media\_x \* media\_y))

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ;

(xx - (cant\_datos \* (media\_x \* media\_x)))

retorne (resultado\_B1);

}

|  |  |
| --- | --- |
| **Referencias de**  **Diseño** | Metodo Obtener valor B0: |
|  | En este método recibe la media de la lista ligada x, la media de la lista ligada y y el |
|  | valor calculado B1. |
|  | Con estos elementos se realiza el cálculo del valor B0 a través de la formula que se |
|  | encuentra en la documentación entregada |

|  |  |
| --- | --- |
| **Parámetros** | Doble media de X, Doble Media de Y y Doble B1 |
|  |  |

Método double Obtener\_B0(doble media\_x, doble media\_y, doble B1)

{

doble resultado\_B0 = 0;

resultado\_B1= media\_y - (B1 \* media\_x);

retorne (resultado\_B0);

}

|  |  |
| --- | --- |
| **Referencias de**  **Diseño** | Método Obtener valor R de lista X y Y: |
|  | En este método recibe la cantidad de datos, el producto de las listas XY, el valor de la |
|  | lista X, el valor de la lista Y, el producto de las listas XX y el producto de las listas YY. |
|  | Con estos valores se realiza el cálculo de valor de R teniendo como base la fórmula |
|  | que se encuentra en la documentación entregada. |

|  |  |
| --- | --- |
| **Parámetros** | Entero Cantidad de datos en el sistema, Doble XY , Doble X, Doble Y, Doble XX y |
|  | Doble YY. |
|  |  |

Método double Obtener\_R\_XY(entero cant\_datos, doble xy, doble x, doble y, doble xx, doble yy)

{

doble resultado\_R = 0;

resultado\_R = (cant\_datos \* xy) - (x \* y)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ;

Raiz([ (cont\_datos \* xy) - (x \* x) ] \* [ (cant\_datos \* yy) - (y \* y) ])

retorne (resultado\_R);

}

|  |  |
| --- | --- |
| **Referencias de**  **Diseño** | Metodo Mostrar Resultados: |
|  | En este método recibe el valor de B0, el valor de B1, el valor de R, el valor del producto |
|  | de RR y el valor de Y\_Calculado. Con ayuda de estos parametros se va a realizar la |
|  | impresion en pantalla de los valores que el sistema calculo y son la respuesta esperada |
|  | por el usuario. |

|  |  |
| --- | --- |
| **Parámetros** | Doble B0, Doble B1, Doble R, Doble RR y Doble Y\_Calculado |
|  |  |

Método Mostrar\_Resultados(doble B0, doble B1, doble R, doble RR, doble y\_calculado)

{

Muestre: “El B0 calculado: “ + B0;

Muestre: “El B1 calculado: “ + B1;

Muestre: “El R calculado: “ + R;

Muestre: “El RR calculado: “ + RR;

Muestre: “El valor de Y calculado: “ + y\_calculado;

}

|  |  |
| --- | --- |
| **Referencias de**  **Diseño** | Metodo Main: |
|  | En este método se realiza la interacción con el usuario para el ingreso de los datos |
|  | necesarios por el sistema para brindar el resultado esperado. En este método se realizan |
|  | los proceso, invocación de métodos y cálculos necesarios para verificar la entrada del |
|  | usuario y brindar la respuesta de cálculos esperados por el cliente. |

|  |  |
| --- | --- |
| **Parámetros** | Este método no tiene parámetros para su ejecución. |
|  |  |

Método Principal()

{

listaLigada lista\_x = nuevo listaLigada();

listaLigada lista\_y = nuevo listaLigada();

doble estimador = 0;

direccionArchivo1= Muestre: "Introduzca la ruta del primer conjunto de datos: ";

lista\_x = LeerArchivo(direccionArchivo1);

direccionArchivo2= Muestre: "Introdusca la ruta del segundo conjunto de datos:";

lista\_y = LeerArchivo(direccionArchivo2);

estimador: Muestre: "Por favor introduzca el valor estimador");

Si (estimador !=numero) entonces

{

Muestre: "Estimador no es un número. El programa se cerrará";

Salir()

}

Si (lista\_x.esVacia() == Verdadero o lista\_y.esVacia()==Verdadero) entonces

{

Muestre: "El archivo esta vacio.";

}

Si No

{

entero cant\_datos = lista\_x.cant\_datos();

doble media\_x = Media(lista\_x);

doble media\_y = Media(lista\_y);

listaLigada xy = Multiplicar\_Listas (datos\_x, datos\_y);

listaLigada xx = Multiplicar\_Listas (datos\_x, datos\_y);

listaLigada yy = Multiplicar\_Listas (datos\_y, datos\_y);

doble sum\_xy = Suma(xy);

doble sum\_xx = Suma(xx);

doble sum\_yy = Suma(yy);

doble sum\_x = Suma(datos\_x);

doble sum\_y = Suma(datos\_y);

doble B1 = Obtener\_B1(sum\_xy, media\_x, media\_y, sum\_xx, cant\_datos);

doble B0 = Obtener\_B0(media\_x, media\_y, B1);

doble R\_XY = Obtener\_R\_XY(cant\_datos, sum\_xy, sum\_x, sum\_y, sum\_xx, sum\_yy);

doble R2 = R\_XY \* R\_XY;

doble Y\_estimado = B0 + (B1\*estimador);

Mostrar\_Resultados(B0, B1, R\_XY, R2, Y\_estimado);

}