**Program Description**

Using PSP1, write a program to

• calculate the linear regression parameters and and correlation coefficients and for a set of *n* pairs of data,



• given an estimate, calculate an improved prediction, where



• enhance the linked list developed in program 1 to store the *n* data sets, where each record holds two real numbers

PSP Time Recording Log

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Deivid Alexander Osorio Barrera | Date | 07/02/2015 |
| Program | Cálculo de regresión lineal coeficiente de variación | Program # | 3 |
| Instructor | Luis Daniel Benavides Navarro | Language | java |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project** | **Phase** | **Start Date and Time** | **Int. Time** | **Stop Date and Time** | **Delta**  **Time** | **Comments** |
| PSP1 | Plan | 08/02/2015 10:50 am |  | 08/02/2015 11:00 am | 10 | Planeación de la solución al programa requerido. |
| PSP1 | Design | 08/02/2015 11:00 am |  | 08/02/2015 11:20 am | 20 | Diseño conceptual de alto nivel con la definición de las clases generales y plan general de ejecución del programa. |
| PSP1 | Code | 08/02/2015 11:20 am | 160 | 08/02/2015 08:20 pm | 380 | Codificación del programa, se tiene un tiempo de interrupción de 40 minutos. |
| PSP1 | Compile | 08/02/2015 |  | 08/02/2015 | 10 | Compilación y validación de errores en el build |
| PSP1 | Unit Test | 08/02/2015 |  | 08/02/2015 | 30 | Ejecución de pruebas unitarias y solución de errores identificados |
| PSP1 | Postmortem | 08/02/2015 |  | 08/02/2015 | 30 | Diligenciamiento de documentos y formatos PSP |
|  |  |  |  |  |  |  |

Time Recording Log Instructions

|  |  |
| --- | --- |
| Purpose | * Use this form to record the time you spend on each project activity. * For the PSP, phases often have only one activity; larger projects usually have multiple activities in a single process phase. * These data are used to complete the Project Plan Summary. * Keep separate logs for each program. |
| General | * Record all of the time you spend on the project. * Record the time in minutes. * Be as accurate as possible. * If you need additional space, use another copy of the form. * If you forget to record the starting, stopping, or interruption time for an activity, promptly enter your best estimate. |
| Header | * Enter your name and the date. * Enter the program name and number. * Enter the instructor’s name and the programming language you are using. |
| Project | Enter the program name or number. |
| Phase | Enter the name of the phase for the activity you worked on, e.g. Planning, Design, Test. |
| Start Date and Time | Enter the date and time when you start working on a process activity. |
| Interruption Time | * Record any interruption time that was not spent on the process activity. * If you have several interruptions, enter their total time. * You may enter the reason for the interrupt in comments. |
| Stop Date and Time | Enter the date and time when you stop working on that process activity. |
| Delta Time | Enter the clock time you actually spent working on the process activity, less the interruption time. |
| Comments | Enter any other pertinent comments that might later remind you of any unusual circumstances regarding this activity. |

PSP Defect Recording Log

|  |  |
| --- | --- |
| Defect Types |  |
| 10 Documentation | 60 Checking |
| 20 Syntax | 70 Data |
| 30 Build, Package | 80 Function |
| 40 Assignment | 90 System |
| 50 Interface | 100 Environment |

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Deivid Alexander Osorio Barrera | Date | 07/02/2015 |
| Program | Cálculo de regresión lineal coeficiente de variación | Program # | 3 |
| Instructor | Luis Daniel Benavides Navarro | Language | java |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Project |  | | Date |  | Number |  | Type |  | Inject |  | Remove |  | Fix Time |  | Fix Ref. |
| PSP1 |  | | 08/02/2015 |  | 001 |  | 80 |  | 0 |  | 1 |  | 5 |  |  |
| Description: | | | Error en el cálculo del valor B1 en la formula, hacía falta el cuadrado del valor x | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
| Project |  | | Date |  | Number |  | Type |  | Inject |  | Remove |  | Fix Time |  | Fix Ref. |
| PSP1 |  | | 08/02/2015 |  | 002 |  | 80 |  | 0 |  | 1 |  | 5 |  |  |
| Description: | | | Error al calcular el valor R(x,y) esto porque en el cálculo quedó pendiente el numerador | | | | | | | | | | | | | |
| de la fórmula | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
| Project |  | | Date |  | Number |  | Type |  | Inject |  | Remove |  | Fix Time |  | Fix Ref. |
| PSP1 |  | | 08/02/2015 |  | 003 |  | 80 |  | 0 |  | 1 |  | 10 |  |  |
| Description: | | | En el cálculo del valor R(x,y) tenía error porque estaba retornando información que ya | | | | | | | | | | | | | |
| Tenía un valor calculado previo, esto duplicaba el valor resultante. | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
| Project |  | | Date |  | Number |  | Type |  | Inject |  | Remove |  | Fix Time |  | Fix Ref. |
| PSP1 |  | | 08/02/2015 |  | 004 |  | 80 |  | 0 |  | 1 |  | 10 |  |  |
| Description: | | | En la ejecución del programa por consola se generó error porque no calculaba el valor | | | | | | | | | | | | | |
| De manera correcta, se validó el error y se identificó que las variables estaban retornando el valor con | | | | | | | | | | | | | | | | |
| Todos los decimales lo cual generaba error en los unit test | | | | | | | | | | | | | | | | |
| Project |  | | Date |  | Number |  | Type |  | Inject |  | Remove |  | Fix Time |  | Fix Ref. |
|  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Description: | | |  | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
| Project |  | | Date |  | Number |  | Type |  | Inject |  | Remove |  | Fix Time |  | Fix Ref. |
|  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Description: | | |  | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
| Project |  | | Date |  | Number |  | Type |  | Inject |  | Remove |  | Fix Time |  | Fix Ref. |
|  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Description: | | |  | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
| Project |  | | Date |  | Number |  | Type |  | Inject |  | Remove |  | Fix Time |  | Fix Ref. |
|  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Description: | | |  | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |

PSP Defect Recording Log Instructions

|  |  |
| --- | --- |
| Purpose | * Use this form to hold data on the defects that you find and correct. * These data are used to complete the Project Plan Summary form. |
| General | * Record each defect separately and completely. * If you need additional space, use another copy of the form. |
| Header | * Enter your name and the date. * Enter the program name and number. * Enter the instructor’s name and the programming language you are using. |
| Project | * Give each program a different name or number. * For example, record test program defects against the test program. |
| Date | Enter the date on which you found the defect. |
| Number | * Enter the defect number. * For each program or module, use a sequential number starting with 1 (or 001, etc.). |
| Type | * Enter the defect type from the defect type list summarized in the top left corner of the form. * Use your best judgment in selecting which type applies. |
| Inject | * Enter the phase when this defect was injected. * Use your best judgment. |
| Remove | Enter the phase during which you fixed the defect. (This will generally be the phase when you found the defect.) |
| Fix Time | * Enter the time that you took to find and fix the defect. * This time can be determined by stopwatch or by judgment. |
| Fix Ref. | * If you or someone else injected this defect while fixing another defect, record the number of the improperly fixed defect. * If you cannot identify the defect number, enter an X. |
| Description | Write a succinct description of the defect that is clear enough to later remind you about the error and help you to remember why you made it. |

PSP Process Improvement Proposal (PIP)

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Deivid Alexander Osorio Barrera | Date | 07/02/2015 |
| Program | Cálculo de regresión lineal coeficiente de variación | Program # | 3 |
| Instructor | Luis Daniel Benavides Navarro | Language | java |

|  |
| --- |
| Problem Description |
| Briefly describe the problems that you encountered. |
|  |
| En la generación del programa se encontró error al momento de calcular los tiempos de desarrollo y los |
| errores solucionados, esto porque se dificulta el registro de la información a medida que se desarrolla y se |
| va compilando el programa. |
|  |
|  |
|  |
| Proposal Description |
| Briefly describe the process improvements that you propose. |
|  |
| El tiempo de solución de los errores se debería enmarcar en la fase de codificación para así definer que el |
| tiempo de code incluye una solución complete y los tiempos de solución de error se enmarcan en la fase |
| de testing. |
|  |
|  |
|  |
|  |
|  |
| Other Notes and Comments |
| Note any other comments or observations that describe your experiences or improvement ideas. |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

PSP Process Improvement Proposal (PIP) Instructions

|  |  |
| --- | --- |
| Purpose | * To provide a way to record process problems and improvement ideas * To provide an orderly record of your process improvement ideas * To record any other noteworthy observations |
| General | Use the PIP form to   * record process improvement ideas as they occur to you * establish priorities for your improvement plans * describe lessons learned and unusual conditions   Keep PIP forms on hand while using the PSP.   * Record process problems even without proposed solutions. * Submit a PIP with each PSP assignment report. |
| Header | * Enter your name and the date. * Enter the program name and number. * Enter the instructor’s name and the programming language you are using. |
| Problem Description | Briefly describe any problems or experiences that led to this PIP. |
| Proposal Description | Describe the proposed improvement as explicitly as possible. |
| Other Notes and Comments | Briefly describe any other observations or facts that would later help you to   * remember what you did while writing this program * remember an idea for a future improvement * explain to your instructor something you did and why you did it |

|  |  |
| --- | --- |
| Estándar de codificación | |
| Purpose | Guía la implementación o desarrollo de un programa en JAVA |
| Program Headers | Toda clase debe tener una cabecera que identifique su function y fecha de desarrollo. |
| Header Format | /\*\*  \* @Number Program: Número de programa  \* @author: Nombre de quien desarrolla la clase  \* @version: Fecha en la que se desarrolla la clase  \* @Description: Breve descripción de la clase  \*/ |
| Listing Contents | Lista de métodos de la clase |
| Contents Example | /\*\*  /\* Listing Methods  \* Method1 (Constructor)  \* Method2  \* …  \*/ |
| Identifiers | Use nombres nemotécnicos que sean descriptivos con la funcionalidad u objetivos de cada variable, método, clase, constante y demás elementos que utilice en el desarrollo del programa.  La estructura de estos nombres debe iniciar en minúscula y si el nombre es compuesto la siguiente palabra inicia en Mayúscula. |
| Identifier Example | metodoCalculador  calculaFormulas |
| Comments | Los comentarios se dan a nivel de clase y métodos, en caso de documentar una parte específica del código se realizará antes de la línea, no al finalizar la línea y se explicará su funcionalidad |
| Good Comment | /\* Condicional para validación de límite \*/  If(record\_count > limit) |
| Bad Comment | If(record\_count > limit) /\* Condicional para validación de límite \*/ |
| Major Sections | En caso de requerirse una descripción de una sección de código se comentará indicando el inicio y fin de la sección y al iniciar su respectiva descripción. |
| Example | /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Section 1: Sección del programa que realiza calculos aritméticos  \*/  Métodos…  /\*\*  \* Fin de section 1  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/ |
| Blank Spaces | - Separe la definición de métodos, definición de variables, definición de rutinas con al menos un espacio.  - Separe cada programa de construcción o método con al menos un espacio. |

|  |  |
| --- | --- |
|  | |
| Indenting | * Idente cada línea de código según su nivel * En caso de requerir abrir un corchete siempre abralo al final de la instrucción y al cerrar cierrelo en una sola línea * Si tiene instrucciones condicionales de una única línea no require corchetes a menos que la instrucción esté dentro de un case |
| Indenting Example | while (miss\_distance > threshold){  success\_code = move\_robot (target \_location);  if (success\_code == MOVE\_FAILED) {  printf(“The robot move has failed.\n”);  }  }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  if(miss\_distance > threshold)  printf(“The robot move has failed.\n”); |
| Capitalization | * Todas las variables o métodos se definen iniciando en minúscula y posterior a la primera palabra con letra capital * No define variables en mayuscula sostenida * Sólo para constants utilice mayusculas sostenida o para numeraciones dentro de una clase |
| Capitalization Examples | Int miVarible = 0;  Public String metodoCalculadora{}  #define DEFAULT-NUMBER-OF-STUDENTS 15  int class-size = DEFAULT-NUMBER-OF-STUDENTS; |
| Methods Declaration | - Toda definición de método debe tener su modificador definido, estos pueden ser:  Public  Private  Protected  - Los nombres de las clases se definen en letra capial |
| Methods Declaration Examples | Public Class MiPrimeraClase{} |

Test Report Template

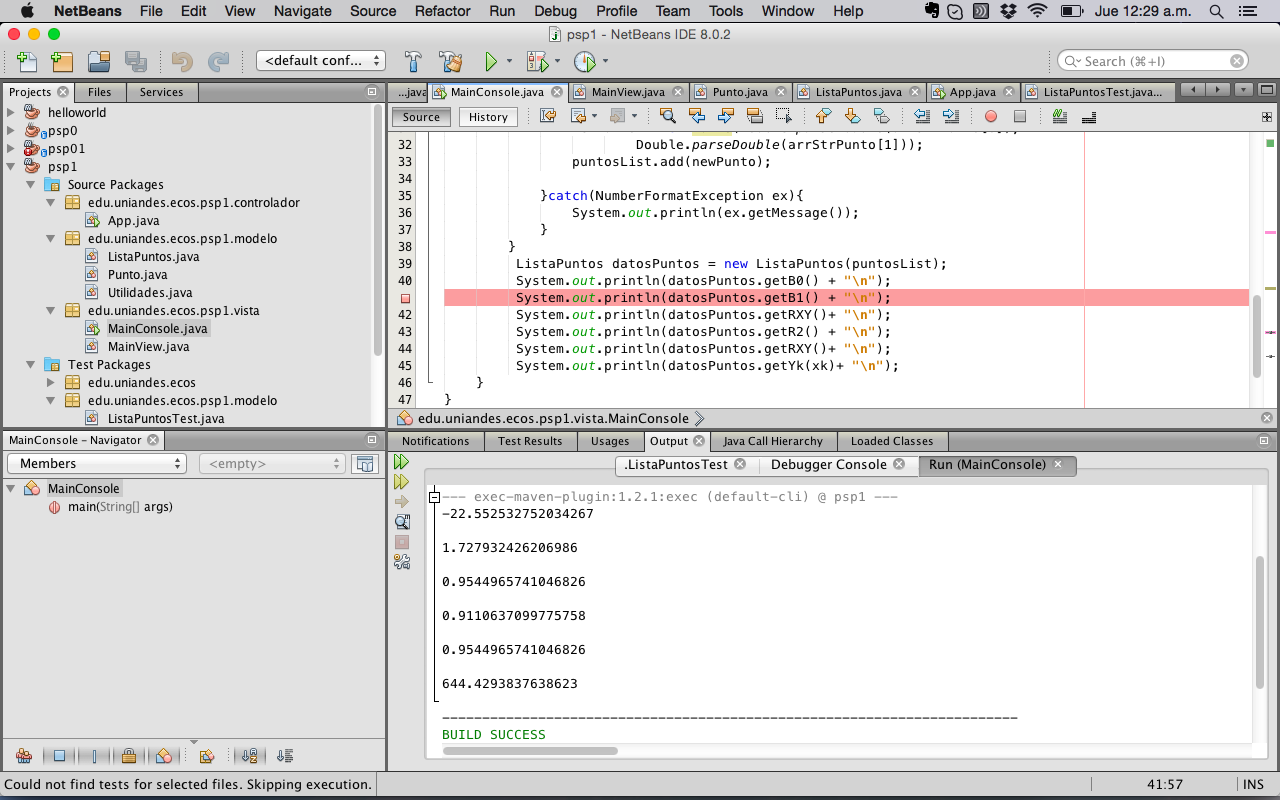
|  |  |  |  |
| --- | --- | --- | --- |
| Student | Deivid Alexander Osorio Barrera | Date | 07/02/2015 |
| Program | Cálculo de regresión lineal coeficiente de variación | Program # | 3 |
| Instructor | Luis Daniel Benavides Navarro | Language | java |

|  |  |
| --- | --- |
| Test Name/Number | Test number 1 |
| Test Objective | Obtener el valor de R(xy), R2, B0 y B1 con los siguientes valores |
|  | "130,186-650,699-99,132-150,272-128,291-302,331-95,199-945,1890-368,788-961,1601" |
| Test Description | Los numeros son parejas de valores X y Y, estos son ingresados separados |
|  | por guión y la pareja x y y separada por coma |
|  |  |
|  |  |
|  |  |
| Test Conditions |  |
|  |  |
|  |  |
|  |  |
|  |  |
| Expected Results | B0 = -22.55 |
|  | B1 = 1.7279 |
|  | R(xy) = 0.9545 |
|  | R2 = 0.9111 |
|  | Yk = 644.429 |
|  |  |
| Actual Results | B0 = -22.5525 |
|  | B1 = 1.7279 |
|  | R(xy) = 0.954496 |
|  | R2 = 0.911063 |
|  | Yk = 644.429 |
| Test Name/Number | Test number 2 |
| Test Objective | Obtener el valor de R(xy), R2, B0 y B1 con los siguientes valores |
|  | "130,15-650,69.9-99,6.5-150,22.4-128,28.4-302,65.9-95,19.4-945,198.7-368,38.8-961,138.2" |
| Test Description | Los numeros son parejas de valores X y Y, estos son ingresados separados |
|  | por guión y la pareja x y y separada por coma |
|  |  |
|  |  |
|  |  |
| Test Conditions |  |
|  |  |
|  |  |
|  |  |
|  |  |
| Expected Results | B0 = -4.039 |
|  | B1 = 0.1681 |
|  | R(xy) = 0.9333 |
|  | R2 = 0.8711 |
|  | Yk = 60.858 |
|  |  |
| Actual Results | B0 = -4.0388 |
|  | B1 = 0.1681266 |
|  | R(xy) = 0.9333068 |
|  | R2 = 0.871061 |
|  | Yk = 60.858 |
|  |  |
| Test Name/Number | Test number 3 |
| Test Objective | Obtener el valor de R(xy), R2, B0 y B1 con los siguientes valores |
|  | "163,186-765,699-141,132-166,272-137,291-355,331-136,199-1206,1890-433,788-1130,1601" |
| Test Description | Los numeros son parejas de valores X y Y, estos son ingresados separados |
|  | por guión y la pareja x y y separada por coma |
|  |  |
|  |  |
|  |  |
| Test Conditions |  |
|  |  |
|  |  |
|  |  |
|  |  |
| Expected Results | B0 = -23.92 |
|  | B1 = 1.43097 |
|  | R(xy) = 0.9631 |
|  | R2 = 0.9276 |
|  | Yk = 528.4294 |
| Actual Results | B0 = -23.923888 |
|  | B1 = 1.43096694 |
|  | R(xy) = 0.9631140 |
|  | R2 = 0.927588 |
|  | Yk = 528.4294 |
|  |  |
| Test Name/Number | Test number 4 |
| Test Objective | Obtener el valor de R(xy), R2, B0 y B1 con los siguientes valores |
|  | "163,15-765,69.9-141,6.5-166,22.4-137,28.4-355,65.9-136,19.4-1206,198.7-433,38.8-1130,138.2" |
| Test Description | Los numeros son parejas de valores X y Y, estos son ingresados separados |
|  | por guión y la pareja x y y separada por coma |
|  |  |
|  |  |
|  |  |
| Test Conditions |  |
|  |  |
|  |  |
|  |  |
|  |  |
| Expected Results | B0 = -4.604 |
|  | B1 = 0.140164 |
|  | R(xy) = 0.948 |
|  | R2 = 0.8988 |
|  | Yk = 49.4994 |
|  |  |
| Actual Results | B0 = -4.60374542 |
|  | B1 = 0.140163526 |
|  | R(xy) = 0.948032 |
|  | R2 = 0.898766 |
|  | Yk = 49.4994 |

Test Report Template Instructions

|  |  |
| --- | --- |
| Purpose | * To maintain a record of the tests run and the results obtained * To be sufficiently complete so that you can later re-run the same tests and get the same results * To facilitate regression testing of modified or reused programs |
| General | * Expand this table or use multiple copies as needed. * Report all the tests that were successfully run. * Be as brief and concise as possible. |
| Header | * Enter your name and the date. * Enter the program name and number. * Enter the instructor’s name and the programming language you are using. |
| Test Name/Number | Uniquely identify each test for each program.   * the same tests with different data * the same data with different tests |
| Test Objective | Briefly describe the objective of the test. |
| Test Description | Describe each test’s data and procedures in sufficient detail to facilitate its later use as a regression test. |
| Test Conditions | * List any special configuration, timing, fix, or other conditions of the test. * When multiple tests are run with different parameters or under varying conditions, separately list each. |
| Expected Results | List the results that the test should produce if it runs properly. |
| Actual Results | List the results that were actually produced. |

Pantallas de test:



Size Estimating Template

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Deivid Alexander Osorio Barrera | Date | 07/02/2015 |
| Program | Cálculo de regresión lineal coeficiente de variación | Program # | 3 |
| Instructor | Luis Daniel Benavides Navarro | Language | java |
| Size Measure |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Estimated | | | | | | | | | | | |
| Base Parts |  | Base | |  | Deleted | |  | Modified | | |  | Added | |
|  |  |  | |  |  | |  |  | | |  |  | |
| App.java |  | 65 | |  | 0 | |  | 0 | | |  | 15 | |
|  |  |  | |  |  | |  |  | | |  |  | |
|  |  |  | |  |  | |  |  | | |  |  | |
| Total | B | | 65 | D | | 0 | M | | 0 | **BA** | | | 15 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Actual | | | | | | | | | | |
| Base Parts |  | Base | |  | Deleted | |  | Modified | |  | Added | |
|  |  |  | |  |  | |  |  | |  |  | |
| App.java |  | 65 | |  | 0 | |  | 0 | |  | 22 | |
|  |  |  | |  |  | |  |  | |  |  | |
|  |  |  | |  |  | |  |  | |  |  | |
| Total |  | | 65 |  | | 0 |  | | 0 |  | | 22 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Estimated | | | | | | | |  | Actual | | |
| Parts Additions |  | Type |  | Items |  | Rel. Size | |  | Size\* |  | Size\* |  | Items |
| Modelo punto |  | Logic |  | 3 |  | S | |  | 20 |  | 15 |  | 3 |
| Modelo de listaPuntos |  | Math |  | 5 |  | L | |  | 50 |  | 162 |  | 12 |
| Vista consola MainConsole |  | ViewC |  | 1 |  | M | |  | 20 |  | 32 |  | 1 |
| Vista web MainView |  | ViewE |  | 2 |  | M | |  | 30 |  | 51 |  | 3 |
| Vista web Captura datos |  | ViewE |  | 2 |  | M | |  | 30 |  | 0 |  | 0 |
|  |  |  |  |  |  |  | |  |  |  |  |  |  |
|  |  |  |  |  |  |  | |  |  |  |  |  |  |
|  |  |  |  |  |  |  | |  |  |  |  |  |  |
|  |  |  |  |  |  |  | |  |  |  |  |  |  |
| Total |  |  |  |  |  |  | PA | | 150 |  | 260 |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  | Estimated |  | Actual |
| Reused Parts | |  | Size |  | Size |
|  | |  |  |  |  |
| Clase de càlculo de promedio de un LinkedList | |  | 10 |  | 10 |
|  | |  |  |  |  |
|  | |  |  |  |  |
|  | |  |  |  |  |
|  | |  |  |  |  |
|  | |  |  |  |  |
| Total | R | | 10 |  | 10 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PROBE Calculation Worksheet (Added and Modified) |  | Size |  | Time |
| Added size (A): A = BA+PA |  | 165 |  |  |
| Estimated Proxy Size (E): E = BA+PA+M |  | 165 |  |  |
| PROBE estimating basis used: (A, B, C, or D) |  | C |  | C |
| Correlation: (R2) |  |  |  |  |
| Regression Parameters: β0 Size and Time |  | 0 |  | 0 |
| Regression Parameters: β1 Size and Time |  | 1,33 |  | 0.39 |
| Projected Added and Modified Size (P): P = β0size + β1size\*E |  | 219.45 |  |  |
| Estimated Total Size (T): T = P + B - D - M + R |  | 294.45 |  |  |
| Estimated Total New Reusable (NR): sum of \* items |  | 10 |  |  |
| Estimated Total Development Time: Time = β0time + β1time\*E |  |  |  | 64.35 |
| Prediction Range: Range |  |  |  |  |
| Upper Prediction Interval: UPI = P + Range |  | 373.065 |  | 412.23 |
| Lower Prediction Interval: LPI = P - Range |  | 153.615 |  | 206.115 |
| Prediction Interval Percent: |  | 70% |  | 70% |

Size Estimating Template Instructions

|  |  |
| --- | --- |
| Purpose | Use this form with the PROBE method to make size estimates. |
| General | * A part could be a module, component, product, or system. * Where parts have a substructure of methods, procedures, functions, or similar elements, these lowest-level elements are called items. * Size values are assumed to be in the unit specified in size measure. * Avoid confusing base size with reuse size. * Reuse parts must be used without modification. * Use base size if additions, modifications, or deletions are planned. * If a part is estimated but not produced, enter its actual values as zero. * If a part is produced that was not estimated, enter it using zero for its planned values. |
| Header | * Enter your name and the date. * Enter the program name and number. * Enter the instructor’s name and the programming language you are using. * Enter the size measure you are using. |
| Base Parts | If this is a modification or enhancement of an existing product   * measure and enter the base size (more than one product may be entered as base) * estimate and enter the size of the deleted, modified, and added size to the base program   After development, measure and enter the actual size of the base program and any deletions, modifications, or additions. |
| Parts Additions | If you plan to add newly developed parts   * enter the part name, type, number of items (or methods), and relative size * for each part, get the size per item from the appropriate relative size table, multiply this value by the number of items, and enter in estimated size * put an asterisk next to the estimated size of any new-reusable additions   After development, measure and enter   * the actual size of each new part or new part items * the number of items for each new part |
| Reused Parts | If you plan to include reused parts, enter the   * name of each unmodified reused part * size of each unmodified reused part   After development, enter the actual size of each unmodified reused part. |

**PSP1 Project Plan Summary**

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Deivid Alexander Osorio Barrera | Date | 07/02/2015 |
| Program | Cálculo de regresión lineal coeficiente de variación | Program # | 3 |
| Instructor | Luis Daniel Benavides Navarro | Language | java |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Summary*** | ***Plan*** | | |  | ***Actual*** | | |  | ***To Date*** | | |
| ***Size/Hour*** |  | | |  |  | | |  |  | | |
|  |  | | |  |  | | |  |  | | |
| **Program Size** | ***Plan*** | | |  | **Actual** | | |  | **To Date** | | |
| Base (B) | 65 | | |  | 65 | | |  |  | | |
|  | ***(Measured)*** | | |  | (Measured) | | |  |  | | |
| Deleted (D) | ***0*** | | |  | 0 | | |  |  | | |
|  | ***(Estimated)*** | | |  | (Counted) | | |  |  | | |
| Modified (M) | ***0*** | | |  | 0 | | |  |  | | |
|  | ***(Estimated)*** | | |  | (Counted) | | |  |  | | |
| Added (A) | ***15*** | | |  | 22 | | |  |  | | |
|  | ***(A+M − M)*** | | |  | (T − B + D − R) | | |  |  | | |
| Reused (R) | ***10*** | | |  | 10 | | |  | 10 | | |
|  | ***(Estimated)*** | | |  | (Counted) | | |  |  | | |
| Added and Modified (A+M) | ***15*** | | |  | 22 | | |  | 22 | | |
|  | ***(Projected)*** | | |  | (A + M) | | |  |  | | |
| Total Size (T) | ***90*** | | |  | 97 | | |  | 97 | | |
|  | ***(A+M + B − M − D + R)*** | | |  | (Measured) | | |  |  | | |
| Total New Reusable |  | | |  |  | | |  |  | | |
|  |  | | |  |  | | |  |  | | |
| ***Estimated Proxy Size (E)*** | 165 | | |  |  | | |  |  | | |
|  |  | | |  |  | | |  |  | | |
| **Time in Phase (min.)** | **Plan** |  | **Actual** | | |  | **To Date** | | |  | **To Date %** |
| Planning | 10 |  | 10 | | |  | 67 | | |  | 5.27% |
| Design | 10 |  | 20 | | |  | 40 | | |  | 3.14% |
| Code | 240 |  | 380 | | |  | 746 | | |  | 58.74% |
| Compile | 0 |  | 10 | | |  | 127 | | |  | 10% |
| Test | 20 |  | 30 | | |  | 115 | | |  | 9.05% |
| Postmortem | 40 |  | 30 | | |  | 175 | | |  | 13.77% |
| Total | 320 |  | 480 | | |  | 1270 | | |  | 100% |
|  |  |  |  | | |  |  | | |  |  |
| **Defects Injected** |  |  | **Actual** | | |  | **To Date** | | |  | **To Date %** |
| Planning |  |  | 0 | | |  | 0 | | |  | 0% |
| Design |  |  | 0 | | |  | 0 | | |  | 0% |
| Code |  |  | 0 | | |  | 0 | | |  | 0% |
| Compile |  |  | 0 | | |  | 11 | | |  | 84.61% |
| Test |  |  | 0 | | |  | 2 | | |  | 15.38% |
| Total Development |  |  | 4 | | |  | 13 | | |  | 100% |
|  |  |  |  | | |  |  | | |  |  |
| **Defects Removed** |  |  | **Actual** | | |  | **To Date** | | |  | **To Date %** |
| Planning |  |  | 0 | | |  | 0 | | |  | 0 |
| Design |  |  | 0 | | |  | 0 | | |  | 0 |
| Code |  |  | 4 | | |  | 4 | | |  | 100% |
| Compile |  |  | 0 | | |  | 0 | | |  | 0 |
| Test |  |  | 0 | | |  | 0 | | |  | 0 |
| Total Development |  |  | 4 | | |  | 4 | | |  | 100% |
| After Development |  |  | 0 | | |  | 0 | | |  |  |

|  |  |  |
| --- | --- | --- |
| PSP1 Plan Summary Instructions | |  |
| Purpose | To hold the plan and actual data for programs or program parts | |
| General | * Use the most appropriate size measure, either LOC or element count. * “To Date” is the total actual to-date values for all products developed. * ***A*** ***part could be a module, component, product, or system.*** | |
| Header | * Enter your name and the date. * Enter the program name and number. * Enter the instructor’s name and the programming language you are using. | |
| *Summary* | * ***Enter the added and modified size per hour planned, actual, and to-date.*** | |
| Program Size | * ***Enter plan base, deleted, modified, reused, new reusable, and total size from the Size Estimating template.*** * Enter the plan added and modified size value (A+M) ***from projected added and modified size (P) on the Size Estimating template.*** * ***from the Size Estimating template***. * ***Calculate plan added size as A+M – M.*** * ***Enter estimated proxy size (E) from the Size Estimating template***. * Enter actual base, deleted, modified, reused, total, and new reusable size Calculate actual added size as T-B+D-R and actual added and modified size as A+M. * Enter to-date reused, added and modified, total, and new reusable size. | |
| Time in Phase | * Enter ***plan total time in phase*** ***from*** the estimated total development time ***on the Size Estimating template***. * Distribute the estimated total time across the development phases according to the To Date % for the most recently developed program. * Enter the actual time by phase and the total time. * To Date: Enter the sum of the actual times for this program plus the to-date times from the most recently developed program. * To Date %: Enter the percentage of to-date time in each phase. | |
| Defects Injected | * Enter the actual defects by phase and the total actual defects. * To Date: Enter the sum of the actual defects injected by phase and the to-date values for the most recent previously developed program. * To Date %: Enter the percentage of the to-date defects injected by phase. | |
| Defects Removed | * To Date: Enter the actual defects removed by phase plus the to-date values for the most recent previously developed program. * To Date %: Enter the percentage of the to-date defects removed by phase. * After development, record any defects subsequently found during program testing, use, reuse, or modification. | |