**PSP1 Project Plan Summary**

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Wilman Rincon | Date | 09/02/2015 |
| Program | 3 | Program # | 3 |
| Instructor | Luis Daniel Benavides | Language | Java |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Summary*** | ***Plan*** | | |  | ***Actual*** | | |  | ***To Date*** | | |
| ***Size/Hour*** | 195 | | |  | 152 | | |  | 553 | | |
|  |  | | |  |  | | |  |  | | |
| **Program Size** | ***Plan*** | | |  | **Actual** | | |  | **To Date** | | |
| Base (B) | 0 | | |  |  | | |  |  | | |
|  | ***(Measured)*** | | |  | (Measured) | | |  |  | | |
| Deleted (D) | ***0*** | | |  |  | | |  |  | | |
|  | ***(Estimated)*** | | |  | (Counted) | | |  |  | | |
| Modified (M) | ***0*** | | |  |  | | |  |  | | |
|  | ***(Estimated)*** | | |  | (Counted) | | |  |  | | |
| Added (A) | ***86*** | | |  | 86 | | |  |  | | |
|  | ***(A+M − M)*** | | |  | (T − B + D − R) | | |  |  | | |
| Reused (R) | ***20*** | | |  | 20 | | |  | 20 | | |
|  | ***(Estimated)*** | | |  | (Counted) | | |  |  | | |
| Added and Modified (A+M) | ***86*** | | |  | 106 | | |  | 196 | | |
|  | ***(Projected)*** | | |  | (A + M) | | |  |  | | |
| Total Size (T) | ***106*** | | |  | 106 | | |  | 106 | | |
|  | ***(A+M + B − M − D + R)*** | | |  | (Measured) | | |  |  | | |
| Total New Reusable | 106 | | |  | 106 | | |  | 106 | | |
|  |  | | |  |  | | |  |  | | |
| ***Estimated Proxy Size (E)*** |  | | |  |  | | |  |  | | |
|  |  | | |  |  | | |  |  | | |
| **Time in Phase (min.)** | **Plan** |  | **Actual** | | |  | **To Date** | | |  | **To Date %** |
| Planning | 10 |  | 14 | | |  | 47 | | |  | 9 |
| Design | 30 |  | 20 | | |  | 70 | | |  | 12 |
| Code | 100 |  | 50 | | |  | 290 | | |  | 52 |
| Compile | 0 |  | 0 | | |  | 0 | | |  | 0 |
| Test | 25 |  | 18 | | |  | 56 | | |  | 10 |
| Postmortem | 30 |  | 50 | | |  | 90 | | |  | 16 |
| Total | 195 |  | 152 | | |  | 553 | | |  | 100 |
|  |  |  |  | | |  |  | | |  |  |
| **Defects Injected** |  |  | **Actual** | | |  | **To Date** | | |  | **To Date %** |
| Planning |  |  | 0 | | |  | 0 | | |  | 0 |
| Design |  |  | 0 | | |  | 0 | | |  | 0 |
| Code |  |  | 2 | | |  | 7 | | |  | 100 |
| Compile |  |  | 0 | | |  | 0 | | |  | 0 |
| Test |  |  | 0 | | |  | 0 | | |  | 0 |
| Total Development |  |  | 2 | | |  | 7 | | |  | 100 |
|  |  |  |  | | |  |  | | |  |  |
| **Defects Removed** |  |  | **Actual** | | |  | **To Date** | | |  | **To Date %** |
| Planning |  |  | 0 | | |  | 0 | | |  | 0 |
| Design |  |  | 0 | | |  | 0 | | |  | 0 |
| Code |  |  | 0 | | |  | 0 | | |  | 0 |
| Compile |  |  | 0 | | |  | 0 | | |  | 0 |
| Test |  |  | 2 | | |  | 7 | | |  | 100 |
| Total Development |  |  | 2 | | |  | 7 | | |  | 100 |
| After Development |  |  |  | | |  |  | | |  |  |

PSP Time Recording Log

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Wilman Rincon | Date | 09/02/2015 |
| Program | 3 | Program # | 3 |
| Instructor | Luis Daniel Benavides | Language | java |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project** | **Phase** | **Start Date and Time** | **Int. Time** | **Stop Date and Time** | **Delta**  **Time** | **Comments** |
| 3 | Planing | 18:00 | 18:14 | 0 | 14 |  |
| 3 | Design | 18:30 | 18:50 | 0 | 20 |  |
| 3 | Code | 20:00 | 20:50 | 0 | 50 |  |
| 3 | Test | 21:00 | 21:15 | 3 | 18 |  |
| 3 | PostMortem | 21:20 | 22:00 | 10 | 50 |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

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 | Wilman Rincon | Date | 09/02/2015 |
| Program | 3 | Program # | 3 |
| Instructor | Luis Daniel Benavides | Language | Java |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Project |  | | Date |  | Number |  | Type |  | Inject |  | Remove |  | Fix Time |  | Fix Ref. |
| 3 |  | | 9/2 |  | 1 |  | 80 |  | Code |  | Test |  | 1 |  |  |
| Description: | | | En el metodo ConsoleInput de la clase App, estaba calculando primero el B0 y después el | | | | | | | | | | | | | |
| B1, proceso que está mal debido que para calcular el B0 antes toca calcular el B1. | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
| Project |  | | Date |  | Number |  | Type |  | Inject |  | Remove |  | Fix Time |  | Fix Ref. |
| 3 |  | | 9/2 |  | 2 |  | 80 |  | Code |  | Test |  | 2 |  |  |
| Description: | | | En el metodo CalcularRxy de la clase CalcularDatos, tenía mal la fórmula para calcular | | | | | | | | | | | | | |
| Ese valor, ya que no estaba multiplicando por la cantidad de datos. | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
| 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: | | |  | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
| Project |  | | Date |  | Number |  | Type |  | Inject |  | Remove |  | Fix Time |  | Fix Ref. |
|  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Description: | | |  | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
| Project |  | | Date |  | Number |  | Type |  | Inject |  | Remove |  | Fix Time |  | Fix Ref. |
|  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Description: | | |  | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |

Test Report Template

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Wilman Rincon | Date | 09/02 |
| Program | 3 | Program # | 3 |
| Instructor | Luis Daniel Benavides | Language | Java |

|  |  |
| --- | --- |
| Test Name/Number | Test 1/1 |
| Test Objective | Ingresar un conjunto de datos para X y Y, y esperar unos resultados esperados |
|  | De acuerdo a las tablas dadas para el ejercicio. |
| Test Description | Encontrar los valores de B0, B1, Rxy, Yk |
|  |  |
|  |  |
|  |  |
|  |  |
| Test Conditions | Valores de **X:** [130.0, 650.0, 99.0, 150.0, 128.0, 302.0, 95.0, 945.0, 368.0, 961.0] |
|  |  |
|  | Valores de **Y:** [186.0, 699.0, 132.0, 272.0, 291.0, 331.0, 199.0, 1890.0, 788.0, 1601.0] |
|  |  |
|  |  |
| Expected Results | B0 = -22.55 B1=1.7279 Rxy =0.9545 R2=0.9111 Yk=644.429 |
|  |  |
|  |  |
|  |  |
| Actual Results | B0 = -22.5525 B1= 1.727 Rxy = 0.9544 R2= 0.9110  Yk= 644.42 |
|  |  |
|  |  |
|  |  |
|  |  |
| Test Name/Number | Test 2 / 2 |
| Test Objective | Ingresar un conjunto de datos para X y Y, y esperar unos resultados esperados |
|  | De acuerdo a las tablas dadas para el ejercicio. |
| Test Description | Encontrar los valores de B0, B1, Rxy, Yk |
|  |  |
|  |  |
|  |  |
|  |  |
| Test Conditions | Valores **X:** [130.0, 650.0, 99.0, 150.0, 128.0, 302.0, 95.0, 945.0, 368.0, 961.0] |
|  |  |
|  | Valores **Y:** [15.0, 69.9, 6.5, 22.4, 28.4, 65.9, 19.4, 198.7, 38.8, 138.2] |
|  |  |
|  |  |
| Expected Results | B0 = -4.039 B1=0.1681 Rxy =0.9333 R2=.8711 Yk=60.858 |
|  |  |
|  |  |
|  |  |
| Actual Results | B0 = -4.038B1= 0.1681Rxy = 0.933 R2= 0.8710  Yk= 60.85 |
|  |  |
|  |  |
|  |  |
|  |  |

Size Estimating Template

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Wilman Rincon | Date | 09/02 |
| Program | 3 | Program # | 3 |
| Instructor | Luis Daniel Benavides | Language | Java |
| Size Measure |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Estimated | | | | | | | | | | | |
| Base Parts |  | Base | |  | Deleted | |  | Modified | | |  | Added | |
|  |  |  | |  |  | |  |  | | |  |  | |
|  |  |  | |  |  | |  |  | | |  |  | |
|  |  |  | |  |  | |  |  | | |  |  | |
|  |  |  | |  |  | |  |  | | |  |  | |
| Total | B | |  | D | |  | M | |  | **BA** | | |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Actual | | | | | | | | | | |
| Base Parts |  | Base | |  | Deleted | |  | Modified | |  | Added | |
|  |  |  | |  |  | |  |  | |  |  | |
|  |  |  | |  |  | |  |  | |  |  | |
|  |  |  | |  |  | |  |  | |  |  | |
|  |  |  | |  |  | |  |  | |  |  | |
| Total |  | |  |  | |  |  | |  |  | |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Estimated | | | | | | | |  | Actual | | |
| Parts Additions |  | Type |  | Items |  | Rel. Size | |  | Size\* |  | Size\* |  | Items |
| Calcular |  | Calc |  | 4 |  | Medio | |  | 40 |  | 50 |  | 5 |
| CalcularDatos |  | Calc |  | 5 |  | Medio | |  | 50 |  | 60 |  | 6 |
| App |  | Logic |  | 2 |  | Small | |  | 10 |  | 20 |  | 4 |
| MainView |  | Logic |  | 2 |  | Small | |  | 10 |  | 15 |  | 3 |
|  |  |  |  |  |  |  | |  |  |  |  |  |  |
|  |  |  |  |  |  |  | |  |  |  |  |  |  |
|  |  |  |  |  |  |  | |  |  |  |  |  |  |
|  |  |  |  |  |  |  | |  |  |  |  |  |  |
|  |  |  |  |  |  |  | |  |  |  |  |  |  |
|  |  |  |  |  |  |  | |  |  |  |  |  |  |
| Total |  |  |  |  |  |  | PA | | 110 |  | 145 |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  | Estimated |  | Actual |
| Reused Parts | |  | Size |  | Size |
|  | |  |  |  |  |
|  | |  |  |  |  |
|  | |  |  |  |  |
|  | |  |  |  |  |
|  | |  |  |  |  |
|  | |  |  |  |  |
|  | |  |  |  |  |
| Total | R | |  |  |  |

Size Estimating Template (continued)

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Wilman Rincon | Program | 3 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PROBE Calculation Worksheet (Added and Modified) |  | Size |  | Time |
| Added size (A): A = BA+PA |  |  |  |  |
| Estimated Proxy Size (E): E = BA+PA+M |  | 29 |  |  |
| PROBE estimating basis used: (A, B, C, or D) |  | C |  |  |
| Correlation: (R2) |  |  |  |  |
| Regression Parameters: β0 Size and Time |  | 30.12 |  |  |
| Regression Parameters: β1 Size and Time |  | 0.94 |  |  |
| Projected Added and Modified Size (P): P = β0size + β1size\*E |  | 57.38 |  |  |
| Estimated Total Size (T): T = P + B - D - M + R |  | 547.38 |  |  |
| Estimated Total New Reusable (NR): sum of \* items |  |  |  |  |
| Estimated Total Development Time: Time = β0time + β1time\*E |  |  |  |  |
| Prediction Range: Range |  |  |  |  |
| Upper Prediction Interval: UPI = P + Range |  |  |  |  |
| Lower Prediction Interval: LPI = P - Range |  |  |  |  |
| Prediction Interval Percent: |  |  |  |  |

PSP Process Improvement Proposal (PIP)

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Wilman Rincon | Date | 09/02/2015 |
| Program | 3 | Program # | 3 |
| Instructor | Luis Daniel Benavides | Language | Java |

|  |
| --- |
| Problem Description |
| Briefly describe the problems that you encountered. |
| Para este ejercicio se me dificulto al principio el cálculo de los valores, ya que no había realizado |
| Correctamente la fórmula que indicaba el ejercicio. |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| Proposal Description |
| Briefly describe the process improvements that you propose. |
|  |
| En la etapa de diseño realizer un correcto entendimiento del requerimiento, para evitar problemas en la |
| Parte de implementación. |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| Other Notes and Comments |
| Note any other comments or observations that describe your experiences or improvement ideas. |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Coding/Counting Standard Template**

|  |  |
| --- | --- |
| Purpose | To guide the development of programs |
| Counting Standard | * Count each physical line as one LOC. * Do not count blank lines and comment-only lines. * Be consistent about what you put on each physical line. * Do not count import nor package lines (at the beginning) * Do not count special flag-marked lines. We are definifing the below ones:   + //M for modified lines   + //D for deleted lines |
| Program Headers | Begin all programs with a descriptive header. |
| Header Format | /\*  \* Classname  \*  \* Version information  \*  \* Date  \*  \*/ |
| Reuse Instructions | * Describe how the program is used. Provide the declaration format, parameter values and types, and parameter limits. * Provide warnings of illegal values, overflow conditions, or other conditions that could potentially result in improper operation. |
| Reuse Example | /\*\*   \* Brief description of the method purpose an operation  \* @author Engineer’s name  \* @param p\_fechaDesde Fecha desde la cual se deben buscar datos  \* @return valor del resultado  \*/ |
| Identifiers | Use descriptive names for all variables, method names, constants, and other identifiers. Avoid single letter variables, use them only when they are short lived. |
| Identifier Naming Table | |  |  |  | | --- | --- | --- | | Classes | Class names should be nouns, in mixed case with the first letter of each internal word capitalized. Try to keep your class names simple and descriptive. Use whole words-avoid acronyms and abbreviations | class Raster;  class ImageSprite; | | Interfaces | Interface names should be capitalized like class names. | interface RasterDelegate;  interface Storing; | | Methods | Methods should be verbs, in mixed case with the first letter lowercase, with the first letter of each internal word capitalized. | run();  runFast();  getBackground(); | | Variables | Except for variables, all instance, class, and class constants are in mixed case with a lowercase first letter. Internal words start with capital letters. Variable names should not start with underscore \_ or dollar sign $ characters, even though both are allowed.  Variable names should be short yet meaningful. The choice of a variable name should be mnemonic- that is, designed to indicate to the casual observer the intent of its use. One-character variable names should be avoided except for temporary "throwaway" variables. Common names for temporary variables are i, j, k, m, and n for integers; c, d, and e for characters. | int i;  char c;  float myWidth; | | Constants | Names should be in uppercase. | static final int MIN\_WIDTH = 4;  static final int MAX\_WIDTH = 999;  static final int GET\_THE\_CPU = 1; | |

(continued)**Coding Standard Template (continued)**

|  |  |
| --- | --- |
| Comments | * Document the code so that the reader can understand its operation. * Comments should explain both the purpose and behavior of the code. * Comment variable declarations to indicate their purpose. |
| Comments types | A block comment should be preceded by a blank line to set it apart from the rest of the code.  /\*  \* Here is a block comment.  \*/  A single-line comment should be preceded by a blank line.  if (condition) {  /\* Handle the condition. \*/  ...  }  Very short comments can appear on the same line as the code they describe, but should be shifted far enough to separate them from the statements. If more than one short comment appears in a chunk of code, they should all be indented to the same tab setting.  Here's an example of a trailing comment in Java code:  if (a == 2) {  return TRUE; /\* special case \*/  } else {  return isPrime(a); /\* works only for odd a \*/  }  The // comment delimiter can comment out a complete line or only a partial line. It shouldn't be used on consecutive multiple lines for text comments  if (foo > 1) {  // Do a double-flip.  ...  }  else {  return false; // Explain why here.  }  //if (bar > 1) {  //  // // Do a triple-flip.  // ...  //}  //else {  // return false;  //} |
| Good Comment | If(record\_count > limit) /\* have all records been processed?  \*/ |
| Bad Comment | If(record\_count > limit) /\* check if record count exceeds limit  \*/ |
| Major Sections | Precede major program sections by a block comment that describes the processing that is done in the next section |
| Example | /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  /  /\* The program section examines the contents of the array ‘grades’ and calcu- \*/  /\* lates the average class grade.  \*/  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  / |
| Blank Spaces | * Write programs with sufficient spacing so they do not appear crowded. * Separate every program construct with at least one space. |
| Indenting | * Avoid lines longer than 80 characters, since they're not handled well by many terminals and tools. * Indent every level of brace from the previous one. |
| Indenting  Example | someMethod(longExpression1, longExpression2, longExpression3,  longExpression4, longExpression5);  if ((condition1 && condition2)  || (condition3 && condition4)  ||!(condition5 && condition6)) {  doSomethingAboutIt();  } |