PSP0.1 Project Plan Summary

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Wilman Rincon | Date | 02/02/2015 |
| Program | Conteo de lineas | Program # | 2 |
| Instructor | Luis Daniel | Language | Java |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Program Size*** | ***Plan*** | | |  | ***Actual*** | | |  | ***To Date*** | | |
| ***Base (B)*** |  | | |  | 0 | | |  |  | | |
|  |  | | |  | ***(Measured)*** | | |  |  | | |
| ***Deleted (D)*** |  | | |  | 0 | | |  |  | | |
|  |  | | |  | ***(Counted)*** | | |  |  | | |
| ***Modified (M)*** |  | | |  | 0 | | |  |  | | |
|  |  | | |  | ***(Counted)*** | | |  |  | | |
| ***Added (A)*** |  | | |  | 0 | | |  |  | | |
|  |  | | |  | ***(T − B + D − R)*** | | |  |  | | |
| ***Reused (R)*** |  | | |  | 0 | | |  |  | | |
|  |  | | |  | ***(Counted)*** | | |  |  | | |
| ***Added and Modified (A+M)*** | 90 | | |  | 0 | | |  |  | | |
|  |  | | |  | ***(A + M)*** | | |  |  | | |
| ***Total Size (T)*** |  | | |  |  | | |  |  | | |
|  |  | | |  | ***(Measured)*** | | |  |  | | |
| ***Total New Reusable*** |  | | |  |  | | |  |  | | |
|  |  | | |  |  | | |  |  | | |
| **Time in Phase (min.)** | ***Plan*** |  | **Actual** | | |  | **To Date** | | |  | **To Date %** |
| Planning | 24 |  | 26 | | |  | 33 | | |  | 8 |
| Design | 72 |  | 30 | | |  | 50 | | |  | 13 |
| Code | 141 |  | 200 | | |  | 240 | | |  | 59 |
| Compile | 0 |  | 0 | | |  | 0 | | |  | 0 |
| Test | 27 |  | 30 | | |  | 38 | | |  | 9 |
| Postmortem | 36 |  | 30 | | |  | 40 | | |  | 11 |
| Total | 300 |  | 316 | | |  | 401 | | |  | 100 |
|  |  |  |  | | |  |  | | |  |  |
| **Defects Injected** |  |  | **Actual** | | |  | **To Date** | | |  | **To Date %** |
| Planning |  |  | 0 | | |  | 0 | | |  | 0 |
| Design |  |  | 0 | | |  | 0 | | |  | 0 |
| Code |  |  | 3 | | |  | 5 | | |  | 100 |
| Compile |  |  | 0 | | |  | 0 | | |  | 0 |
| Test |  |  | 0 | | |  | 0 | | |  | 0 |
| Total Development |  |  | 3 | | |  | 5 | | |  | 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 |  |  | 3 | | |  | 5 | | |  | 100 |
| Total Development |  |  | 3 | | |  | 5 | | |  | 100 |
| After Development |  |  |  | | |  |  | | |  |  |

PSP Time Recording Log

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Wilman Rincon | Date | 02/02/2015 |
| Program | Conteo Lineas | Program # | 2 |
| Instructor | Luis Daniel | Language | Java |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project** | **Phase** | **Start Date and Time** | **Int. Time** | **Stop Date and Time** | **Delta**  **Time** | **Comments** |
| 2 | Planing | 18:00 | 0 | 18:26 | 26 |  |
| 2 | Design | 18:40 | 0 | 19:10 | 30 |  |
| 2 | Code | 8:00 | 0 | 11:20 | 200 |  |
| 2 | Test | 12:00 | 0 | 12:30 | 30 |  |
| 2 | PostMortem | 13:00 | 0 | 13:30 | 30 |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

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 | 03/02/2015 |
| Program | Conteo lineas de codigo | Program # | 2 |
| Instructor | Luis Daniel | Language | Java |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Project |  | | Date |  | Number |  | Type |  | Inject |  | Remove |  | Fix Time |  | Fix Ref. |
| 2 |  | | 02/03 |  | 1 |  | 80 |  | code |  | test |  | 1 |  |  |
| Description: | | | En el metodo contador linea no quedo bien declarado el objeto HsahTable. | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
| Project |  | | Date |  | Number |  | Type |  | Inject |  | Remove |  | Fix Time |  | Fix Ref. |
| 2 |  | | 02/03 |  | 2 |  | 80 |  | code |  | test |  | 1 |  |  |
| Description: | | | En el metodo RetornarArchivo no se inicializo correctamente la variable lineaTemp | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
| Project |  | | Date |  | Number |  | Type |  | Inject |  | Remove |  | Fix Time |  | Fix Ref. |
| 2 |  | | 02/03 |  | 3 |  | 80 |  | code |  | test |  | 1 |  |  |
| Description: | | | En el metodo retornarProperties no tenía inicializado correctamente el objeto Properties. | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
| 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: | | |  | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |

PSP Process Improvement Proposal (PIP)

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Wilman Rincon | Date | 02/03/2015 |
| Program | Conteo de lineas | Program # | 2 |
| Instructor | Luis Daniel | Language | Java |

|  |
| --- |
| Problem Description |
| Briefly describe the problems that you encountered. |
|  |
| Para este proyecto encontre dificultados al momento de implementar la búsqueda de las clases que iba a |
| Leer de los programas, debido a que estoy incursionando en el lenguaje y no conozco todas las sentencias |
| de Java. |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| Proposal Description |
| Briefly describe the process improvements that you propose. |
|  |
| Para evitar estos retraso en el proceso se profundizara más en lenguaje para mitigar los tiempos tan altos |
| de desarrollo que pueden ocasionar el desconocimiento del lenguaje. |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| 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();  } |