PSP2.1 Project Plan Summary

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Deivis Enrique Vergel Arenas | Date | 07/03/2015 |
| Program | ECOS | Program # | 6 |
| Instructor | Luis Daniel Benavides Navarro | Language | JAVA |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Summary** | | **Plan** | | |  | **Actual** | | |  | **To Date** | | |
| Size/Hour | | 1012/27.33333334 | | |  | 372/4.383333333 | | |  | 1023/23.23333333 | | |
| Planned Time | | 1640 | | |  |  | | |  | 1640 | | |
| Actual Time | |  | | |  | 263 | | |  | 1781 | | |
| CPI (Cost-Performance Index) | |  | | |  |  | | |  | 0.920830993 | | |
|  | |  | | |  |  | | |  | (Planned/Actual) | | |
| % Reuse | | 0 | | |  | 0 | | |  | 6.26349892 | | |
| % New Reusable | | 11.03955842 | | |  | 65.23178808 | | |  | 34.77321814 | | |
| Test Defects/KLOC or equivalent | |  | | |  |  | | |  |  | | |
| Total Defects/KLOC or equivalent | | 3/265 | | |  | 1/302 | | |  | 14/1389 | | |
| Yield % | | 50 | | |  | 0 | | |  | 0 | | |
| ***% Appraisal COQ*** | | 1,334557553 | | |  | 4,562737643 | | |  | 4,94129159 | | |
| ***% Failure COQ*** | | 7,973048833 | | |  | 17,11026616 | | |  | 9,14872798 | | |
| ***COQ A/F Ratio*** | | 0,16738359 | | |  | 0,26666667 | | |  | 0,54010695 | | |
| ***PQI*** | |  | | |  |  | | |  |  | | |
|  | |  | | |  |  | | |  |  | | |
| **Program Size** | | **Plan** | | |  | **Actual** | | |  | **To Date** | | |
| Base (B) | | 0 | | |  | 0 | | |  |  | | |
|  | | (Measured) | | |  | (Measured) | | |  |  | | |
| Deleted (D) | | 1 | | |  | 0 | | |  |  | | |
|  | | (Estimated) | | |  | (Counted) | | |  |  | | |
| Modified (M) | | 1 | | |  | 2 | | |  |  | | |
|  | | (Estimated) | | |  | (Counted) | | |  |  | | |
| Added (A) | 100 | | | |  | 175 | | |  |  | | |
|  | (A+M − M) | | | |  | (T − B + D − R) | | |  |  | | |
| Reused (R) | 197 | | | |  | 197 | | |  | 284 | | |
|  | (Estimated) | | | |  | (Counted) | | |  |  | | |
| Added and Modified (A+M) | 120 | | | |  | 177 | | |  | 1362 | | |
|  | (Projected) | | | |  | (A + M) | | |  |  | | |
| Total Size (T) | 417 | | | |  | 372 | | |  | 1761 | | |
|  | (A+M + B − M − D + R) | | | |  | (Measured) | | |  |  | | |
| Total New Reusable | 250 | | | |  | 267 | | |  | 750 | | |
|  |  | | | |  |  | | |  |  | | |
| Estimated Proxy Size (E) |  | | | |  |  | | |  |  | | |
|  | |  | | |  |  | | |  |  | | |
| ***Upper Prediction Interval (70%)*** | | 542.1 | | |  |  | | |  |  | | |
| ***Lower Prediction Interval (70%)*** | | 291.9 | | |  |  | | |  |  | | |
|  | |  |  |  | | |  |  | | |  |  |

**(continued)**

**PSP2.1 Project Plan Summary (continued)**

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Deivis Enrique Vergel Arenas | Program # | 6 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Time in Phase (min.)** | **Plan** |  | | **Actual** | | | | |  | | **To Date** | | |  | | **To Date %** | |
| Planning | 26,6142617 |  | | 49 | | | | |  | | 207 | | |  | | 10,1272016 | |
| Design | 56,9343066 |  | | 12 | | | | |  | | 350 | | |  | | 17,1232877 | |
| Design Review | 4,04267266 |  | | 12 | | | | |  | | 36 | | |  | | 1,76125245 | |
| Code | 142,672656 |  | | 35 | | | | |  | | 882 | | |  | | 43,1506849 | |
| Code Review | 7,07467715 |  | | 23 | | | | |  | | 65 | | |  | | 3,18003914 | |
| Compile | 0 |  | |  | | | | |  | | 0 | | |  | | 0 | |
| Test | 23,9191465 |  | | 45 | | | | |  | | 187 | | |  | | 9,14872798 | |
| Postmortem | 38,7422796 |  | | 87 | | | | |  | | 317 | | |  | | 15,5088063 | |
| Total | 300 |  | | 263 | | | | |  | | 2044 | | |  | | 100 | |
| ***Total Time UPI (70%)*** | 390 |  | |  | | | | |  | |  | | |  | |  | |
| ***Total Time LPI (70%)*** | 210 |  | |  | | | | |  | |  | | |  | |  | |
|  |  | |  | |  | | |  | |  | | | | |  | |  |
| **Defects Injected** | **Plan** | |  | | **Actual** | | |  | | **To Date** | | | | |  | | **To Date %** |
| Planning | 0,07142857 | |  | |  | | |  | | 1 | | | | |  | | 6,66666667 |
| Design | 0,14285714 | |  | |  | | |  | | 2 | | | | |  | | 13,3333333 |
| Design Review | 0 | |  | |  | | |  | |  | | | | |  | |  |
| Code | 0,78571429 | |  | | 1 | | |  | | 12 | | | | |  | | 80 |
| Code Review |  | |  | |  | | |  | |  | | | | |  | |  |
| Compile |  | |  | |  | | |  | |  | | | | |  | |  |
| Test |  | |  | |  | | |  | |  | | | | |  | |  |
| Total Development | 1 | |  | | 1 | | |  | | 15 | | | | |  | | 100 |
|  |  | |  | |  | | |  | |  | | | | |  | |  |
| **Defects Removed** | ***Plan*** | |  | | **Actual** | | |  | | **To Date** | | | | |  | | **To Date %** |
| Planning | 0,07142857 | |  | |  | | |  | | 1 | | | | |  | | 6,66666667 |
| Design |  | |  | |  | | |  | |  | | | | |  | |  |
| Design Review |  | |  | |  | | |  | |  | | | | |  | |  |
| Code | 0,42857143 | |  | |  | | |  | | 6 | | | | |  | | 40 |
| Code Review |  | |  | |  | | |  | |  | | | | |  | |  |
| Compile |  | |  | |  | | |  | |  | | | | |  | |  |
| Test | 0,5 | |  | | 1 | | |  | | 8 | | | | |  | | 53,3333333 |
| Total Development | 1 | |  | | 1 | | |  | | 15 | | | | |  | | 100 |
| After Development |  | |  | |  | | |  | |  | | | | |  | |  |
|  |  | | | | |  |  | | | | |  |  | | | | |
| **Defect Removal Efficiency** | **Plan** | | | | |  | **Actual** | | | | |  | **To Date** | | | | |
| Defects/Hour − Design Review |  | | | | |  |  | | | | |  |  | | | | |
| Defects/Hour − Code Review |  | | | | |  |  | | | | |  |  | | | | |
| Defects/Hour − Compile |  | | | | |  |  | | | | |  |  | | | | |
| Defects/Hour − Test | 1,38461538/5 | | | | |  | 1/4,383333333 | | | | |  | 2/8.966666666 | | | | |
| DRL (DLDR/UT) |  | | | | |  |  | | | | |  |  | | | | |
| DRL (Code Review/UT) |  | | | | |  |  | | | | |  |  | | | | |
| DRL (Compile/UT) |  | | | | |  |  | | | | |  |  | | | | |

PSP Time Recording Log

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Deivis Enrique Vergel Arenas | Date | 07/03/2015 |
| Program | ECOS | Program # | 6 |
| Instructor | Luis Daniel Benavides Navarro | Language | JAVA |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project** | **Phase** | **Start Date and Time** | **Int. Time** | **Stop Date and Time** | **Delta**  **Time** | **Comments** |
| PSP2.1 | Plan | 28/02/15 8:36 PM |  | 28/02/15 9:25 AM | 49 | Analisis de las nuevas reglas para la proyeccion de p y la clase CalcularSimpsomRule |
| PSP2.1 | Desing | 28/02/15 9:26 PM |  | 28/02/15 9:38 PM | 12 | Diseño método agregado proyectarX y nuevos atributos en la clase CalcularSimpsomRule , diseño de la diseño de MVC |
| PSP2.1 | Desing Review | 28/02/15 9:39 PM |  | 28/02/15 9:51 PM | 12 |  |
| PSP2 | Code | 28/02/15 9:52 PM | 24 | 28/02/15 10:51 PM | 35 | Desarrollo método agregado proyectarX y nuevos atributos en la clase CalcularSimpsomRule , codificacion del MVC |
| PSP2 | Code Review | 01/03/15 03:25 PM |  | 01/03/15 03:48 PM | 23 |  |
| PSP2 | Test | 01/03/15 03:49 PM | 12 | 01/03/15 04:46 PM | 45 |  |
| PSP2 | Postmortem | 07/03/15 07:37 PM | 29 | 07/03/15 09:33 PM | 87 |  |

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 | Deivis Enrique Vergel Arenas | Date | 07/03/2015 |
| Program | ECOS | Program # | 6 |
| Instructor | Luis Daniel Benavides Navarro | Language | JAVA |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Project |  | | Date |  | Number |  | Type |  | Inject |  | Remove |  | Fix Time |  | Fix Ref. |
| PSP2 |  | | 01/03/2015 |  | 01 |  | 40 |  | Code |  | Test |  | 2 |  |  |
| Description: | | | se aplico escala de decimales en el cálculo de las formulas parciales lo que | | | | | | | | | | | | | |
| causaba diferencias de decimales en el resultado final | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |

PSP Defect Type Standard

|  |  |  |
| --- | --- | --- |
| **Type Number** | **Type Name** | **Description** |
| 10 | Documentation | Comments, messages |
| 20 | Syntax | Spelling, punctuation, typos, instruction formats |
| 30 | Build, Package | Change management, library, version control |
| 40 | Assignment | Declaration, duplicate names, scope, limits |
| 50 | Interface | Procedure calls and references, I/O, user formats |
| 60 | Checking | Error messages, inadequate checks |
| 70 | Data | Structure, content |
| 80 | Function | Logic, pointers, loops, recursion, computation, function defects |
| 90 | System | Configuration, timing, memory |
| 100 | Environment | Design, compile, test, or other support system problems |

PSP Process Improvement Proposal (PIP)

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Deivis Enrique Vergel Arenas | Date | 07/03/2015 |
| Program | ECOS | Program # | 6 |
| Instructor | Luis Daniel Benavides Navarro | Language | JAVA |

|  |
| --- |
| Problem Description |
| Briefly describe the problems that you encountered. |
| Se presentaron errores de definición de datos importantes para resolver correctamente el ejercicio |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| Proposal Description |
| Briefly describe the process improvements that you propose. |
| El redondeo de los valores debe realizarse al finalizar los cálculos del proceso completo para evitar |
| desviación del resultado final por redondeos parciales |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| Other Notes and Comments |
| Note any other comments or observations that describe your experiences or improvement ideas. |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Test Report Template

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Deivis Enrique Vergel Arenas | Date | 07/03/2015 |
| Program | ECOS | Program # | 6 |
| Instructor | Luis Daniel Benavides Navarro | Language | JAVA |

|  |  |
| --- | --- |
| Test Name/Number | CasosPropuesto /01 |
| Test Objective | la solución debe cumplir con los casos de prueba propuestos por el ejercicio |
|  |  |
| Test Description | Calcular los datos expuestos en la tabla 1 con los datos allí expuestos |
|  |  |
|  |  |
|  |  |
|  |  |
| Test Conditions | Cargar automáticamente los datos de la tabla 1 |
|  |  |
|  |  |
|  |  |
|  |  |
| Expected Results | Generar la salida esperada en la tabla 1 |
|  |  |
|  |  |
|  |  |
| Actual Results | Generara los datos de salida esperada en la tabla 1 |
|  |  |
|  |  |
|  |  |
|  |  |

**Size Counting Standard Template**

|  |  |  |  |
| --- | --- | --- | --- |
| Definition Name: | LOC java estandar | Language: | JAVA |
| Author: | DeivisVergel | Date: | 04/02/2015 |

|  |  |  |
| --- | --- | --- |
| **Count Type** | **Type** | **Comments** |
| Physical/Logical | logical |  |
| **Statement Type** | **Included** | **Comments** |
| Executable | si |  |
| Nonexecutable: |  |  |
| Declarations | Si, Nota 1 | Private String var1; |
| Compiler Directives | No, Nota 3 | Import package |
| Comments | No,Nota 2 | // /\* \*/ |
| Blank lines | no |  |
| Other elements |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| **Clarifications** |  | **Examples/Cases** |
| Espacios en blanco | no |  |
| Inicios de deciciones ,ciclos , o metodos | si | Se incluyen los inicios de ciclo CASE, DO, ELSE, ENUM, FOR, IF, SWITCH, WHILE etc, el inicio debe estar en la misma línea “{” en caso contrario se contaran los dos como inicio |
| fin de decisiones o ciclos | no | No se contaran los finales de ciclo que estén solos en una línea así “}” |
| Sentencias multilineas | si | Un métodos o sentencia multilinea será contada individualmente |
| Clases Embebidas | no | Se debe crear una clase por archivo .java |
|  |  |  |
|  |  |  |
| Note 1 |  | Se identifica como atributo de una clase una declaración por línea , si son del mismo tipo se deberá hacer en una línea independiente |
| Note 2 |  | Se excluyen los comentarios realizados en una lineacompleta con los símbolos “//” y los comentarios multilinea con los símbolos “/\*” y “\*/”, y las líneas comentarías intermedias siempre que comiencen por el símbolo “\*” |
| Note 3 |  | No se incluyen las lineas de imports y package |
| Note |  |  |

Size Estimating Template

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Student | | Deivis Enrique Vergel Arenas | | | Date | | 07/03/2015 | |
| Program | | ECOS | | | Program # | | 6 | |
| Instructor | | Luis Daniel Benavides Navarro | | | Language | | JAVA | |
| Size Measure | |  |  | |  | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Estimated | | | | | | | | | | | |
| Base Parts |  | Base | |  | Deleted | |  | Modified | | |  | Added | |
| App |  | 0 | |  | 2 | |  | 2 | | |  | 20 | |
| MainView |  | 0 | |  | 0 | |  | 0 | | |  | 100 | |
| CalcularSimpsomRule |  | 0 | |  | 0 | |  | 0 | | |  | 145 | |
|  |  |  | |  |  | |  |  | | |  |  | |
| Total | B | | 0 | D | | 2 | M | | 2 | **BA** | | | 265 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Actual | | | | | | | | | | |
| Base Parts |  | Base | |  | Deleted | |  | Modified | |  | Added | |
| App |  |  | |  |  | |  |  | |  | 38 | |
| MainView |  |  | |  |  | |  |  | |  | 64 | |
| CalcularSimpsomRule |  | 197 | |  |  | |  |  | |  | 70 | |
|  |  |  | |  |  | |  |  | |  |  | |
| Total |  | | 197 |  | | 0 |  | | 0 |  | | 169 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Estimated | | | | | | | |  | Actual | | |
| Parts Additions |  | Type |  | Items |  | Rel. Size | |  | Size\* |  | Size\* |  | Items |
|  |  |  |  |  |  |  | |  |  |  |  |  |  |
|  |  |  |  |  |  |  | |  |  |  |  |  |  |
|  |  |  |  |  |  |  | |  |  |  |  |  |  |
|  |  |  |  |  |  |  | |  |  |  |  |  |  |
|  |  |  |  |  |  |  | |  |  |  |  |  |  |
|  |  |  |  |  |  |  | |  |  |  |  |  |  |
|  |  |  |  |  |  |  | |  |  |  |  |  |  |
|  |  |  |  |  |  |  | |  |  |  |  |  |  |
|  |  |  |  |  |  |  | |  |  |  |  |  |  |
|  |  |  |  |  |  |  | |  |  |  |  |  |  |
| Total |  |  |  |  |  |  | PA | |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  | Estimated |  | Actual |
| Reused Parts | |  | Size |  | Size |
| CalcularSimpsomRule | |  | 197 |  | 267 |
|  | |  |  |  |  |
|  | |  |  |  |  |
|  | |  |  |  |  |
|  | |  |  |  |  |
|  | |  |  |  |  |
|  | |  |  |  |  |
| Total | R | | 197 |  | 267 |

Size Estimating Template

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Deivis Enrique Vergel Arenas | Program | JAVA |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PROBE Calculation Worksheet (Added and Modified) |  | Size |  | Time |
| Added size (A): A = BA+PA |  | 316 |  |  |
| Estimated Proxy Size (E): E = BA+PA+M |  | 225 |  |  |
| PROBE estimating basis used: (A, B, C, or D) |  | D |  |  |
| Correlation: (R2) |  | 0.9111 |  |  |
| Regression Parameters: β0 Size and Time |  | -22.55 |  |  |
| Regression Parameters: β1 Size and Time |  | 1.7279 |  |  |
| Projected Added and Modified Size (P): P = β0size + β1size\*E |  | 644.429 |  |  |
| Estimated Total Size (T): T = P + B - D - M + R |  |  |  |  |
| 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: |  |  |  |  |

**PSP2 Design Review Checklist**

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Deivis Enrique Vergel Arenas | Date | 07/03/2015 |
| Program | ECOS | Program # | 6 |
| Instructor | Luis Daniel Benavides Navarro | Language | JAVA |

|  |  |
| --- | --- |
| Purpose | To guide you in conducting an effective design review |
| General | * Review the entire program for each checklist category; do not attempt to review for more than one category at a time! * As you complete each review step, check off that item in the box at the right. * Complete the checklist for one program or program unit before reviewing the next. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Complete | Verify that the design covers all of the applicable requirements.   * All specified outputs are produced. * All needed inputs are furnished. * All required includes are stated. | x | x | x |  |
| External Limits | Where the design assumes or relies upon external limits, determine if behavior is correct at nominal values, at limits, and beyond limits. | x |  |  |  |
| Logic | * Verify that program sequencing is proper.   Stacks, lists, and so on are in the proper order.  Recursion unwinds properly.   * Verify that all loops are properly initiated, incremented, and terminated. * Examine each conditional statement and verify all cases. | x | x | x |  |
| Internal Limits | Where the design assumes or relies upon internal limits, determine if behavior is correct at nominal values, at limits, and beyond limits. | x |  |  |  |
| Special Cases | * Check all special cases. * Ensure proper operation with empty, full, minimum, maximum, negative, and Cero values for all variables. * Protect against out-of-limits, overflow, and underflow conditions. * Ensure “impossible” conditions are absolutely impossible. * Handle all possible incorrect or error conditions. | x | x | x | x |
| Functional Use | * Verify that all functions, procedures, or methods are fully understood and properly used. * Verify that all externally referenced abstractions are precisely defined. | x | x |  |  |
| System Considerations | * Verify that the program does not cause system limits to be exceeded. * Verify that all security-sensitive data are from trusted sources. * Verify that all safety conditions conform to the safety specifications. | x | x | x |  |
| Names | Verify that   * all special names are clear, defined, and authenticated * the scopes of all variables and parameters are self-evident or defined * all named items are used within their declared scopes | x | x | x |  |
| Standards | Ensure that the design conforms to all applicable design standards. | x |  |  |  |

**Code Review Checklist**

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Deivis Enrique Vergel Arenas | Date | 07/03/2015 |
| Program | ECOS | Program # | 6 |
| Instructor | Luis Daniel Benavides Navarro | Language | JAVA |

|  |  |
| --- | --- |
| Purpose | To guide you in conducting an effective code review |
| General | * Review the entire program for each checklist category; do not attempt to review for more than one category at a time! * As you complete each review step, check off that item in the box at the right. * Complete the checklist for one program or program unit before reviewing the next. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Complete | Verify that the code covers all of the design. | x |  |  |  |
| Includes | Verify that the includes are complete. | x |  |  |  |
| Initialization | Check variable and parameter initialization.   * at program initiation * at start of every loop * at class/function/procedure entry | x | x | x |  |
| Calls | Check function call formats.   * pointers * parameters | x | x |  |  |
| Names | Check name spelling and use.   * Is it consistent? * Is it within the declared scope? * Do all structures and classes use ‘.’ reference? | x | x | x |  |
| Strings | Check that all strings are   * identified by pointers * terminated by NULL | x | x |  |  |
| Pointers | Check that all   * pointers are initialized NULL * pointers are deleted only after new * new pointers are always deleted after use | x | x | x |  |
| Output Format | Check the output format.   * Line stepping is proper. * Spacing is proper. | x | x |  |  |
| () Pairs | Ensure that () are proper and matched. | x |  |  |  |
| Logic Operators | * Verify the proper use of ==, =, ||, and so on. * Check every logic function for (). | x | x |  |  |
| Line-by-line check | Check every line of code for   * instruction syntax * proper punctuation | x | x |  |  |
| Standards | Ensure that the code conforms to the coding standards. | x |  |  |  |
| File Open and Close | Verify that all files are   * properly declared * opened * closed | x | x | x |  |

**Operational Specification Template**

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Deivis Enrique Vergel Arenas | Date | 07/03/2015 |
| Program | ECOS | Program # | 6 |
| Instructor | Luis Daniel Benavides Navarro | Language | JAVA |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scenario Number** | **1** | **User Objective** | Usuario funcional | |
| **Scenario Objective** | | Detalle de scenario de prueba | | |
| **Source** | **Step** | **Action** | | **Comments** |
| Heroku | 1 | Ingresar a la URL : https://ecosdeve6.herokuapp.com/ | | Pueden ocurrir errores de dispoibilidad, o no han sido asignado dynos a la aplicacion |
| Programa | 2 | Comprobar que los calculos coincidan con los resultados esperados | |  |

State Specification Template

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Student | | Deivis Enrique Vergel Arenas | | Date | | 07/03/2015 |
| Program | | ECOS | | Program # | | 6 |
| Instructor | | Luis Daniel Benavides Navarro | | Language | | JAVA |
|  |  | | | | | |
| **State Name** | | | **Description** | | | |
| Valido | | | El estado es valido cuando la clase retorna los valores esperados de la prueba propuesta con los siguientes valores:   * prueba 1:e =.00001, x=1, num\_seg=10, dof=6 y p=0.20 , obtiene el resultado esperado de x en 0.55338 * prueba 2: e =.00001, x=1, num\_seg=10, dof=6 y p=0.45 , obtiene el resultado esperado de x en 1.75305 * prueba 3 : e =.00001, x=1, num\_seg=10, dof=6 y p=0.495 obtiene el resultado esperado de x en 4.60352 | | | |
| Invalido | | | El estado es invalid cuando lo valores esperados para x no coinciden con los de la prueba | | | |
| Funcional | | | El estado es no functional cuando la aplicacion no despliega o no esta disponible en la URL : https://ecosdeve6.herokuapp.com/ | | | |
|  | | |  | | | |
|  | | |  | | | |
| **Function/Parameter** | | | **Description** | | | |
| public void proyectarX() throws Exception | | | Esta function es la encargada de obtener los atributos de la clase y ejecutar la regla de simpson interativamente hasta obtener el valor esperado de p | | | |
|  | | |  | | | |
|  | | |  | | | |
|  | | |  | | | |
|  | | |  | | | |
|  | | |  | | | |
| **States/Next States** | | | **Transition Condition** | | **Action** | |
| Funcional | | | Ingresar a la URL : <https://ecosdeve6.herokuapp.com/> | | Se ejecuta el programa y arroja la pantalla con los resultados | |
| Valido | | | Ejecutar la function proyectarX en tres instancioas de la clase CalcularSimpsomRule los parametros y resultados deben ser :   * instancia 1:e =.00001, x=1, num\_seg=10, dof=6 y p=0.20 , obtiene el resultado esperado de x en 0.55338 * instancia 2: e =.00001, x=1, num\_seg=10, dof=6 y p=0.45 , obtiene el resultado esperado de x en 1.75305 * instancia 3 : e =.00001, x=1, num\_seg=10, dof=6 y p=0.495 obtiene el resultado esperado de x en 4.60352 | | Terminar el proceso | |
| Invalido | | | Ejecutar la function proyectarX en tres instancioas de la clase CalcularSimpsomRule y no obtener los resultados esperados en el estado valido | | Terminar el proceso | |

**Logic Specification Template**

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Deivis Enrique Vergel Arenas | Date | 07/03/2015 |
| Program | ECOS | Program # | 6 |
| Instructor | Luis Daniel Benavides Navarro | Language | JAVA |

|  |  |
| --- | --- |
| **Design** | Requerimientos de programa |
| **References** | Se reutiliza la clase CalcularSimpsomRule y se adiciona la funcion proyectarX para |
|  | calcular p hasta llegar al valor deseado |
|  |  |
|  |  |

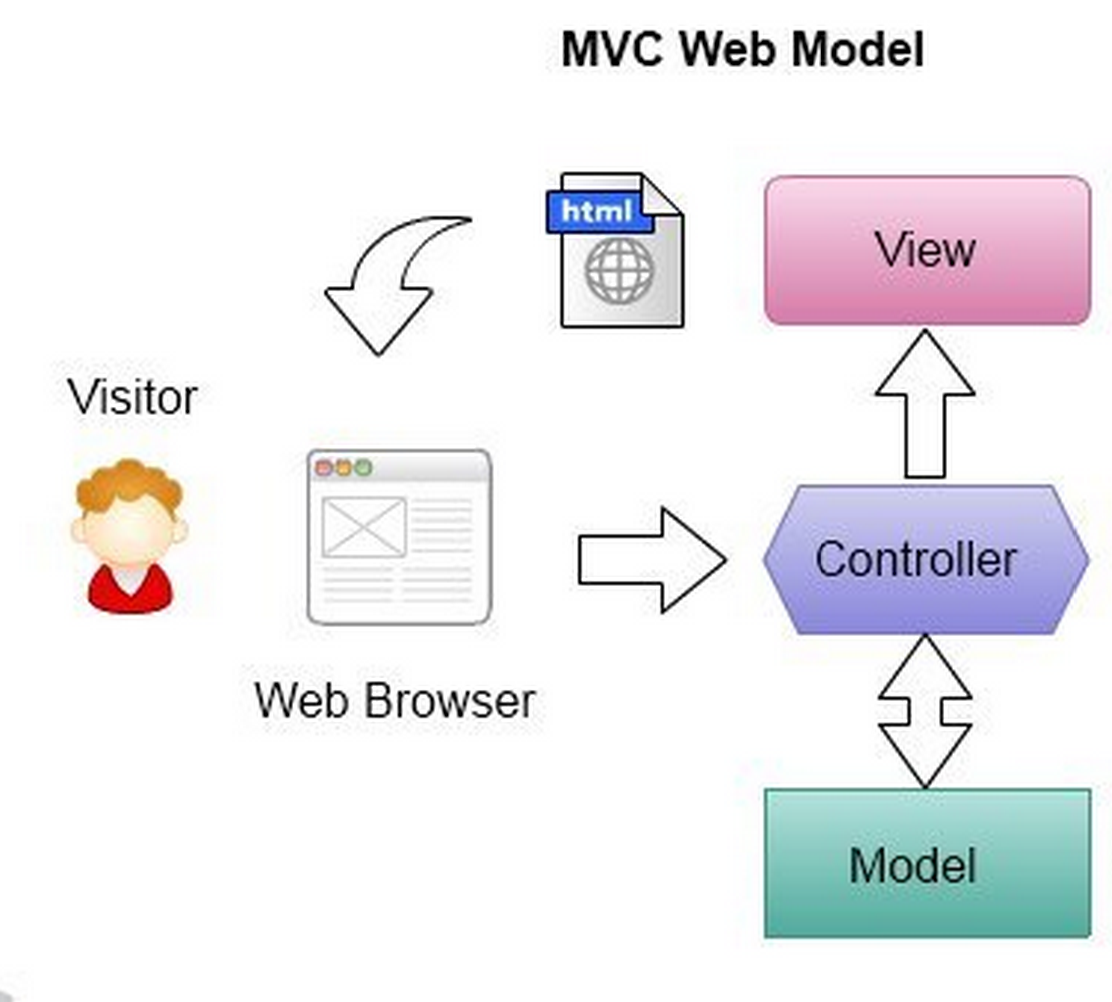
|  |  |
| --- | --- |
| **Parameters** | Ninguno |
|  |  |
|  |  |
|  |  |
|  |  |
| public void proyectarX() throws Exception {  try {  CalcularSimpsonRule intento = new CalcularSimpsonRule(this.e, this.x, this.num\_seg, this.dof);  BigDecimal acumulador\_num\_seg = BigDecimal.ZERO;  int signo;  intento.calcularP();  signo = intento.getP().setScale(5, RoundingMode.HALF\_UP).compareTo(getProyectarP().setScale(5, RoundingMode.HALF\_UP));  getIntentos().add(intento);  do {  acumulador\_num\_seg = acumulador\_num\_seg.add(this.num\_seg.multiply(new BigDecimal(2)));  if (!getProyectarP().equals(BigDecimal.ZERO) && getIntentos().size() >= 2) {  setAuxiliar(getResultado());  if ((intento.getP().setScale(5, RoundingMode.HALF\_UP).compareTo(getProyectarP().setScale(5, RoundingMode.HALF\_UP)) < 0) && !intento.getP().setScale(5, RoundingMode.HALF\_UP).equals(getAuxiliar().setScale(5, RoundingMode.HALF\_UP))) {  setX(getX().add(getD()));  } else if ((intento.getP().setScale(5, RoundingMode.HALF\_UP).compareTo(getProyectarP().setScale(5, RoundingMode.HALF\_UP)) > 0) && !intento.getP().setScale(5, RoundingMode.HALF\_UP).equals(getAuxiliar().setScale(5, RoundingMode.HALF\_UP))) {  setX(getX().subtract(getD()));  }  }  intento = new CalcularSimpsonRule(this.e, this.x, this.num\_seg.multiply(new BigDecimal(2)), this.dof);  intento.calcularP();  if (signo != intento.getP().setScale(5, RoundingMode.HALF\_UP).compareTo(getProyectarP().setScale(5, RoundingMode.HALF\_UP))) {  setD(d.divide(new BigDecimal(2), MathContext.DECIMAL64));  }  signo = intento.getP().setScale(5, RoundingMode.HALF\_UP).compareTo(getProyectarP().setScale(5, RoundingMode.HALF\_UP));  if (intento.getP().setScale(5, RoundingMode.HALF\_UP).equals(getProyectarP().setScale(5, RoundingMode.HALF\_UP))) {  setFlag(true);  }  getIntentos().add(intento);  } while ((intentos.get(intentos.size() - 2).getP().setScale(5, RoundingMode.HALF\_UP).subtract(intentos.get(intentos.size() - 1).getP().setScale(5, RoundingMode.HALF\_UP)).abs().setScale(5, RoundingMode.HALF\_UP).compareTo(getE().setScale(5, RoundingMode.HALF\_UP)) >= 0) || (!isFlag()));  setResultado(intentos.get(intentos.size() - 1).getP().setScale(5, RoundingMode.HALF\_UP));  } catch (Exception ex) {  throw new ExceptionApp("Error al Calcular la variables:" + ex.getMessage());  }  } | |

**Metaphor/Architecture Specification Template**

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Deivis Enrique Vergel Arenas | Date | 07/03/2015 |
| Program | ECOS | Program # | 6 |
| Instructor | Luis Daniel Benavides Navarro | Language | JAVA |

|  |  |
| --- | --- |
| **Design** | Operacional |
| **References** |  |
|  |  |
|  |  |
|  |  |

**Graphical representation of the metaphor**

****

**Textual representation of metaphor**

La aplicación tiene una arquitectura basada en el patrón MVC, he implementa tres capas dispuestas según el proceso lógico que debe ejecutar cada componente para mantener principios de seguridad , encapsulamiento y rehusó entre otros, el usuario interactúa con dos de las capas, con una vista que provee todos los componentes gráficos necesarios para su interacción y con un controlador que atiende todas las peticiones del usuario realizadas desde la vista por medio de las acciones propuestas por el comportamiento de cada acción , desde este punto el controlador se encargara de gestionar la petición e invocar a la capa de modelo que contiene la lógica de comportamiento de la funcionalidad, y retornar los datos atreves del controlador para que sean interpretados y graficados por la vista de nuevo