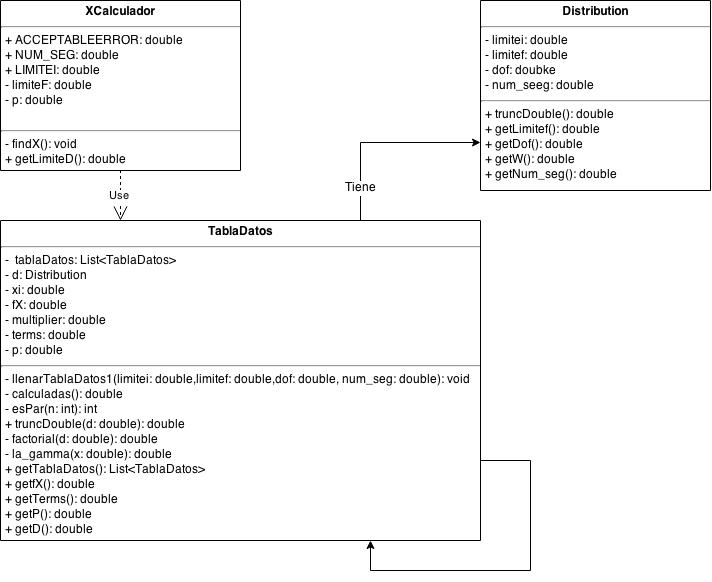
**Metaphor/Architecture Specification Template**

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Alvaro Suarez | Date | 01 Marzo 2015 |
| Program | Tarea 6 | Program # | CSOF5101\_01\_6 |
| Instructor | Luis Daniel Benavides Navarro | Language | JAVA |

|  |  |
| --- | --- |
| **Design** | 1. Distribution |
| **References** | 2. TablaDatos |
|  | 3. XCalculador |
|  |  |
|  |  |

**Graphical representation of the metaphor**

****

**Textual representation of metaphor**

|  |
| --- |
| Se crearon 3 clases en el diseño 1. Distribution que contiene los datos que se van a calcular en la integral, estos datos son limite inferior y limite superior de la integral, grados de libertad, asi como el numero de segmentos a utilizer |
| 2. TablaDatos que es la que guarda los datos propios de la integral para dicha distribucion, contiene los atributos si misma, la distribucion, limite inferior, f(x), el multiplicador, y el resultado de la integral. |
| 3. XCalculador Es la clase encargada de hacer la busqueda del limite superior de la integral para un valor dado p, |

Metaphor/Architecture Specification Template Instructions

|  |  |
| --- | --- |
| Purpose | * To contain the metaphor for a program, component, or system * To enable precise, rapid and complete design understanding * To facilitate thorough design and implementation reviews and inspections |
| General | * Use this template to document the program’s high-level metaphor. * The metaphor could be based in common programming patterns as MVC, or architectural styles as tree layer design, client-server, or inversion of control frameworks * After implementation and testing, update the template to reflect the actual implemented product. * Use plain language and avoid using programming instructions wherever practical. |
| 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. |
| Design References | List the references used to produce the program’s logical design.   * the Operational, Functional, and State templates * the program’s requirements * any other pertinent source |
| Graphical representation of the metaphor/Architecture | * Create a graphical representation of the main program parts and its interactions * Use clear names for each part * Use edges with arrows to show interactions * Use descriptive names for the interactions |
| Textual representation of metaphor | * Use text to describe the main idea and metaphor used in your design * Describe the graphical representation using common language |