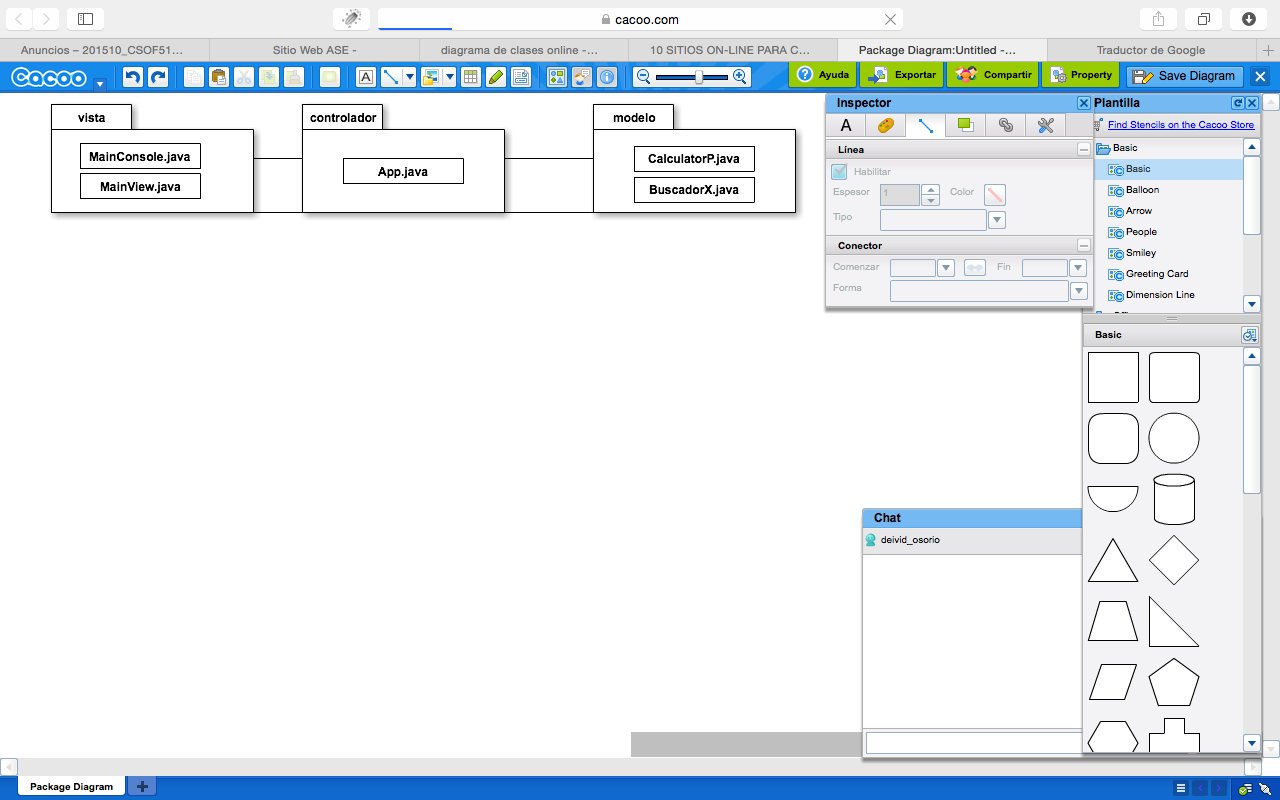
**Metaphor/Architecture Specification Template**

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Deivid Alexander Osorio Barrera | Date | 27/02/2015 |
| Program | Programa para encontrar el valor de x para el que la integral de t de 0 a x da un resultado de p | Program # | 6 |
| Instructor | Luis Daniel Benavides Navarro | Language | Java |

|  |  |
| --- | --- |
| **Design** |  |
| **References** | Se basa la construcción de este diseño en el patron de diseño modelo - vista – controlador el cual ha permitido organizer las clases de acuerdo a su especificación funcional o la especialidad de la clase, lo que se llama alta especialización de las clases. |
|  |  |

**Graphical representation of the metaphor**

****

**Textual representation of metaphor**

**Se crean tres paquetes que representen a su vez la vista o interafe que tendran los usuarios, que para este taller, hay una interface de consola y una web.**

**El siguiente paquete es el controlador, que tiene las clases necesarias para hacer comunicar la vista con las clases que representan el modelo del sistema.**

**El paquete de modelo tiene la especificación de los objetos que permiten realizar los calculus, este package tiene las clases que son encargadas de realizar las formulas y procesos matemáticos de acuerdo al objeto o la definición que representa cada clase.**

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 |