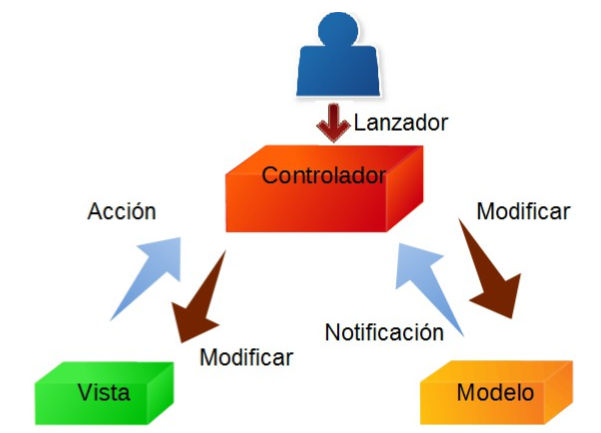
**Metaphor/Architecture Specification Template**

|  |  |  |  |
| --- | --- | --- | --- |
| Student | Making Solutions | Date | 25/04/2015 |
| Program | TSP | Program # | 1 |
| Instructor | Daniel Benavidez | Language | Java |

|  |  |
| --- | --- |
| **Design** | MVC |
| **References** | Se utiliza un patrón MVC para el desarrollo de un analizador de programas |
|  |  |
|  |  |
|  |  |

**Graphical representation of the metaphor**

****

**Textual representation of metaphor**

El modelo utilizado para el diseño y desarrollo de programa es MVC “Modelo-Vista-Controlador” que lo que busca es separar todo el código del programa en tres partes:

1. Modelo: Va la lógica del negocio, se gestiona el acceso a datos y sus formatos.

2. Vista: Interactua con el usuario, renderiza los datos que provienen del modelo.

3 Controlador: Se encarga del flujo de la aplicación, es el mediador entre el modelo y la vista ya que entre ellos no se pueden comunicar, también responde a las peticiones del usuario realizados desde la vista y la pasa al modelo para que sean procesados.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 |