**DISEÑO CONCEPTUAL**

**ITERACIÓN 1 TSP**

**STATUS QUO**

**ZAMIR ANDREI GARCÍA ROMERO**

**ÁLVARO DAVID LÓPEZ PINILLA**

**DIEGO ANDRÉS LOZANO ROLDÁN**

**WILMAN RINCON BAUTISTA**

**DEIVIS ENRIQUE VERGEL ARENAS**

**CONCEPTOS AVANZADOS DE INGENIERÍA DE SOFTWARE**

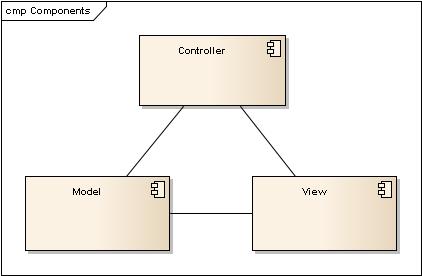
**UNIVERSIDAD DE LOS ANDES**

**2.015**

**A continuación se muestra el diseño conceptual correspondiente a la primera iteración del proyecto:**

# Diseño

El diseño que se selecciono fue la división en componentes organizados con el patrón MVC.



# Componentes

En la imagen se tienen tres componentes que funcionan como agrupadores, representan las áreas de trabajo que se especificaron. El componente Model, representa el mundo y en este caso, mantendrá las clases encargadas de realizar los cálculos que soportaran el negocio, es decir, los conteos de líneas, clases, y soporte al lenguaje. El componente Controller, estará compuesto por las clases que orquestarán el flujo de la aplicación, y que servirán de intermediarios entre las salidas y entradas del programa. Por último, se encuentra el componente View, que es la clase que contiene la interfaz de usuario; esta interfaz está contenida en una clase, y muestra la bienvenida al usuario junto con los datos calculados que genere la aplicación.

Las relaciones entre los diferentes componente son de manera bidireccional, y se relacionan tal cual como se ve en la imagen. En este orden, el componente Modelo se relaciona con el componente Controller y con View, lo cual establece los flujos de información de la aplicación.

# Tamaño de los componentes

Los tamaños estimados de cada componente se encuentran especificados en el formato STRAT de la iteración.

# Funciones

Las funciones que se implementaran son las siguientes:

1. Conteo de líneas lógicas por cada procedimiento, función o rutina.
2. Conteo de clases.
3. Mostrar una descripción de clases (métodos, atributos, visibilidad).
4. Mostrar una descripción del programa (paquetes, clases por paquete).
5. Leer una configuración de reglas de análisis de complejidad, relaciones, acoplamiento y accesibilidad de un programa en java.
6. Mostrar las excepciones al estándar de desarrollo.

Nota: En este caso, el programa que se usara estará embebido en el proyecto.