Metrics

Los resultados del análisis en Metrics son satisfactorios; un único parámetro (Number of Parameters) no cumple con los límites máximos aconsejados.

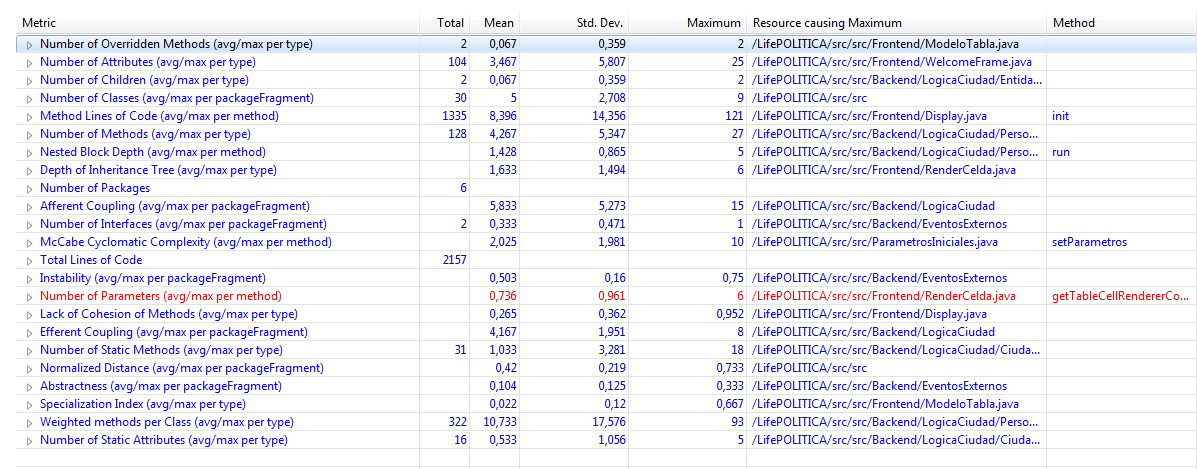


Figure 1 - Metrics Entrega 5

Logramos mejorar la los otros dos parámetros que resultaban rojos en la versión anterior del código (Nested Block Depth y McCabe Cyclomatic Complexity).

Con respecto al parámetro Nested Block Depth, disminuimos los niveles de anidamiento en el método placeComponentes en la clase Persona de 7 a 5. Este parámetro mide el número de niveles de anidamiento y se supone que mientras más profundo sea el anidamiento, más complejo resulta el código. Más niveles de anidamiento hacen el código más complicado de entender, como también aumenta las posibilidades de causar errores en la lógica del programa. El número máximo aconsejado es 5, exactamente el número que logramos para la nueva versión del software. Si el número es superior, es aconsejable dividir el método. De hecho partimos el método anterior entre submetodos más sencillos.

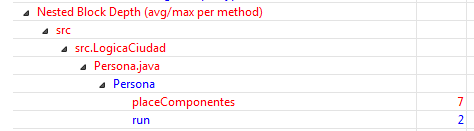


Figure - Nested Block Depth Entrega 4

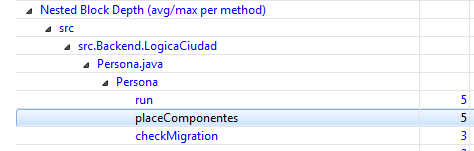


Figure - Nested Block Depth Entrega 5

El parámetro Nested Block Depth está conectado con el parámetro McCabe Cyclomatic Complexity, de hecho simplificando el método placeComponentes, mejoró también la McCabe Cyclomatic Complexity de 26 a 10. En este caso también conseguimos alcanzar el número máximo aconsejado: 10 ciclos. Si el método tiene más de 10 ciclos, es mejor dividir el método. Cada vez que ocurre un flujo alternativo (if, for, while, etc.) este parámetro es incrementado en una unidad. Como para la entrega anterior el valor era igual a 26, lo consideramos alarmante por su complejidad. Una mayor complejidad podría causar errores y baja entendibilidad. Además, métodos más complejos toman más tiempo para testarlos.

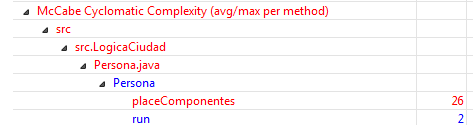


Figure - Cyclomatic Complexity Entrega 4

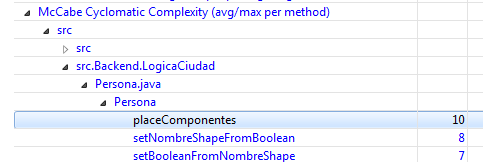


Figure 5 - Cyclomatic Complexity Entrega 5

Logramos también disminuir la complejidad del método setParametros, modificando las condiciones de control sobre los parametros iniciales ingresados del usuario (velocidad, número de partidos y porcentajes por cada partido). En lugar de ocupar una condición if por cada valor ingresado, creamos 3 parametros boolean y un único control if sobre la conjunción lógica entre ellos.

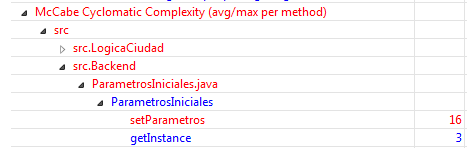


Figure - Cyclomatic Complexity Entrega 4

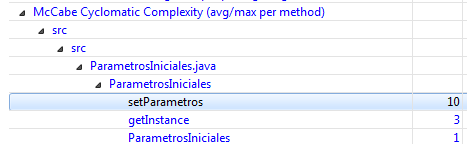


Figure 7 - Cyclomatic Complexity Entrega 5

El único parámetro que no pudimos mejorar fue el Number of Parameters en el método getTableCellRendererComponent, quedando igual a 6. No fue posible modificarlo porque proviene de un método que no fue hecho por nosotros. Además, como el máximo aconsejable es 5, no consideramos este dato alarmante.



Figure - Number of Parameters Entrega 5