



Taller: CODEMR

G671 - Calidad y Auditoría - Grupo 1



Tras finalizar este taller deberíais saber...

1. ¿Qué es CodeMR?
2. ¿Cómo se instala CodeMR en Eclipse?
3. ¿Cómo se instala CodeMR en Android Studio?
4. ¿Por qué es importante CodeMR?
5. ¿Qué nos puede proporcionar CodeMR?

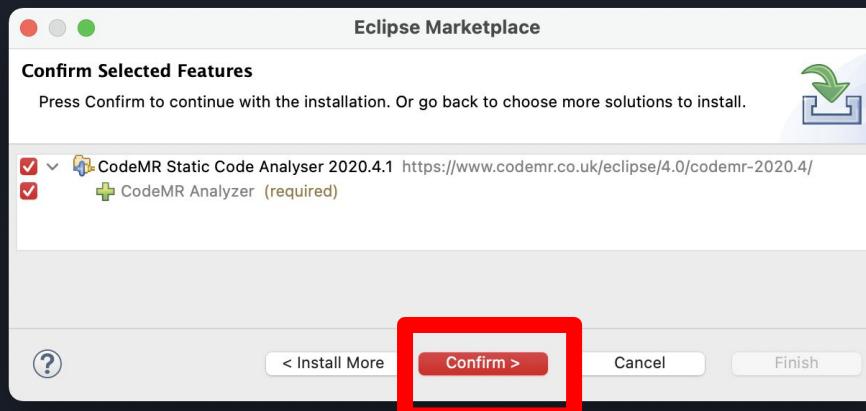


1. ¿Qué es CodeMR?

- Es una herramienta software que escanea un código software y muestra el nivel de calidad del mismo.
- CodeMR analiza el código fuente de manera local y guarda los ficheros de análisis en tu directorio de trabajo. Aunque existe una versión que permite analizar tu servidor o contenedor de docker.

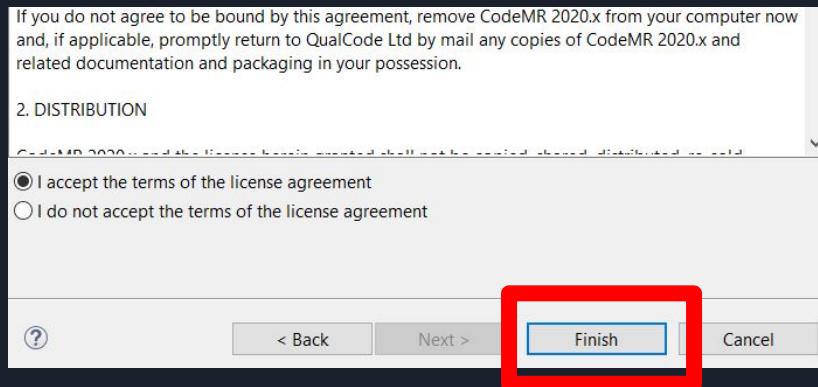
2. ¿Cómo se instala CodeMR en Eclipse?

1. Acceder a la página de descargas codemr.co.uk/downloads/
2. Arrastrar el botón  en una ventana de código en Eclipse de un proyecto abierto.
3. Se abrirá automáticamente el menú de Instalar Software nuevo.



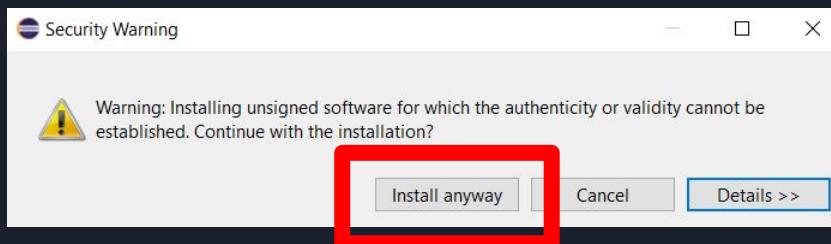
2. ¿Cómo se instala CodeMR en Eclipse?

4. Sigue las instrucciones de instalación de Eclipse. (Para C++ es necesario seleccionar la opción 'CodeMR C++ Extension y tener EclipseCDT instalado).
5. Se deben aceptar los términos de licencia y darle al botón de *Finish*.

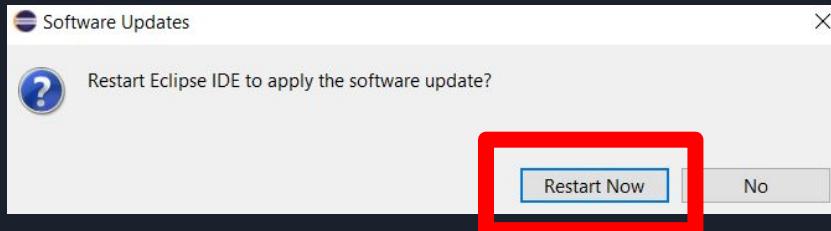


2. ¿Cómo se instala CodeMR en Eclipse?

6. Aunque aparezca un warning a mitad de la instalación, hemos comprobado que es seguro, por lo que se puede pulsar el botón *Install anyway*.

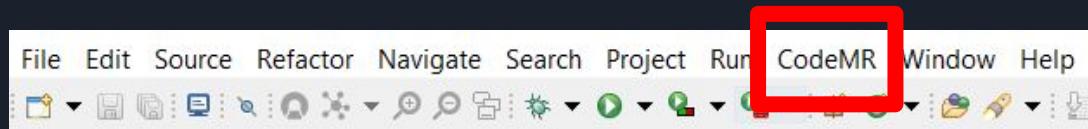


7. Despues se debe reiniciar eclipse para aplicar la actualización del software.



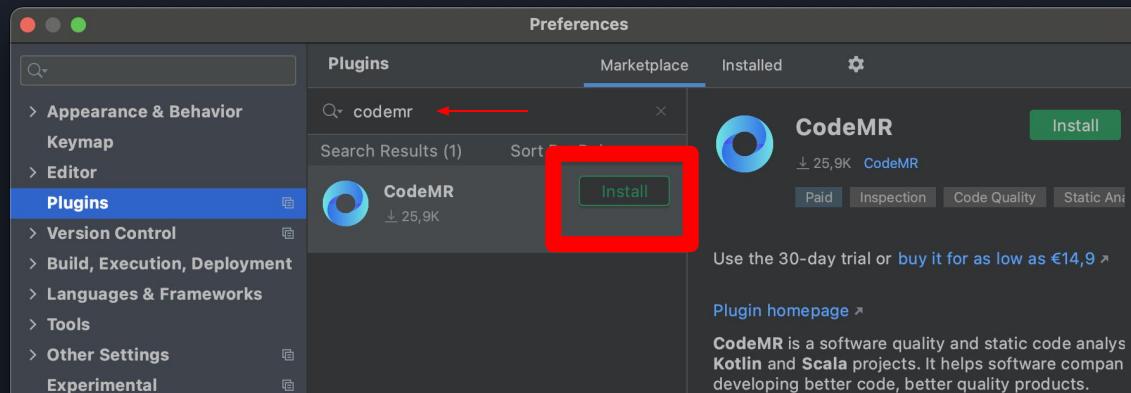
2. ¿Cómo se instala CodeMR en Eclipse?

8. Tras reiniciar, aparece el apartado CodeMR en la barra superior de eclipse



3. ¿Cómo se instala CodeMR en AndroidStudio?

1. Ir a las preferencias de Plugins de AndroidStudio.
2. Buscar “CodeMR” en el MarketPlace e instalarlo.



3. Tras reiniciar el Android Studio se selecciona la opción de prueba gratuita de 30 días.



4. ¿Por qué es importante CodeMR?

- **Mide las métricas de calidad** del código:
 - Reduce el tiempo en revisar código.
- Permite **visualizar el software en una sola imagen**:
 - Facilita la integración de nuevos integrantes en el equipo de desarrollo.
 - Asegura conocimiento de la calidad del software.
 - Permite comparar herramientas o librerías parecidas fácilmente.
- Permite **extraer módulos y rediseñar la estructura** del paquete.
 - Permite que el equipo esté en constante auto-superación.
 - Permite comprobar el producto en cada sprint (en el momento que tu elijas).

Además:

Compatibilidad con diversos lenguajes y plataformas:





5. Información que proporciona CodeMR

- **Model Editor**
- **Graph Editor**
- **Quality Attributes**
- **Module Extraction Features**
- **Report Generation Features**

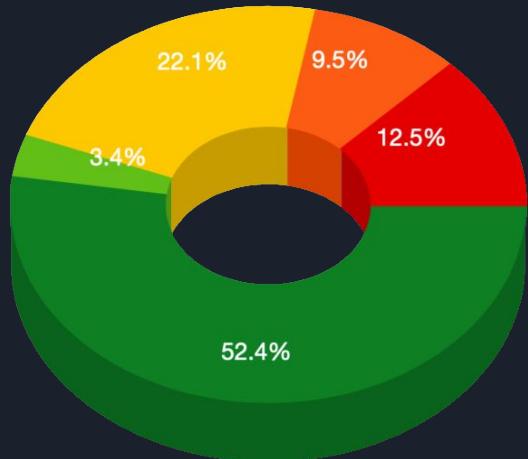


Model Editor

Nos proporciona diversos tipos de gráficos

- Metric Distribution
- Package Structure
- Sunburst
- Package Dependencies
- TreeMap
- Project Outline

Model Editor: Metric Distribution



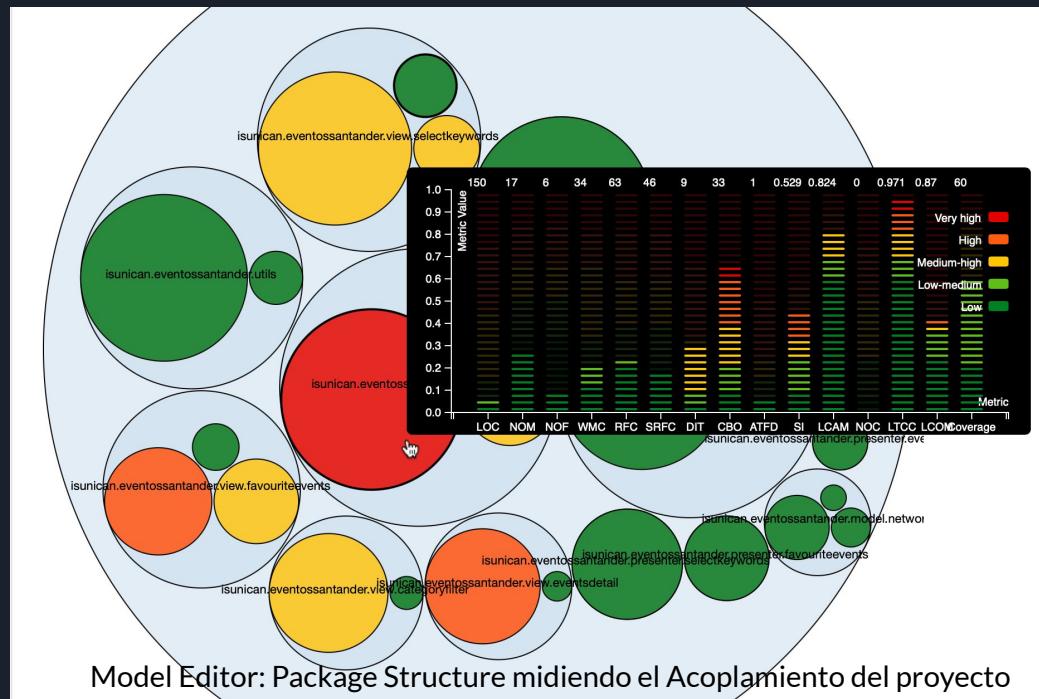
Distribución de la métrica seleccionada por línea de código.

Hay 5 niveles (bajo, bajo-medio, medio, medio-alto, alto), que corresponden con los colores (verde oscuro, verde claro, amarillo, naranja y rojo), respectivamente

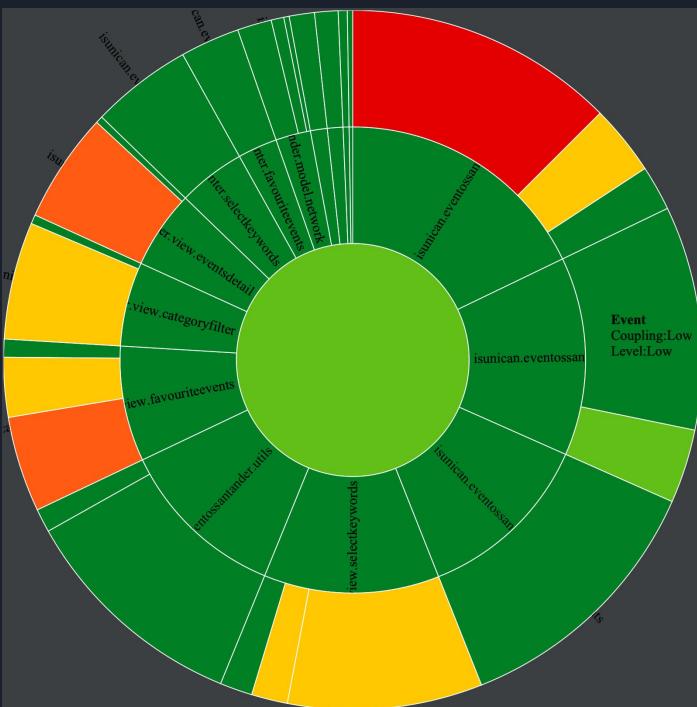
Model Editor: Package Structure

La estructura de paquetes del Model Editor se muestra como una vista en burbujas para así tener una visión global del proyecto.

Hay tres niveles de arbol: Proyecto, Paquete y Clase.

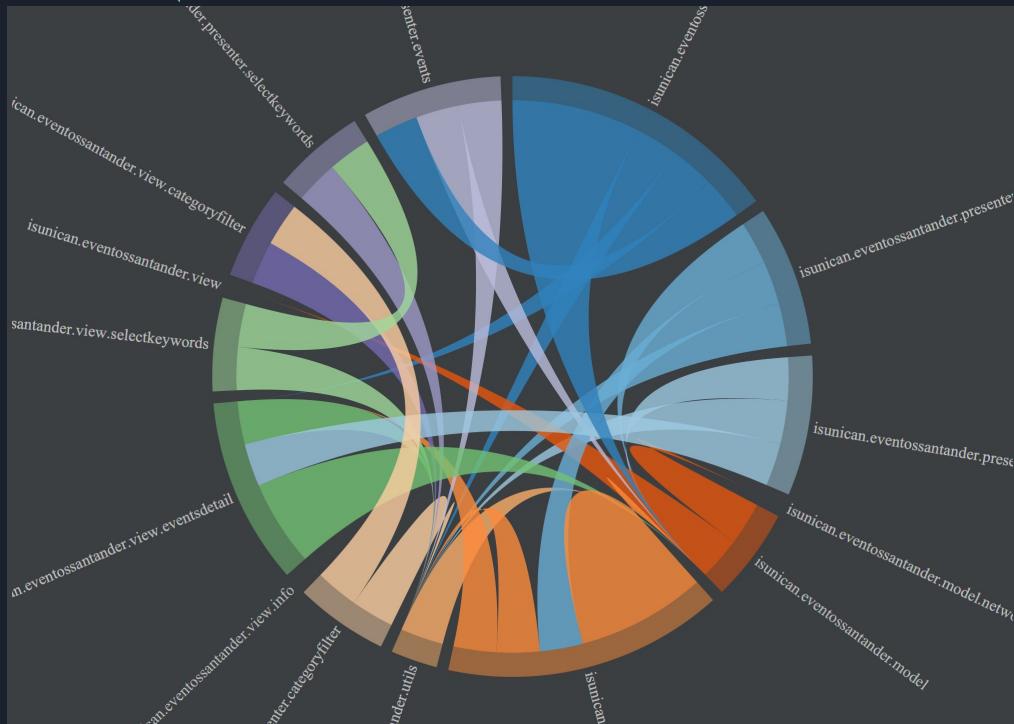


Model Editor: Sunburst



Las métricas de proyecto, paquete y clase se muestran en una vista que representa la jerarquía de paquetes de la aplicación con sus clases.

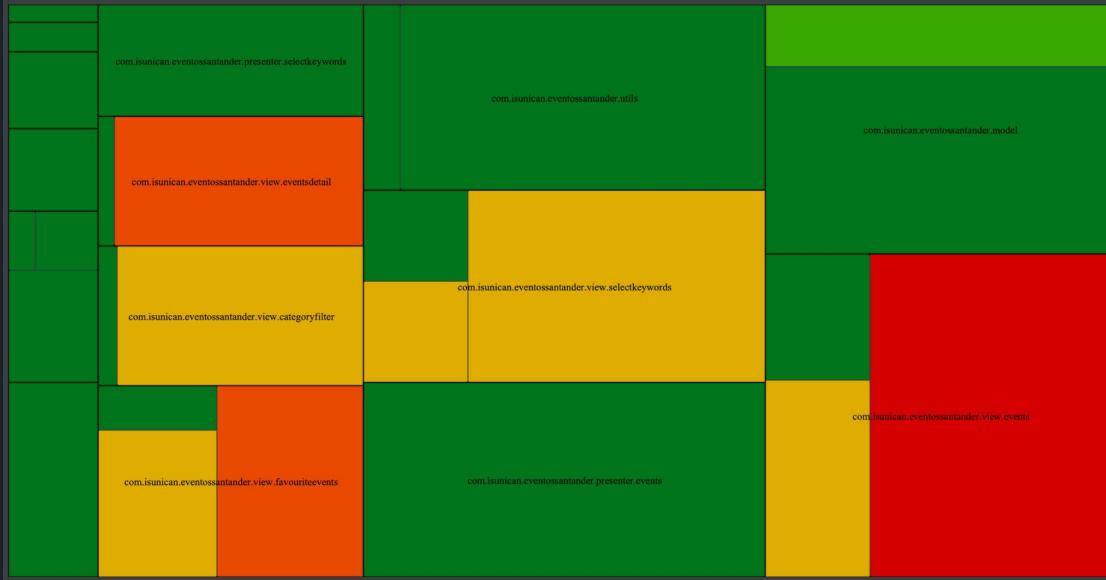
Model Editor: Package Dependency



Permite ver de manera sencilla las dependencias entre paquetes con esta vista.

Se puede seleccionar un paquete para ver las dependencias del que estés interesado.

Model Editor: TreeMap



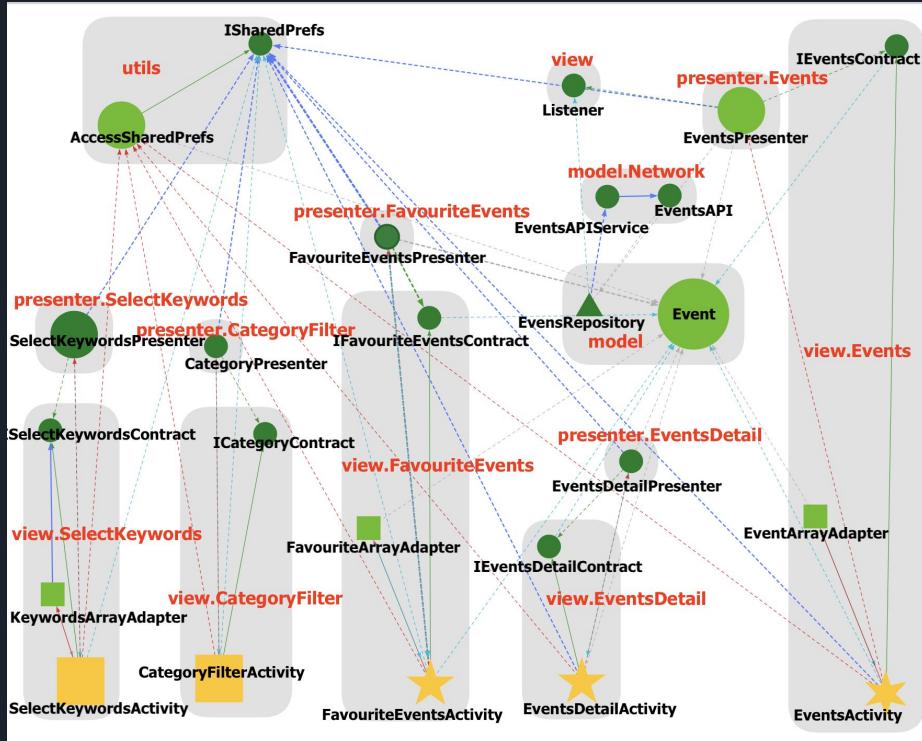
Esta vista da la oportunidad de investigar proyectos grandes de un vistazo. Se muestran los paquetes y su nivel de calidad.

Model Editor: Project Outline

ID	CLASS	COUPLING	COMPLEXITY	LACK OF COHESION	SIZE	LOC	COMPLEXITY	COUPLING	LACK OF COHESION	SIZE
1	EventsActivity	██████	██████	██████	██████	150	medium-high	very-high	high	low-medium
2	EventsDetailActivity	██████	██████	██████	██████	61	medium-high	high	low	low-medium
3	FavouriteEventsAc...	██████	██████	██████	██████	53	medium-high	high	low-medium	low-medium
4	SelectKeywordsAct...	██████	██████	██████	██████	108	medium-high	medium-high	low-medium	low-medium
5	CategoryFilterAct...	██████	██████	██████	██████	65	medium-high	medium-high	low	low-medium
6	EventArrayAdapter	██████	██████	██████	██████	39	low-medium	medium-high	low	low
7	FavouriteEventArr...	██████	██████	██████	██████	33	low-medium	medium-high	low	low
8	KeywordsArrayAdapter	██████	██████	██████	██████	20	low-medium	medium-high	low	low
9	EventsRepository	██████	██████	██████	██████	41	low	low-medium	low	low
10	InfoActivity	██████	██████	██████	██████	5	medium-high	low	low	low
11	EventsPresenter	██████	██████	██████	██████	148	low-medium	low	medium-high	low-medium

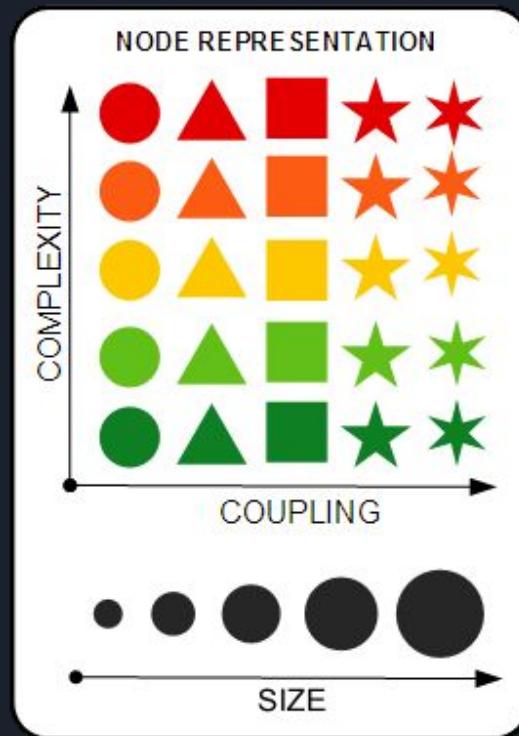
Disponemos de una vista avanzada en la que se proporciona toda la información de todas las clases de todos los paquetes, con sus métricas y niveles de calidad

Graph Editor



CodeMR extrae automáticamente métricas de calidad y relaciones de clases del código fuente y las visualiza en un entorno interactivo basado en gráficos

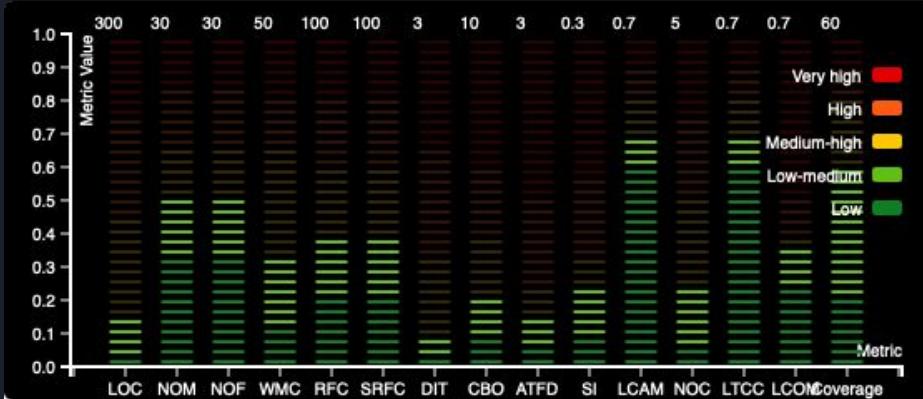
Quality Attributes



Las propiedades visuales de 1 nodo en 1 gráfico son su color, tamaño y patrón de forma.

- Color: complejidad de la entidad
- Forma: acoplamiento
- Tamaño: tamaño de la clase

Quality Attributes



La calidad externa del software muestra los síntomas visibles cuando hay problemas, pero esos problemas son derivados de atributos de calidad internos como el acoplamiento, la complejidad, la cohesión y el tamaño.



Métricas

- Tamaño
- Complejidad
- Acoplamiento



Métricas - Tamaño

- **CLOC**: número de líneas no vacías y sin comentarios
- **NOF**: número de atributos en la clase
- **NOM**: número de métodos en la clase
- **NOSF**: número de atributos estáticos
- **NOSM**: número de métodos estáticos
- **CM-LOC**: número de líneas de métodos no vacías y sin comentarios
- **Nol**: número total de interfaces



Métricas - Tamaño

- NoCIs: número total de clases
- NoE: número de entidades (interfaces + clases)
- NORM: número de métodos anulados
- NofP: número de paquetes en el proyecto
- NofPa: número de paquetes externos a los que hace referencia el proyecto
- NofEE: número de entidades externas a las que hace referencia el proyecto



Métricas - Complejidad

- **WMC**: suma de las complejidades de todos los métodos de la clase. Estima el esfuerzo necesario que se necesita para el desarrollo, mantenimiento y realización de pruebas una clase.
- **DIT**: Indica la posición de la clase en el árbol de herencia
- **RFC**: Número de métodos que pueden ser invocados externamente.
- **SI**: mide el grado en el que las subclases implementan las clases de sus clases 'ancestras'.



Métricas - Acoplamiento

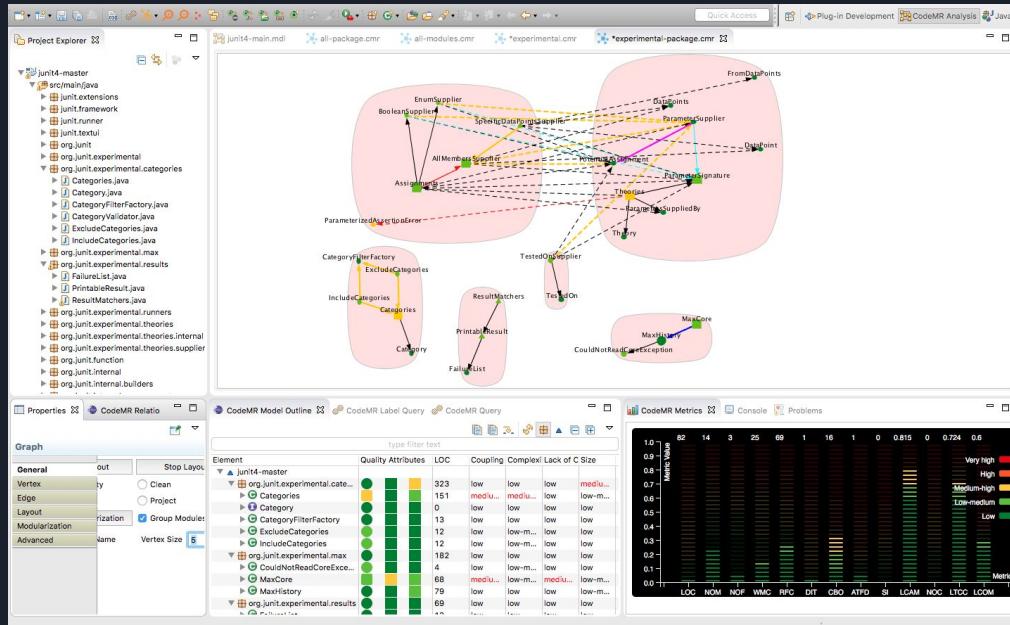
- **NOC**: número de subclases directas de una clase.
- **CBO**: número de clases a las que una clase está acoplada.
- **CBO Lib**: número de clases dependientes de una librería.
- **CBO App**: número de clases dependientes en la aplicación.
- **EC**: (acoplamiento saliente). número de clases en otros paquetes en las que las clases de ese paquete depende. (Dependencia)
- **AC**: (acoplamiento entrante). número de clases en otros paquetes que dependen de clases del mismo paquete. (Responsabilidad)



Métricas - Cohesión

- **LCOM**: cómo los métodos de una clase están relacionados entre sí.
- **LCAM**: nivel de ausencia de cohesión entre métodos.
- **LTCC**: nivel de ausencia de cohesión estrecha entre métodos públicos de una clase.

Module Extraction

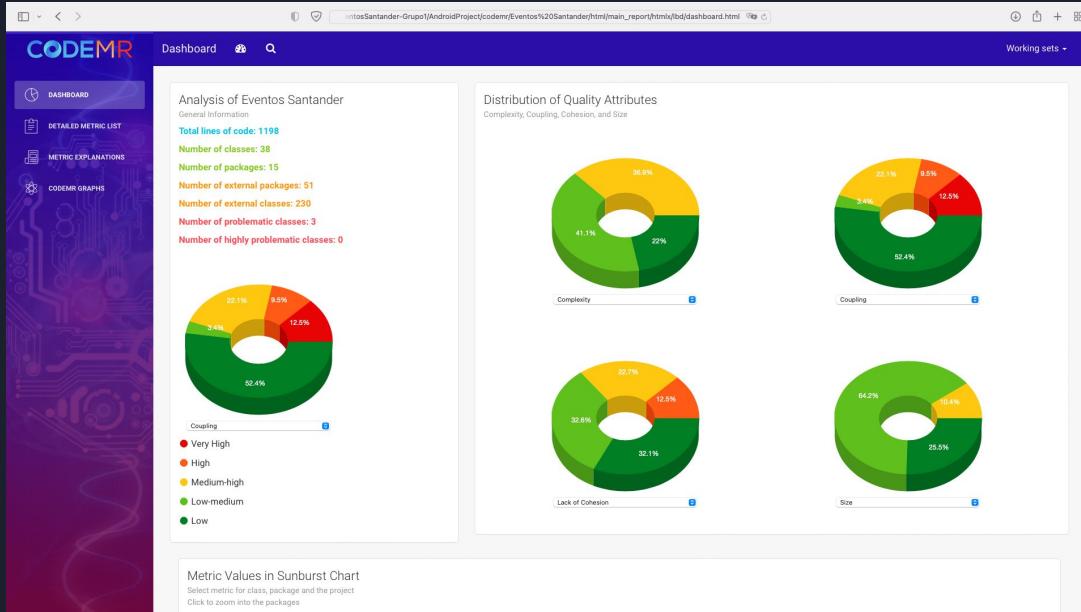


Permite incrementar la estructura de paquetes de dominio y migrarla a otras arquitecturas de microservicios desde una aplicación monolítica.

Permite subdividir el proyecto en subsistemas definiendo sets de trabajo y analizarlos por separado.

Permite definir todo esto desde una interfaz de usuario o simplemente editando ficheros JSON.

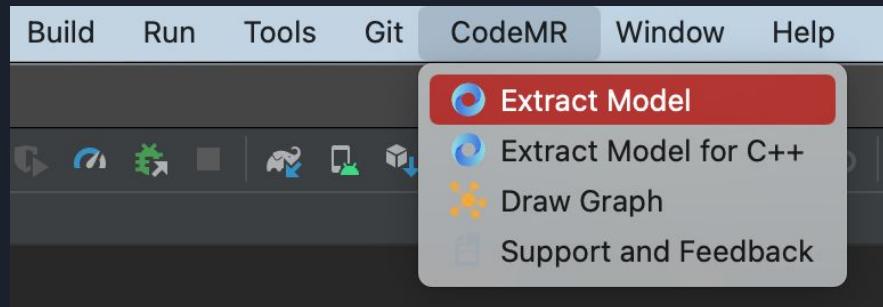
Report Generation Features



CodeMR nos permite crear un informe en formato *html* en el que se nos muestran todos los datos y las métricas explicadas anteriormente.

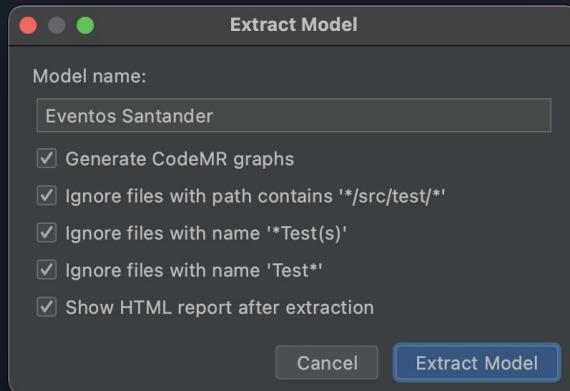
¿Cómo usar CodeMR en Android Studio?

1. Debemos tener seleccionada la carpeta raíz del proyecto en la vista de la estructura del proyecto
2. En el menú superior, en la nueva entrada de CodeMR tenemos que clicar en el apartado “Extract Model”



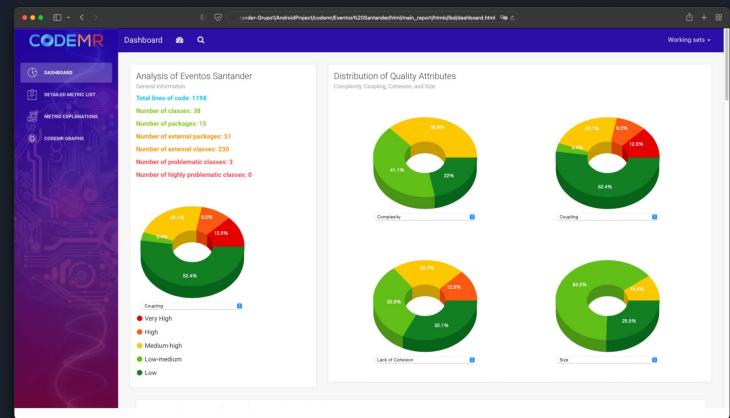
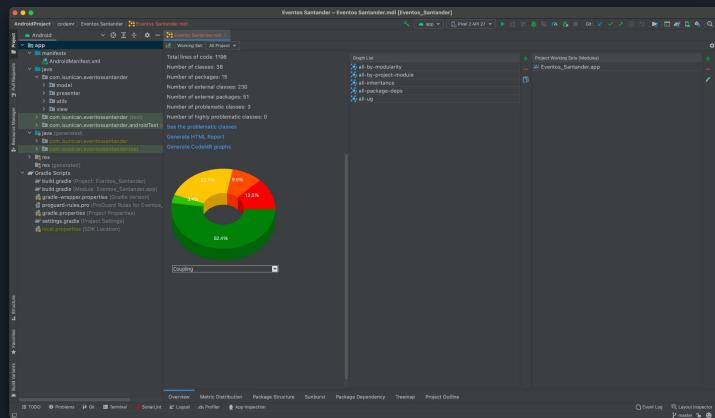
¿Cómo usar CodeMR en Android Studio?

3. A continuación nos muestra una ventana en la que seleccionar el nombre que le queremos dar al proyecto y la opciones que queremos usar.



¿Cómo usar CodeMR en Android Studio?

4. Tras esto la herramienta analiza el código y nos abre el proyecto de CodeMR en Android Studio. También nos abre un informe en el navegador web.



CODEMR

Muchas gracias por su atención.



¿Preguntas?