

## INFORME SOLVER.

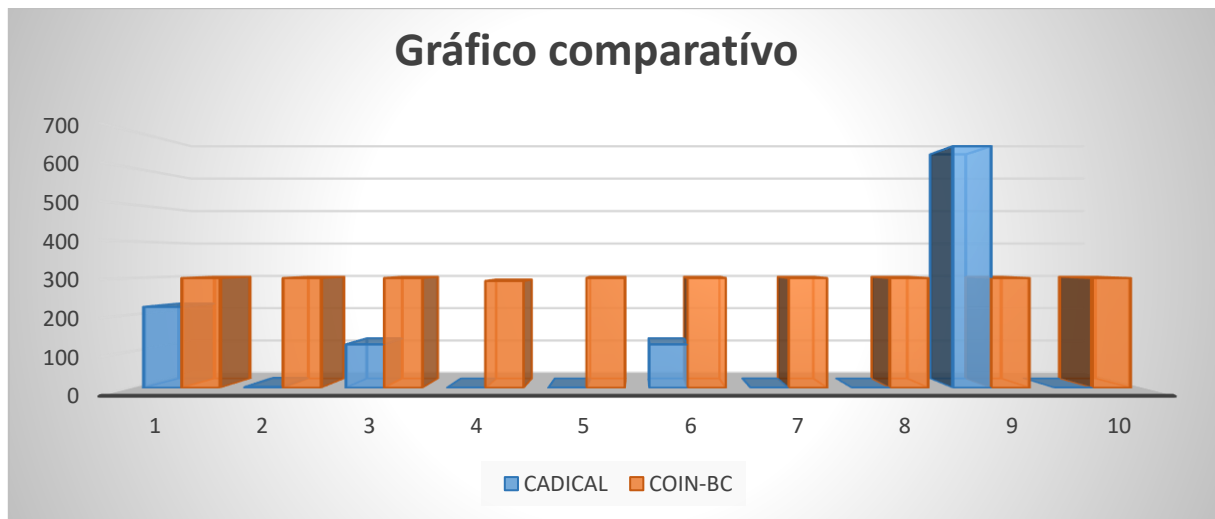
### CADICAL.

El Solver está implementado desde cero de forma modular en el lenguaje de programación C++. También existe una interfaz API para C, pero el core de la aplicación todavía no está listo para el uso incremental, ya que carece de manejo de presunciones. El código fuente está cuidadosamente documentado y consta de aproximadamente 10 000 líneas de código.

La parte de procesamiento consta de tres Métodos de procesamiento programados individualmente: sondeo, subsunción y eliminación (limitada) de variables. Durante (literal fallido) sondeando solo las raíces del sondeo binario gráfico de implicación y se aprenden cláusulas binarias a través de resolución hiper binaria. Estos se utilizan para eliminar literales equivalentes después de descomponer el gráfico de implicación binario en componentes fuertemente conectados por lo tanto solo sobrevivirán a lo sumo una reducción de una cláusula. También se eliminan explícitamente los duplicados de cláusulas binarias antes de sondear.

### COIN-OR CBC.

Es un Solver de programación lineal de enteros mixtos de código abierto escrito en C++. Puede usarse como una biblioteca invocable o mediante un ejecutable independiente. Se puede llamar a través de AIMMS. Cbc está escrito en C++ y se publica como código fuente abierto bajo la Licencia Pública de Eclipse (EPL) y se puede redistribuir libremente.



El Solver de Minizinc solo logró resolver una instancia en el tiempo límite de 5 minutos. El Solver SAT, CADICAL si logró resolver todas las instancias en los tiempos indicados en el gráfico, cabe resaltar que debido que la compilación de los solvers SAT se realizó en Linux se usó una máquina virtual con recursos limitados lo que puede haber influido en el tiempo de solución de las instancias.

Se puede agregar desde ésta perspectiva personal de implementación, que los Solvers SAT de la SATcompetition son Solvers especializados para combinaciones altamente difíciles de resolver, mientras que los Solvers de Minizinc son más de un propósito general de combinaciones de variables más cortas y el número restricciones mucho menor, por lo cual puede que su tiempo de respuesta sea mucho más alto, además de que el tiempo de respuesta también se ve afectado por poder de computo de la maquina en donde se esté ejecutando Minizinc ya que a menos poder de computo más tiempo se requiere para encontrar una respuesta