

## Informe de rendimiento

### RESUME

Para el informe de rendimiento realizamos 40 ejecuciones del solver entre las que veinte eran instancias X-SAT y otras 20 eran las reducciones de estas instancias en 5-SAT para asegurarnos que en la ejecución habrían instancias satisfactorias y no satisfactorias descargamos archivos que especificaran el resultado , en este proceso pudimos afirmar que las transformaciones están bien reducidas pues cada instancia satisfactoria resultaba en una reducción satisfactoria y en cada instancia no satisfactoria su resultado también fue no satisfactorio Lastimosamente para cuestiones de estudio en rendimiento los tiempos son en promedio iguales a continuación presentaremos la descripción del solver y los resultados.

### DESCRIPCION DEL SOLVER

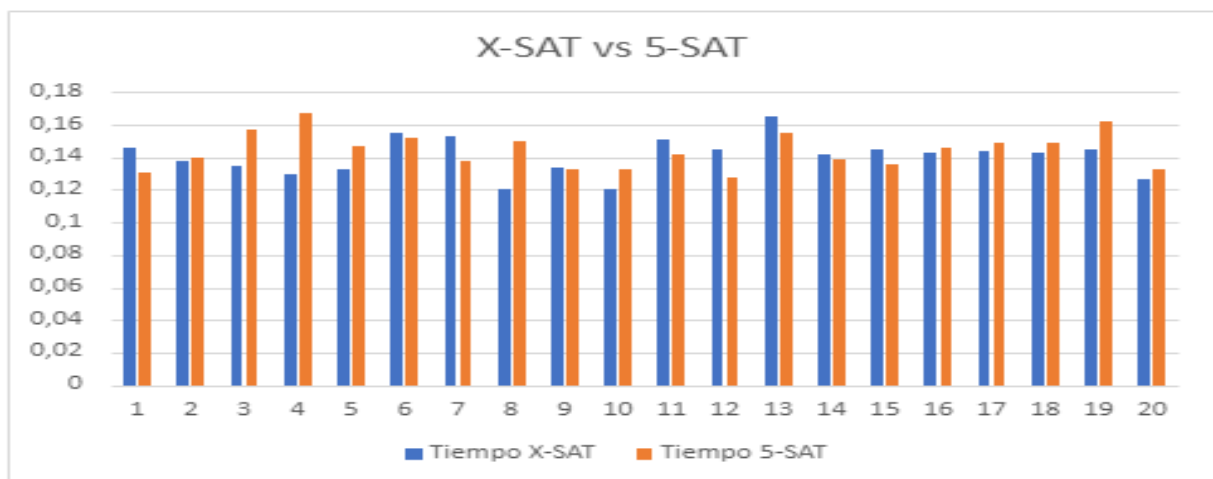
El solver usado en nuestro informe se llama Candy, es uno de los solvers ganadores para resolver problemas SAT en el 2017 sacado de la siguiente

pagina : <https://baldur.iti.kit.edu/sat-competition-2017/solvers/>

Es el primero de los de la lista, una vez construimos el ejecutable en la consola el solver espera un archivo o la dirección de un archivo que contenga un problema SAT en formato dimacs y al ejecutarlo lo que hace es intentar hallar la solución posible del problema reemplazando en las cláusulas todas las variables por true y false.

La salida del solver Da mucha información que realmente no logramos entender, así que solo sacamos al final el tiempo que se demoró el proceso en ejecución y si es satisfactible o no.

### X-SAT vs 5-SAT



### CONCLUSIONES

Los problemas iniciales y sus versiones reducidas fueron muy sencillos de resolver por el solver , debido a eso los tiempos de ejecución son muy parecidos en ambos y las diferencias no logran apreciarse , cogiendo algunos casos particulares como el del problema 4 me parece muy curioso que aunque son décimas de segundo la reducción demora mucho más con respecto a la inicial y pasan casos similares con el 8 y el 19 las reducciones en estos casos aumentaron el tiempo considerablemente lo que me hace pensar que en mi algoritmo la manera en que yo reduzco es ineficiente y podría mejorar.