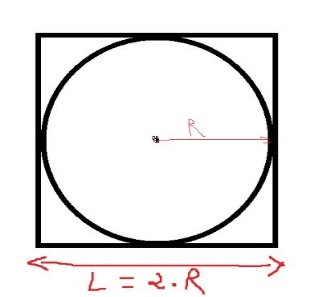
**Trabalho Final de Algoritmos**

# **INTRODUÇÃO**

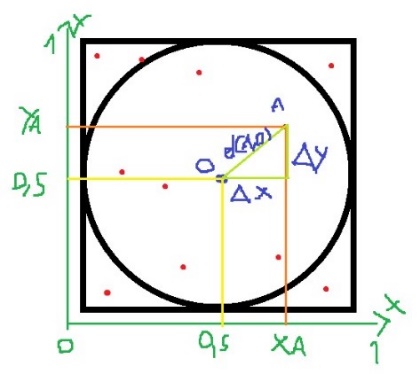
Nesse trabalho irei resolver o PI pelo método de aproximação proposto por Monte Carlo. O trabalho foi feito na linguagem C, e de todos os requisitos só não consegui utilizar matriz/ponteiro e alocação de memória.

# **DESENVOLVIMENTO**

Bom eu comecei usando a fórmula pra descobrir a área do círculo que no caso é

AC=, em seguida a área do quadrado (Essa equação é a mesma coisa que resolvendo fica ), para achar o Pi usei a fórmula eliminado os R ficou .

Depois eu usei a dispersão dos pontos, pra assim no código eu gerar posições aleatórias para isso eu utilizei do plano cartesiano, peguei entre o número 0 e o 1, pelo número 0,5 ser o número central, localizações dos pontos com números menores que 0,5 estarão dentro do círculo e maiores fora. No Plano Cartesiano as coordenadas das localizações dos pontos foram definidas por e que são as posições dos pontos. Usei o Teorema de Pitágoras foi utilizado para descobrir a hipotenusa, que é a distância da localização de um ponto (dentro do círculo) até o seu centro, a fórmula usada foi



Agora vêm a parte usada no código, comecei fazendo um registro, armazenando duas variáveis dentro para depois eu conseguir pegar essas duas variáveis e usar em qualquer função ou módulo, depois de declarada as variáveis eu criei o tipo de dados delas. Depois eu criei um módulo de ação para gerar pontos em posições aleatórias com duas coordenadas (X, Y), essas coordenadas eu transformei de float pra double para ter um valor mais preciso, o “rand” que eu coloquei foi pra gerar as posições aleatórias que eu havia falado, já o “RAND\_MAX” eu usei pra pegar o maior valor possível gerado pelo “rand”, e a divisão de rand/RAND\_MAX foi para gerar valores entre 0 e 1 (usei o 1 e o 0 pois eu fico com valores mais previsíveis, portanto mais fácil pra calcular). Em seguida eu criei outro módulo de ação para calcular a posição dos pontos, no caso o teorema de Pitágoras, usei a passagem de Parâmetros para pegar os valores das coordenadas X e Y já que nesse módulo como não havia a declaração de variáveis a equação não saberia que valores seriam essas duas coordenadas, em seguida usei um “if” para pegar os pontos que tivessem menos que 0,5 de distância, os pontos com menos de 0.5 ele retornava em 1 e “armazenava” eles, e os que retornavam em números maiores que 0.5 ele retornava em 0 e tirava do procedimento

Em seguida parti pro módulo e fiz o último módulo, que nesse caso era o módulo de procedimento comecei declarando as variáveis e iniciando elas, uma para limitar a quantidade de pontos (Nesse caso a variável pra limitar a quantidade de pontos eu coloquei 1078000 pra só pegar até essa quantidade de pontos), e outra para me dizer quantos pontos havia no círculo e uma outra variável contadora para ir contando e armazenando a quantidade de pontos que tem dentro do círculo e a última variável criada foi pra fazer o cálculo de PI. Iniciei um laço de repetição, no laço de repetição eu coloquei pra somar a quantidade de pontos dentro do círculo com a distância dos pontos (O módulo de ação que calcula a posição dos pontos) e ir contando e acumulando os pontos, depois eu calculei o PI dividindo número de pontos do círculo pelo número de pontos do quadrado e multiplicando esse resultado por 4 e por último eu pedi pro programa imprimir o valor de PI pra mim com quatro casas decimais.

**\*Obs: Professor, no vídeo de explicação eu falei que criei uma struct pra usar as variáveis em outros lugares, eu me confundi, eu criei a struct pra cria um novo tipo de dados.**