



Guia do Usuário Final

Rafael Nasser <rnasser@inf.puc-rio.br>

Karin Breitman <karin@inf.puc-rio.br>

Hélio Lopes <lopes@inf.puc-rio.br>

Índice

1. INTRODUÇÃO.....	3
2. CRIANDO SUA CONTA NO AZURE	4
3. CRIANDO SUA CONTA DE ARMAZENAMENTO	6
4. CRIANDO SEU PRIMEIRO SERVIÇO (SIMULAÇÕES EM C#).....	7
5. CONSUMINDO UM SERVIÇO	9
6. INSTALANDO PRÉ-REQUISITOS (SIMULAÇÕES EM R-PROJECT)	10
7. USANDO CÓDIGO COMPILADO (SIMULAÇÕES EM MATLAB)	11
ANEXO I.....	14

1. Introdução

Este guia é destinado a profissionais que desejam utilizar o poder da nuvem para executar suas simulações de Monte Carlo com alta precisão e baixo custo, realizando mínimas alterações em seus códigos.

A ideia essencial de computação na nuvem é que recursos computacionais, como processamento, armazenamento e banda, estão disponíveis sob demanda e você só paga pelo tempo utilizado. Criando a ilusão de escalabilidade quase infinita e uma oportunidade de economia sem precedentes.

Com o pacote "McCloudRelease.zip", disponível no menu "Downloads" do site oficial do McCloud (<http://mccloud.codeplex.com>), é muito fácil executar simulações escritas em C#, Matlab ou R-Project no Windows Azure (www.windowsazure.com.br), plataforma de computação na nuvem da Microsoft.

Nenhum conhecimento de computação na nuvem ou específico de programação é necessário, basta **seguir sequencialmente com atenção os passos apresentados nesse guia** para rapidamente desfrutar dos benefícios da nuvem em suas simulações.

O presente guia foi desenvolvido para ser seguido em uma máquina com sistema operacional Windows. Ao utilizar outros sistemas operacionais será necessário identificar ferramentas similares as recomendadas que suportem o ambiente em questão.

Recomendamos fortemente que na sua primeira leitura deste guia você utilize os códigos de simulação fornecidos na pasta "PiSamples" do pacote em questão, deixando para utilizar seus códigos de simulação após o completo entendimento do funcionamento do McCloud.

O código de exemplo fornecido nesta pasta, refere-se a um problema matemático clássico, a aproximação do valor da constante π , obviamente, utilizando Monte Carlo. O racional deste exemplo é apresentado em profundidade no ANEXO I do presente guia.

Boa sorte!

***Observação:** É muito importante para nossa pesquisa saber em quais cenários o McCloud vem ajudando e receber sugestões de melhorados. Dessa forma, constamos com seu retorno via e-mail ou pelo site oficial da solução.*

2. Criando sua Conta no Azure

Você precisará de um celular para receber um código de verificação por mensagem de texto e de um cartão de crédito Visa ou Mastercard para comprovação da sua identidade, mas fique tranquilo, a Microsoft informa que não existe nenhuma obrigação de compra e que nenhuma cobrança é realizada, a menos que você explicitamente remova o limite de gastos, ou seja, passe para uma conta paga. Siga os passos a seguir.

- 1) Acesse o portal do Azure (<https://www.windowsazure.com/pt-br/>);
- 2) Caso necessário, o idioma pode ser alterado no rodapé do site;
- 3) Clique em “ENTRAR” na lateral superior direita da página inicial;
- 4) Uma janela se abrirá sobre o portal (cuidado com o bloqueador de *pop-ups*);
- 5) Clique em “Inscreva-se” no rodapé desta janela de identificação;
- 6) O formulário de criação de um “Windows Live ID” se abrirá;
- 7) Preencha corretamente a ficha de cadastro exibida e salve;
- 8) Não esqueça de memorizar ou anotar o e-mail e senha informados;
- 9) Aguarde o recebimento do e-mail de confirmação do seu endereço de e-mail;
- 10) Quando o e-mail chegar, clique no link de confirmação da sua conta;
- 11) Possivelmente será solicitada a confirmação do seu e-mail e senha;
- 12) A mensagem “Seu endereço de e-mail foi verificado” será exibida;
- 13) Retorne ao portal do Azure e clique novamente em “ENTRAR”
- 14) Desta vez, identifique-se informando o e-mail e senha escolhidos;
- 15) Clique em “gerenciar” no canto superior direito do portal;
- 16) Clique em “Usar o Novo Portal” no botão verde, se aparecer essa pergunta;
- 17) Clique em “Inscrever-se agora mesmo” em um dos botões verdes;
- 18) Clique em “avaliação gratuita” no canto direito do menu do topo em cinza;
- 19) Clique em “testar gratuitamente” no botão verde;
- 20) Escolha a opção “Brasil” na caixa de seleção e clique na seta para avançar;
- 21) Informe seu celular e clique em “Enviar mensagem de texto”;
- 22) Digite o código recebido via SMS no seu celular;
- 23) Clique em “Verificar código” e depois na seta para avançar;
- 24) Preencha os seus dados pessoais e do seu cartão de crédito, e confirme;
- 25) Sua assinatura do Windows Azure será configurada;

- 26) Pronto, clique novamente em “Gerenciar” que a plataforma está disponível;
- 27) Quando precisar acessá-la se identifique no portal e clique em “gerenciar”.

Cabe observar que em vez da opção “avaliação gratuita”, esses mesmos passos podem ser seguidos escolhendo a opção “Pagamento pelo uso > comprar” no passo 18. No entanto, enquanto esse benefício de avaliar gratuitamente estiver disponível, recomendamos fazer uso para conhecer a plataforma, de forma a optar pela versão paga já conhecendo o serviço. No site do Windows Azure você pode consultar o custo da hora de computação, de armazenamento, transações de armazenamento e transferência de dados. Esses são os valores que balizam o custo da simulação.

Se você quiser acompanhar o uso dos recursos contratados, clique em “Cobrança” no canto superior da plataforma. É importante observar que a conta de armazenamento é computada pelo máximo de volume de dados armazenados no mês (*GB/mês*), pelo volume de transferência de dados para fora da plataforma (*download em GB*) e pelo total de transações internas (*em dezenas de milhares*).

Atualmente a versão gratuita contempla:

- 750 horas de computação pequena;
- 20 GB de armazenamento com 1.000.000 de transações;
- 20 GB de transferências de dados de saída (entrada é ilimitada);
- 1 GB de banco de dados (não utilizado pelo McCloud).

Atualmente os custos desses serviços são:

- USD\$0,12 por hora ou fração do relógio de computação da instância padrão*;
- USD \$0,125 por GB armazenado por mês com base na média diária;
- USD \$0,01 por 10.000 transações de armazenamento;
- USD\$0,12 por GB transferido para fora do Azure.

** Uma instância padrão contempla 1 núcleo, 1,75GB de memória, 165GB de armazenamento local e 100Mbps de largura de banda. Instâncias maiores são múltiplos desta instância.*

3. Criando sua Conta de Armazenamento

Com a conta da plataforma criada, você precisa criar uma conta de armazenamento dentro da plataforma. Siga os passos a seguir depois de acessar a plataforma.

- 1) Clique em “Serviços Hospedados, Contas de Armazenamento e CDN”;
- 2) Também na esquerda, acima desta opção, ao clicar, abrirá seus subitens;
- 3) Clique em “Contas de Armazenamento”;
- 4) Clique em “Nova Conta de Armazenamento” no menu do topo;
- 5) Escolha um nome para sua conta dentro dos padrões permitidos;
- 6) Este nome fará parte da URL do armazenamento (*nome.core.windows.net*);
- 7) Escolha também a região onde o seu serviço ficará;
- 8) Por exemplo, “Oeste dos Estados Unidos”, pela proximidade;
- 9) Confirme e aguarde a conta ser criada (Status = Criado);
- 10) Clique com o botão direito sobre a conta de armazenamento;
- 11) Escolha a opção “Exibir Chaves de Acesso”;
- 12) Tome nota da chave primária, junto ao nome e URL da conta criada;
- 13) Acesse <http://azurestorageexplorer.codeplex.com/>;
- 14) Na aba “Downloads”, baixe o arquivo “.msi” mais recente desta ferramenta;
- 15) Instale em seu computador essa ferramenta na pasta desejada;
- 16) Abra a ferramenta e clique no botão “Add Account” dentro da mesma;
- 17) Digite o nome e chave da conta criada e clique “Add Storage Account”;
- 18) Nos botões da direita escolha visualizar *Blobs*, *Queues* ou *Tables*;
- 19) Pronto, você já possui uma conta de armazenamento na nuvem funcionando;

Com essa ferramenta você pode visualizar a tabela “log” e a tabela “mcc”, para acompanhar o que está acontecendo na nuvem. A primeira apresenta o registro das atividades do McCloud e a segunda o registro das simulações solicitadas.

Você também pode com essa ferramenta visualizar as pastas (containers) das simulações realizadas e, se desejar, apagá-las, quando as mesmas não forem mais necessárias. Cada simulação possui exatamente uma pasta com o nome igual a sua chave de identificação com o prefixo “mcc”.

Se você não é um especialista em nuvem, não utilize essa ferramenta para outra finalidade!

4. Criando seu Primeiro Serviço (Simulações em C#)

Sugerimos que o primeiro serviço que você crie seja o que permite simulações na linguagem C# em formato textual. Para isso, siga os passos a seguir.

- 1) Acesse <http://mccloud.codeplex.com>;
- 2) Na aba “Downloads” salve o arquivo “McCloudRelease.zip”;
- 3) Salve o conteúdo deste arquivo na pasta de sua preferência;
- 4) Dentro desta acesse a pasta “McCloudWorker”;
- 5) Abra o arquivo “ServiceConfiguration.Cloud.cscfg” desta pasta;
- 6) Você pode utilizar qualquer editor de texto, por exemplo, o bloco de notas;
- 7) Substitua “MyAccountName” pelo nome da conta anotado na seção anterior;
- 8) Essa informação aparece quatro vezes, sendo três dentro de URLs;
- 9) Substitua “MyAccountSharedKey” pela chave anotada na seção anterior;
- 10) Substitua “MyW” pela quantidade de instâncias (nós) desejadas;
- 11) Em geral, sem previa autorização, esse número não pode ultrapassar 19;
- 12) Substitua “MySimulation” por “CSharp” para identificar a implementação;
- 13) Salve este arquivo de configuração;
- 14) Acesse agora a pasta “McCloudService”;
- 15) Repita os passos de 5 a 13 no arquivo desta pasta, mas o “MyW” deve ser 1;
- 16) Acesse a plataforma na parte de gerenciamento;
- 17) Clique em “Serviços Hospedados, Contas de Armazenamento e CDN”;
- 18) Também na esquerda, acima desta opção, ao clicar, abrirá seus subitens;
- 19) Clique em “Serviços Hospedados”;
- 20) Clique em “Nova Serviço Hospedado” no menu do topo;
- 21) Um formulário de criação de serviço será exibido;
- 22) Primeiramente criaremos o serviço que está na pasta “McCloudWorker”;
- 23) Dê o nome da pasta ao serviço (“McCloudWorker”);
- 24) Escolha uma URL disponível para o serviço (ex. “mccloudworker100”);
- 25) Escolha a mesma região que escolheu para a conta de armazenamento;
- 26) Recomendamos “Em qualquer lugar dos Estados Unidos”;
- 27) Escolha a opção “Implantar em produção”;
- 28) Dê um nome qualquer a implementação (ex. “versao1”);

- 29) Localize pacote “McCloudWorker.cspkg” dentro da pasta e selecione;
- 30) Localize o arquivo “ServiceConfiguration.Cloud.cscfg” desta mesma pasta;
- 31) Confirme (“OK”) que o serviço começará a ser criado;
- 32) Repita os passos de 19 a 31 para o serviço da pasta “McCloudService”;
- 33) Não se preocupe com o alerta de uso de uma única instância;
- 34) Anote o nome que utilizou no endereço deste serviço (“McCloudService”);
- 35) Acompanhe a criação das instâncias até que todas fiquem “Pronto”;
- 36) Provavelmente leve entorno de 7 minutos para que todas estejam prontas;
- 37) Depois disso, seu serviço estará disponível no endereço anotado.

Atenção, os serviços são cobrados enquanto estiverem criados (horas de computação = horas de serviço criado), portanto, se não for utilizá-los, remova-os e posteriormente crie novamente a partir dos arquivos alterados, seguindo as instruções acima a partir do passo 16.

5. Consumindo um Serviço

Nessa seção demonstraremos uma aplicação para consumir o serviço criado na seção anterior e que poderá ser utilizada para consumir todos os serviços apresentados nesse guia. Siga os passos a seguir.

- 1) Abra a pasta que descompactou o conteúdo de “McCloudRelease.zip”;
- 2) Abra a pasta “McCloudClientExec”;
- 3) Abra o arquivo “McCloudClient.exe.config”;
- 4) Você pode utilizar qualquer editor de texto, por exemplo, o bloco de notas;
- 5) Substitua “MyMcCloudService” pelo endereço do serviço criado;
- 6) A estrutura do endereço deve ficar: <http://nome.cloudapp.net/Service.svc>;
- 7) Salve o arquivo e retorne para a pasta;
- 8) Os arquivos “codein.txt” e “codeout.txt” representam a simulação;
- 9) Esses devem ser escritos na linguagem C#, no editor de sua preferência;
- 10) Como exemplo disponibilizamos o código da aproximação do π ;
- 11) O racional deste exemplo está disponível no ANEXO I;
- 12) Copie o “codein.txt” e “codeout.txt” da pasta “PiSample> CSharp” para esta;
- 13) É fortemente recomendado experimentar esse exemplo antes de fazer outros;
- 14) Se você não possui, instale o “.Net Framework 4”;
- 15) Disponível em: <http://www.microsoft.com/download/en/details.aspx?id=17851>;
- 16) Abra o arquivo “McCloudClient.exe”;
- 17) Informe a quantidade de realizações, por exemplo, “1000”;
- 18) Informe o máximo de realizações por tarefas, por exemplo, “100”;
- 19) Aguarde a execução terminar e acompanhe em tela as evoluções;
- 20) Ao final um arquivo com o resultado será criado nessa mesma página;
- 21) Pronto, você conseguiu fazer uso da nuvem para realizar sua simulação.

Atenção, uma tarefa não pode demorar mais de 2 horas para ser executada, portanto, coloque um número de realizações por tarefa que seja sempre exequível nesse tempo. Além disso, informações adicionais da execução estarão disponíveis na conta de armazenamento, como, por exemplo, a instrumentação na tabela “log”.

6. Instalando Pré-requisitos (Simulações em R-Project)

Nessa seção criaremos outro serviço, dessa vez para permitir simulações na linguagem R-Project em formato textual. Este serviço demanda a instalação do programa R em cada instância para sua adequada execução. Dessa forma, ensinaremos como instalar este pré-requisito para funcionamento deste serviço. Siga os passos a seguir.

- 1) Faça download do instalador do R para Windows em <http://www.r-project.com>;
- 2) Renomeie o arquivo para “R.exe”;
- 3) Acesse a sua conta de armazenamento com o Azure Explorer*;
- 4) Selecione o ícone “Blobs” no canto superior direito;
- 5) Clique no ícone “New” no canto superior esquerdo;
- 6) Dê o nome de “startup” e acessibilidade “Public Blob”;
- 7) Clique “Create Container” para confirmar a criação da pasta;
- 8) Clique sobre a pasta criada;
- 9) Clique no ícone “Upload” e selecione o “R.exe” criado;
- 10) Esse carregamento pode demorar, pois o arquivo é grande;
- 11) Pronto, o McCloud vai instalar o R em cada instância ao iniciar o serviço;

Agora que os pré-requisitos foram atendidos, os passos para criar este serviço são os mesmos do primeiro serviço que criamos neste guia (seção 4), no entanto, substitua “MySimulation” por “RProject” para identificar a implementação em questão.

A aplicação cliente para esse serviço também é exatamente igual à utilizada no primeiro (seção 5), mas com o endereço deste novo serviço e com “codein.txt” e “codeout.txt” da pasta “PiSample >RProject”.

Cabe observar que o tempo para as instâncias ficarem prontas é maior, pois se faz necessário instalar os pré-requisitos, o que deve demorar em torno de 10 minutos.

** O Azure Explorer não é a melhor ferramenta para transferência de arquivos grandes. Uma sugestão disponível gratuitamente para teste é o CloudBerry Explorer for Azure Blob Store (<http://www.cloudberrylab.com>), cujo funcionamento é similar, mas exclusivo para esse tipo de armazenamento.*

7. Usando Código Compilado (Simulações em Matlab)

Nessa seção criaremos outro serviço, dessa vez para permitir simulações na linguagem MatLab em formato compilado. Em particular, este serviço demanda cuidados adicionais devido aos requisitos para que o código compilado execute adequadamente. Dessa forma, ensinaremos como instalar os pré-requisitos para funcionamento deste serviço e como enviar os códigos compilados. Siga os passos a seguir.

Parte I – Criando seus Executáveis MatLab

- 12) Abra a pasta que descompactou o conteúdo de “McCloudRelease.zip”;
- 13) Abra a pasta “PiSample >MatLab”;
- 14) Abra o MatLab e visualize os arquivos “process.m” e “merge.m”;
- 15) Como exemplo disponibilizamos o código da aproximação do π ;
- 16) O racional deste exemplo está disponível na seção ANEXO I;
- 17) Digite “deploytool” na linha de comando do MatLab;
- 18) Digite “process.prj” como nome do projeto;
- 19) Escolha a opção “Windows Standalone Application” no tipo;
- 20) Ao confirmar um painel se abrirá à direita;
- 21) Clique em “Add main file” nesse painel e escolha o arquivo “process.m”;
- 22) Clique no ícone “Build” acima do arquivo que foi adicionado;
- 23) Aguarde que o executável deste arquivo esteja pronto;
- 24) Este estará disponível na subpasta “process\distrib”;
- 25) Copie o “process.exe” para a pasta raiz que estamos usando;
- 26) Repita de 6 a 14 para o arquivo substituindo “process” por “merge”;
- 27) Copie também para essa pasta o “MCRInstaller.exe”;
- 28) Este arquivo fica dentro da sua instalação do MatLab no seu computador;
- 29) Informações: <http://www.mathworks.com/products/compiler/mcr/index.html>;
- 30) Faça também download para essa pasta do “vcredist_x86.exe”;
- 31) Disponível em <http://www.microsoft.com/download/en/details.aspx?id=5555>.

Obviamente que para criar seus executáveis personalizados você precisará do MatLab instalado em sua máquina. Além disso, fique atento, pois o MCRInstaller precisa ser na mesma versão do MatLab que você utilizou para compilar os executáveis, incluindo o fato de ser 32/64-bit. No entanto, sempre deve ser para Windows, pois é o sistema operacional do Azure.

Se você não possui MatLab instalado, pode seguir esse guia utilizando o “process.exe” e “merge.exe” disponibilizados na pasta em questão, no entanto, deve utilizar o “MCRInstaller.exe” na versão 7.1.5 para Windows 32-bit, versão utilizada para compilá-los. Pelo tamanho (~200MB) não podemos disponibilizá-lo nesta pasta do pacote em questão, mas você deve conseguir localizá-lo sem dificuldade na Internet.

Parte II – Carregando os Executáveis na Nuvem

- 1) Acesse a sua conta de armazenamento com o Azure Explorer;
- 2) Selecione o ícone “Blobs” no canto superior direito;
- 3) Clique no ícone “New” no canto superior esquerdo;
- 4) Dê o nome de “onstart” e acessibilidade “Private”;
- 5) Clique “Create Container” para confirmar a criação da pasta;
- 6) Clique sobre a pasta criada;
- 7) Clique no ícone “Upload” e selecione o “process.exe” criado;
- 8) Clique no ícone “Upload” e selecione o “merge.exe” criado;
- 9) Pronto, o McCloud vai copiar esses arquivos para cada instância;
- 10) Clique novamente no ícone “New” no canto superior esquerdo;
- 11) Dê o nome de “startup” e acessibilidade “Public Blob”;
- 12) Clique “Create Container” para confirmar a criação da pasta;
- 13) Clique sobre a pasta criada;
- 14) Clique no ícone “Upload” e selecione o “vcredist_x86.exe” criado;
- 15) Clique no ícone “Upload” e selecione o “MCRInstaller.exe” criado;
- 16) Esse carregamento pode demorar, pois os arquivos são grandes;
- 17) Pronto, o McCloud vai instalar esses arquivos em cada instância.

Agora que os pré-requisitos foram atendidos, os passos para criar este serviço são os mesmos do primeiro serviço que criamos neste guia (seção 4), no entanto, substitua “MySimulation” por “MatLab” para identificar a implementação em questão.

A aplicação cliente para esse serviço também é exatamente igual à utilizada no primeiro (seção 5), mas com o endereço deste novo serviço e com “codein.txt” e “codeout.txt” da pasta “PiSample >MatLab”. Cabe observar que o tempo para as instâncias ficarem prontas é muito maior, pois se faz necessário instalar os pré-requisitos e reiniciar as instâncias, o que deve demorar em torno de 20 minutos.

8. Desenvolvendo sua Própria Simulação

Finalmente você está apto a rodar suas próprias simulações na nuvem com o McCloud, no entanto, algumas dicas são fundamentais:

- 1) Conforme ilustrado no exemplo (ANEXO I), cujos códigos estão na pasta “PiSample”, as suas simulações devem ser divididas em duas partes, onde:
 - a. A primeira deve gerar e retornar um conjunto de amostras. Esta parte será, de forma transparente, executada diversas vezes, provavelmente em paralelo, pelo McCloud;
 - b. A segunda deve calcular o resultado da simulação. O McCloud agrega todas as amostras geradas em um único arquivo que é fornecido a essa parte.
- 2) No caso de simulações em R ou MATLAB, o protótipo da função de cada parte deve respeitar a utilizada nos exemplos.
- 3) Fique atento ao controle da semente dos geradores de números pseudo aleatórios, como feito nos exemplos, senão você corre o risco de gerar amostras a partir das mesmas sementes.
- 4) Retorne apenas as informações essenciais na primeira parte, para evitar uso desnecessário de armazenamento ou mesmo travamento do seu código. Por exemplo, se o interesse é a média das amostras, para cada execução retorne apenas a média e não todas as amostras, pois a partir da média é possível fazer a média das médias na segunda parte. Esse mesmo cuidado pode ser tomado com variância, histograma e outros comportamentos estatísticos que se deseje extrair.
- 5) Caso você precise fornecer outros arquivos a serem utilizados na sua simulação, você pode enviá-los para cada instância através do container “onstart”.

Obrigado pelo interesse no McCloud e boa sorte nas suas pesquisas!

ANEXO I

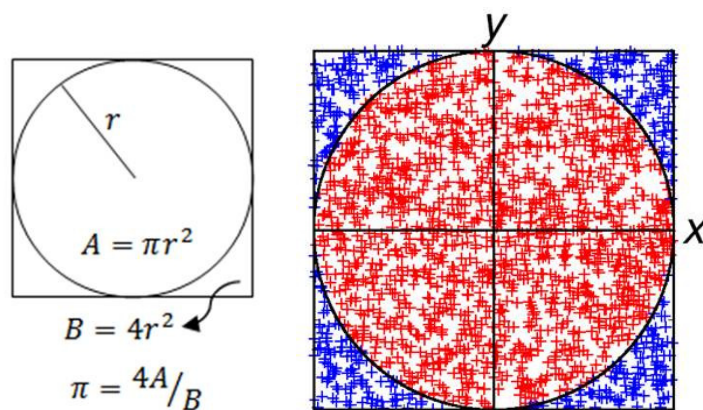
O racional do cálculo da aproximação do π , utilizando o Método de Monte Carlo, encontra-se detalhado abaixo e ilustrado na figura a seguir.

1) Imagine um círculo de raio r inscrito dentro de um quadrado de lado $2 \times r$, ou seja, de lado duas vezes o raio do círculo ou, simplesmente, o diâmetro do círculo;

2) A geometria básica nos ensina que a área do círculo, nomeada de A , é $A = \pi.r^2$, e a área do quadrado, nomeada de B , é $B = \text{lado}^2 = (2.r)^2 = 4.r^2$.

3) Se dividirmos as áreas do círculo pela área do quadrado encontramos a relação $A / B = \pi.r^2 / 4.r^2 = \pi / 4$, então, π pode ser expresso por $\pi = 4 A / B$.

4) Assim, se sortearmos pontos dentro da área do quadrado, este pode ou não estar dentro da área do círculo. Desta forma, sorteando uma quantidade suficientemente grande de pontos (amostra) obtemos a relação de pontos dentro do círculo e o total de pontos sorteados (obrigatoriamente dentro do quadrado), aproximando, assim, a relação A / B pelo Método de Monte Carlo e, consequentemente, o valor de π .



Os passos lógicos da simulação, considerando o racional descrito acima, são:

i. A incerteza envolvida é o ponto a ser sorteado, ou melhor, as duas coordenadas cartesianas x - y que definem esse ponto;

ii. Considerando o centro do quadrado como o ponto $(0,0)$, as coordenadas podem variar entre $-r$ e r , pois o lado do quadrado tem comprimento $2 \times r$ e estamos interessados apenas em pontos dentro da área do quadrado;

iii. Como estamos interessados em aproximar a relação (A / B) , podemos considerar, para efeito de simplificação, apenas um quarto do quadrado e consequentemente do círculo. Neste caso, escolhemos o quarto onde os eixos são positivos;

iv. Como a relação A / B se mantém independente da escolha do tamanho do quadrado, ou seja, o valor de r . Supondo que $r = 1$, a distribuição de probabilidade das coordenadas x e y são os números entre 0 e r , ou melhor, entre 0 e 1;

A geração de dois valores pseudo-aleatórios, aderentes a essa distribuição, representará o resultado obtido ao se sortear um ponto, ou seja, dois valores entre 0 e 1, que representaram x e y ;

i. O resultado determinístico corresponde a encontrar um resultado que indique se o ponto está ou não dentro do quadrado;

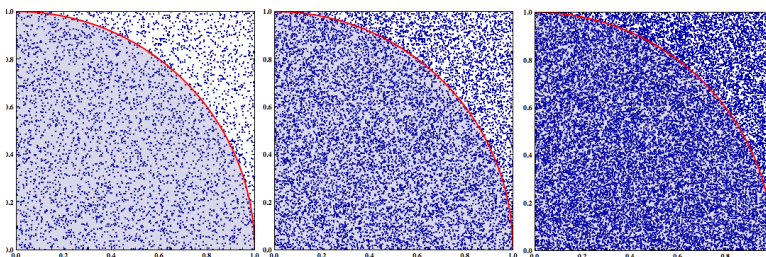
ii. Utilizando o Teorema de Pitágoras podemos extrair a relação $z = \sqrt{x^2 + y^2}$;

iii. Se o z é menor que r , i.e., o raio do círculo, no caso em questão, o número 1, o ponto está dentro da área do círculo e o resultado é positivo, caso contrário, o resultado é negativo.

Repetindo o passo B e C conseguiremos obter uma amostra suficientemente grande, ou seja, com n resultados positivos ou negativos.

Com base na amostra podemos contar quantas vezes ocorreu o Sim e dividindo pelo tamanho da amostra (n) aproximamos a relação A / B , que multiplicado por 4, resulta na aproximação de π .

Nesta figura abaixo ilustramos três diferentes momentos da geração de pontos para o quarto do quadrado, onde alguns se encontram dentro do círculo e outros não.



Note que quanto maior o número de pontos (amostra), mais próximo estaremos de cobrir “toda” a área e, conseqüentemente, de uma aproximação mais precisa. Evidente, portanto, que quanto maior o esforço computacional melhor a aproximação obtida.