

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE  
INSTITUTO METRÓPOLE DIGITAL  
ORGANIZAÇÃO E ARQUITETURA DE COMPUTADORES

PROJETO FINAL DA 3ª UNIDADE:  
JUMP'S PREDICTION - SIMULADOR DE PREDIÇÃO DE SALTOS

PROFESSOR:

GUSTAVO GIRÃO BARRETO DA SILVA

ALUNOS:

JOÃO VICTOR MARQUES DE OLIVEIRA

MATHEUS FELLIPE DA COSTA ANDRADE

NATAL, DEZEMBRO DE 2015

## **1. RESUMO**

A interdisciplinaridade é um aspecto de grande importância para uma formação acadêmica de nível conceituado. Em meio a aplicação da interdisciplinaridade entre a matéria de Organização e Arquitetura de Computadores e o conhecimento prático e teórico adquirido pela equipe durante a disciplina de Linguagem de Programação II e o curso técnico integrado em Informática pelo Instituto Federal de Ciência e Tecnologia do Rio Grande do Norte, surgiu o projeto denominado de “JUMP’S PREDICTION”. O projeto consiste no desenvolvimento de uma aplicação que simule o funcionamento de um processador nas suas tomadas de desvio.

## 2. INTRODUÇÃO

Dentro do contexto de Organização e Arquitetura de Computadores, surge a necessidade de projetar máquinas que sejam cada vez mais eficientes, de modo que se aproveitem de novas tecnologias e maior capacidade de processamento ao mesmo tempo em que usem os recursos disponíveis da melhor forma, nunca perdendo desempenho. Com foco em componentes ociosos de um processador, surgiu o pipeline, uma técnica de hardware que, atualmente, é fundamental para fazer máquinas mais velozes.

Em resumo, o pipeline faz com que as instruções de um computador sejam divididas em etapas menores (estágios), de modo que regiões diferentes de uma CPU sejam responsáveis por estágios distintos. Assim, a técnica promove uma espécie de “paralelismo” entre as operações, fazendo com que em um mesmo ciclo de clock etapas de várias instruções (usando componentes da CPU distintos entre si) sejam realizadas e, em um ciclo seguinte, cada instrução avance um estágio na sua execução.

Embora faça uso mais racional do processador, o pipeline pode gerar conflitos. Uma instrução, por exemplo, não pode ser executada simultaneamente a outra da qual ela seja dependente. Outro problema recorrente são instruções desvio: Como saber se um trecho de código (um loop, por exemplo) será encerrado ou continuará? O processador deve carregar as instruções seguintes ou haverá um desvio condicional? Para não atrasar o processamento de dados, o processador precisa tomar determinadas decisões, e esse é exatamente o foco do projeto Jump's Prediction, um sistema de predição de desvios.

Uma vez que não é possível prever os desvios que ocorrerão na execução de um programa, uma solução é adotar técnicas que objetivam acertar se o próximo desvio será ou não tomado. O programa simula 4 técnicas de predição: adaptativa de dois níveis, 1 bit, 2 bits e com memória de saltos. De forma simplificada, a adaptativa de dois níveis tem uma tabela que guarda padrões. Assim, se após determinada sequência, um desvio for tomado, na

próxima vez que tal sequência se repetir a predição também será tomado. A de 1 e 2 bits funcionam de maneira análoga, nelas, existe uma memória que guarda as últimas predições. Na de 1 bit, por exemplo, a predição é sempre igual ao último desvio, enquanto a de 2 bits verifica os dois últimos resultados. A memória de saltos, por sua vez, foi criada pela equipe e toma a decisão que tiver sido mais usada até o momento.

### 3. IMPLEMENTAÇÃO

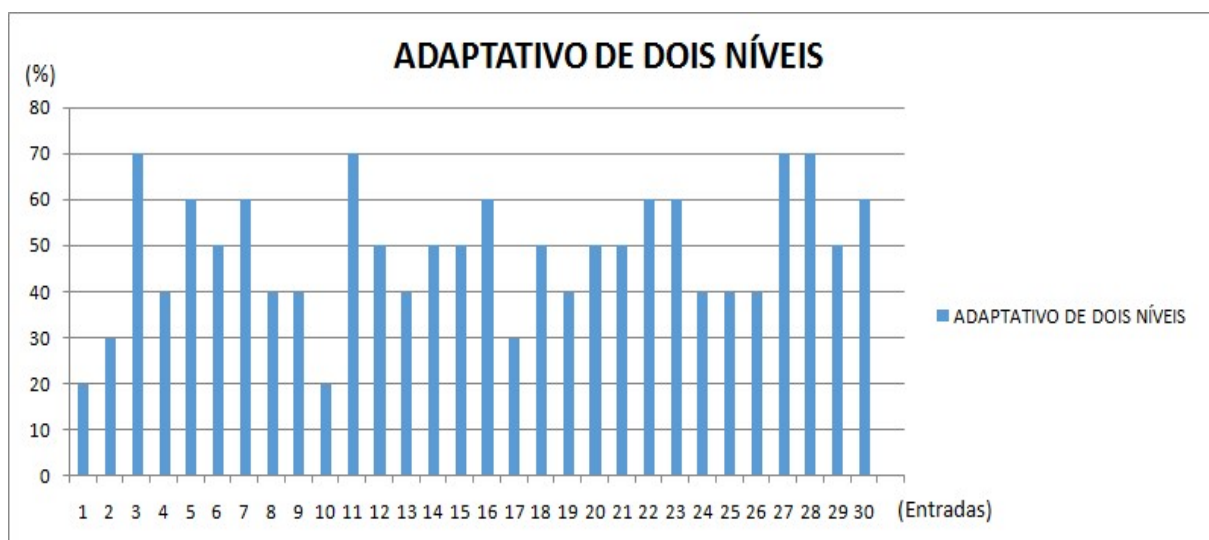
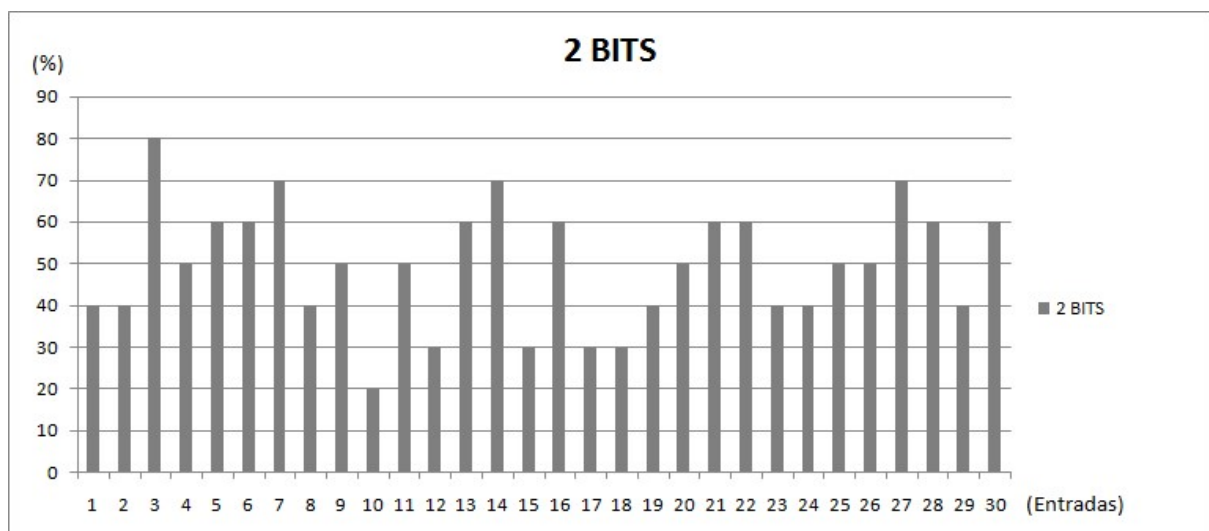
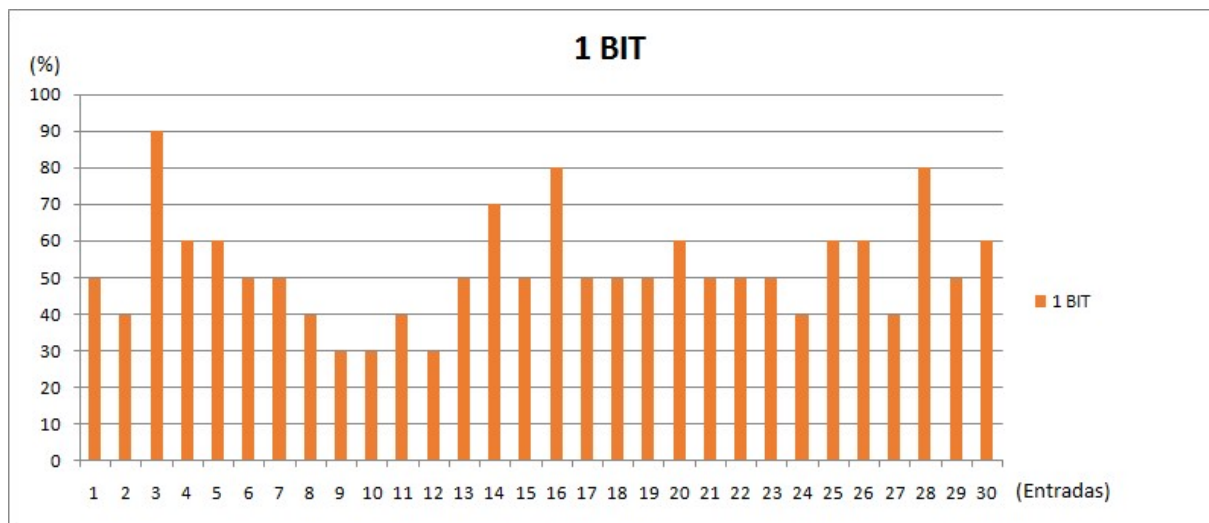
Desenvolvido na linguagem de programação C#, com a IDE para desenvolvimento .NET, Visual Studio 2010, e com subsistema gráfico Windows Presentation Foundation (WPF), a aplicação aborda e soluciona técnicas de predição de desvios através do uso da carga teórica adquirida na disciplina. Quatro tipos de predições foram utilizadas na composição do simulador e, para cada um delas, foram aplicadas diversas estruturas de dados. Dentre elas, destacamos o uso de dicionários, listas e arrays.

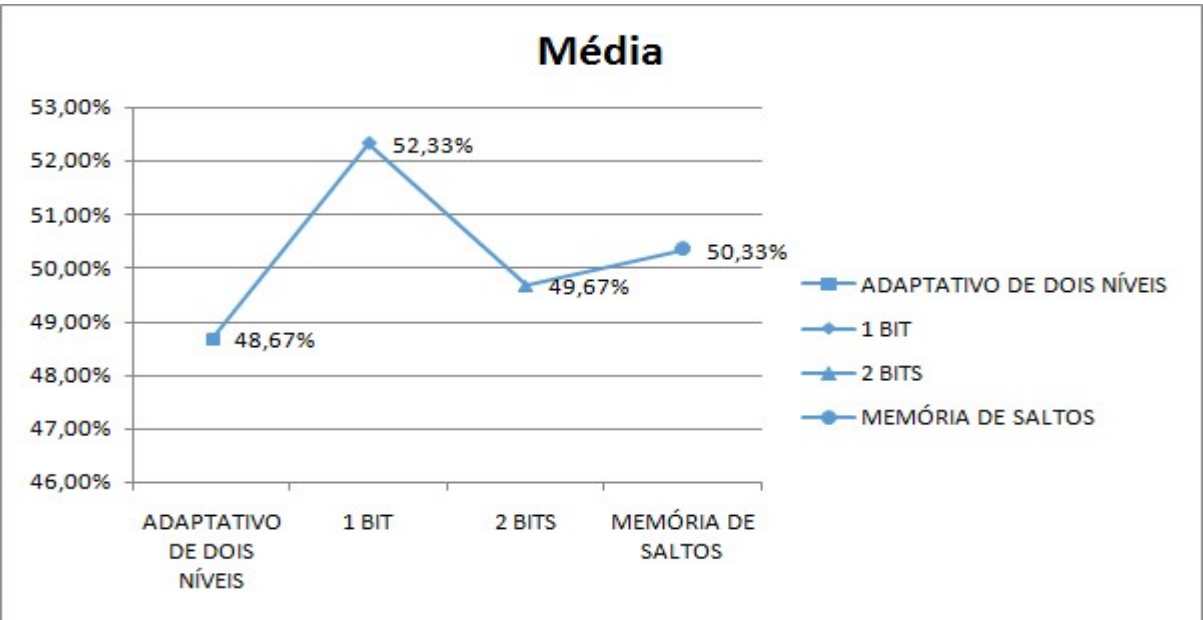
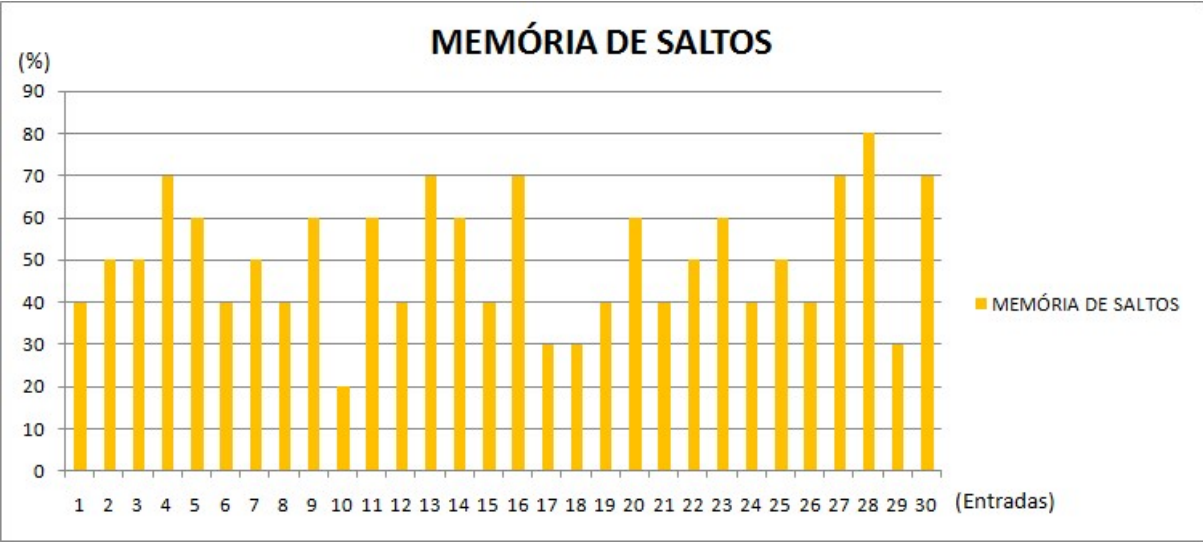
Em um primeiro momento, a predição mais simples e básica foi implementada, no caso, a de 1 bit. Nela, um loop itera toda entrada dos dados e para cada uma é previsto que a ultima decisão será repetida. A de 2 bits, por sua vez, funciona de maneira análoga. Para tal implementação, foi necessária estrutura similar, nela, entretanto, apenas dois erros de tomada de decisão em sequência ocasionam uma mudança na predição.

Consequentemente, os tipos de predições mais complexas foram desenvolvidas: a adaptativa de dois níveis e a de memória de saltos, esta proposta pela equipe. Na adaptativa, um padrão nas tomadas de decisões é constituído. Com o uso desse conjunto de padrões, tomadas seguintes poderão ser previstas de acordo com tomadas que já ocorreram. Por exemplo, caso um dos padrões armazenados seja “T N” e preceda a decisão “N”, em situações posteriores, cujo padrão se repita, a mesma decisão “N” será tida como mais provável. Para tal implementação, a estrutura de dados dicionário foi utilizada, de modo que sua chave é uma concatenação das suas duas últimas entradas anteriores. Por último, a predição com memória de saltos foi criada pelo grupo. Nesse tipo de predição, duas variáveis funcionam a nível de software como registradores, onde uma armazena a quantidade de decisões “T” e outra as de decisões “N”. O valor de predição é tomado de acordo com o maior valor de decisões tomadas anteriormente. Por exemplo, em uma entrada “T T T N N T T T N” o próximo valor de predição será “T”.

Além disso, foi acrescentado ao software diversos recursos que geram uma experiência mais intuitiva, simples e explicativa para o usuário final. Três tipos de entrada de dados foram implementadas. Uma para gerar valores aleatórios, uma através de inserção manual e outra por meio de escolha de um arquivo “.txt”, no formato “ T N N N N T N T ....”. Para promover um uso não mecanicista do aplicativo, uma tela de ajuda também foi feita, explicando de forma resumida cada um dos tipos de predição.

## 4. RESULTADOS







## **5. CONCLUSÃO**

O desenvolvimento do projeto permitiu a equipe entender de uma forma mais esclarecedora os conceitos de Pipeline e de predição de desvios para melhorar seu desempenho. Com os resultados obtidos, foi gerada uma contradição ao pensamento do grupo em relação a eficiência de cada predição realizada. O uso de valores aleatórios para gerar os casos de teste, nos concebeu entradas cujo desempenho da predição de 1 bit teve maior percentual de acertos em relação aos outros tipos de predição. Esperava-se que a de 2 bits tivesse a maior quantidade final de acerto, porém, dentre os 30 casos de testes, com 10 entradas cada, tal predição teve o segundo pior desempenho. Isso se deve ao fato de que a geração aleatória de números pela linguagem de programação não simula o real comportamento de um software, justamente onde os tipos de predição se baseiam.

## 6. REFERÊNCIAS

[1] Stack Overflow.

<http://stackoverflow.com/>. Último acesso em 05/12/2015

[2] Youtube: CodingMadeEasy

<https://msdn.microsoft.com/pt-br/library/618ayhy6.aspx>. Último acesso em 05/12/2015

Hennessy, John. L.; Patterson, David A. Organização e Projeto de Computadores, 3a Ed, 2005.