

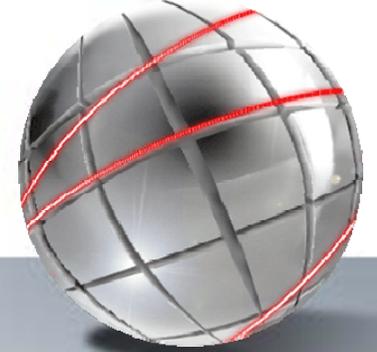
# ***Guia para Análises***

Experimento do DZero (CMS?)



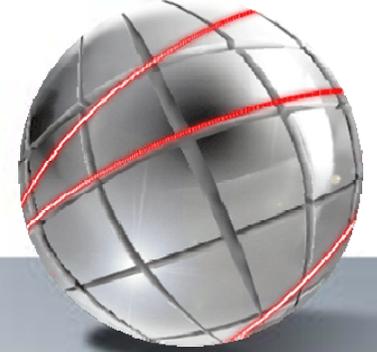
Pedro Mercadante (UNESP)

# Sumário



- Dados
- Sinal
  - Simulação por MC
- Definição dos objetos Físicos
- Background
  - Simulação por MC
  - Estimativa a partir dos dados
- Comparação dados vs background
  - Limites
  - Medidas

# **Dados**



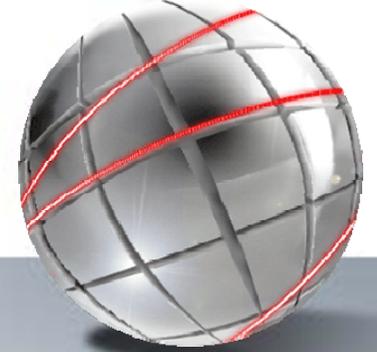
- Definição do data Sample
  - Skimmed (pre selection)
- Definição dos Triggers
- Data Quality
  - Algumas exigências básicas
- Cálculo da Luminosidade

# Simulação por Monte Carlo



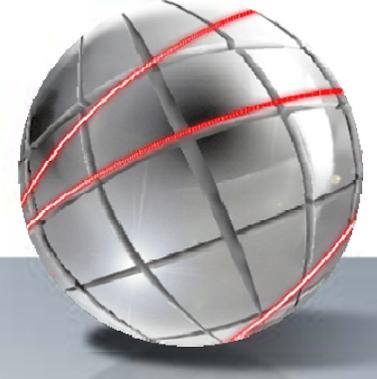
- Pythia fornece a simulação do evento completo (todas as partículas produzidas com seus momentos)
- Como comparar com o experimento?
- Simulação do detector:
  - d0gstar: Simulação do detector. O programa segue a trajetória de cada partícula, calculando, para cada elemento do detector, a energia depositada, mudanças na trajetória e outros efeitos de interação com a matéria
  - d0sim: Transforma a informação do d0gstar em dados idênticos ao que o detector real nos daria (pulsos, contagens ADC)
  - d0reco: Reconstrução do sinal. Transforma as informações do detector em objetos físicos. Exatamente o mesmo programa é utilizado sobre os dados reais.

# Compilar e rodar



- Precisamos compilar os programas `mcpp_gen` e `D0_mcpp_gen` incluindo nossas bibliotecas do Pythia:
  - Incluir as bibliotecas geradas `libpylib.a` e `aldata.o` no diretório `/lib/Linux2.4-GCC_3_4/`
  - Definir as variáveis de pythia para o nosso diretório:  
>setenv PYTHIA\_DIR /rooms/safe/brazil/mercadan/monte\_carlo/test/pythia  
>setenv PYLIB /rooms/safe/brazil/mercadan/monte\_carlo/test/pythia/lib
- Rodar o programa utilizando a simulação completa do DZero:
  - Utilizei `mc_runjob v06-05-26` e o macro:  
`/rooms/safe/brazil/mercadan/monte_carlo/mc_p17.09.01/pythia_ued.macro`
  - Utilizei o card file:  
`/work/lafex-clued0/mercadan/pythia_cards/pythia_ued_incl.cards`

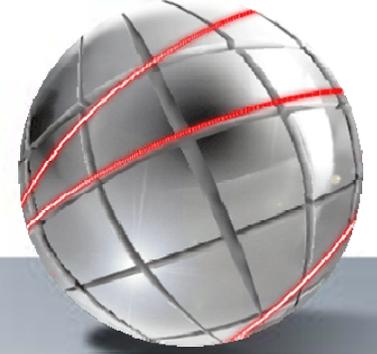
# Análise



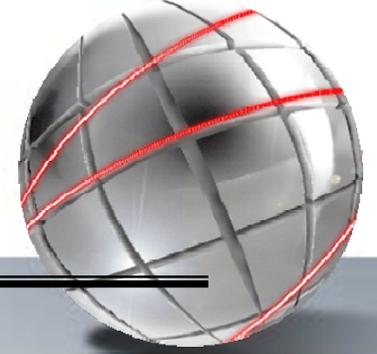
- Reconstruir os objetos físicos
  - Dados → d0reco → Tmbanalyze → root
  - MC → d0reco → Tmbanalyze → root
- Ferramentas comuns (cafe). Executamos a partir das rootuplas geradas após o Tmbanalyze.
- Os diversos grupos de algoritmos preparam as definições dos objetos.
  - Muons
  - Missing Et

# ***Objetos Físicos***

- Leptons
- Jatos
- Missing Et
- .....



# Métodos de detecção



Signature	Detector Type	Particle
Jet of hadrons	Calorimeter	u, c, d, s b, g
'Missing' energy	Calorimeter	$\nu_e, \nu_\mu, \nu_\tau$
Electromagnetic shower	EM Calorimeter	e, $\gamma$
Purely ionization interactions, $dE/dx$	Muon Absorber	$\mu, \tau \rightarrow \mu\nu\nu$
Decays, $c\tau \geq 100\mu\text{m}$	Si tracking	c, b, $\tau$

# Muons



- Podem ser identificados em três subsistemas independentes
  - Câmara de Muons (3 camadas): associa um track, chamado muon local
  - O central track system (consiste do SMT e CFT) mede tracks com precisão. Um muon local com um track associado é chamado de central track matched muon
  - Partícula MIP no calorímetro (método em desenvolvimento, eficiência de apenas .5)
- Qualidade
  - Tight: Sinal na camada A e BC do sistema de muons, consistente com um track no CTS
  - Medium: Pode falhar em um dos critérios do tight
  - Loose: Outras combinações
- Isolamento

# Configuração do muon

```
+cafe.Run: MuonSelector(muon)
```

```
+cafe.Packages: caf_util
```

```
##### MUON SELECTOR #####
```

```
muon.DebugLevel: 0
```

```
# input muon branch name
```

```
muon.From: Muon
```

```
# output tree name
```

```
muon.Tree: SelectedObjects
```

```
# output muon branch name
```

```
muon.To: selectedMuons
```

```
#sort branch in pT:
```

```
muon.Sort: 1
```

```
##### pT cut and eta cut are not set by std MuonID
```

```
# pT cut in GeV (set in main config file)
```

```
#muon.pT: 15.0
```

```
# eta cut
```

```
muon.eta: -1
```

```
# minimal number of muons in events
```

```
muon.nMuons: 2
```

```
muon.nMuonsMax: -1
```

```
# Beam spot info
```

```
muon.BeamSpotFile: caf_util/configs/beamSpot-2.09
```

```
# Do you want to reject runs with no beam spot entry
```

```
muon.RejectNoBeamSpot: 1
```

```
# Do you want to correct Muon pT (with or without smt hits)
```

```
muon.CorrectPtSMT: 0
```

```
muon.CorrectPtNoSMT: 1
```

```
##### Standard Muon ID
```

```
# Set Muon Quality name & version
```

```
muon.MuonQualityName: Loose
```

```
muon.MuonQualityVersion: 1
```

```
# Set Muon Track Quality name & version
```

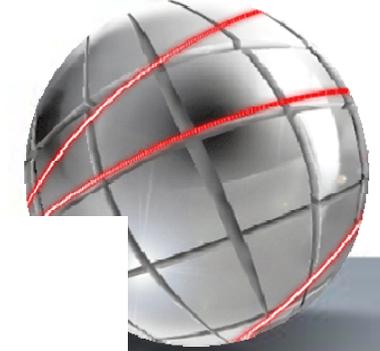
```
muon.TrkQualityName: Loose
```

```
muon.TrkQualityVersion: 1
```

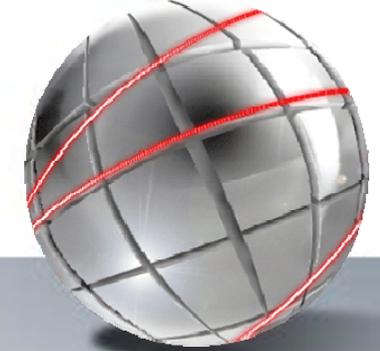
```
# Set Muon Isolation name & version
```

```
muon.IsoQualityName: NPTight
```

```
muon.IsoQualityVersion: 1
```



# Configuração para MC



```
+cafe.Run: MuonSelector(muon_pre)
```

```
+cafe.Run: ApplyMuonSmear(mu_smear)
```

```
+cafe.Run: MuonSelector(muon)
```

```
##### FIRST MUON SELECTOR #####
```

```
configure first muon selector:
```

- apply pT correction
- no pt cut
- select 2 loose muons, loose tracks, NPTight isolation

```
##### MUON SMEARING #####
```

```
mu_smear.From: preselectedMuons
```

```
mu_smear.Tree: TMBTree
```

```
mu_smear.To: smearedMuons
```

```
mu_smear.conf_name: p17_09
```

```
#mu_smear.Smear: post #set in main config file
```

```
##### FIRST MUON SELECTOR #####
```

```
configure second muon selector:
```

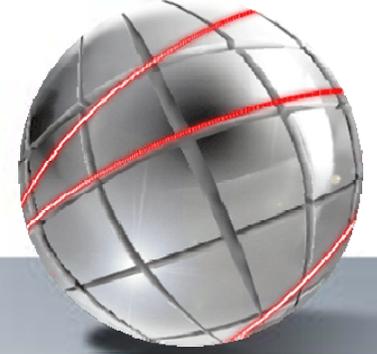
- DON'T apply pT correction
- DO pt cut
- other cuts the same (select 2 loose muons, loose track, NPTight isolation)

# Missing $E_T$



- Calculada a partir das células do calorímetro. Várias definições dependendo se consideramos todas as células ou só acima de certos thresholds (usualmente desconsidera-se as células do CH)
- Várias correções a partir dos objetos reconstruídos:
  - Jatos: Em um jato as células do CH são consideradas sinal e não ruído. A energia de um jato também é recomputada com maior precisão
  - Electrons e fótons: Objetos reconstruídos têm a energia recomputada com maior precisão
  - Muons: Muons depositam pouca energia no calorímetro. Muons reconstruídos devem ter o momento subtraído do cálculo de missing  $E_T$  após ser deduzido a quantidade de energia depositada no calorímetro

# Jatos



- Algoritmo do Jato
- Pt
- Rapidez
- Correções

# Background



- Monte Carlo
  - Gerar os bg (nem sempre necessário)
  - Utilizar bg gerados pela colaboração
  - Corrigir as eficiências a partir de processos conhecidos
- Dados
  - Muitas vezes é necessário estimar os bg a partir dos dados (bg de QCD ou bg instrumental)
  - Escolhe-se uma região dominada por bg e parametriza-se a possibilidade de um sinal ser obtido a partir dos dados.

# ***Comparação dados vs bg***



- Após uma pré seleção compara-se os dados com o bg (ou seja, com a estimativa do MP). Uma boa concordância significa que entendemos os dados.
- Aplicamos os cortes na tentativa de extrair um sinal
- Compara-se dados com bg após os cortes. Obtemos limites de exclusão ou extraímos medidas.