

**INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**  
RIO GRANDE DO NORTE



REDE FEDERAL  
DE EDUCAÇÃO  
PROFISSIONAL  
E TECNOLÓGICA  
1909-2009

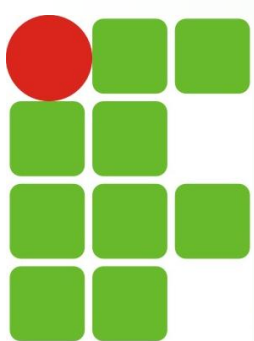
# Curso Superior em Redes de Computadores

**SMTP**



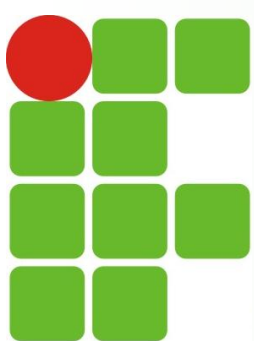
# Objetivo

- Conhecer as características, funcionalidades e componentes do serviço de correio eletrônico (***S**imple **M**essage **T**ransfer **P**rotocol - **SMTP***)



# Introdução

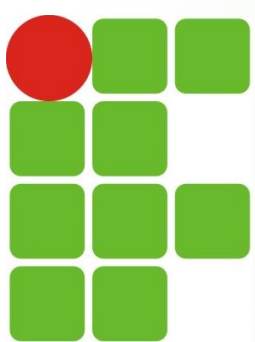
- Camada de aplicação
  - Usa os serviços da camada de transporte para permitir a comunicação entre os processos de aplicação
    - Serviço de datagramas
    - Serviço de circuito virtual
  - O desenvolvedor da aplicação deve selecionar o serviço de transporte a ser adotado
  - Serviço de transporte sem conexão
    - Utiliza o protocolo UDP
  - Serviço de transporte com conexão
    - Utiliza o protocolo TCP



# SMTP (*Simple Message Transfer Protocol*)

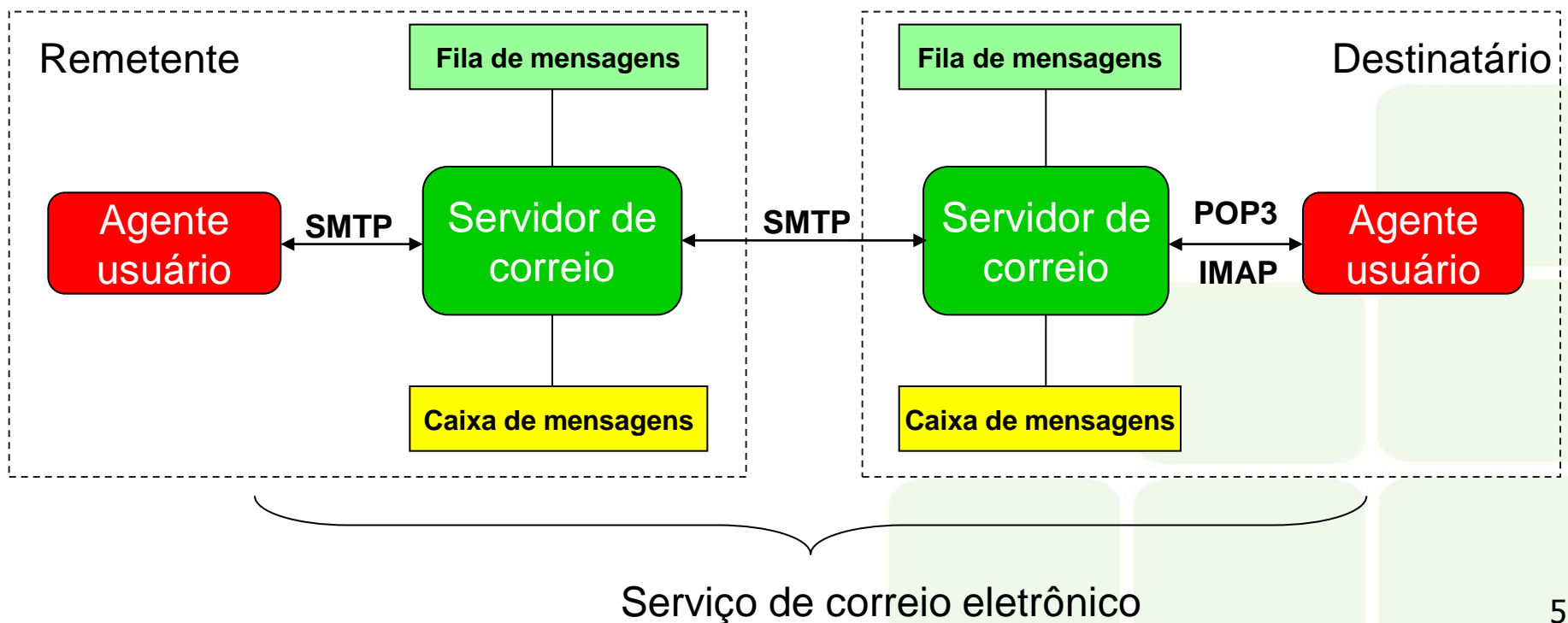
## ■ Características

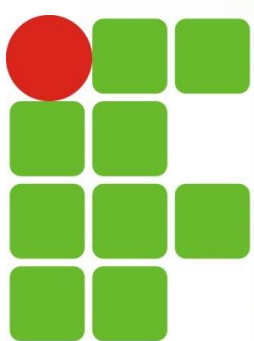
- Implementa o serviço de correio eletrônico da arquitetura TCP/IP (RFCs 821 e 822)
- Baseado no modelo Cliente-Servidor
- Utiliza os serviços de transporte
  - Com conexão
    - Envio e recebimento de mensagens



# SMTP (*Simple Message Transfer Protocol*)

## ■ Características





# SMTP (*Simple Message Transfer Protocol*)

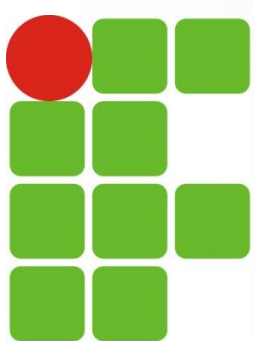
## ■ Componentes

### ■ Agente usuário

- Programa usado pelo usuário para ler, compor e enviar mensagens
- Usado pelo usuário remetente e destinatário
- Também denominado leitor de correio ou UA
- Exemplos
  - Pine, mutt, outlook express, mozilla tunder bird

### ■ Servidor de correio

- Realiza roteamento de mensagens
- Configurado pelo administrador do domínio
- Também denominado agente de transferência de mensagens (MTA)
- Unix: Sendmail



# SMTP (*Simple Message Transfer Protocol*)

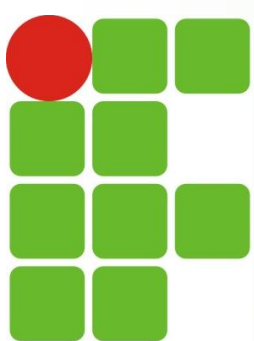
## ■ Componentes

### ■ Caixa de mensagens (mailbox)

- Mantém as mensagens enviadas aos respectivos usuários
- Cada usuário possui uma caixa de mensagens
- Viabiliza o modelo de **comunicação assíncrona**

### ■ Fila de mensagens

- Armazena temporariamente as mensagens até que seja possível entregá-las
- Adota a técnica de *spooling* (*Simultaneous Peripheral Operation Online*) para tratar falhas temporárias nos servidores de correio



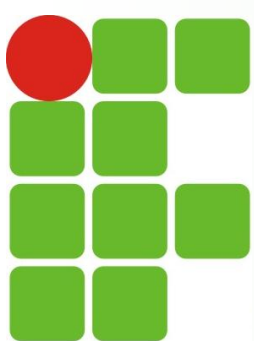
# SMTP (*Simple Message Transfer Protocol*)

## ■ Componentes

### ■ Protocolo SMTP

- Protocolo de aplicação do serviço de correio eletrônico da arquitetura TCP/IP
- Define um conjunto de comandos e resposta
- É especificado no RFC 821
- Utiliza a porta TCP 25
- É adotado para transportar as mensagens nos seguintes estágios
  - Agente e servidor de correio do usuário remetente
  - Servidores de correio dos usuários remetentes e destinatários

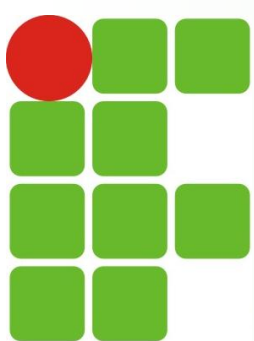




# SMTP (*Simple Message Transfer Protocol*)

## ■ Envio de mensagens

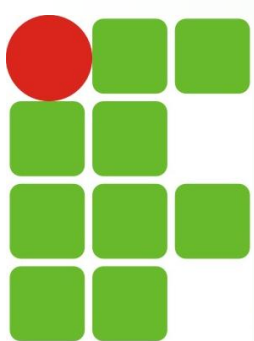
- Agente do usuário envia a mensagem para o servidor do remetente
- Servidor do remetente armazena a mensagem na fila
- Servidor do remetente envia a mensagem para o servidor do destinatário
  - Consulta o DNS, solicitando os registros MX associados ao domínio do usuário destinatário
  - Em caso de falha, servidor do remetente mantém a mensagem na fila e tenta novamente (default: a cada 30min, por 4 dias)
  - Caso insucesso, retira mensagem da fila e notifica remetente
- Servidor do destinatário armazena a mensagem na respectiva caixa de mensagens



# SMTP (*Simple Message Transfer Protocol*)

## ■ Leitura de mensagens

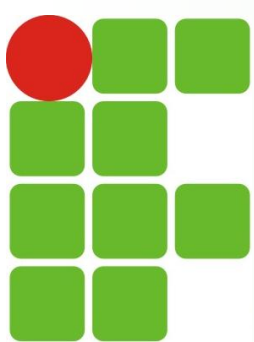
- Agente do usuário recupera mensagens da caixa de mensagens do servidor de correio do usuário
- Acesso direto
  - Agente usuário executa na mesma estação em que reside o arquivo que contém a caixa de mensagens do usuário
- Acesso via protocolo de acesso
  - Agente usuário pode executar em estação diferente daquela em que reside o arquivo que contém a caixa de mensagens do usuário
  - Adota os protocolos POP3 ou IMAP



# SMTP (*Simple Message Transfer Protocol*)

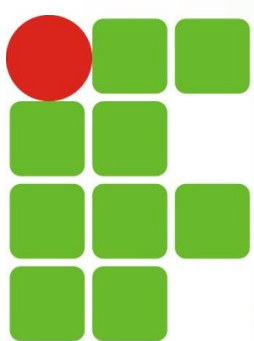
## ■ Modelo de interação

- Através de COMANDOS enviados pelos Clientes e RESPOSTAS enviadas pelos Servidores;
- Servidor envia uma resposta para cada comando recebido;
- Cada resposta possui um código numérico que sinaliza o correto processamento ou erro. Opcionalmente, descreve seu significado;
- Na comunicação entre Servidores, o Servidor remetente se comporta como Cliente e o destinatário como Servidor;
- As UAs se comportam como Cliente;



# SMTP (*Simple Message Transfer Protocol*)

- Modelo de Interação - Comandos do SMTP
  - Helo – identifica o cliente ao servidor
  - Mail – Indica o endereço de correio do remetente
  - RCPT – Informa o endereço de correio do destinatário
  - Data – Cliente solicita autorização para envio da mensagem
  - Quit – Finaliza a sessão
  - Turn – Solicita a inversão de papéis cliente /servidor
  - Rset – Aborta a transação de correio
  - Vrfy – Verifica a validade de um usuário no servidor
  - Expn – Identifica a composição de uma lista de distribuição

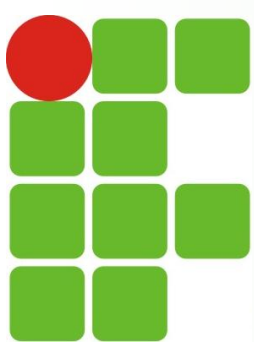


# SMTP (*Simple Message Transfer Protocol*)

## ■ Modelo de interação

```
C:\> telnet mail.ifrn.local 25

S: 220 mail.ifrn.local
C: Helo mail.ifrn.local
S: 250 mail.ifrn.local Hello mail.ifrn.local, pleased to meet you
C: Mail From: <salesfilho@ifrn.local>
S: 250 <salesfilho@ifrn.local> ... Sender ok
C: RCPT To: <bob@ifrn.local>
S: 250 <bob@ifrn.local> ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on line by itself
C: Ola, isto eh um Teste...
C: .
S: 250 Message accepted for delivery
C: Quit
S: 221 mail.ifrn.local closing connection
```



# Exemplo de interação SMTP

{estabelecida a conexão TCP com o Server}

S: 220 hamburger.edu

C: HELO crepes.fr

S: 250 Hello crepes.fr, pleased to meet you {Cliente/Server se apresentam}

C: MAIL FROM: <alice@crepes.fr> {início de cada mensagem do Cliente}

S: 250 alice@crepes.fr... Sender ok {remetente}

C: RCPT TO: <bob@hamburger.edu> {destinatário}

S: 250 bob@hamburger.edu ... Recipient ok

C: DATA

S: 354 Enter mail, end with "." on a line by itself

C: Do you like ketchup?

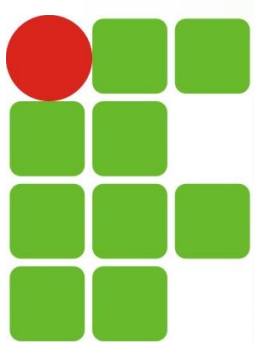
C: How about pickles?

C: .

S: 250 Message accepted for delivery

C: QUIT

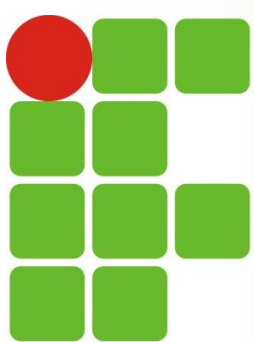
S: 221 hamburger.edu closing connection



# SMTP (*Simple Message Transfer Protocol*)

## ■ Modelo de interação

- Cliente estabelece conexão na Porta 25 – TCP do Servidor;
- Servidor responde com saudação 220. Indicando que está pronto para iniciar;
  - A resposta de saudação identifica o servidor
- Cliente identifica-se ao servidor através do HELLO;
- Servidor aceita comando e responde com 250 contendo nomes absolutos do servidor e do cliente;
  - Caso não aceite, servidor responde com mensagem 501.
- Cliente envia comando MAIL, informando correio remetente, para indicar início da transação;
- Servidor responde com 250 se aceitar endereço do remetente ou, caso contrário com 501;

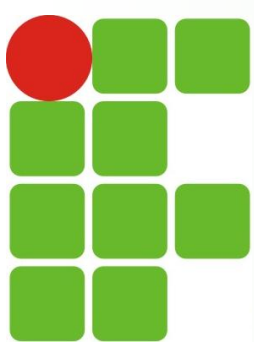


# SMTP (*Simple Message Transfer Protocol*)

## ■ Modelo de interação

- É iniciada a transação. Cliente envia RCPT que identifica correio destinatário;
- Servidor responde com 250 ou, caso não aceite correio destinatário, responde com erro 550;
  - Múltiplos RCPTs são enviados para mensagens multi-destinatários;
  - Há em 250 ou 550 para cada RCPT;
- Cliente envia comando DATA;
- Servidor responde com 354 – Pronto para receber;
- Cliente envia mensagem e encerra com <CR><LF>.<CR><LF>;
- Servidor ao detectar fim da mensagem, responde com 250 informando mensagem na fila
- Cliente envia QUIT solicitando encerramento da sessão;
- Servidor responde com 221, indicando encaminhamento da mensagem

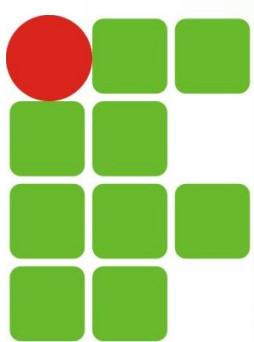




# SMTP (*Simple Message Transfer Protocol*)

## ■ Modelo de interação

- Em uma única sessão podem ser enviadas várias mensagens:
  - Cliente utiliza comando MAIL ao invés de QUIT entre mensagens;
  - Cliente utiliza QUIT no final de todas mensagens para fechar a sessão.
- Podem existir diversas transações em uma mesma sessão:
  - Cada transação consiste de um MAIL, um ou vários RCPT e um DATA;
- Antes de fechar uma sessão, opcionalmente, o cliente pode inverter os papéis Cliente/Servidor através do comando TURN. Servidor aceita com resposta 250 ou não com resposta 502;
- O comando RSET aborta a transação e descarta informação armazenada no servidor;
- O comando VRFY permite verificar validade de um usuário no servidor;
- O comando EXPN permite expansão de uma lista de distribuição, identificando seus membros
- VRFY e EXPN podem ser desabilitados pelo administrador (segurança)<sup>17</sup>



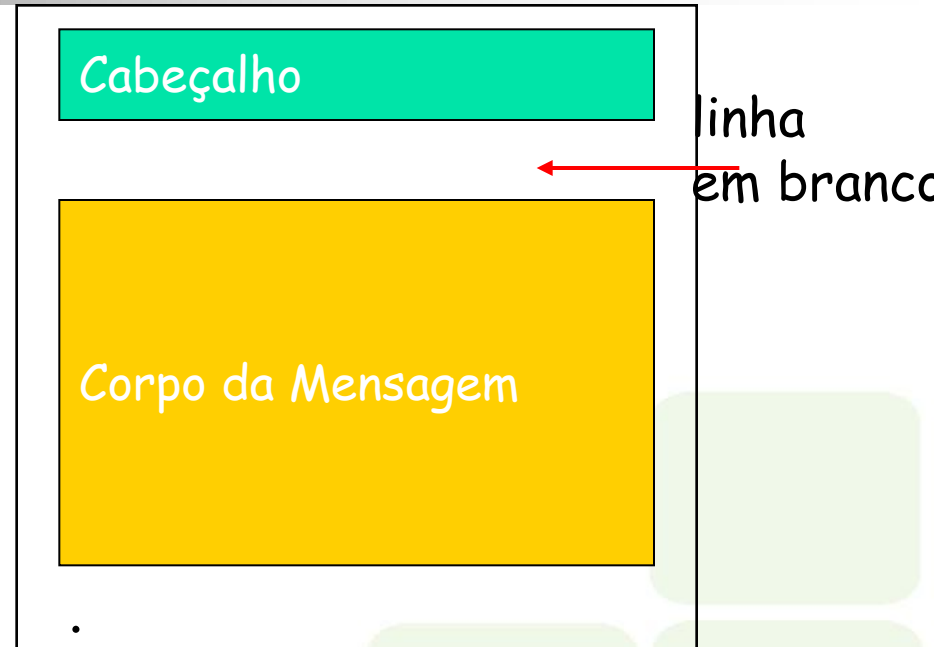
# SMTP (*Simple Message Transfer Protocol*)

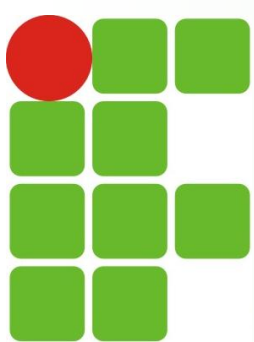
- Estrutura da mensagem
  - Envelope
    - Contém os endereços dos usuários remetente e destinatário (usada pelo Servidor)
  - Cabeçalho
    - Descreve características das mensagens (From, To, Subject)
  - Corpo
    - Conteúdo propriamente dito da mensagem
    - Contém apenas caracteres ASCII de 7 bits
  - O UA adiciona o Cabeçalho ao Corpo e envia ao servidor do remetente;
  - O Servidor remetente adiciona entradas ao cabeçalho e envia ao servidor destinatário;
  - O Servidor destinatário adiciona entradas no cabeçalho e armazena na caixa postal do destinatário

# Estrutura da Mensagem

- SMTP: protocolo para trocar mensagens de e-mail
- RFC 822: Padrão para mensagens do tipo texto:
- Linhas de cabeçalho, Ex.,
  - To:
  - From:
  - Subject: {opcional}

*Diferentemente dos comandos SMTP, fazem parte do próprio email*
- Corpo
  - "Mensagem" ASCII somente com caracteres



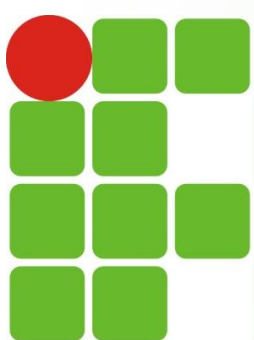


# Estrutura da Mensagem

## ■ Estrutura da mensagem

### ■ O Corpo pode conter caracteres binários

- MIME (Multipurpose Internet Mail Extension), RFCs 2045 e 2046, permite mensagens com múltiplas partes e diferentes formatos (ASCII, Binário).
- Independente do conteúdo da mensagem, o cabeçalho e conteúdo MIME são transmitidos em ASCII de 7 bits;
- MIME codifica dados binários em ASCII;
- O agente MIME adiciona entradas ao cabeçalho e ao corpo da mensagem para indicar ao agente MIME destinatário a estrutura de codificação.
- MIME não impõe modificação nos servidores, pois só os UAs devem interpretar a extensão MIME.



# Formato das Mensagens

## Extensões Multimídia

- MIME: Multipurpose Internet Mail Extension (RFCs 2045 e 2056)
- Linhas adicionais no cabeçalho que declaram o tipo de conteúdo diferente do texto padrão ASCII (imagem, vídeo, áudio)

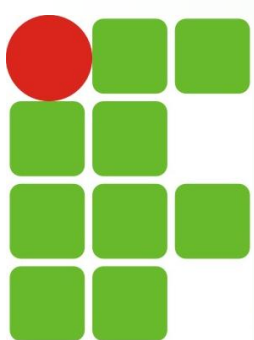
MIME versão

método usado  
para codificar dados

multimedia data  
tipo, subtipo,  
declaração de parâmetro

dados codificados

```
From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Imagem de um gostoso
crepe.
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: image/jpeg
base64 encoded data .....
.....base64 encoded data
```



# Formato das Mensagens

## Extensões Multimídia

- Content-Type e Content-Transfer-Encoding são cabeçalhos-chave MIME para suporte à Multimídia.
- Content-Transfer-Encoding, indica o formato da codificação ASCII usada para codificar a mensagem não-ASCII (imagem, áudio e vídeo).
- Content-Type, permite que o Agente Usuário destinatário execute uma ação adequada sobre a mensagem. Por exemplo, descompressão de uma imagem JPEG.
- Geralmente as mensagens MIME são enviadas em anexos.
- Base64 e quoted-printable, são padrões de codificação para o formato ASCII de 7 bits da RFC 2045 da MIME (evita linhas indesejáveis como contendo apenas um ponto final)
- Após o cabeçalho há uma linha em branco e, após o corpo da mensagem uma linha com apenas um ponto indicando fim da mensagem.

**Content-Type: type/subtype; parâmetros**

Usado para especificar a natureza dos dados no corpo da entidade MIME

Type: declara o tipo geral dos dados

Subtype: define o formato específico para o Type de dados

Parâmetros: são modificadores do Subtype. Não afetam a natureza do conteúdo

Seguem os Types mais comuns:

**TEXT:** indica que o corpo da mensagem contém textos.

**PLAIN:** indica texto simples sem comando de formato ou instrução. Deve ser mostrado exatamente como está, sem tratamento algum de software.

**HTML:** indica que tags embutidas na mensagem devem ser interpretadas. Isso permite a apresentação da mensagem em formato de Página WEB.

**IMAGE:** indica que o corpo da mensagem contém uma imagem. Os subtypes GIF ou JPEG indicam ao destinatário para decodificar antes de apresentar.

**Content-Type: type/subtype; parâmetros**

Seguem os Types mais comuns:

**ÁUDIO:** requer um dispositivo de saída (alto-falante, telefone). O subtype BASIC (codificado 8-bit  $\mu$ -law ) e 32kadpcm (codificação 32 kbps definido na RFC 1911)

**VÍDEO:** exemplo de subtipos: MPEG, QUICKTIME

**APPLICATION:** Usado para dados que não se enquadram nas demais categorias e precisam ser processados por uma Aplicação antes de ser mostrados.

O subtype MSWORD indica que a mensagem MIME deve ser passada ao WORD.

Já o OCTET-STREAM é usado para indicar que o conteúdo são dados binários (programas). Assim, deve ser gravado em disco para posterior processamento.



# Tipo Multiparte

From: alice@crepes.fr  
To: bob@hamburger.edu  
Subject: Picture of yummy crepe.  
MIME-Version: 1.0  
Content-Type: multipart/mixed; boundary=98766789

--98766789

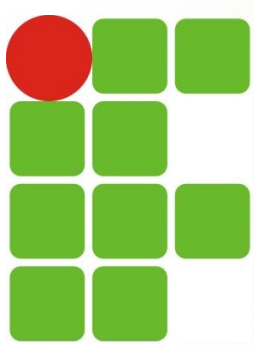
Content-Transfer-Encoding: quoted-printable  
Content-Type: text/plain

Dear Bob,  
Please find a picture of a crepe.

--98766789

Content-Transfer-Encoding: base64  
Content-Type: image/jpeg

base64 encoded data .....  
.....  
.....base64 encoded data  
--98766789--



Usado para indicar que corpo da mensagem possui objetos de várias naturezas (Texto, Imagem, Programa, etc).

O email, diferentemente da WEB, coloca todos os objetos na mesma mensagem. Assim, como o Agente Usuário Destinatário irá saber:

Onde inicia e termina cada objeto?

Como cada objeto não ASCII foi codificado?

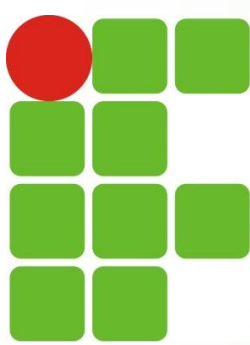
Qual o tipo de conteúdo de cada objeto?

Isso é conseguido precedendo cada objeto com linhas de cabeçalho Content-type: e Content-Transfer-Encoding:, e colocando-se CARACTERES DE FRONTEIRA entre eles.

A separação começa sempre com -- (dois traços) e termina com <CR><LF>.

- O Servidor destinatário insere a linha de cabeçalho **Received:** ao topo da mensagem especificando:
  - O Servidor Origem (from);
  - O Servidor destino (to); e
  - A data/hora da recepção.

# Mensagem Recebida



Received: from crepes.fr by hamburger.edu; 12 Oct 98 15:27:39 GMT  
From: alice@crepes.fr  
To: bob@hamburger.edu  
Subject: Picture of yummy crepe.  
MIME-Version: 1.0  
Content-Type: multipart/mixed; boundary=98766789

--98766789  
Content-Transfer-Encoding: quoted-printable  
Content-Type: text/plain

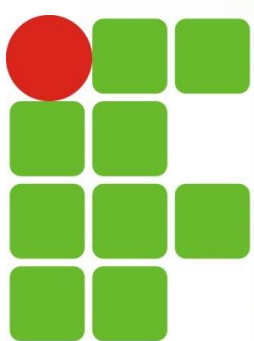
Dear Bob,  
Please find a picture of a crepe.

--98766789  
Content-Transfer-Encoding: base64  
Content-Type: image/jpeg

base64 encoded data .....  
.....  
.....base64 encoded data  
--98766789--

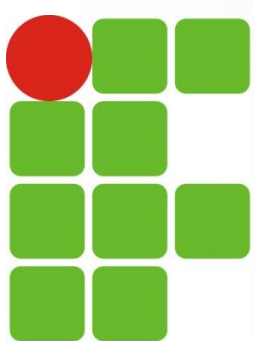
# Protocolos de e-mail

- SMTP: entrega e armazena no servidor do destino
- Protocolo de acesso: recupera mensagens do servidor
  - POP: Post Office Protocol [RFC 1939] - Porta 110
    - autorização (agente <-->servidor), download e atualização
  - IMAP: Internet Mail Access Protocol [RFC 2060] - Porta 143
    - maiores recursos (mais complexo)
    - manipulação de mensagens armazenadas no servidor
  - HTTP: Hotmail , Yahoo! Mail, etc. (conveniente / lento)



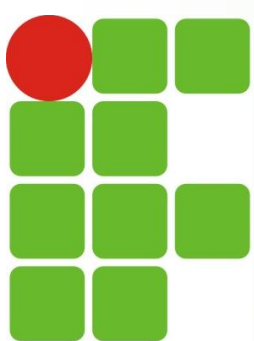
# POP3 (*Post Office Protocol*)

- O Post Office Protocol (POP3) é um protocolo utilizado no acesso remoto a uma caixa de correio eletrônico
- Ele está definido no RFC 1225
- Permite que todas as mensagens contidas numa caixa de correio eletrônico possam ser transferidas sequencialmente para um computador local



# POP3 (*Post Office Protocol*)

- Comandos do protocolo POP3
  - USER – Envia a identificação do usuário
  - PASS – Envia a senha do usuário
  - LIST – Lista as mensagens do usuário
  - RETR – Recupera uma mensagem pelo índice
  - TOP – Mostra uma quantidade de linhas do corpo da mensagem
  - DELE – Apaga uma mensagem da caixa



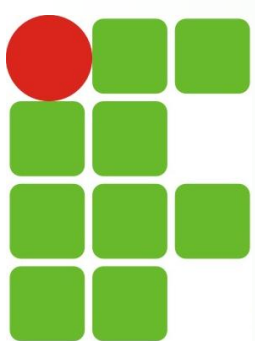
# POP3 (*Post Office Protocol*)

## ■ Modelo de interação

```
C:\> telnet mail.ifrn.local 110

C: Trying 192.168.0.146...
   Connected to freedom.ifrn.local.
   Escape character is '^]'.
S: +OK ready <5146.1231533216@freedom.ifrn.local>
C: user bob
S: +OK Password required for bob.
C: pass bob
S: +OK bob has 2 visible messages (0 hidden) in 2593 octets.
C: list
S: +OK 2 visible messages (2593 octets)
   1 1746
   2 847
C: retr 2
S: ... <Aqui vem o conteúdo da mensagem>
C: dele 1
```





# Atividade

- Fazer envio e recebimento de mensagens utilizando o comandos via telnet
- Monitorar as conexões e os comandos dos protocolos SMTP e POP3 com o analisador de protocolos



# Referências

- Comer, Douglas E., Interligação de Redes Com Tcp/ip
- James F. Kurose, Redes de Computadores e a Internet
- Escola Superior de Redes, Arquitetura e Protocolos de Redes TCP/IP