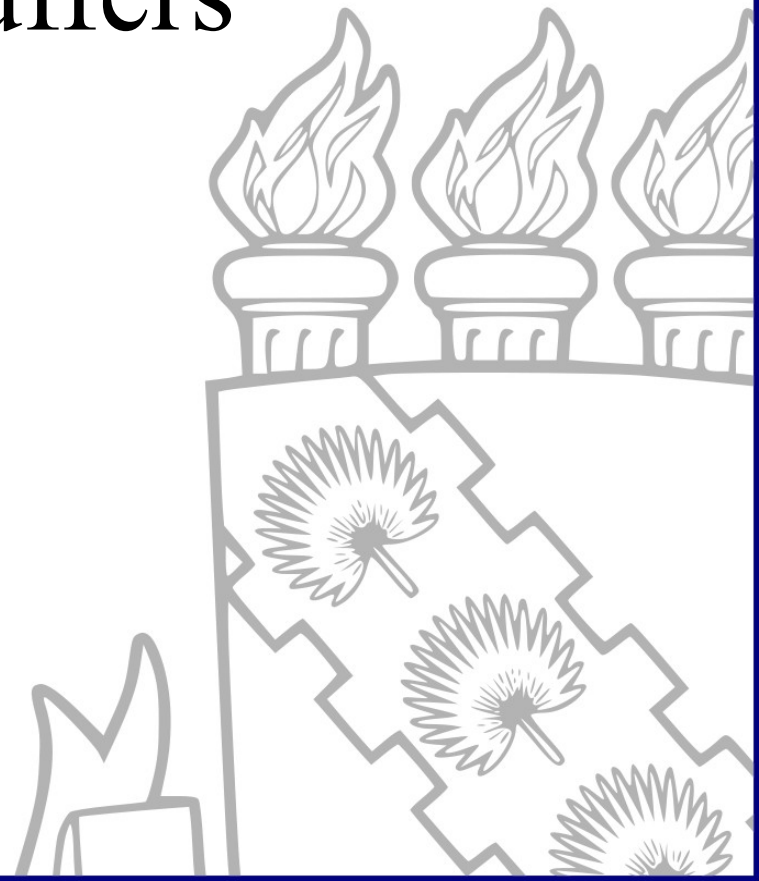


# Laboratório - Protocolo Buffers

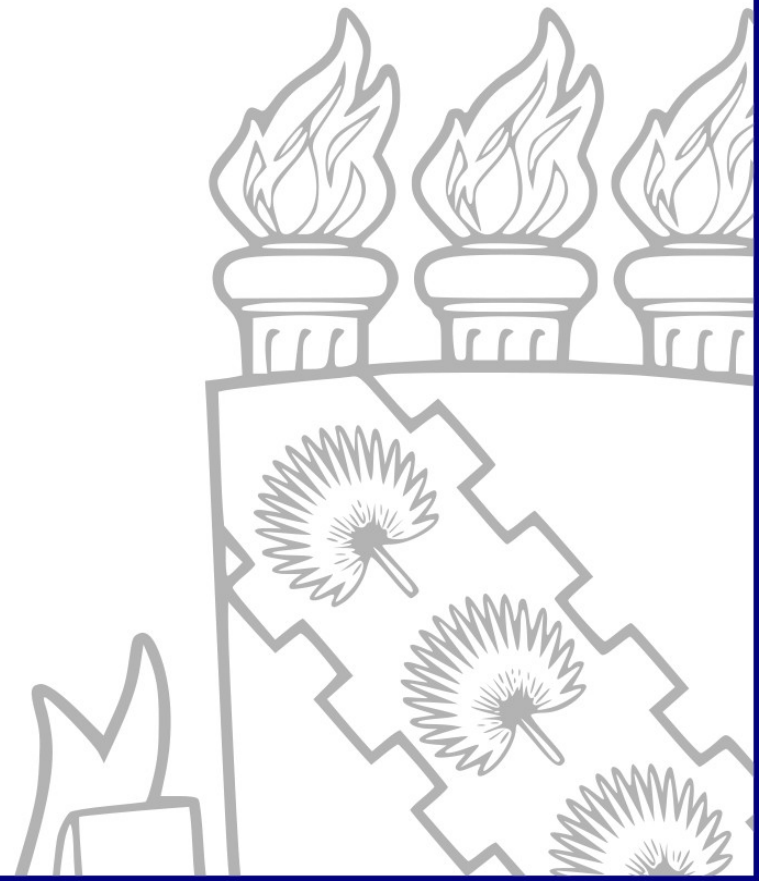
Estagiário: Delano José Holanda Maia  
Mat.: 0497834



# Linguagem abordada

- Java
- Buffers de protocolo

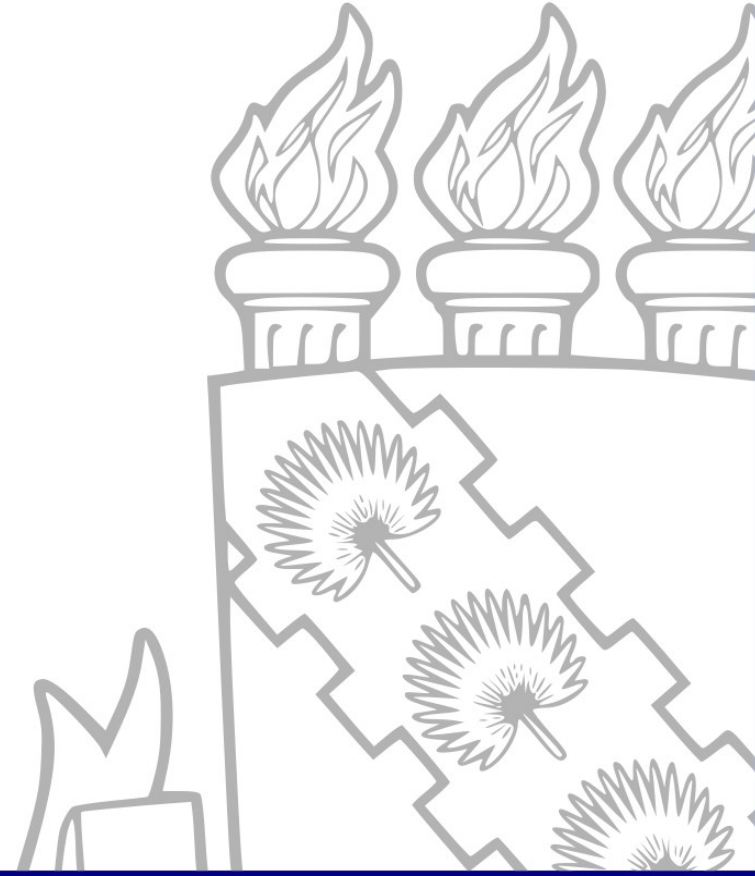
•



# Programas utilizados

- Eclipse
- Compilador Proto
- Gerenciador Maven

•



# Passos

- Defina formatos de mensagem em um .proto arquivo.
- Use o compilador de buffer de protocolo.
- Use a API de buffer de protocolo Java para escrever e ler mensagens



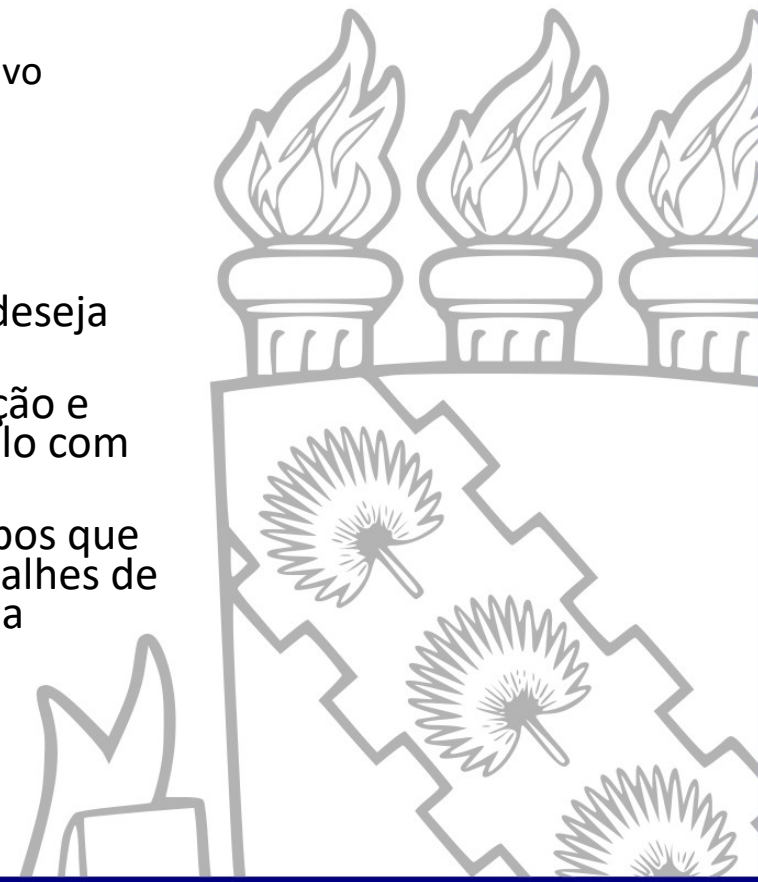
# Porque utilizar buffers de protocolo?

- Utilizaremos um Catálogo de endereços

- Gravar e Ler detalhes dos contatos das pessoas em um arquivo
- Cada pessoa tem: nome, ID, email e telefone

- Serializar e recuperar dados estruturados

- Solução flexível
- “.proto” é uma descrição da estrutura de dados que deseja armazenar.
- Compilador cria uma classe que implementa codificação e análise automática dos dados do buffer de protocolo com um formato binário eficiente.
- A classe gerada fornece getters e setters para os campos que constituem um buffer de protocolo e cuida dos detalhes de leitura e gravação do buffer de protocolo como uma unidade



# Utilização do .proto

## • Modificadores dos campos

- **Message:** indica um dado estruturado, semelhante à Struct em C.
- **Opcional:** o campo pode ou não ser definido
- **Repeated:** o campo pode ser repetido qualquer número de vezes (incluindo zero)
- **Required:** um valor para o campo deve ser fornecido, caso contrário, a mensagem será considerada "não inicializada"

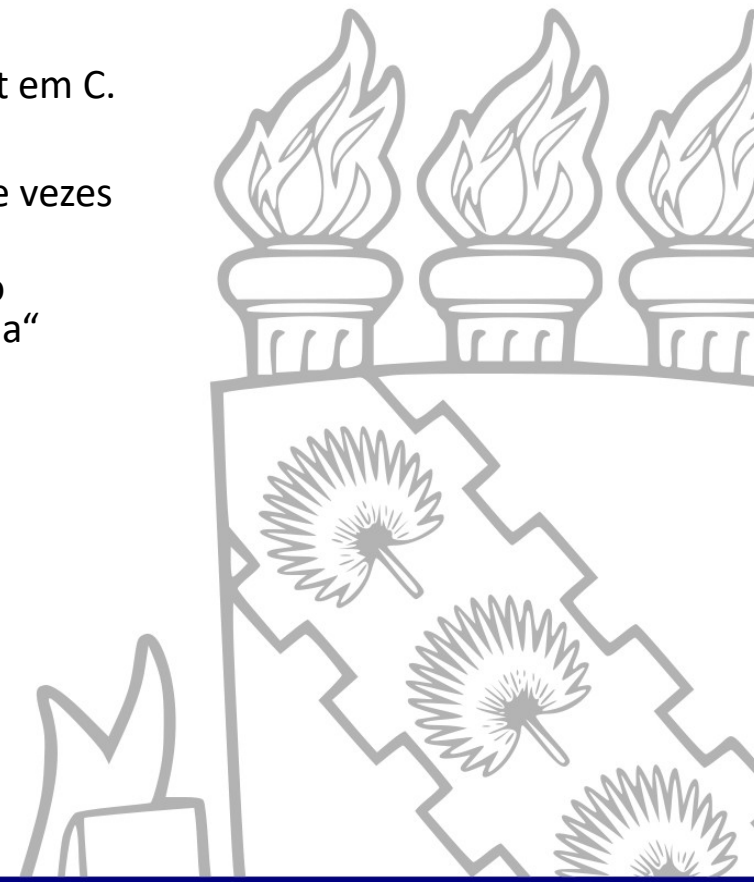
```
syntax = "proto2";

package tutorial;

option java_multiple_files = true;
option java_package = "com.example.tutorial.protos";
option java_outer_classname = "AddressBookProtos";

message Person {
  optional string name = 1;
  optional int32 id = 2;
  optional string email = 3;

  enum PhoneType {
    MOBILE = 0;
    HOME = 1;
    WORK = 2;
  }
}
```



# Compilador Protoc

- Compilando o nosso arquivo .proto

```
protoc -I=$SRC_DIR --java_out=$DST_DIR $SRC_DIR/addressbook.proto
```

- **\$SRC\_DIR**: Path do diretório da aplicação
- **\$DST\_DIR**: Path do diretório que receberá os códigos gerados (geralmente igual a SRC\_DIR)
- **\$SRC\_DIR**: Path do diretório onde está armazenado o arquivo .proto
- 

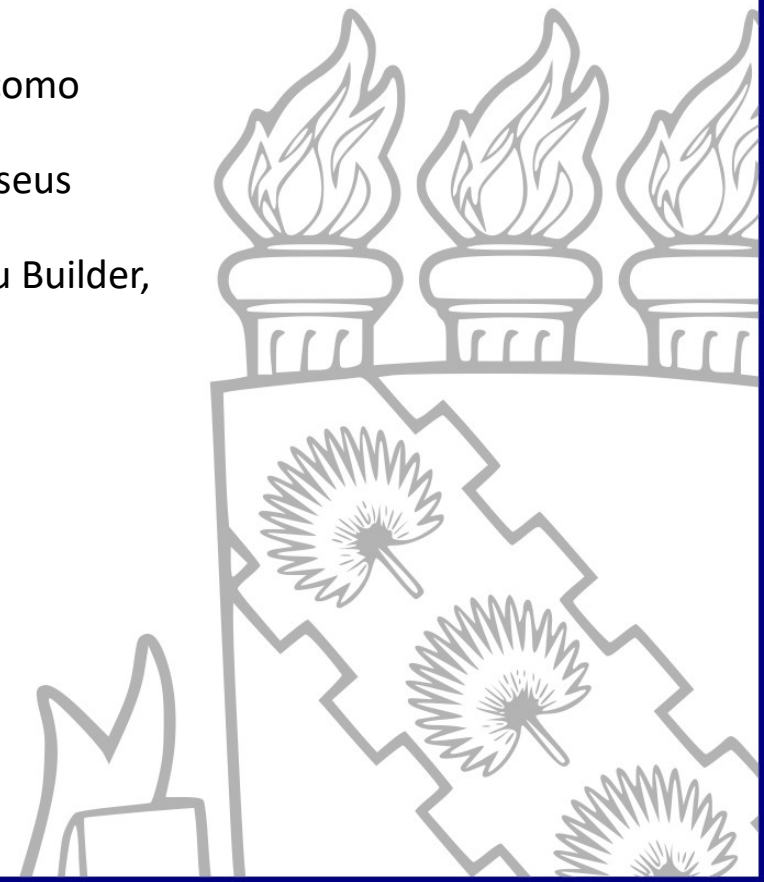




# API

## •Builder

- As classes geradas pelo compilador protoc são imutáveis (como String em java).
- Uma vez que o objeto é construído, ele não pode mais ter seus atributos modificados.
- Para construir um objeto, temos que primeiro construir seu Builder, “setar” os atributos e finalmente chamar build().





# API

## • Empacotamento e Desempacotamento

- **byte[] toByteArray();**
  - Serializa o objeto em array de bytes.
- **static Object parseFrom(byte[] data);**
  - A partir de uma array de bytes, constrói o objeto correspondente.
- **void writeTo(OutputStream output);**
  - serializa o objetos e escreve em um OutputStream.
- **static Object parseFrom(InputStream input);**
  - Lê um InputStream e constrói o objeto relacionado.

# Dúvidas?



Obrigado!

