

Objetivos

- Suporte para internacionalização (multilíngua) **i18n**;
- Traduzir uma aplicação em vários idiomas utilizando o software **poedit**;

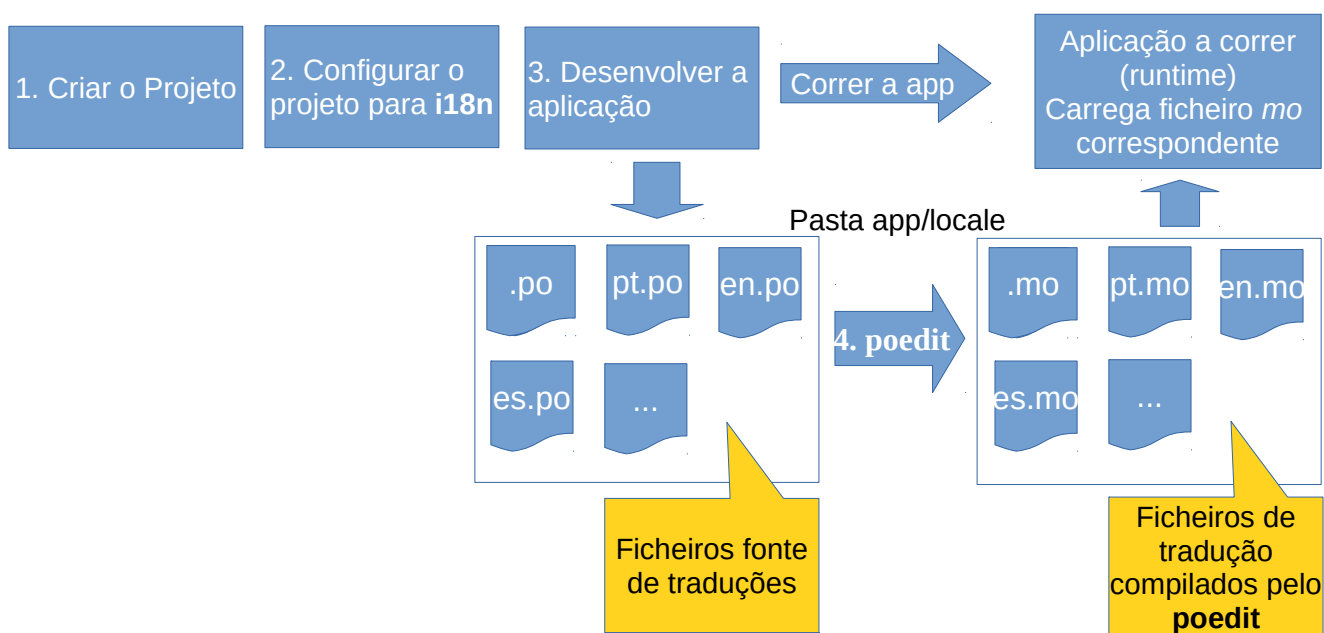
Introdução

Atualmente não faz sentido criar aplicações que não suportem internacionalização. Qualquer programador deve preparar o seu software de modo a que possa ser utilizado em qualquer parte do mundo. A internacionalização é uma dessas formas pois permite, sem grande esforço traduzir toda a interface com o utilizador para um determinado idioma. A internacionalização de um software também é conhecida pelo acrónimo **i18n**.



O *numeronym* para *internationalization* é **i18n** (significa que a palavra *internationalization* começa pela letra **i** seguem-se **18** caracteres e termina com um **n** ficando **i18n**)

No Lazarus, para que um projeto tenha suporte para i18n é necessário efetuar os seguintes passos:



Em primeiro lugar cria-se o projeto. De seguida configura-se o projeto para utilizar i18n e cria-se a pasta de nome *locale* dentro da pasta da aplicação. Desenvolve-se a aplicação criando forms e adicionando controlos aos forms e ao compilar o programa, a pasta locale fica com um ficheiro com o nome da aplicação e com a extensão “.po”, por exemplo “app.po”.

Para se editar (traduzir) os ficheiros “.po”, deve-se utilizar a ferramenta opensource poedit (<http://www.poedit.net/>). Através desta ferramenta podemos gravar os ficheiros em vários idiomas inserindo o código do país no meio, por exemplo (app.pt.po ou app.es.po, para português ou espanhol,

respetivamente).

Mediante o idioma configurado no sistema operativo para o utilizador a aplicação carrega o ficheiro de língua compilado “.mo” e apresenta a interface nesse idioma.



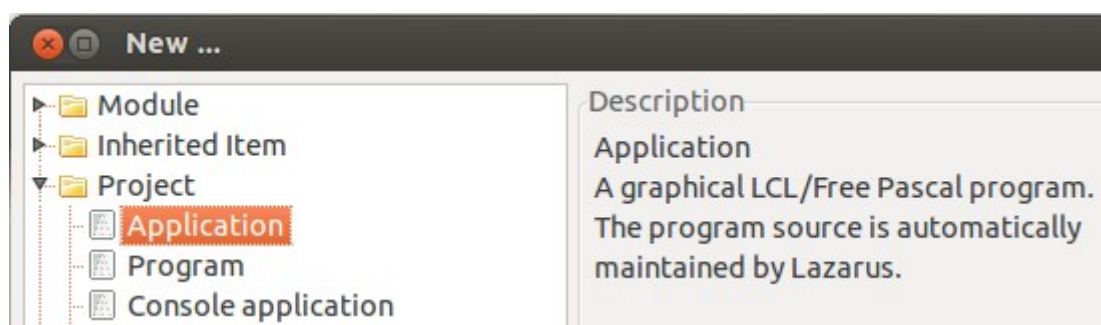
Para testar uma aplicação em várias línguas, deve-se compilar a aplicação e corrê-la a partir da linha de comandos passando o idioma por parâmetro:

Por exemplo, para correr a aplicação *app* em inglês, escrever o seguinte comando na pasta onde está o ficheiro executável:

```
./app -l en
```

Tarefa 1 - Criar o projeto

1. Crie uma nova pasta para este projeto. Por exemplo i18n.
2. Crie um novo projeto no lazarus do tipo “Application”;



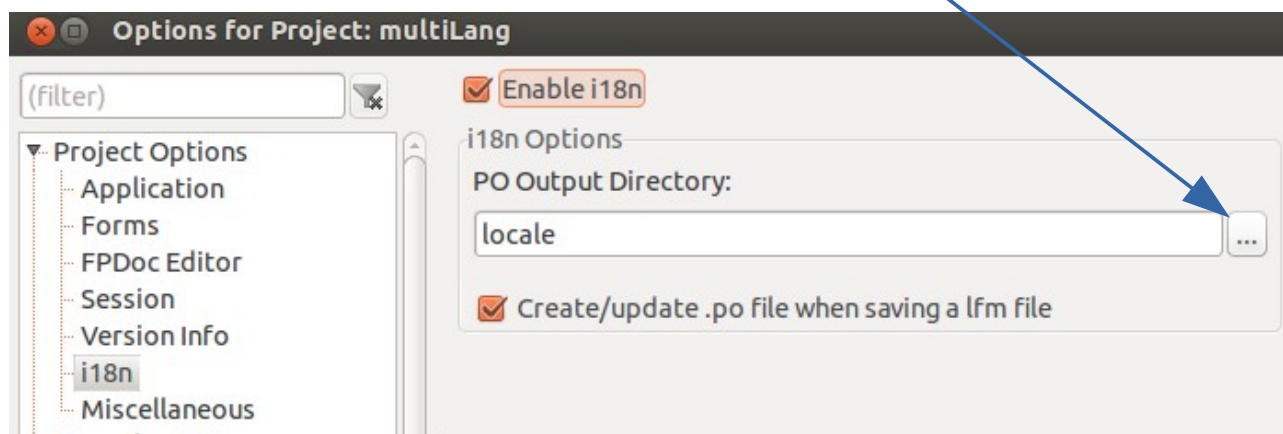
3. Grave o projeto como “multiLang”;
nota: atribua o nome “multiLang” ao projeto e “Main” à unit1;

Tarefa 2 – Ativar suporte para i18n

4. Através do menu “Project” do Lazarus, selecione “Project options...”;

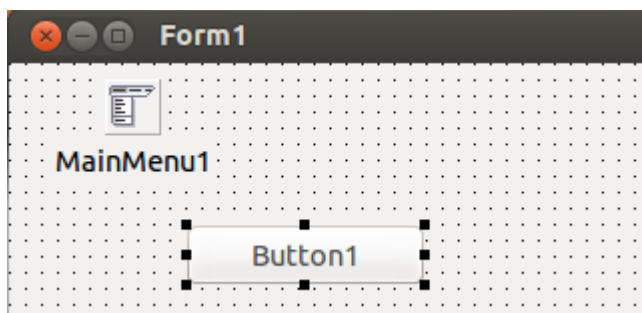


Atenção: deverá criar a pasta *locale* clicando no botão [...]



Tarefa 3 – Desenvolver a aplicação

5. Insira um botão e menu no seu form:



Não altere as *Captions* dos controlos. Fará isso depois quando traduzir o ficheiro.

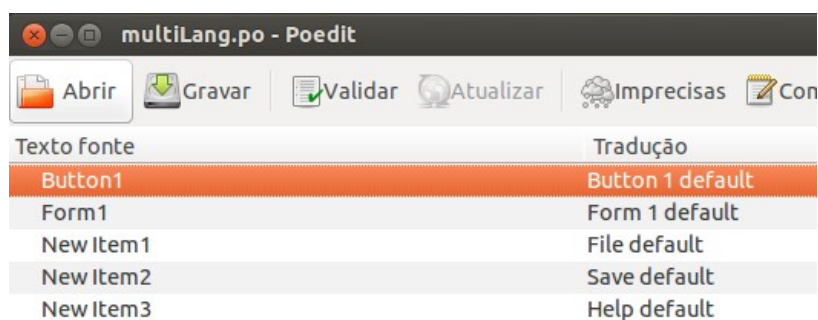
6. No início da unit Main.pas, acrescente o *default translator* na lista de uses:

```
Uses (... ) DefaultTranslator;
```

7. Compile e execute o programa (carregando em F9);
8. Note que depois de compilar o programa a pasta *locale* passa a ter o ficheiro **multiLang.po**

**Tarefa 4 – Traduzir o ficheiro po em vários idiomas**

9. Através da aplicação poedit abra o ficheiro **multiLang/locale/multilang.po**

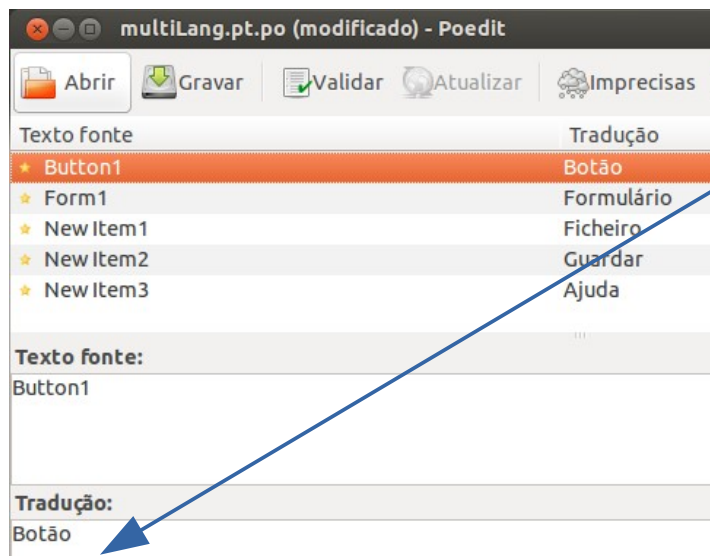


10. Faça uma cópia para a tradução em português (pt). Para isso escolha "Ficheiro > Guardar como..." e escreva o novo nome **multiLang.pt.po**



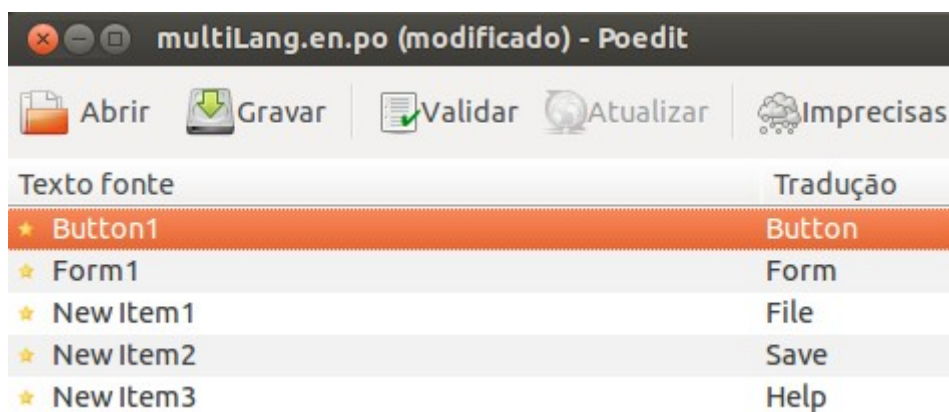
Atenção: deverá guardar este ficheiro na mesma pasta *locale*

11. Faça a tradução dos diferentes componentes neste novo ficheiro, em Português:



Faça a tradução na última caixa que diz "tradução"

12. Depois de traduzir todos e gravar a tradução multiLang.pt.po, faça uma nova cópia agora para o idioma inglês em multiLang.en.po



Repare que na pasta locale passou a ter igualmente ficheiro com a extensão ".mo" que são os ficheiros de língua compilados.

13. Corra o programa e verifique o funcionamento com o idioma pt;

14. Corra o programa no idioma inglês;