SISTEMA DE GERENCIAMENTO DE TAREFAS EM C++ COM INTEGRAÇÃO PYTHON

LINGUAGENS DE PROGRAMAÇÃO

LÍGIA BONIFÁCIO

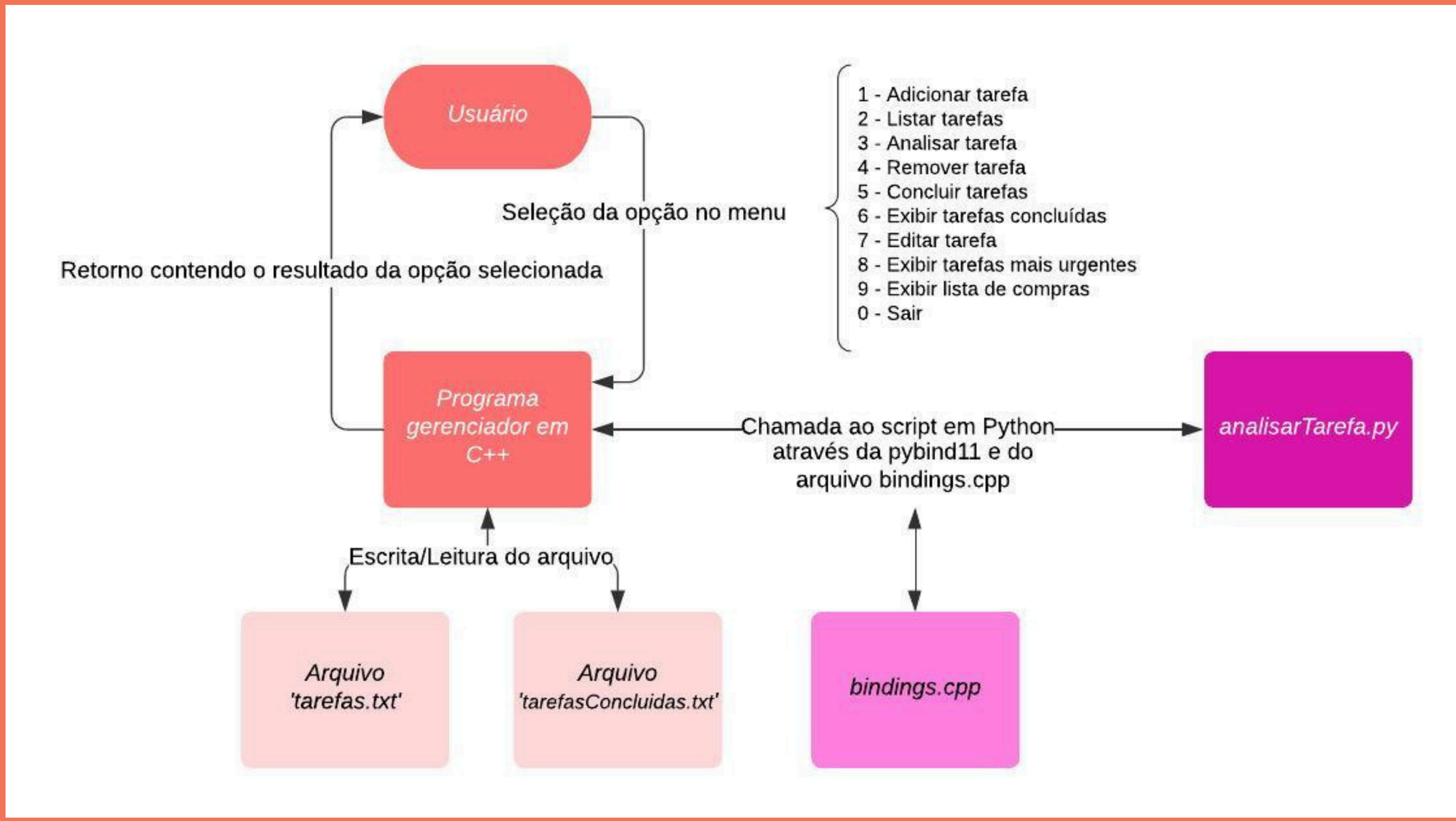
OBJETIVOS

- Desenvolver um **sistema organizacional pessoal** baseado em listas de tarefas (to-do lists) para aumentar a produtividade e organizar atividades diárias.
- Compreender a **interligação entre duas diferentes linguagens de programação**.

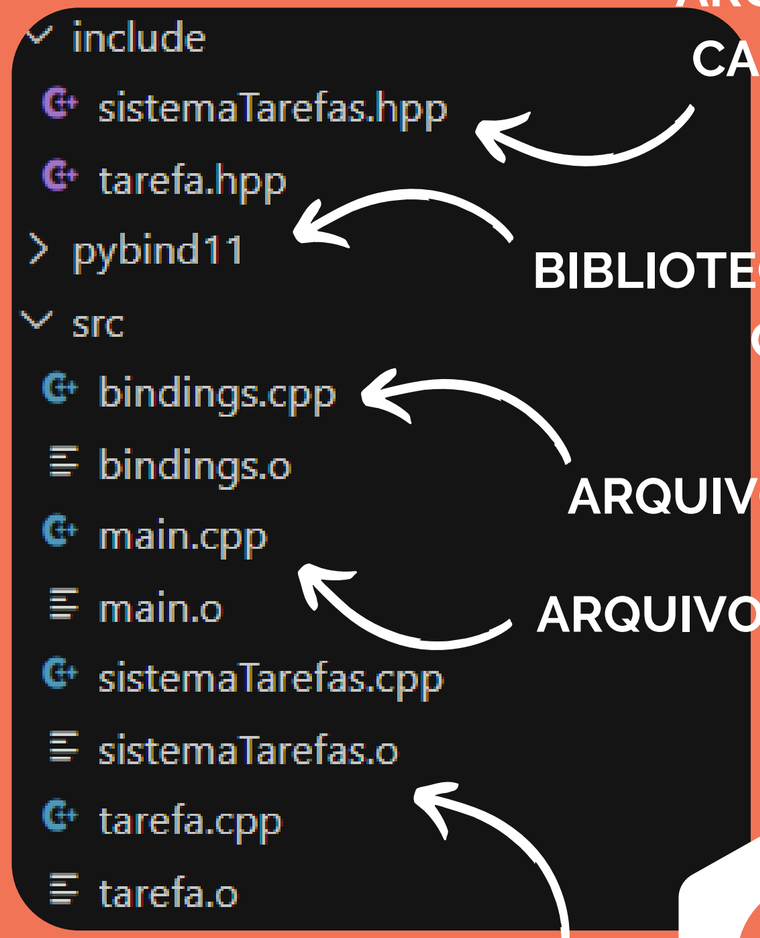
FUNCIONALIDADES

- **Adição de Tarefas**
- **Remoção de Tarefas**
- **Edição de Tarefas**
- **Conclusão de Tarefas**
- **Análise de Tarefas (utilizando Python)**

ESTRUTURA DO PROJETO



ESTRUTURA DO PROJETO



ARQUIVOS DE CABEÇALHO

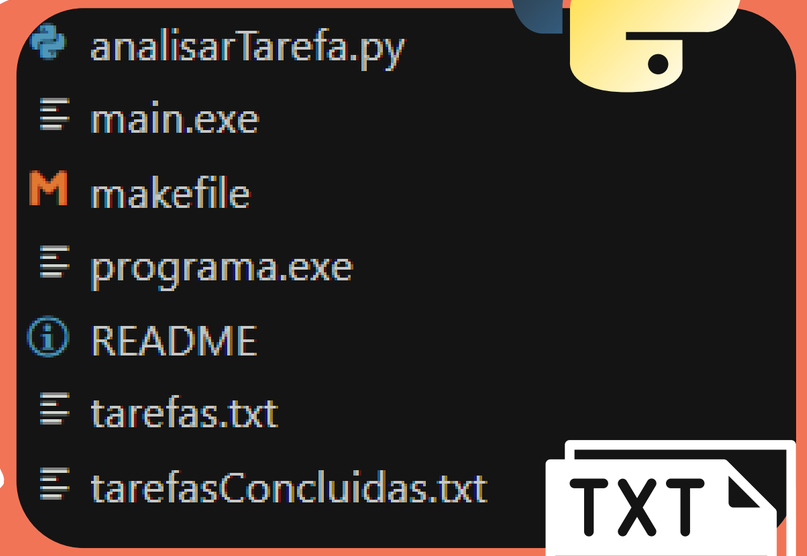
BIBLIOTECA PARA INTEGRAÇÃO COM PYTHON

ARQUIVO DE BINDING

ARQUIVO PRINCIPAL

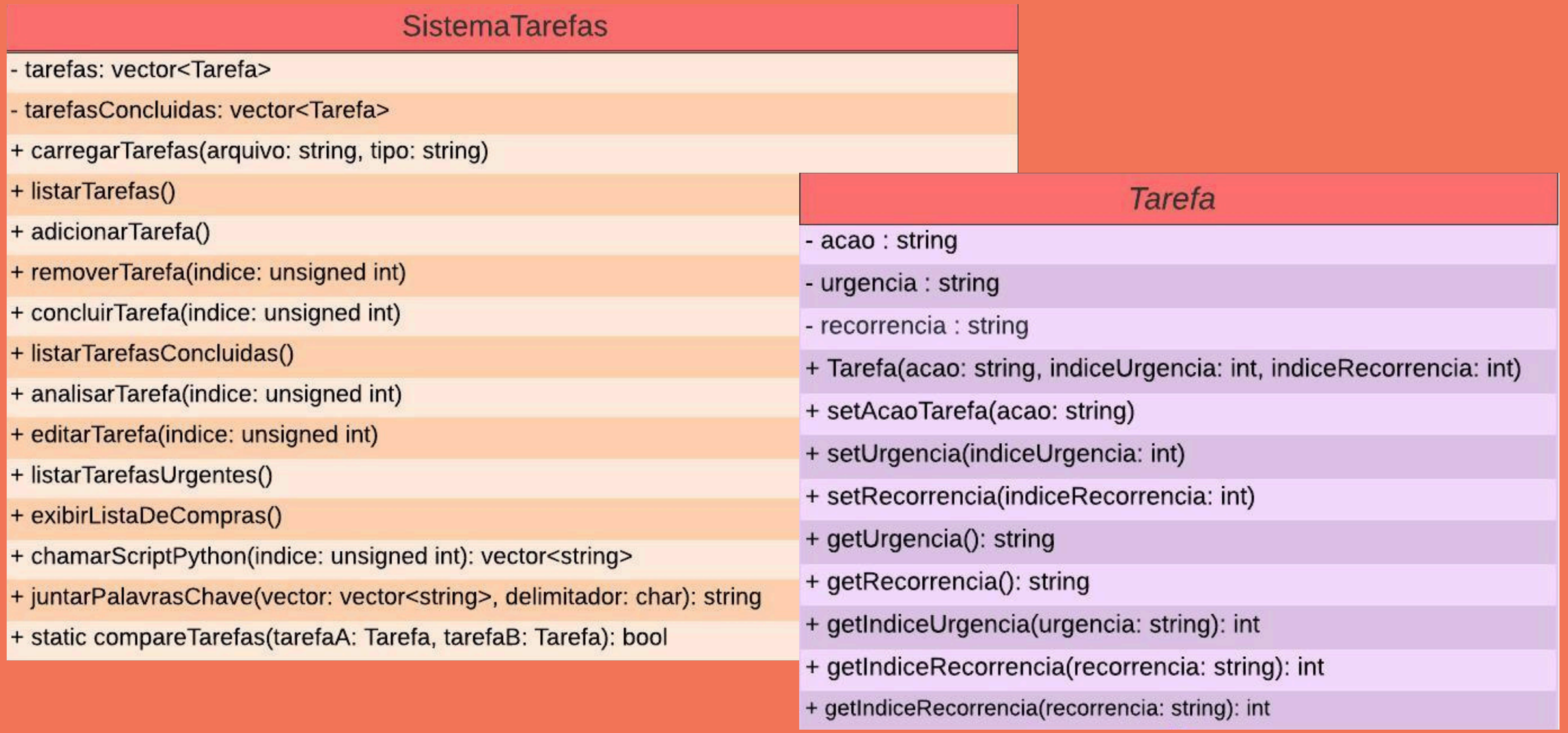
iMPLEMENTAÇÃO DAS CLASSES

FUNÇÕES EM PYTHON



PERSISTÊNCIA DE DADOS

DIAGRAMA UML



INTERAÇÃO COM O USUÁRIO

```
Prompt de Comando - make
Microsoft Windows [versão 10.0.22621.3593]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\ligia>cd C:\Users\ligia\Documents\UFRJ\Linguagens de Programacao\projects\Trabalho Final\Código

C:\Users\ligia\Documents\UFRJ\Linguagens de Programacao\projects\Trabalho Final\Código>make
make: Nothing to be done for 'all'.

C:\Users\ligia\Documents\UFRJ\Linguagens de Programacao\projects\Trabalho Final\Código>make
g++ -std=c++11 -O2 -I./pybind11/include -IC:/msys64/ucrt64/include/python3.11 -I./include -c src/main.cpp -o src/main.o
g++ -std=c++11 -O2 -I./pybind11/include -IC:/msys64/ucrt64/include/python3.11 -I./include src/bindings.o src/sistemaTare
fas.o src/tarefa.o src/main.o -o main -LC:/msys64/ucrt64/lib -lpython3.11
cmd /c copy ./data\tarefas.txt .
Microsoft Windows [versão 10.0.22621.3593]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\ligia\Documents\UFRJ\Linguagens de Programacao\projects\Trabalho Final\Código>
```

PYTHON

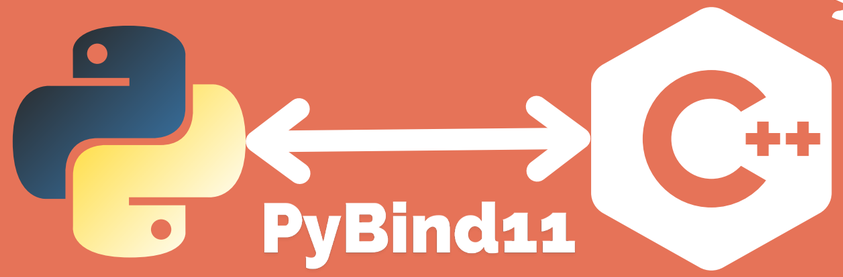
C++

analisarTarefa.py > ...

```python
import string

def contar_palavras(texto):
    palavras = texto.split()
    return len(palavras)
```

```cpp
vector<string> SistemaTarefas::chamarScriptPython(unsigned int indice) {
    vector<string> resultado;
    py::initialize_interpreter();
    if (indice < 1 || indice > tarefas.size()) {
        cout << "Indice de tarefa invalido." << endl;
        return resultado;
    }

    string conteudoTarefa = tarefas[indice - 1].getAcao();
```

```cpp
try {
    py::module_ script = py::module_::import("analisarTarefa");
    auto contar_palavras_func = script.attr("contar_palavras");
    auto identificar_palavras_chave_func = script.attr("identificar_palavras_chave");

    int contagemPalavras = contar_palavras_func(conteudoTarefa).cast<int>();
```




```cpp
    resultado.push_back(conteudoTarefa);
    resultado.push_back(to_string(contagemPalavras));
    resultado.push_back(juntarPalavrasChave(palavrasChave, ' '));
} catch (const py::cast_error& e) {
    cerr << "Erro de casting ao chamar o script Python: " << e.what() << endl;
} catch (const py::error_already_set& e) {
    cerr << "Erro ao chamar o script Python: " << e.what() << endl;
}
py::finalize_interpreter();
return resultado;
```

CONCLUSÃO

- Este projeto demonstrou a **integração entre C++ e Python** para a criação de um sistema de gerenciamento de tarefas. Utilizando C++ e Python, consegui desenvolver uma aplicação capaz de organizar, analisar e persistir dados de tarefas de forma eficiente. A **biblioteca Pybind11** foi essencial para facilitar essa comunicação, permitindo a **chamada de funções Python diretamente do código C++**. Através deste trabalho, entendi a importância da combinação de diferentes tecnologias para atender a requisitos específicos de processamento de dados.

MUITO OBRIGADO !