UNIVERSIDADE FEDERAL DE PERNAMBUCO
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
CENTRO DE INFORMÁTICA

Implementação de biblioteca de tempo

JEFERSON SEVERINO DE ARAUJO
MARIA LUÍSA DOS SANTOS SILVA
TALES VINÍCIUS ALVES DA CUNHA

PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO

RECIFE
2025

Motivação

As bibliotecas de manipulação de tempo, como a chrono em C++ e a time em Python, são fundamentais para aplicações que necessitam de funcionalidades baseadas em tempo, como obter timestamps atuais, calcular a diferença entre eles ou realizar conversões. Inspirado por essa necessidade, nosso projeto consiste em estender a Linguagem Imperativa 2 (LI2) com um sistema de tipos temporais robusto, adicionando não apenas o tipo Timestamp, mas também o tipo Duration e um conjunto completo de operações sobre eles.

Tipos

Para modelar o tempo de forma precisa, propomos a criação de dois novos tipos:

- Timestamp: Representa um ponto específico no tempo, no formato (hora, minuto, segundo, [fuso_horário_opcional]). O fuso horário será um componente opcional para maior flexibilidade.
- Duration: Representa um intervalo ou uma quantidade de tempo, permitindo uma álgebra temporal mais rica e clara.

A introdução desses dois tipos é inspirada em projetos anteriores de extensão de tipos na LI2, como BigInt e BigFraction, que exigiram a implementação e manipulação de valores complexos de forma estruturada.

Álgebra Temporal

A base da nossa extensão será um conjunto de operações essenciais que permitirão a manipulação dos novos tipos. Para isso, será necessário estender as regras de `ExpBinaria` e `ExpUnaria` na gramática da LI2. As operações planejadas são:

- **Adição e Subtração de Timestamps**: Operações que resultam em um `Duration`.
- **Operações de Comparação**: Implementação dos operadores lógicos (`==`, `!=`, `>`, `<`, `>=`, `<=`) para comparar dois `Timestamps` (para ordenação cronológica) ou duas `Durations` (para comparar durações). Esta é uma operação binária fundamental que também estenderá a gramática `ExpBinaria`.
- **Diferença entre Timestamps**: Cálculo preciso do intervalo entre dois pontos no tempo.

- **Conversão de Fuso Horário**: Uma função para converter um `Timestamp` de um fuso para outro (ex: `convert_tz(timestamp, fuso_destino)`).

Operações Complexas

Para além da álgebra básica, implementaremos operações binárias que envolvem a interação entre os novos tipos e os tipos existentes, como:

- Timestamp + Duration -> Timestamp
- Timestamp - Duration -> Timestamp
- Duration * Int -> Duration
- Duration / Int -> Duration

Adicionalmente, criaremos funções para extrair componentes específicos, como get_hour(Timestamp) ou get_day(Timestamp), seguindo o padrão de extensões de tipos numéricos realizadas no projeto BigInt & BigFraction.

Parsing e I/O

Finalmente, para garantir a usabilidade, implementaremos funcionalidades de entrada e saída (I/O):

- **Parsing**: Uma função unária para converter uma string em um Timestamp (ex: parse_timestamp("10:30:00")).
- **Formatação**: Uma função para converter um Timestamp em uma string formatada (ex: format(timestamp, "HH:MM:SS")).