



UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ARTES, CIÊNCIAS E HUMANIDADES
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

OTÁVIO RODRIGUES BAMBANS

**Gerador Automático de Código (programas) Baseado em Grafos de
Conhecimento**

São Paulo

2022

OTÁVIO RODRIGUES BAMBANS

**Gerador Automático de Código (programas) Baseado em Grafos de
Conhecimento**

Projeto de Iniciação Científica apresentado à
Escola de Artes, Ciências e Humanidades da
Universidade de São Paulo.

Orientador: Prof. Dr. Fábio Nakano

Coorientador: Prof. Dr. José de Jesús Pérez
Alcázar

São Paulo

2022

Introdução

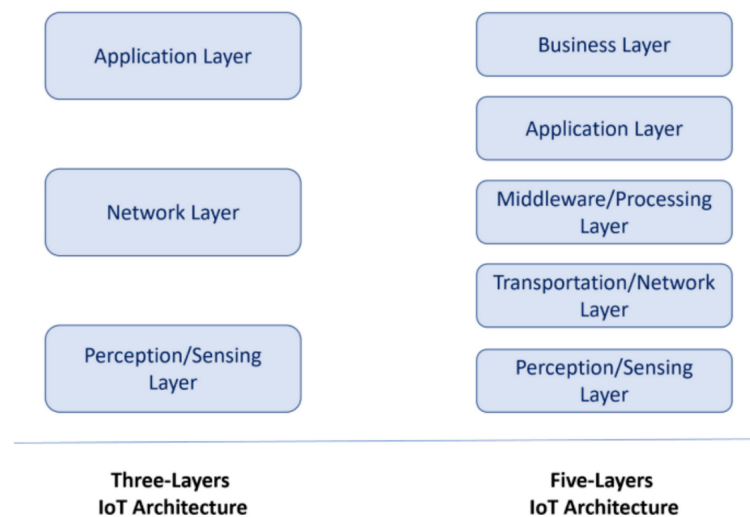
Hoje em dia, as cidades têm se tornado estruturas socioeconômicas cada vez mais complexas de se administrar, ao passo que demandam mais recursos, em geral, escassos. Tendo isso em vista, surgiu, na era da computação, o contexto de “Cidades Inteligentes” (BELLINI; NESI; PANTALEO, 2022). Estas são cidades que fazem uso de sensores e atuadores eletroeletrônicos conectados à redes computacionais que processam, em sentido amplo, os dados coletados. Dessa forma, a interpretação das informações geradas torna-se um importante fator na tomada de decisão em relação aos mais diversos aspectos, nos setores públicos e privados, nos mais diversos níveis hierárquicos, pois têm alta taxa de atualização, alta qualidade e granularidade fina.

Não só, dentro do contexto de Cidades Inteligentes, existe o conceito de IoE - *Internet of Everything* (Internet de Tudo) - definido como uma conexão entre pessoas, processos, dados/informações e coisas (CISCO, 2022), podendo este ser considerado uma generalização do conceito de IoT - *Internet of Things* (Internet das Coisas) - definido, de forma simples, como a interconexão digital entre objetos (ORACLE, 2022) - estes objetos podem ser das mais diversas naturezas, como geladeiras, fogões, micro-ondas, televisões, lâmpadas, relógios, aspiradores de pó, aparelhos de ar-condicionado etc.

Dada uma alta demanda para se desenvolver sistemas que integrem diversas soluções de IoT, surgiu a necessidade de que fossem criadas pesquisas na área, discorrendo sobre como aumentar a eficiência nos aspectos gerenciais e computacionais. Dessa forma, foi percebido que todo o trabalho inteligente é feito ainda por pessoas e o trabalho mecanizado, pelos computadores. Assim, nasceu um novo campo de estudo, hoje tratado como “Web Semântica” (BERNERS-LEE *et al.*, 2001). Este novo campo de estudo trata das relações entre pessoas e computadores, estabelecendo padrões semânticos na criação de vocabulários e regras para interoperação dos dados, *id est*, as informações possuem significado explícito, permitindo que os computadores entendam as informações tanto quanto os humanos que os operam (ANTONIOU, 2012). Não só, estruturas como Grafos de Conhecimento são usadas, por sua generalidade, para armazenar dados e informações as quais, por exemplo, podem ser aproveitadas por ferramentas CASE (*Computer-Aided Software Engineering*) na criação de ferramentas de apoio da Engenharia de Software.

Complementarmente, a Web Semântica se faz necessária, uma vez que o conhecimento é gerado de forma dinâmica e os computadores, na Web 3.0, trabalham de forma estática, baseados em *keywords* (ANTONIOU, 2012). Portanto, são necessárias ferramentas que reduzam os empasses e o esforço a ser empregado no desenvolvimento de novos sistemas, principalmente nos que fazem parte de "ecossistemas inteligentes", proporcionando dinamismo, facilitação de uso amplo, e.g. no mercado de trabalho, e redução de latência, uma vez que soluções integradas inteligentes são custosas e complexas demais, resultando na demora da entrega da solução.

Figura 1 – Arquiteturas comuns em IoT.

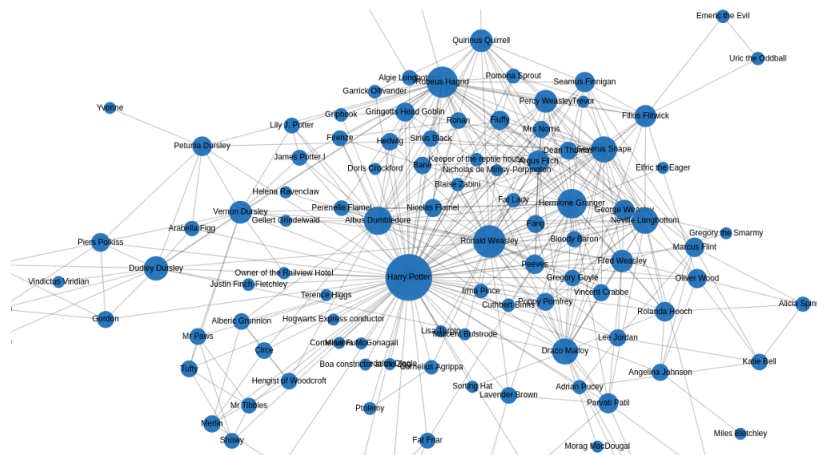


Fonte – (BELLINI; NESI; PANTALEO, 2022)

Sendo assim, existindo os elementos de geração de dados (sensores e atuadores), conexão/transmissão dos dados (IoT) e os padrões estabelecidos (Web Semântica), organizados em alguma arquitetura - tal como na Figura 1 - precisa-se ainda de alguma forma armazenar estes dados/informações, em alguma estrutura que seja, em certos aspectos, geral e abstrata. Para tanto, nascem os Grafos de Conhecimento: estruturas que detém o poder de guardar informações de fontes estruturadas, semi-estruturadas ou não estruturadas, tendo a capacidade de generalizar e abstrair o conhecimento (*e.g.*, o livro *Harry Potter and the Philosopher's Stone* na Figura 2). Por definição, estes grafos são construídos por vértices, os quais armazenam conceitos ou entidades - como categorias ou interfaces com objetos físicos do mundo real - e arestas, as quais representam as relações semânticas entre os vértices do grafo - daí a necessidade de se construir um vocabulário/ontologia (dada a definição de T. R. Gruber e R. Studer, uma ontologia é uma especificação explícita

e formal de uma concepção compartilhada (STAAB; STUDER, 2009)), de forma a se standardizar as interações entre os nodos. Não só, as informações são armazenadas de forma fragmentada - com baixa granularidade, possibilitando que as análises, partindo das relações, gerem conhecimentos mais precisos, formando soluções completas e estruturadas, o que facilita a gestão do conhecimento para uso e compreensão.

Figura 2 – O livro *Harry Potter and the Philosopher's Stone* transformado em um Grafo de Conhecimento.



Fonte – <https://neo4j.com/developer-blog/turn-a-harry-potter-book-into-a-knowledge-graph/>

Dada a estrutura dos grafos de conhecimento, a W3C padronizou o RDF (*Resource Description Framework*) como o *framework* de representação de informação na Internet (CARROLL; KLYNE, 2004). Neste *framework*, as informações são tais que, para qualquer relação P, chamada de Predicado, temos um Sujeito x atrelado a um Objeto y, portanto, triplas (x,P,y) ou fórmulas lógicas $P(x,y)$ são formadas, relacionando dados na construção de informação. Não só, a W3C ainda padronizou o SPARQL (*SPARQL Protocol and RDF Query Language*) como a linguagem de manipulação de banco de dados orientado à grafos padrão para uso em ferramentas que fazem uso do RDF (SEABORNE; HARRIS, 2013).

Note-se que a aplicação da Web Semântica no armazenamento e análise dos dados gerados pelos sensores em uma cidade inteligente é assunto razoavelmente estudado, que já conta com arcabouços comerciais como NEO4J (NEO4J, 2021). Por outro lado, arcabouços baseados em Web Semântica para prototipação e implantação de programas visando acelerar essas etapas, logo, armazenando módulos reusáveis do código-fonte em um grafo de conhecimento para subsequente recuperação, parametrização e composição, seja automática ou semi-automática, ainda estão por chegar a esse grau de maturidade. Por exemplo, *SWoTSuite* (PATEL *et al.*, 2017), apresenta um arcabouço para prototipação

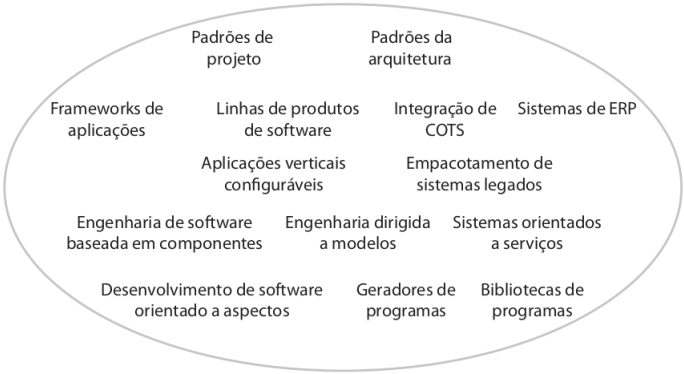
de aplicações em Internet das Coisas, mas apresenta mais detalhes sobre a arquitetura geral da aplicação que à parametrização e construção ¹ dos programas que são executados nos elementos dessa arquitetura.

A fim de direcionar este projeto para uma primeira exploração sobre parametrização e construção automática ou semi-automática de programas baseados em grafos de conhecimento, há necessidade de especificar alguns aspectos do desenvolvimento de programas.

De acordo com Engenharia de Software, existem diversas ferramentas de apoio (Ferramentas CASE), dentre as quais há um grupo conhecido como Ferramentas de Geração de Código e, de forma mais específica, usar-se-á inicialmente no projeto um grupo determinado como "Geradores de código baseados em modelos". Estes geradores têm por característica gerar de forma dinâmica códigos para fins especificados no projetos, preenchendo modelos estáticos previamente formulados - os modelos podem variar, neste caso, a depender dos parâmetros de entrada que o gerador recebe.

Das mais variadas metologias ágeis de desenvolvimento, SCRUM é a que mais se adéqua ao trabalho a ser desenvolvido, uma vez que é uma metodologia iterativa e não prescreve métodos de programação específicos, se encaixando muito bem em abordagens mais técnicas (SOMMERVILLE, 2022). Não só, parte crucial do desenvolvimento que se está construindo galga-se no reúso de software, a partir de modelos de software, assim como no panorama exibido na Figura 3, preenchidos com informações sobre o domínio do problema a partir de um Grafo de Conhecimento. Assim, esta combinação torna o projeto possível, uma vez que SCRUM é flexível e o reúso de software é um método concordante com a metodologia apresentada.

Figura 3 – O panorama de reúso.



Fonte – (SOMMERVILLE, 2022)

¹ em Web Semântica, composição poderia ser aplicado

Por fim, tecnicamente falando, o objetivo desta iniciação científica é explorar a automatização da parametrização e construção de código usando grafos de conhecimento. Com um pouco mais de detalhes, criar um gerador de código baseado em grafos de conhecimento, de certa forma, uma extensão do conceito de geradores de código baseados em modelos, uma vez que os modelos (fragmentos de programas) e dados usados pelo sistema a ser desenvolvido estarão armazenados em um banco de dados orientado a grafos e o uso, gerenciamento e interoperabilidade dos dados estará galgada nos conceitos de IoT e nos padrões da Web Semântica. Colocando de maneira mais ampla, espera-se agilizar a programação e implantação de serviços em Cidades Inteligentes.

Objetivos gerais

Agilizar a programação e implantação de serviços de software em Cidades Inteligentes, automatizando a parametrização e construção de código usando grafos de conhecimento.

Objetivos específicos

- Prototipar, em Python, o gerador de código, seguindo os princípios da Web Semântica, RDF e SPARQL;
- Testar o protótipo da ferramenta, gerando código para programar sensores IoT de temperatura e umidade do ar;
- Disponibilizar o código-fonte do gerador automático no Github;
- Desenvolver as adjacências do projeto (API e aplicação *frontend*);
- Avaliar o desempenho e os resultados;
- Mantê-la.

Metodologia

- Reuniões semanais com orientadores para avaliação de andamento, com citações, e ajuste de atividades;
- Documentação do andamento em formato de semanário;

- Armazenamento de código em um repositório de código (Github) público: [Github](https://github.com/otaviobambans/CodeGenFromKnowledgeGraphs): <https://github.com/otaviobambans/CodeGenFromKnowledgeGraphs>;
- Documentação do código junto à ele no repositório do projeto;
- Consolidação da documentação na ferramenta [Overleaf](#).

Entregáveis

- Documentação do andamento atualizado do projeto;
- Código e sua respectiva documentação e especificação;
 - Manual do desenvolvedor: como instalar o ambiente de desenvolvimento, executar a aplicação, arquitetura e organização do código, referências usadas para o desenvolvimento e demais especificações;
- Relatórios parcial e final do projeto;
- Resumo para SIICUSP;
- Poster para SIICUSP.

Cronograma

Atividades	Meses do projeto
Idealização e projeção do trabalho	01, **, **, **, **, **, **, **, **, **, **, **
Planejamento e prototipação	**, 02, 03, **, **, **, **, **, **, **, **, **
Relatórios parciais	**, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12
Documentação atualizada do projeto	**, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12
Códigos e suas respectivas documentações	**, **, **, 04, 05, 06, 07, 08, 09, 10, 11, 12
Manual do usuário/desenvolvedor	**, **, **, **, 05, 06, 07, 08, 09, 10, 11, 12
Bateria de testes	**, **, **, **, **, 06, **, 08, **, 10, 11, 12
Entrega dos <i>pre-releases</i> do software	**, **, **, **, **, **, **, 08, 09, 10, 11, 12
Resumo para a SIICUSP	**, **, **, **, **, **, **, **, **, 10, 11, 12
Pôster para a SIICUSP	**, **, **, **, **, **, **, **, **, 10, 11, 12
Relatório final	**, **, **, **, **, **, **, **, **, 11, 12
Entrega do <i>final-release</i> do software	**, **, **, **, **, **, **, **, **, 11, 12

Referências

- ANTONIOU, G. e. a. *A Semantic Web primer*. 3. ed. [S.l.]: Massachusetts: Massachusetts Institute of Technology, 2012. Citado 2 vezes nas páginas 2 e 3.
- BELLINI, P.; NESI, P.; PANTALEO, G. IoT-enabled smart cities: A review of concepts, frameworks and key technologies. *Applied Sciences*, MDPI AG, v. 12, n. 3, p. 1607, fev. 2022. Disponível em: <https://doi.org/10.3390/app12031607>. Citado 2 vezes nas páginas 2 e 3.
- BERNERS-LEE; TIM; HENDLER; JAMES; LASSILA, O. The semantic web. *Scientific american*, JSTOR, v. 284, n. 5, p. 34–43, 2001. Citado na página 2.
- CARROLL, J.; KLYNE, G. *Resource Description Framework (RDF): Concepts and Abstract Syntax*. [S.l.], 2004. <https://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>. Citado na página 4.
- CISCO. *The internet of everything: Cisco ioe value index study*. 2022. Disponível em: https://www.cisco.com/c/dam/en_us/about/business-insights/docs/ioe-value-index-faq.pdf. Citado na página 2.
- NEO4J. 2021. Disponível em: <https://neo4j.com/company/>. Citado na página 4.
- ORACLE. *What is IoT?* 2022. Disponível em: <https://www.oracle.com/internet-of-things/what-is-iot/>. Citado na página 2.
- PATEL, P.; GYRARD, A.; DATTA, S. K.; ALI, M. I. Swotsuite: A toolkit for prototyping end-to-end semantic web of things applications. In: . Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2017. (WWW '17 Companion), p. 263–267. ISBN 9781450349147. Disponível em: <https://doi.org/10.1145/3041021.3054736>. Citado na página 4.
- SEABORNE, A.; HARRIS, S. *SPARQL 1.1 Query Language*. [S.l.], 2013. <https://www.w3.org/TR/2013/REC-sparql11-query-20130321/>. Citado na página 4.
- SOMMERVILLE, I. *Engenharia de software*. [S.l.: s.n.], 2022. ISBN 9788579361081. Citado na página 5.
- STAAB, S.; STUDER, R. (Ed.). *Handbook on Ontologies*. Springer Berlin Heidelberg, 2009. Disponível em: <https://doi.org/10.1007/978-3-540-92673-3>. Citado na página 4.