Extendiendo el lenguaje de consultas SPARQL

Adrián Soto

Pontificia Universidad Católica de Chile

Outline

Introducción

Recursión en SPARQL

SPARQL y JSON APIs

Ser alumno de doctorado

Recursión en SPARQL

Recursión en SPARQL

Recursión en SPARQL

Recursión en SPARQL

Recursión en SPARQL

Recursión en SPARQL

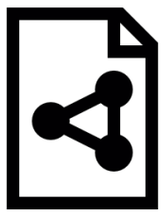
Recursión en SPARQL

JSON APIs en SPARQL

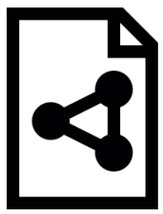
JSON APIs en SPARQL

JSON APIs en SPARQL

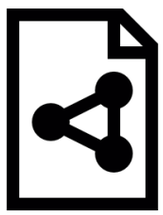
JSON APIs en SPARQL



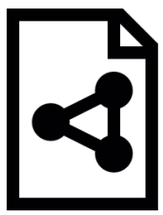
JSON APIs en SPARQL



JSON APIs en SPARQL



JSON APIs en SPARQL



Introducción

El lenguaje de consultas de la web semántica es SPARQL

Introducción

El lenguaje de consultas de la web semántica es SPARQL

Existen muchos endpoints de SPARQL activos en la actualidad

Introducción

El lenguaje de consultas de la web semántica es SPARQL

Existen muchos endpoints de SPARQL activos en la actualidad

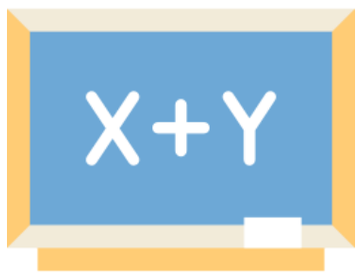
Pero, ¿tiene SPARQL todo lo que necesitamos?

Introducción

En esta charla

Mostraremos una propuesta para que SPARQL pueda:

- Soportar consultas recursivas
- Integrar datos que no necesariamente están en RDF, pero sí están en la web



Pero antes, algunos preliminares

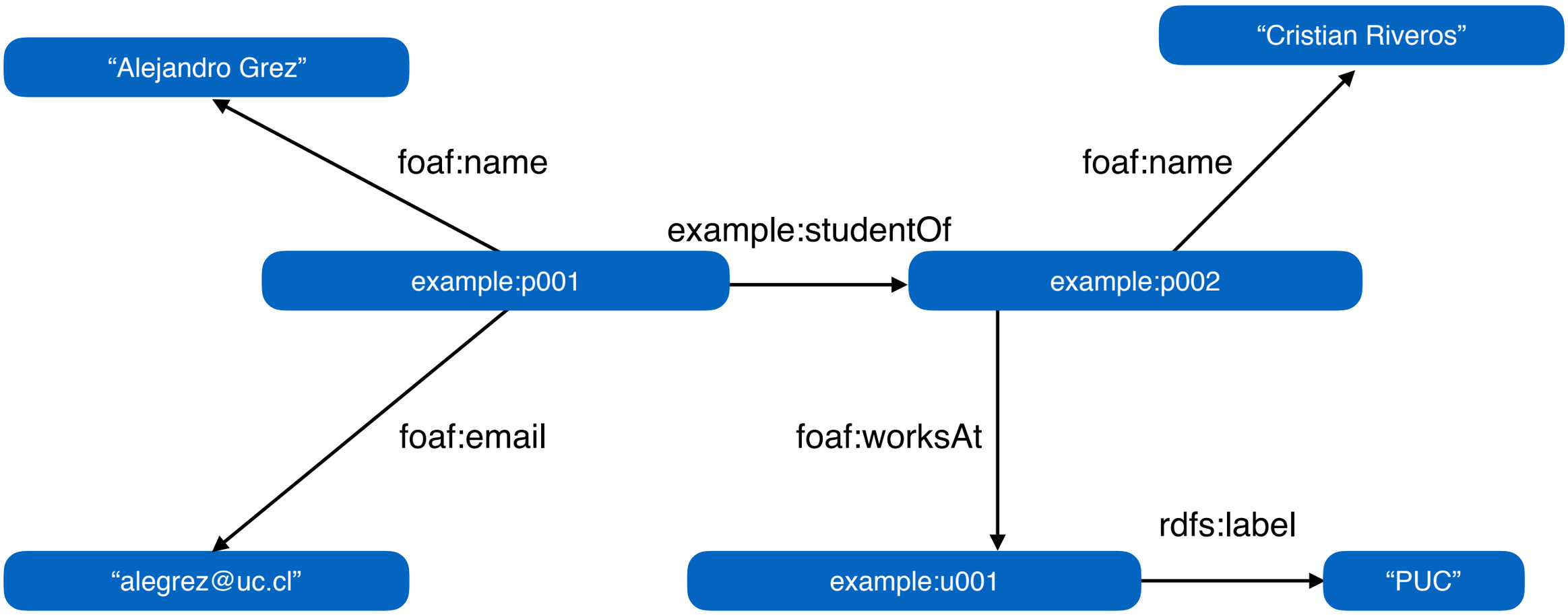
Preliminares

Resource Description Framework (RDF)

Sujeto	Predicado	Objeto
example:p001	foaf:name	“Alejandro Grez”
example:p002	foaf:name	“Cristian Riveros"
example:p001	example:studentOf	example:p002
example:p002	foaf:worksAt	example:u001
example:u001	rdfs:label	“PUC”

Preliminares

Resource Description Framework (RDF)



Preliminares

SPARQL

Todos los profesores de la universidad con label PUC

```
SELECT ?name WHERE
```

Preliminares

SPARQL

Todos los profesores de la universidad con label PUC

```
SELECT ?name WHERE
{
   ?universidad rdfs:label “PUC” .
```

Preliminares

SPARQL

Todos los profesores de la universidad con label PUC

```
SELECT ?name WHERE
{
   ?universidad rdfs:label “PUC” .
   ?prof foaf:worksAt ?universidad .
```

Preliminares

SPARQL

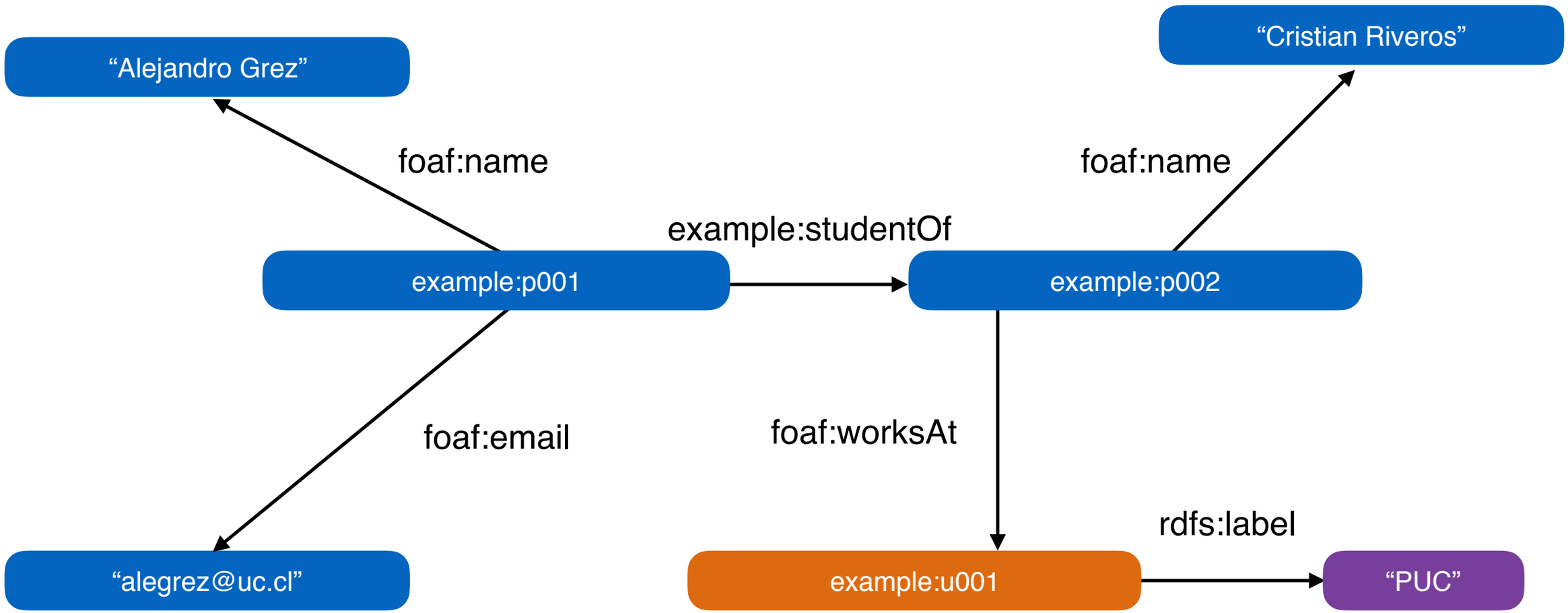
Todos los profesores de la universidad con label PUC

```
SELECT ?name WHERE
{
   ?universidad rdfs:label “PUC” .
   ?prof foaf:worksAt ?universidad .
   ?prof foaf:name ?name
}
```

Preliminares

Resource Description Framework (RDF)

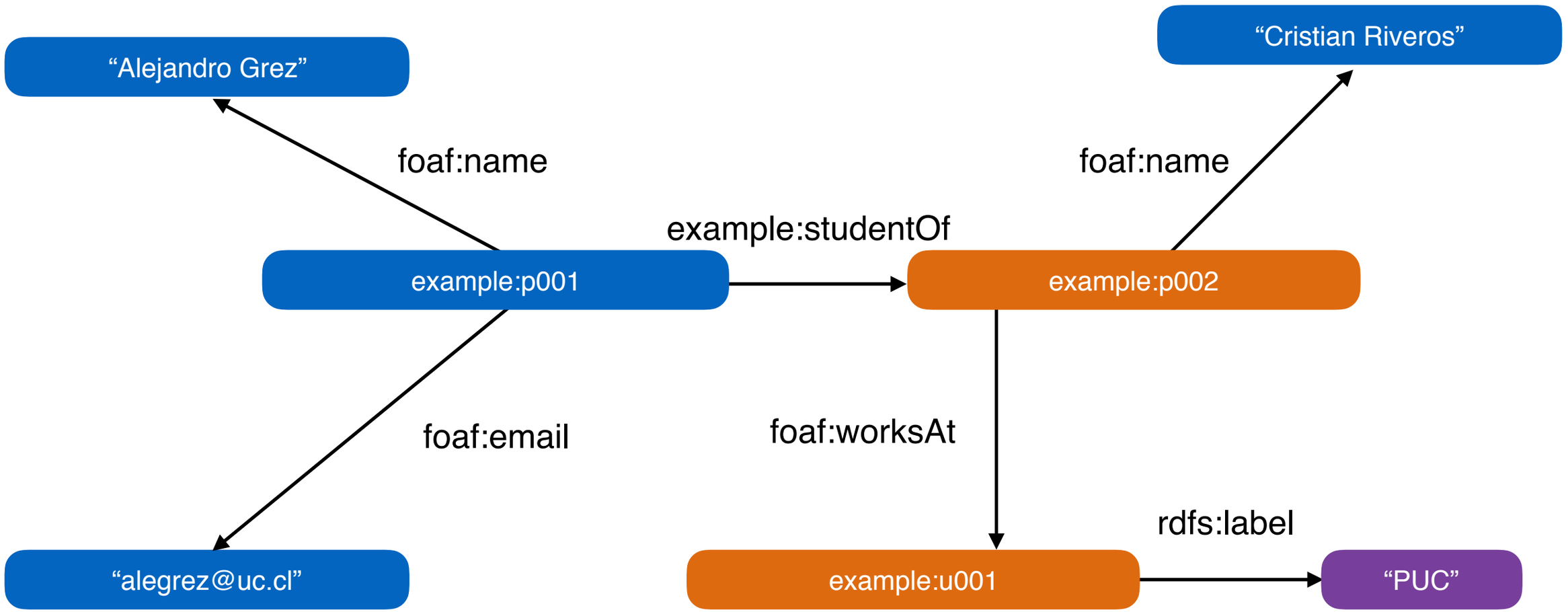
```
?universidad rdfs:label “PUC”
```



Preliminares

Resource Description Framework (RDF)

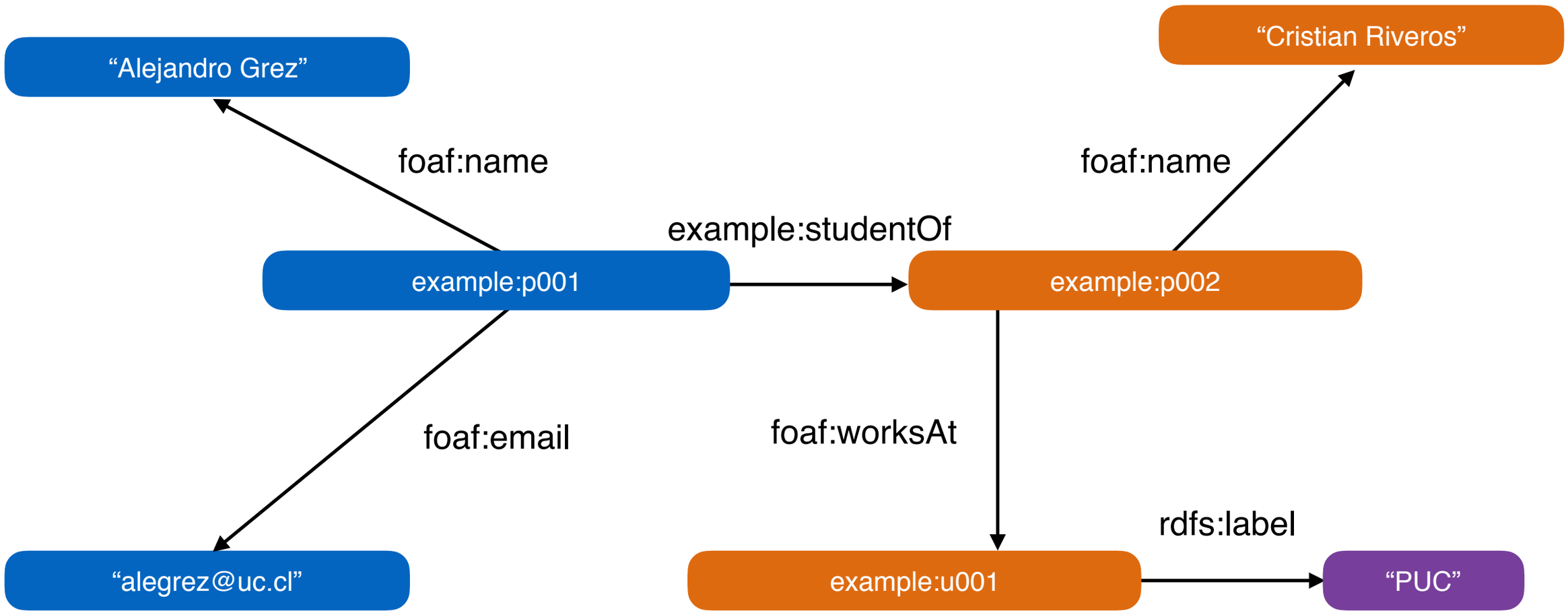
```
?prof foaf:worksAt ?universidad
```



Preliminares

Resource Description Framework (RDF)

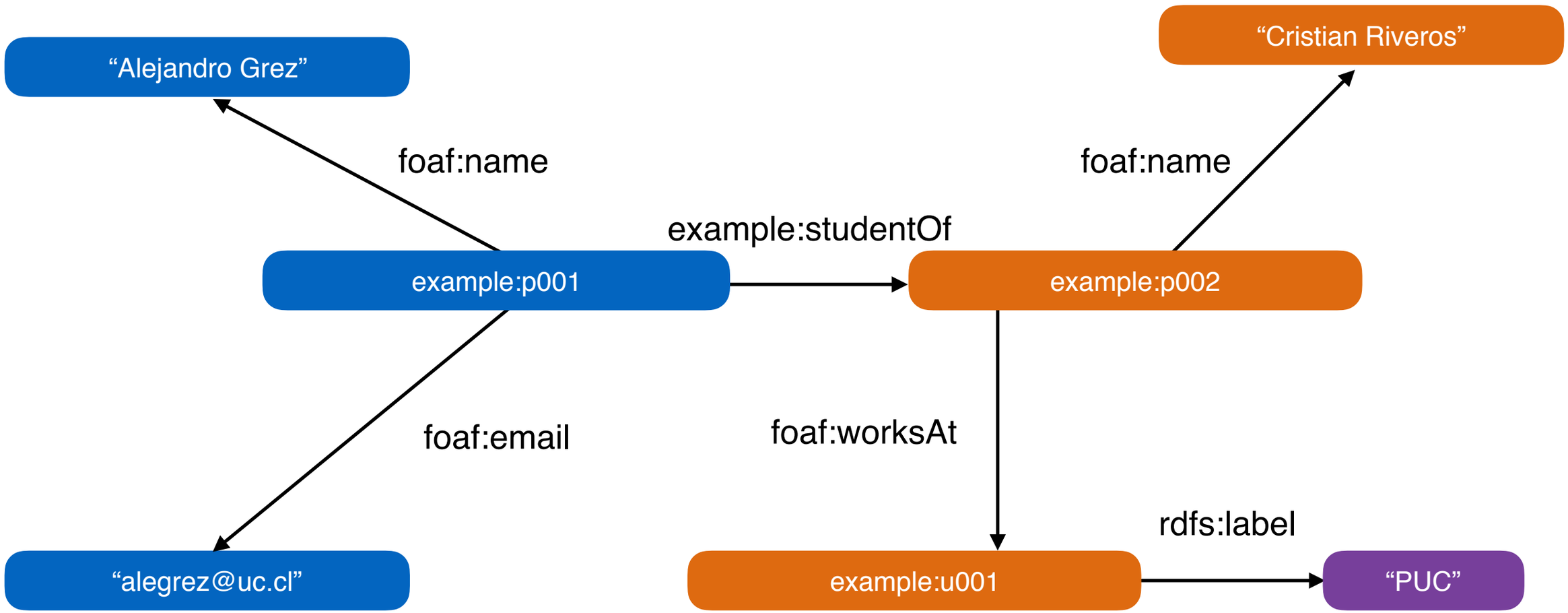
```
?prof foaf:name ?name
```



Preliminares

Resource Description Framework (RDF)

```
SELECT ?name WHERE
```



Preliminares

Resource Description Framework (RDF)

```
SELECT ?name WHERE
```

“Cristian Riveros”

Preliminares

Resource Description Framework (RDF)

```
SELECT ?name WHERE
```

“Cristian Riveros”

Outline

Introducción

Recursión en SPARQL

SPARQL y JSON APIs

Ser alumno de doctorado

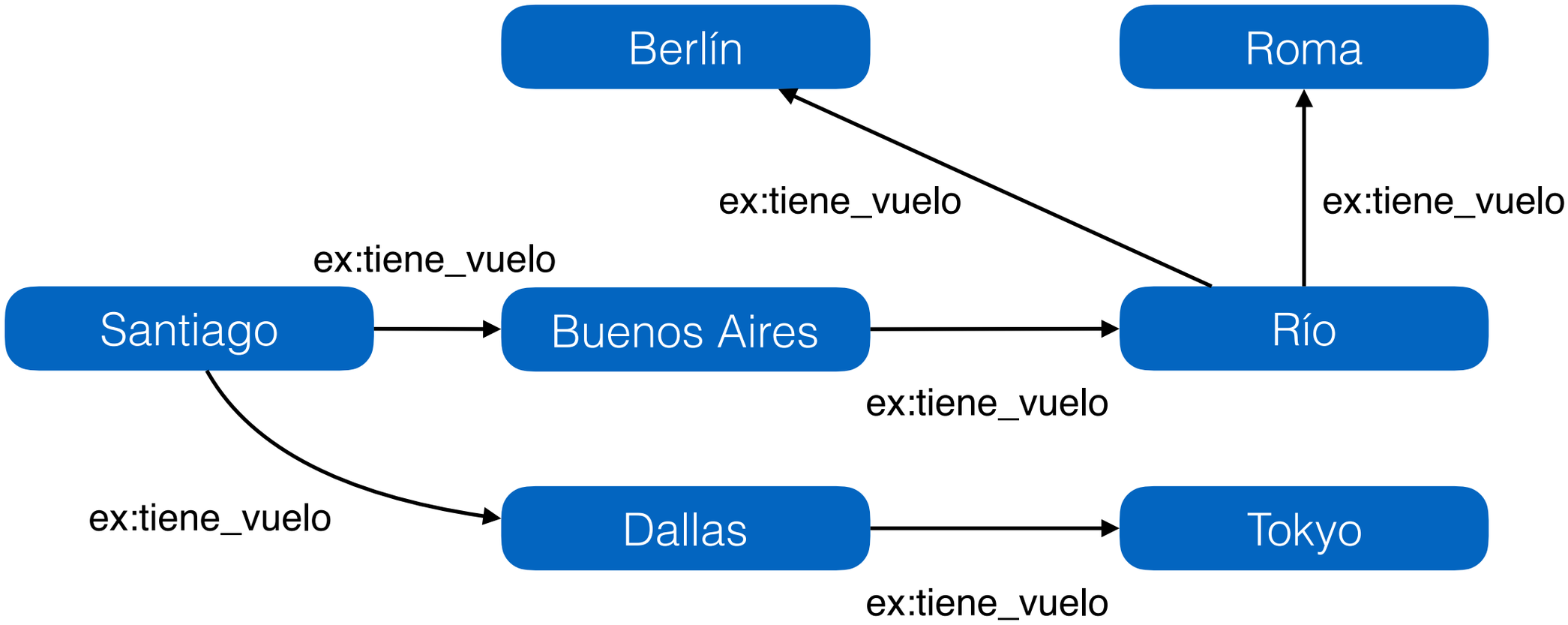
Recursión

Idea Hacer que SPARQL soporte consultas recursivas como las de SQL

Desafío Modificar el código de fuente de un Sistema de Bases de Datos RDF para que soporte tales consultas

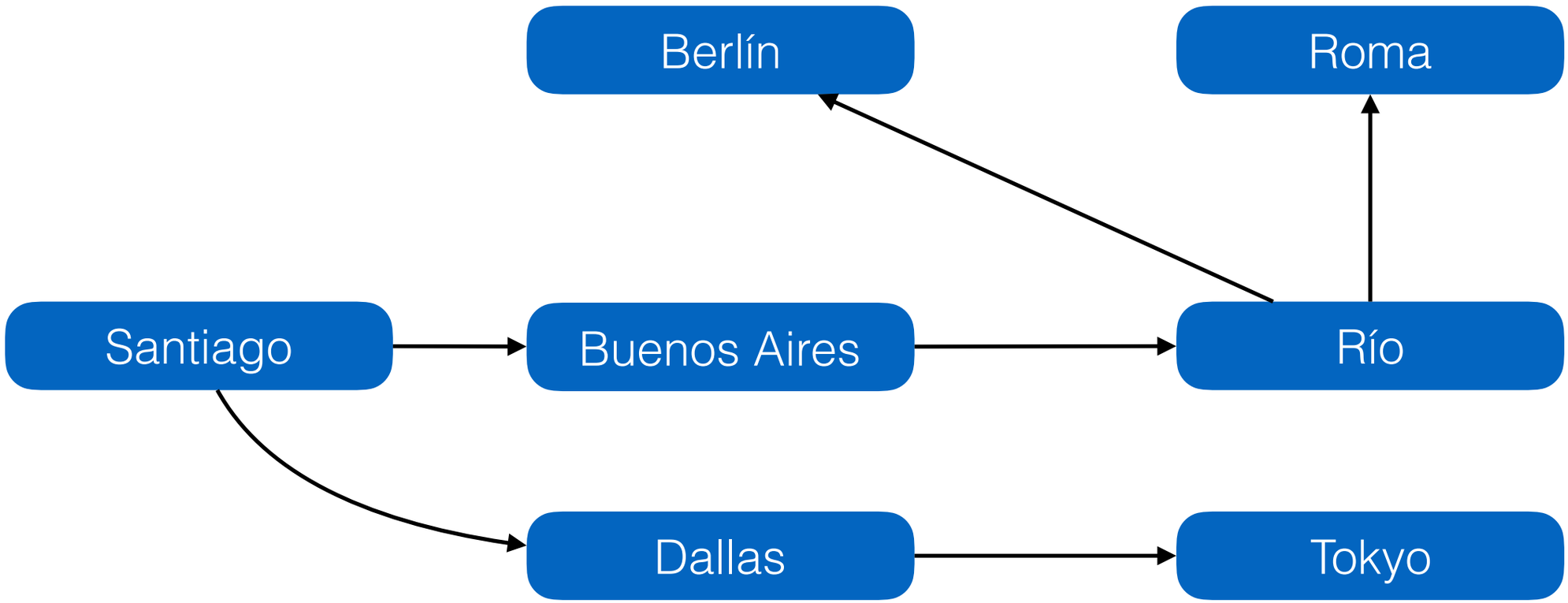
Recursión

Grafo de Vuelos



Recursión

Grafo de Vuelos



Recursión

Recursión

```
WITH RECURSIVE <http://db.puc.cl/temp> AS
{
   CONSTRUCT {
     ?c_origen ex:tiene_vuelo ?c_destino
   }
   FROM NAMED <http://db.puc.cl/temp>
   WHERE {
     {
       ?c_origen ex:tiene_vuelo ?c_destino
     }
     UNION
     {
       ?c_origen ex:tiene_vuelo ?c_intermedia .
       GRAPH <http://db.puc.cl/temp> {
         ?c_intermedia ex:tiene_vuelo ?c_destino
       }
     }
   }
}
```

Pero he escuchado que utilizando *property paths* SPARQL ya podía responder esa consulta!

Recursión

Desde el estándar 1.1 es posible consultar *property paths*, que añadieron algo de recursión a SPARQL

Sin embargo nuestro modelo de consultas recursivas puede expresar muchas consultas que no pueden ser expresadas con *property paths*

Pero si tienen mayor poder expresivo,
¿entonces son más lentos?

Recursión

Mostramos que nuestra implementación corre en tiempos del mismo orden de magnitud, y a veces mucho mejor

Además nuestra implementación puede correr consultas que no caben en memoria, mientras que eso no se puede hacer al usar *property paths*

Recursión

El *paper* "Recursion in SPARQL" fue presentado en la conferencia ISWC 2015

Fue uno de los dos nominados al mejor *paper* de la conferencia

Outline

Introducción

Recursión en SPARQL

SPARQL y JSON APIs

Ser alumno de doctorado

SPARQL y JSON APIs

Idea Extender el operador SERVICE del lenguaje de consulta SPARQL para integrar datos de la web que no están en RDF

Desafío Hacer que la evaluación de este tipo de consultas se óptima

SERVICE

SERVICE

```
SELECT ?name ?weather
WHERE
{
  ?city :comuneOf :Chile .
  SERVICE <http://weather.org/sparql>
  {
    ?city weather:weatherNow ?weather
  }
}
```

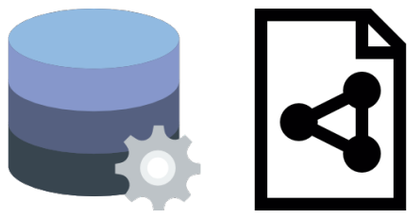
SERVICE

```
SELECT ?name ?weather
WHERE
{
  ?city :comuneOf :Chile .
  SERVICE <http://weather.org/sparql>
  {
    ?city weather:weatherNow ?weather
  }
}
```

SERVICE

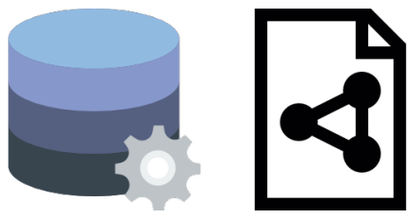
```
SELECT ?name ?weather
WHERE
{
  ?city :comuneOf :Chile .
  SERVICE <http://weather.org/sparql>
  {
    ?city weather:weatherNow ?weather
  }
}
```

SERVICE

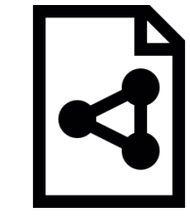


```
SELECT ?name ?weather
WHERE
{
  ?city :comuneOf :Chile .
  SERVICE <http://weather.org/sparql>
  {
    ?city weather:weatherNow ?weather
  }
}
```

SERVICE



```
SELECT ?name ?weather
WHERE
{
  ?city :comuneOf :Chile .
  SERVICE <http://weather.org/sparql>
  {
    ?city weather:weatherNow ?weather
  }
}
```



SERVICE

```
SELECT ?name ?weather
WHERE
{
  ?city :comuneOf :Chile .
  SERVICE <http://weather.org/sparql>
  {
    ?city weather:weatherNow ?weather
  }
}
```

SERVICE

```
SELECT ?name ?weather
WHERE
{
  ?city :comuneOf :Chile .
  SERVICE <http://weather.org/sparql>
  {
    ?city weather:weatherNow ?weather
  }
}
```

SERVICE

```
SELECT ?name ?weather
WHERE
{
  ?city :comuneOf :Chile .
  SERVICE <http://weather.org/sparql>
  {
    ?city weather:weatherNow ?weather
  }
}
```

SERVICE

```
SELECT ?name ?weather
WHERE
{
  ?city :comuneOf :Chile .
  SERVICE <http://weather.org/sparql>
  {
    ?city weather:weatherNow ?weather
  }
}
```

Extensión SERVICE

```
SELECT ?x ?l WHERE {
   ?x wdt:instanceOf wd:mountain .
   ?x wdt:locatedIn wd:Scotland .
   ?x rdfs:label ?l .
   SERVICE <http://weather.api/request?q={?l}>{
      (["description"]) AS (?d)
   }
   FILTER (?d = "Sunny")
}
```

Extensión SERVICE

Resolvemos la primera parte de la consulta:

```
?x wdt:instanceOf wd:mountain .
?x wdt:locatedIn wd:Scotland .
?x rdfs:label ?l
```

?x	?l
wd:Q104674	“Ben Nevis”
wd:Q13130255	“Mount Blair”

Extensión SERVICE

Luego cada fila se extiende según lo que retorne la API

```
SERVICE <http://weather.api/request?q={?l}>{
   (["description"]) AS (?d)
}
```

?x	?l	?d
wd:Q104674	“Ben Nevis”	“Sunny”
wd:Q13130255	“Mount Blair”	“Rain”

Extensión SERVICE

Finalmente se aplican los filtros

```
FILTER (?d = “Sunny")
```

?x	?l	?d
wd:Q104674	“Ben Nevis”	“Sunny”

Experimentos

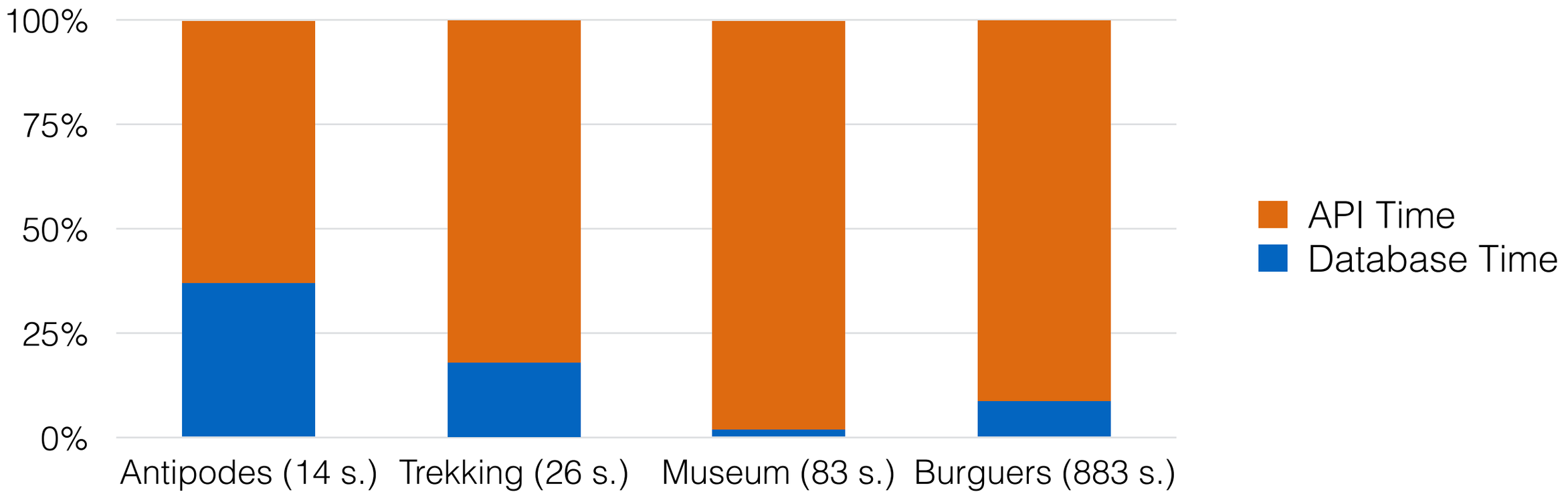


Figura 1: consultas que usaban APIs del mundo real

SPARQL y JSON APIs

En este contexto, el principal cuello de botella es el número de llamados hechos a la API

Por lo mismo, desarrollamos un algoritmo *Worst-Case Optimal* en el número de llamados a la API

SPARQL y JSON APIs

SPARQL y JSON APIs

Probamos 3 algoritmos sobre el benchmark Berlin adaptado para este contexto:

- Vanilla: el algoritmo básico
- Sin duplicados: el algoritmo básico que impedía llamados duplicados
- Algoritmo WCO

SPARQL y JSON APIs

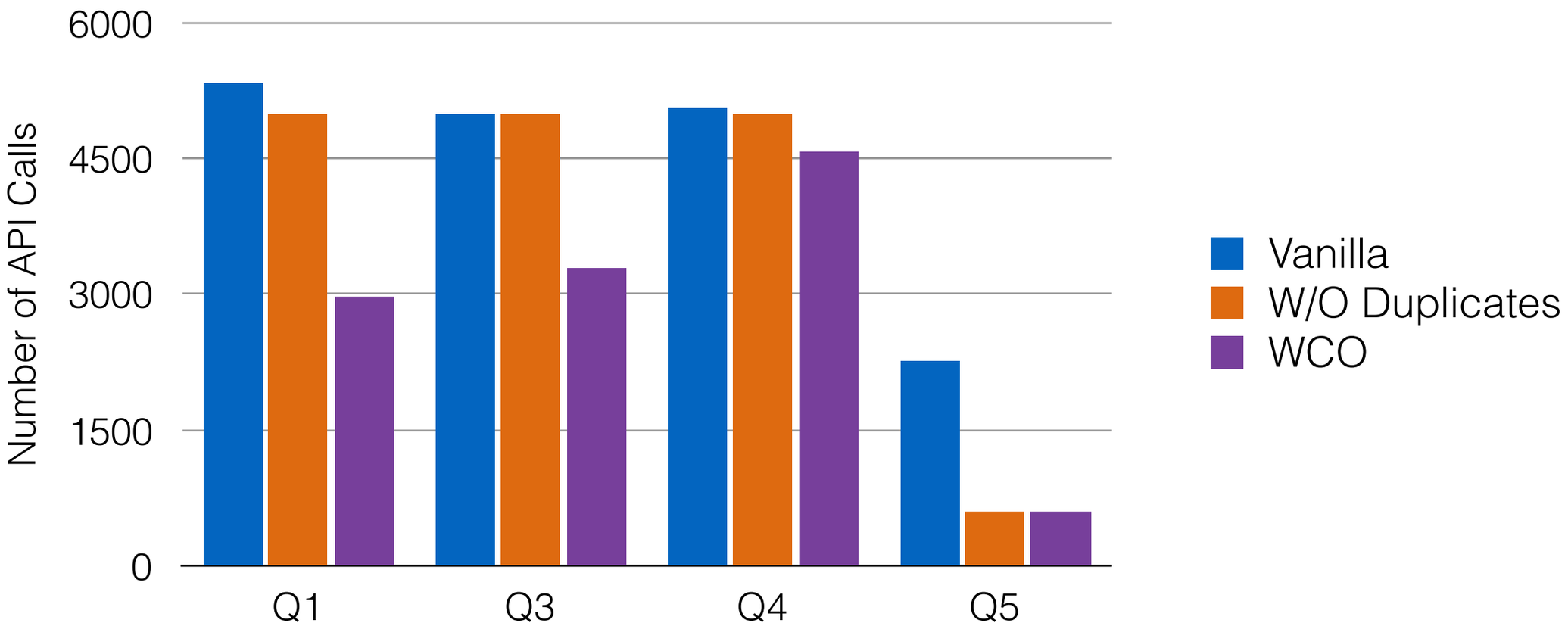


Figura 2: resultados para las consultas del benchmark Berlin

SPARQL y JSON APIs

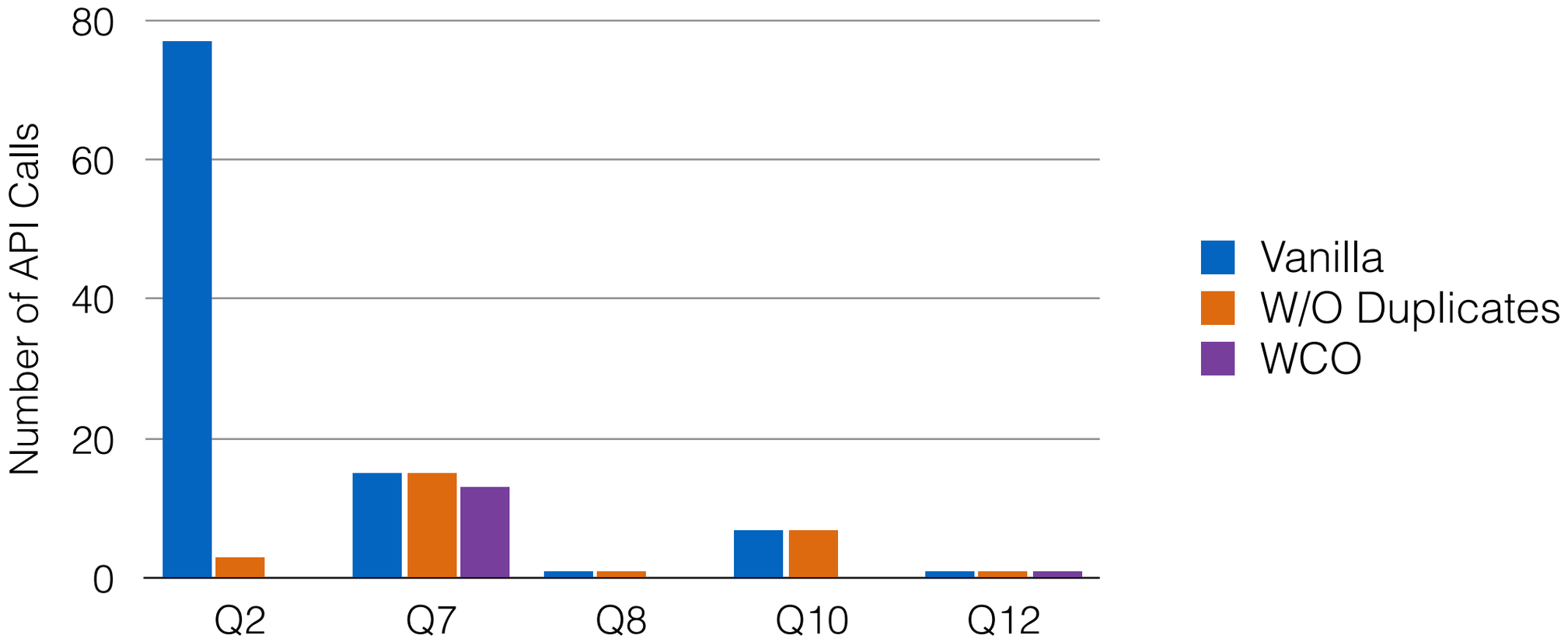


Figura 3: resultados para las consultas más pequeñas del Benchmark Berlin

Resultados

La versión sin duplicados hace 20% menos llamadas

Resultados

La versión sin duplicados hace 20% menos llamadas

La versión WCO hace 53% menos llamadas

Outline

Introducción

Recursión en SPARQL

SPARQL y JSON APIs

Ser alumno de doctorado

ACADEMIA

PRO

YOU CAN WORK WHENEVER YOU WANT! EVERY DAY IS A SATURDAY!

CON

YOU WORK ON SATURDAYS.

Ser Alumno de Doctorado

- Becas y Arancel
- Duración
- Proceso de publicaciones
- Cursos

Datalab UC

Profesores

Marcelo Arenas

Domagoj Vrgoč

Juan L. Reutter

Cristian Riveros

Datalab UC

Temas de Investigación

Datalab UC

Temas de Investigación

- Bases de Datos
- Linked Data
- Teoría de computación
- Blockchain
- Data Science

Extendiendo el lenguaje de consultas SPARQL

Adrián Soto

Pontificia Universidad Católica de Chile