ESCUELA DE INGENIERÍA
FACULTAD DE INGENIERÍA

EDUCACIÓN PROFESIONAL

Bases de Datos NoSQL

Gestión de Datos

Vicente Calisto

Educación Profesional - Escuela de Ingeniería

Hasta ahora

- Bases de datos relacionales
- SQL

Bases de datos relacionales

- Muchas estructura (un esquema fijo)
- Muchas garantías (ACID)
- Generalmente centralizadas (viven en un servidor)

NoSQL

Término común para denominar bases de datos con:

- Menos restricciones que el modelo relacional
- Menos esquema
- Menos garantías de consistencia
- Más adecuadas para la distribución

NoSQL: ¿Por qué?

Sistemas de bases de datos relacionales no están pensadas para un entorno altamente distribuido

- WWW, google, twitter, instagram, etc.

Sistemas distribuidos

Dos problemas fundamentales:

1. Datos no caben en un computador

2. Servidores pueden fallar

Datos no caben en un computador

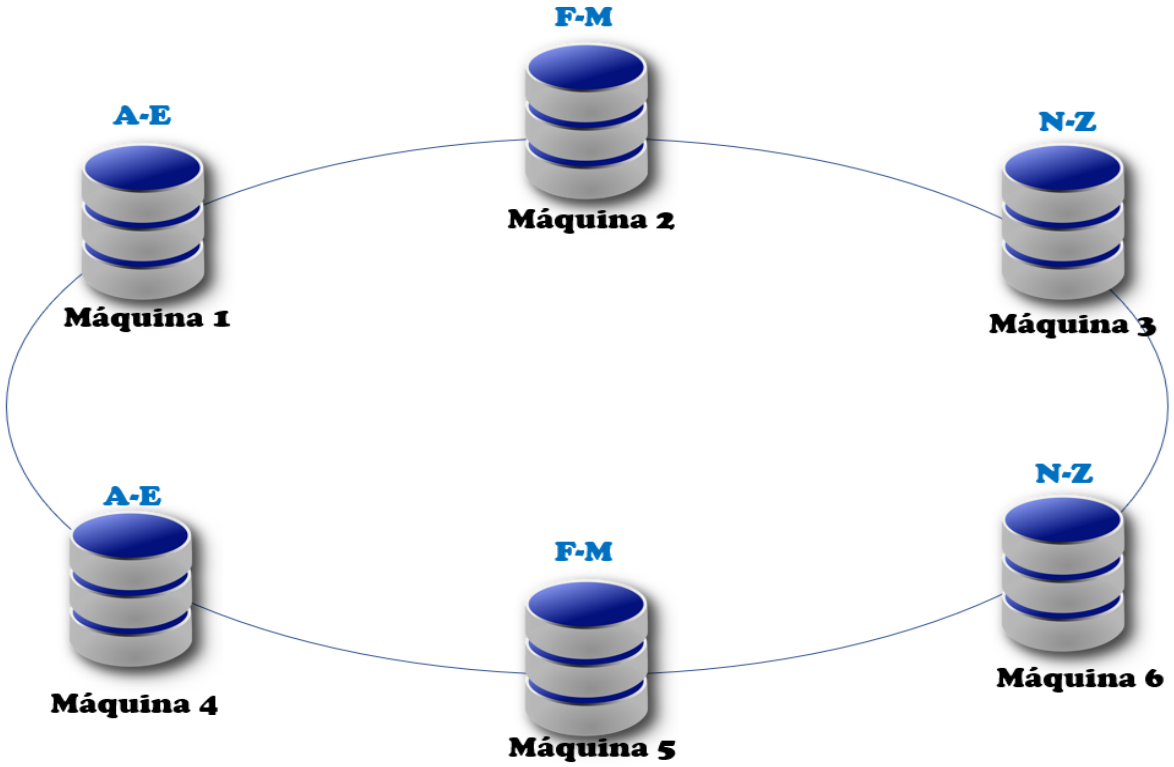
Fragmentación de los datos

Ej: **Usuarios** de twitter

Fragmentación de relación **Usuarios** en tres

Servidores fallan

Replicación de los datos



Replicación en un sistema distribuido

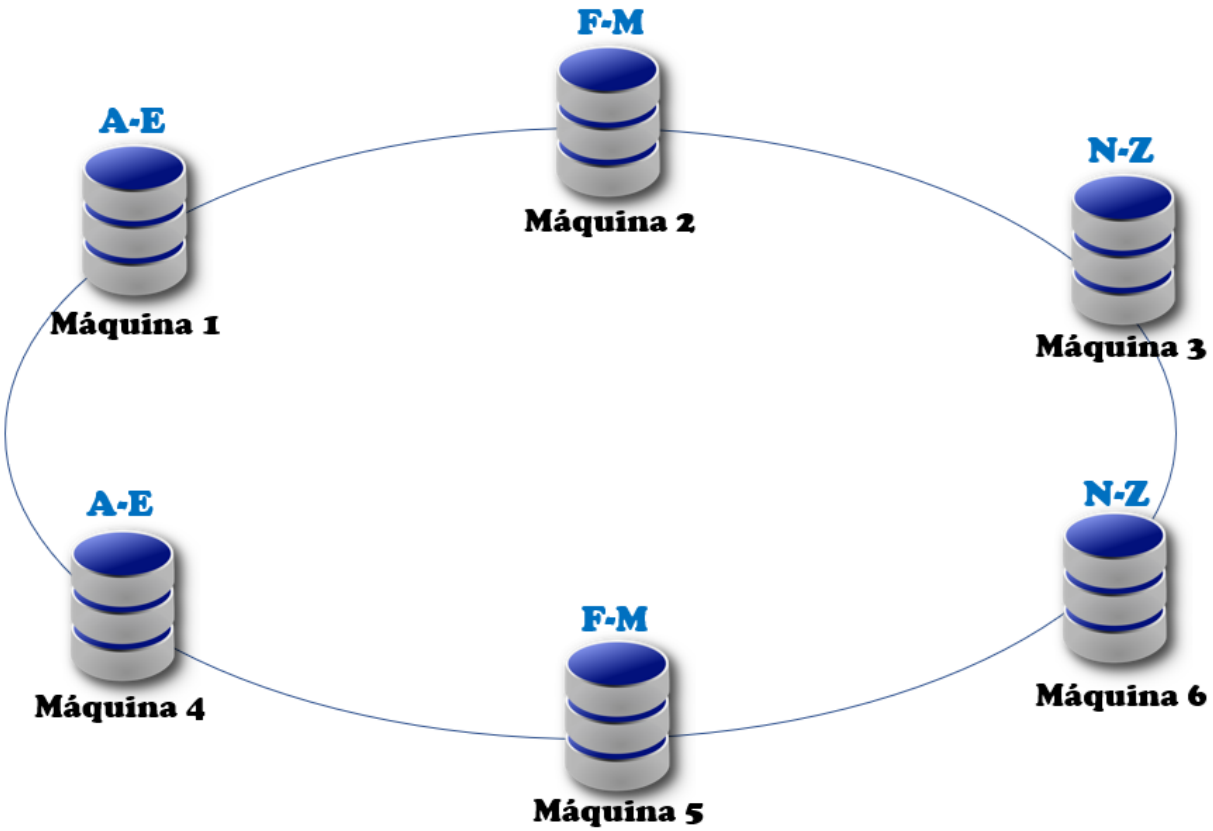
Garantías en un entorno distribuido

Tres propiedades fundamentales:

- Consistency (todos los usuarios ven lo mismo)
- Availability (todas las consultas siempre reciben una respuesta, aunque sea errónea)
- Partition tolerance (el sistema funciona bien pese a estar físicamente dividido)

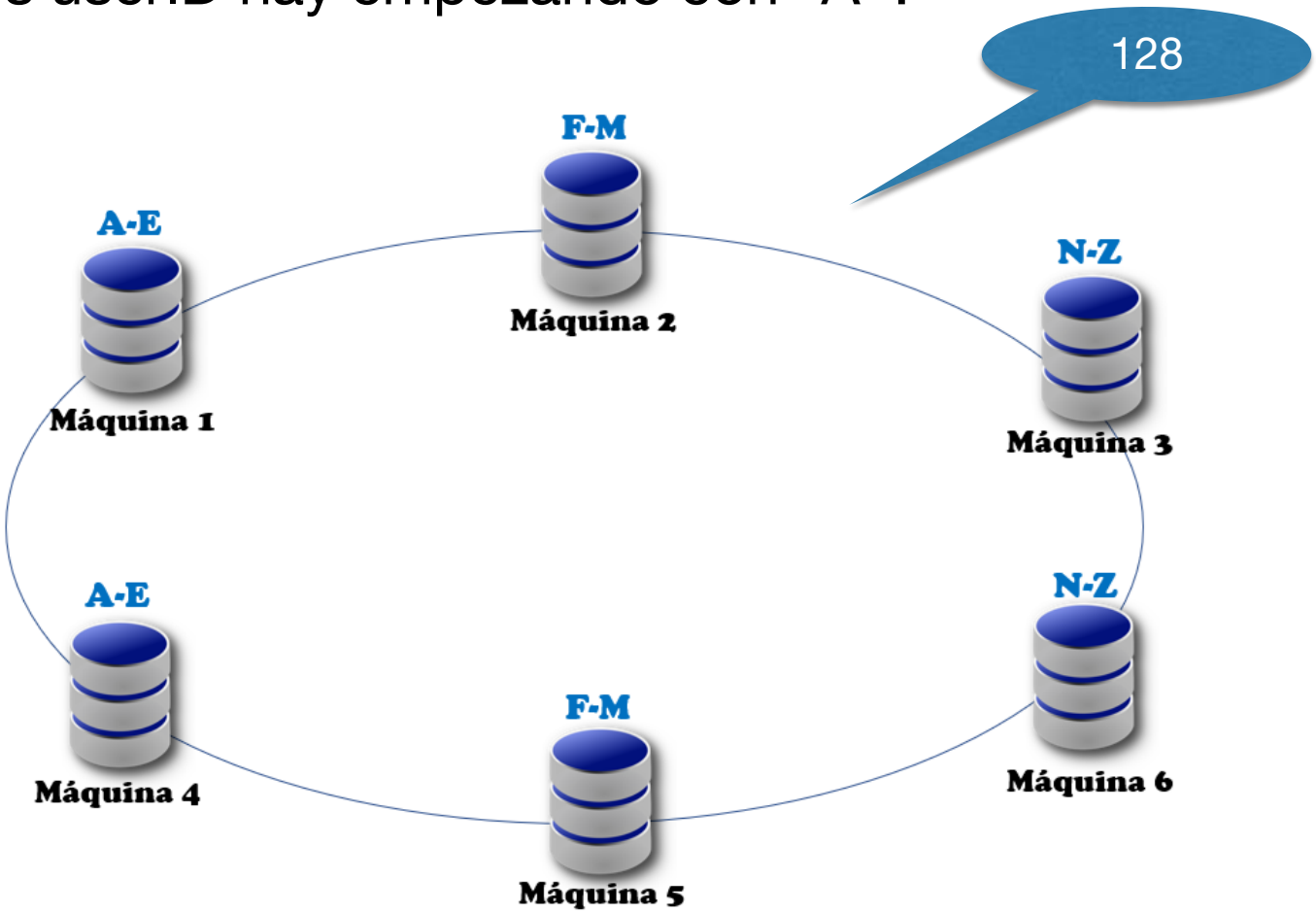
Consistencia

¿Cuántos userID hay empezando con "A"?



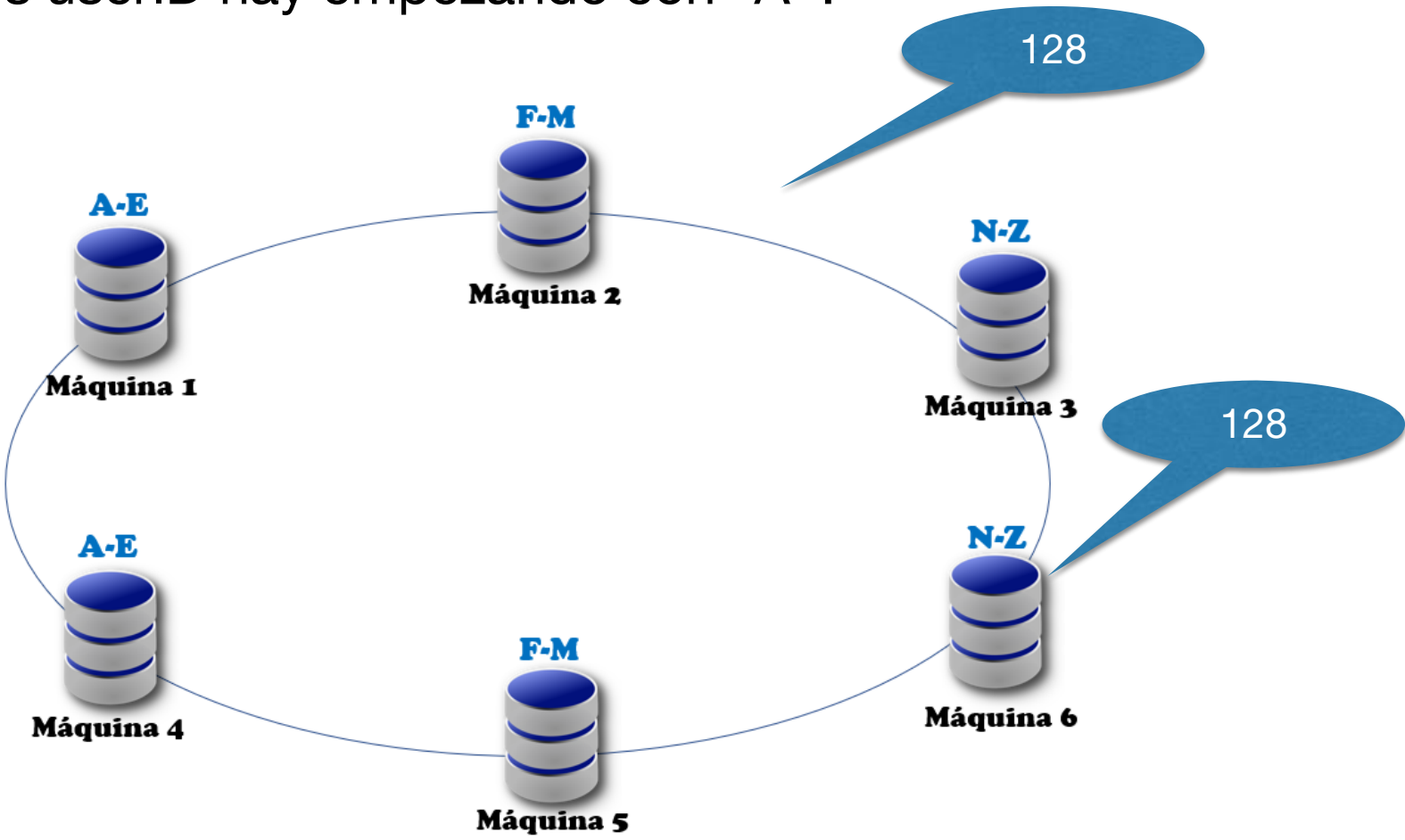
Consistencia

¿Cuántos userID hay empezando con "A"?



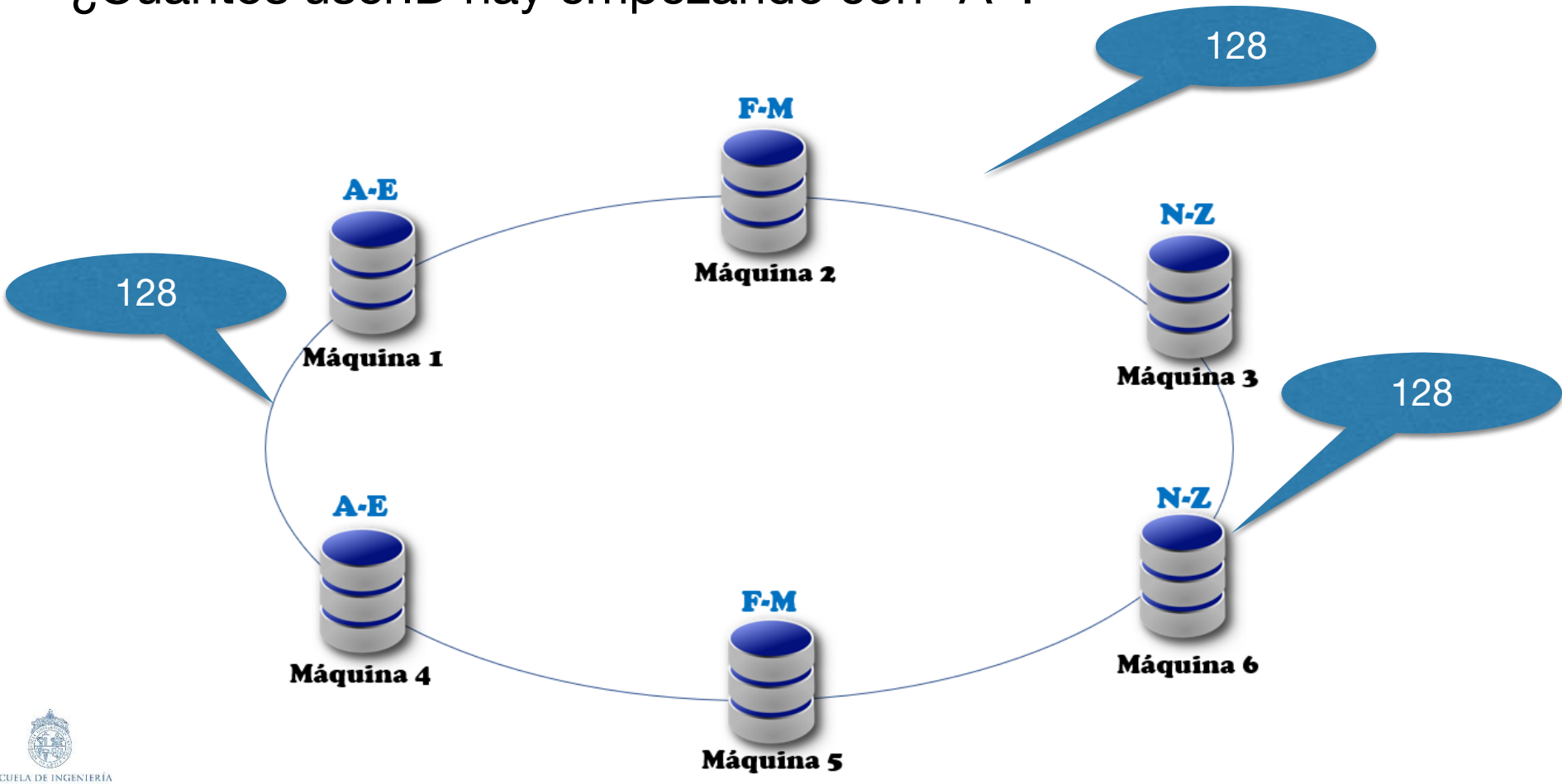
Consistencia

¿Cuántos userID hay empezando con "A"?



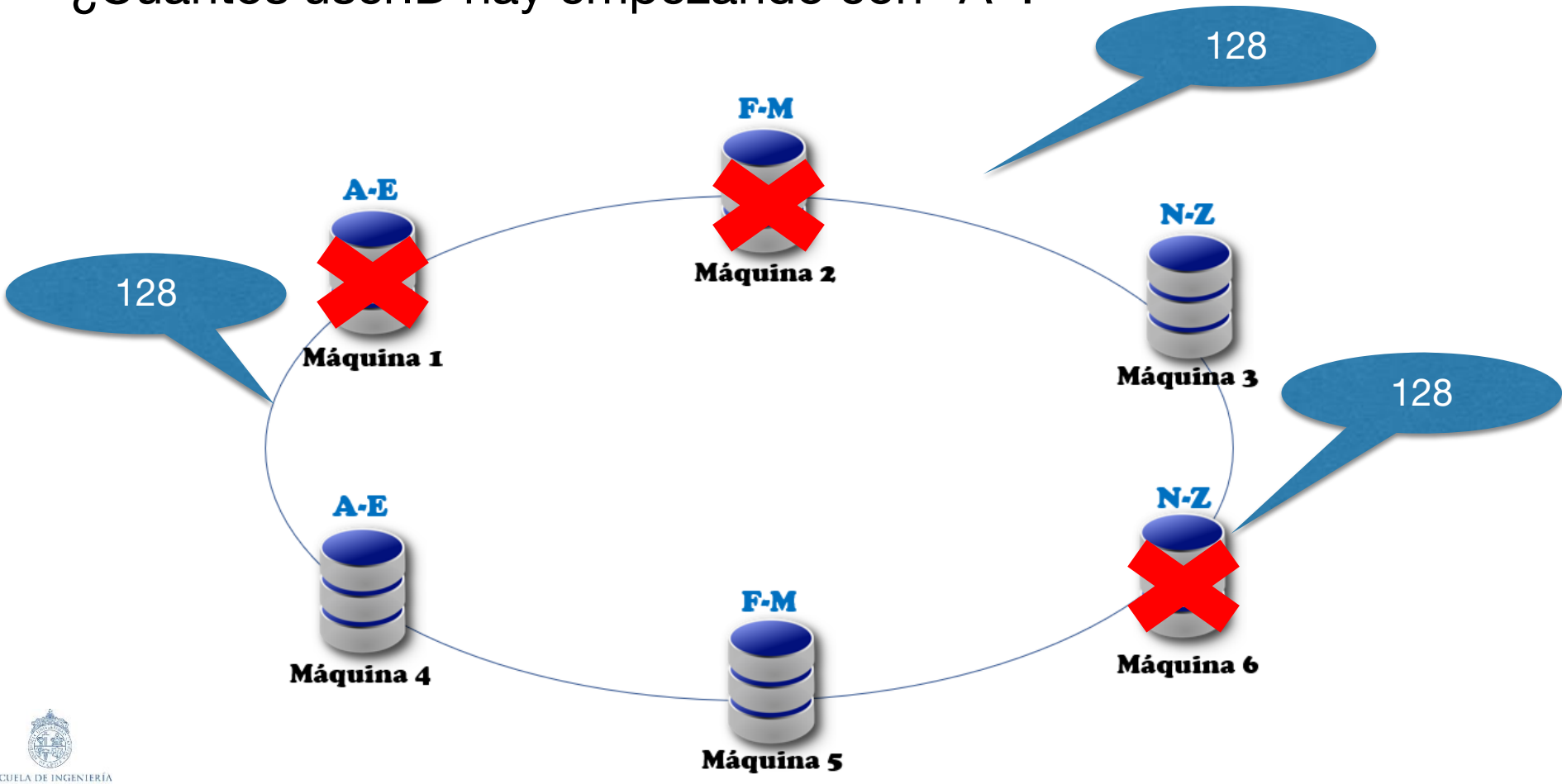
Consistencia

¿Cuántos userID hay empezando con "A"?



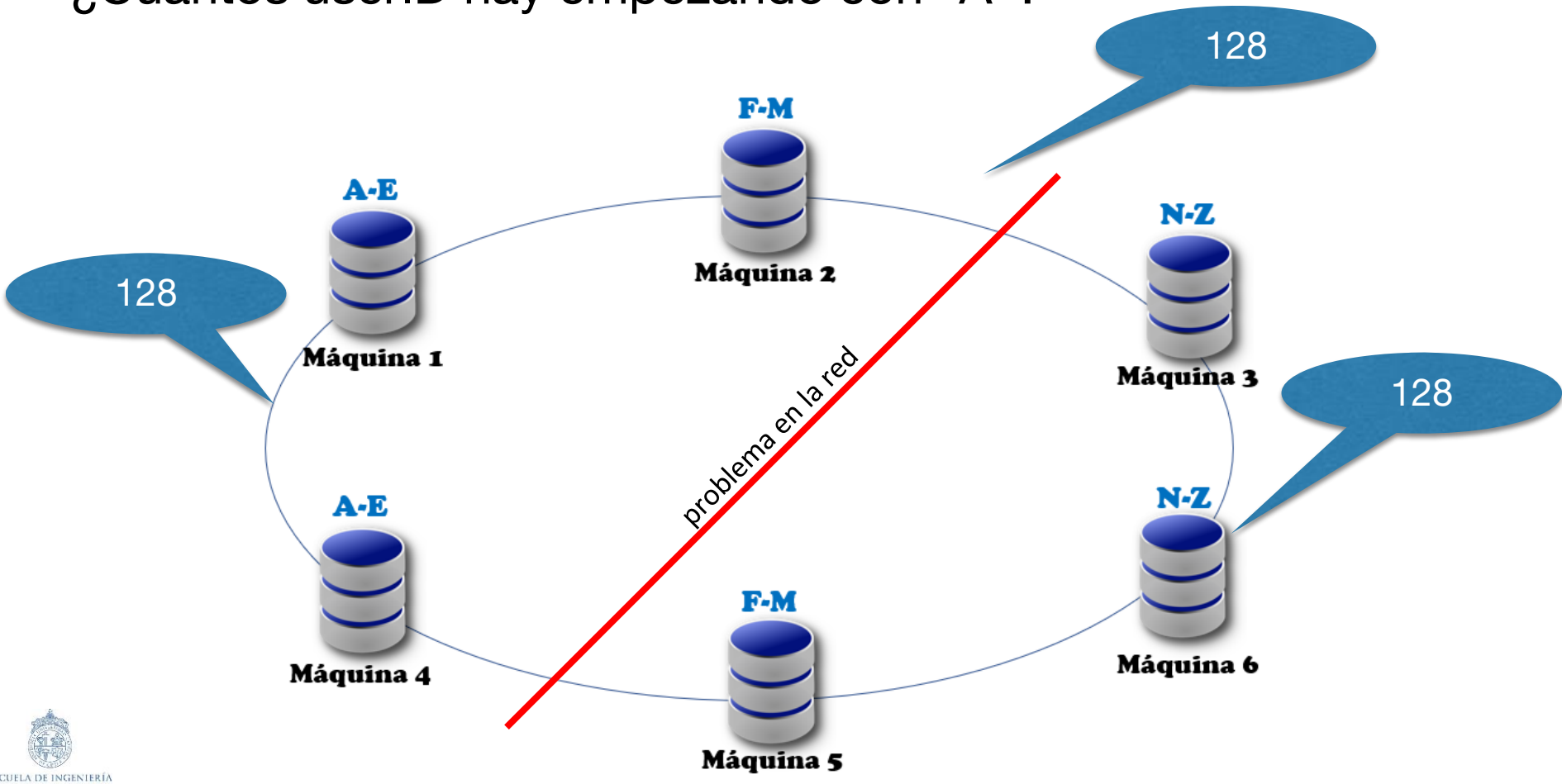
Consistencia

¿Cuántos userID hay empezando con "A"?



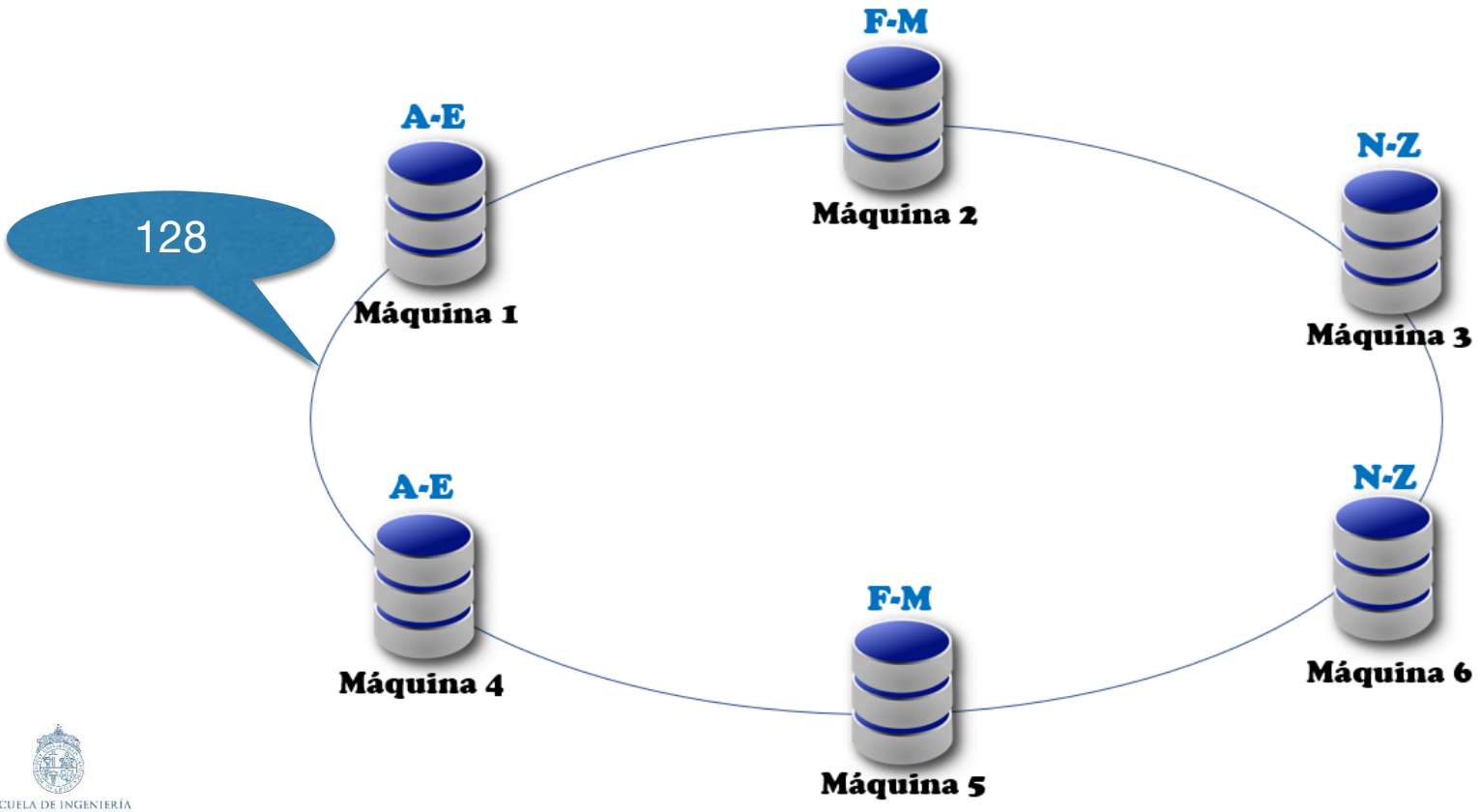
Consistencia

¿Cuántos userID hay empezando con "A"?



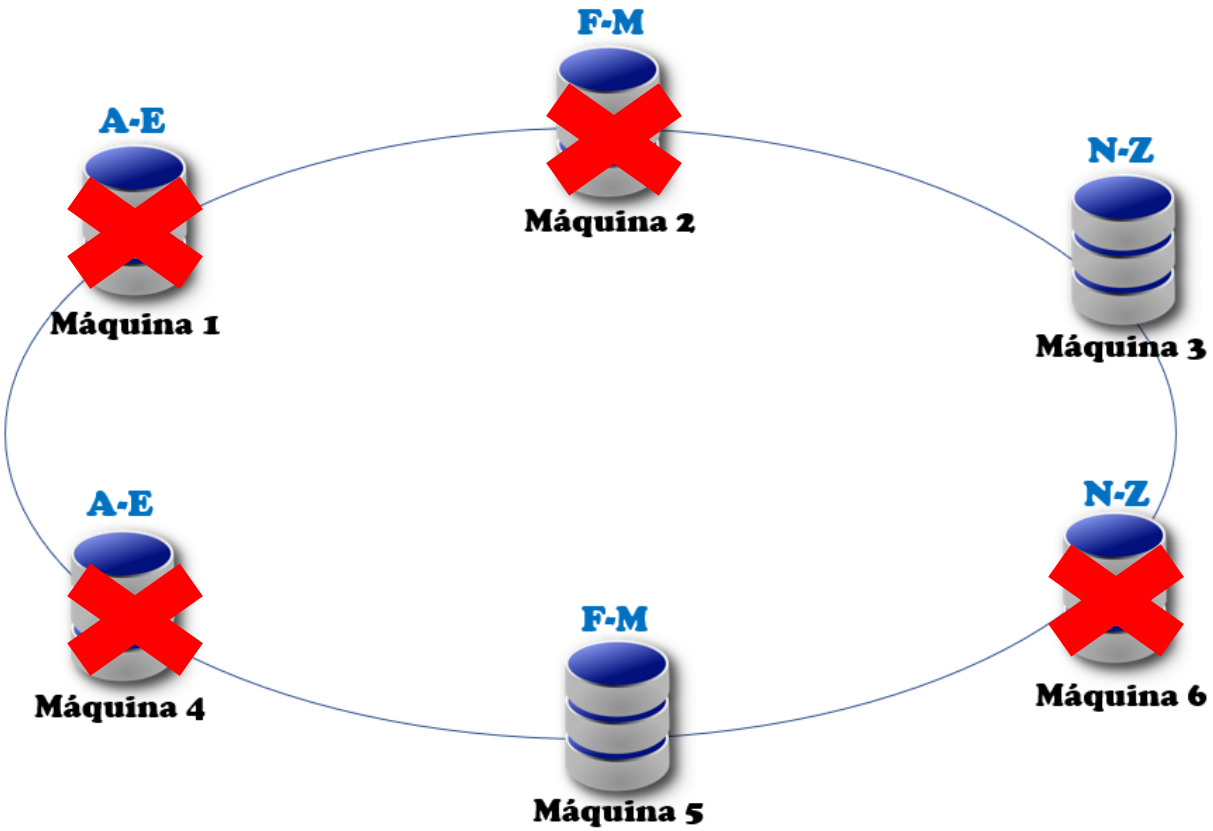
Availability

¿Cuántos userID hay empezando con "A"?



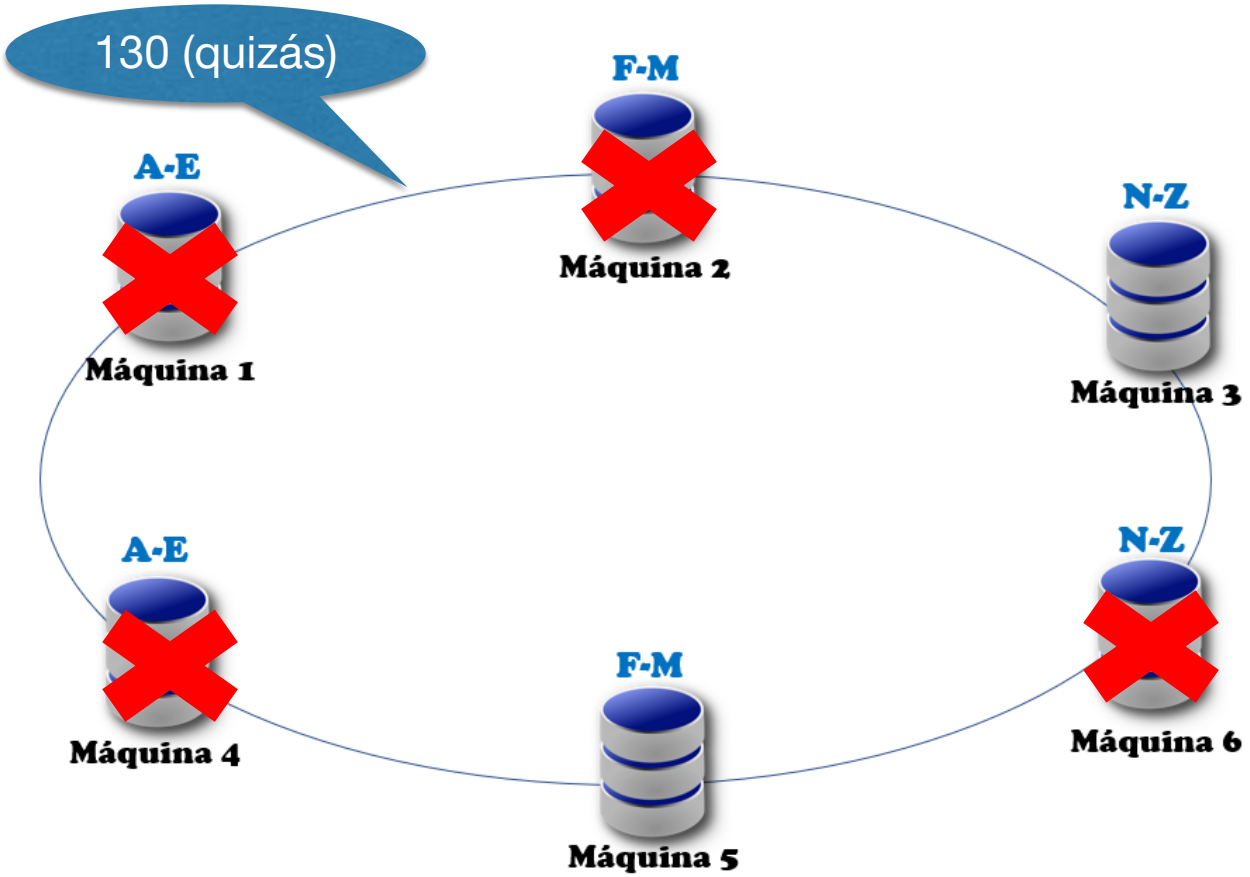
Availability

¿Cuántos userID hay empezando con "A"?



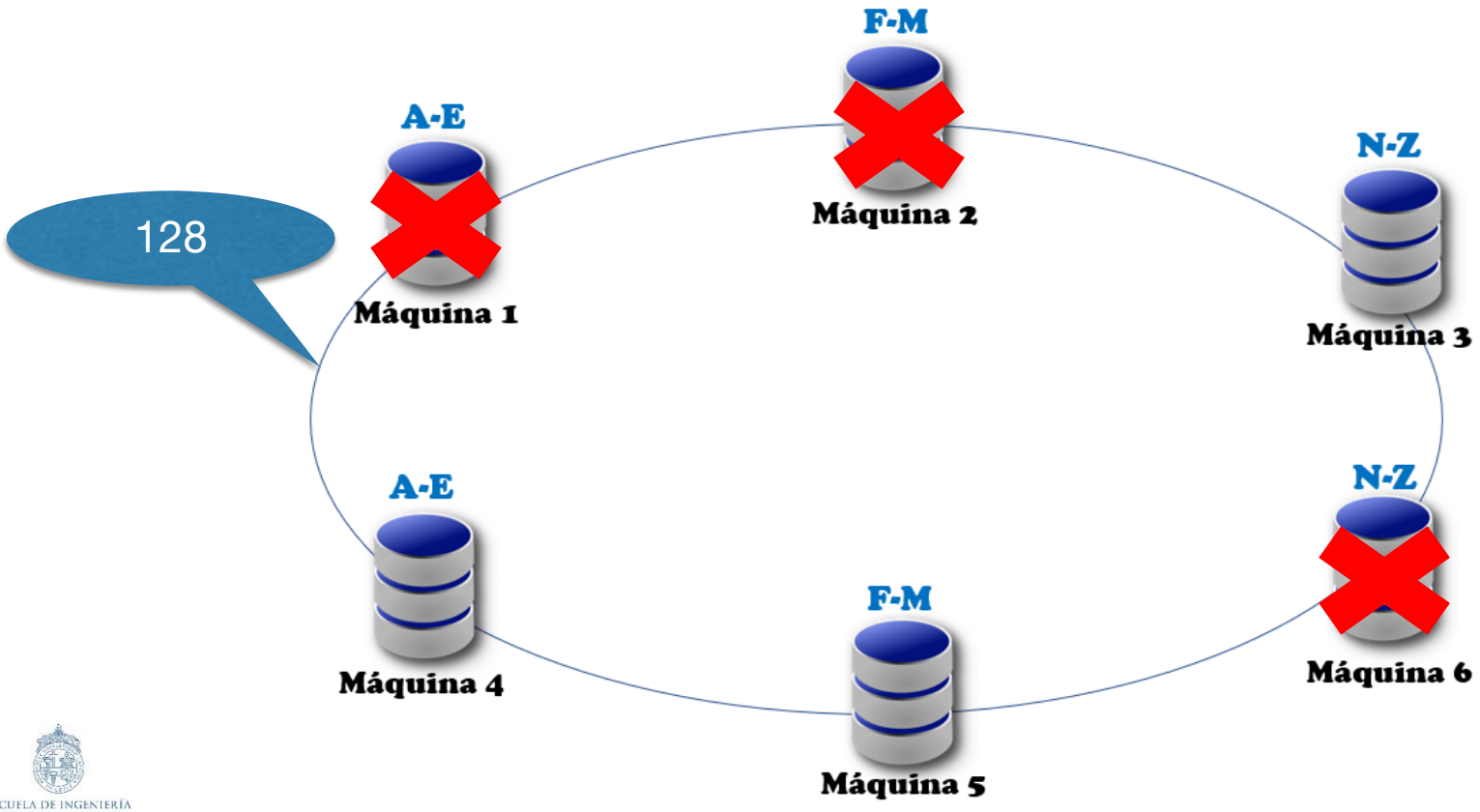
Availability

¿Cuántos userID hay empezando con "A"?



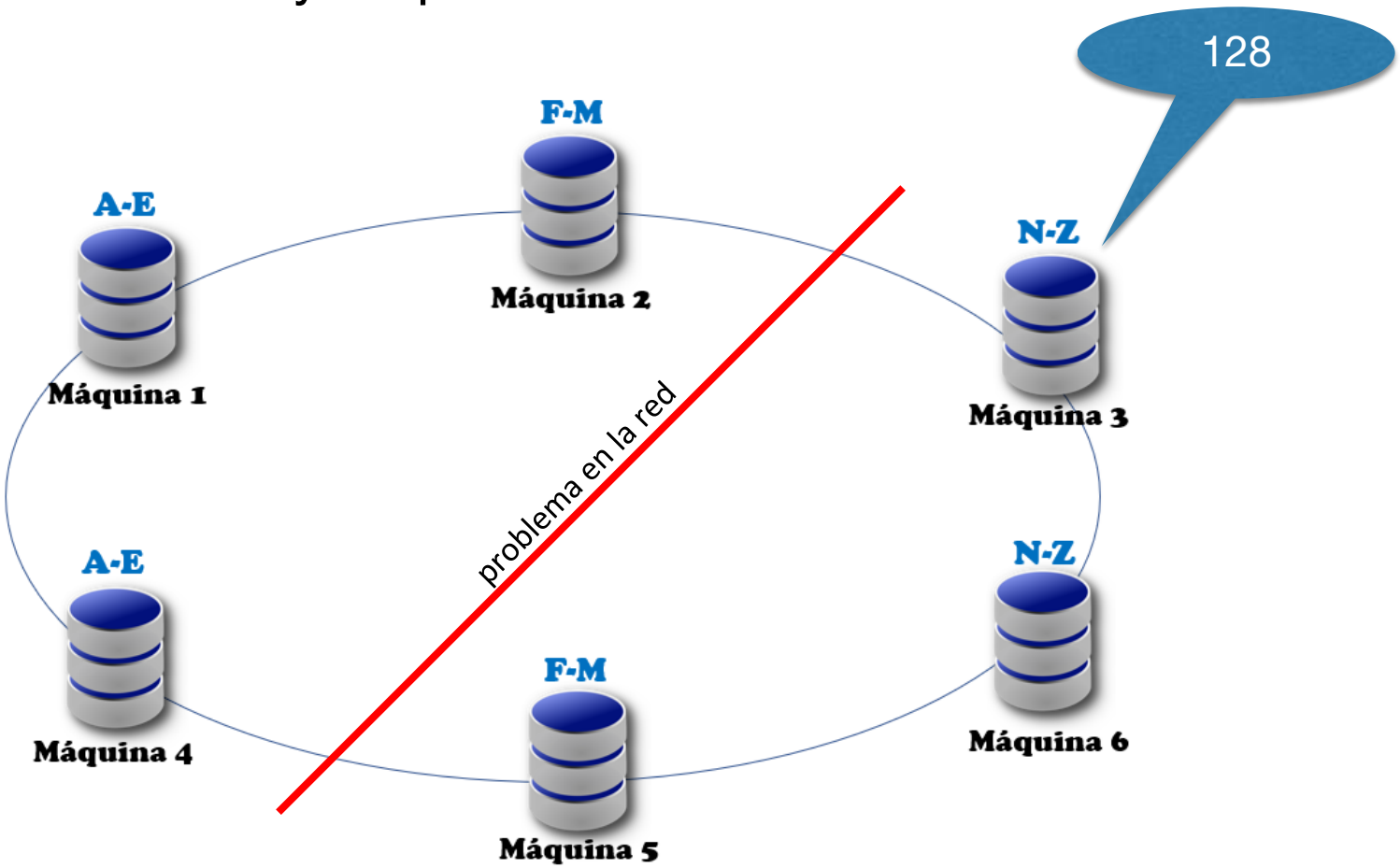
Partition tolerance

¿Cuántos userID hay empezando con "A"?



Partition tolerance

¿Cuántos userID hay empezando con "A"?



Teorema CAP

Plantea que para una base de datos distribuida es imposible mantener simultáneamente estas tres características:

- Consistency
- Availablity
- Partition tolerance

Teorema CAP

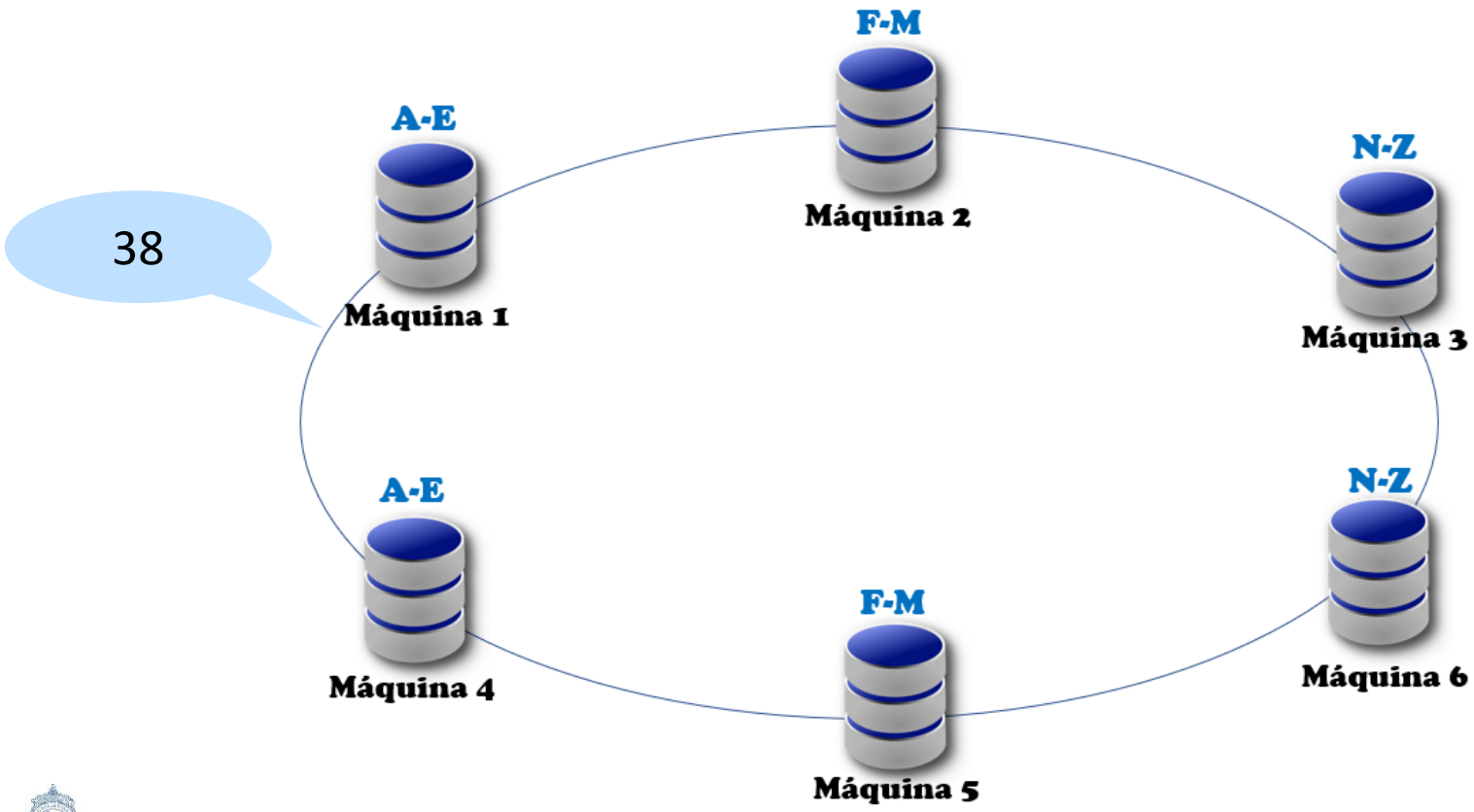
P es dado en cualquier sistema distribuido. Entonces, el Teorema CAP nos dice que hay que elegir entre:

- Consistency
- Availablity

Entonces tenemos sistemas CP y AP

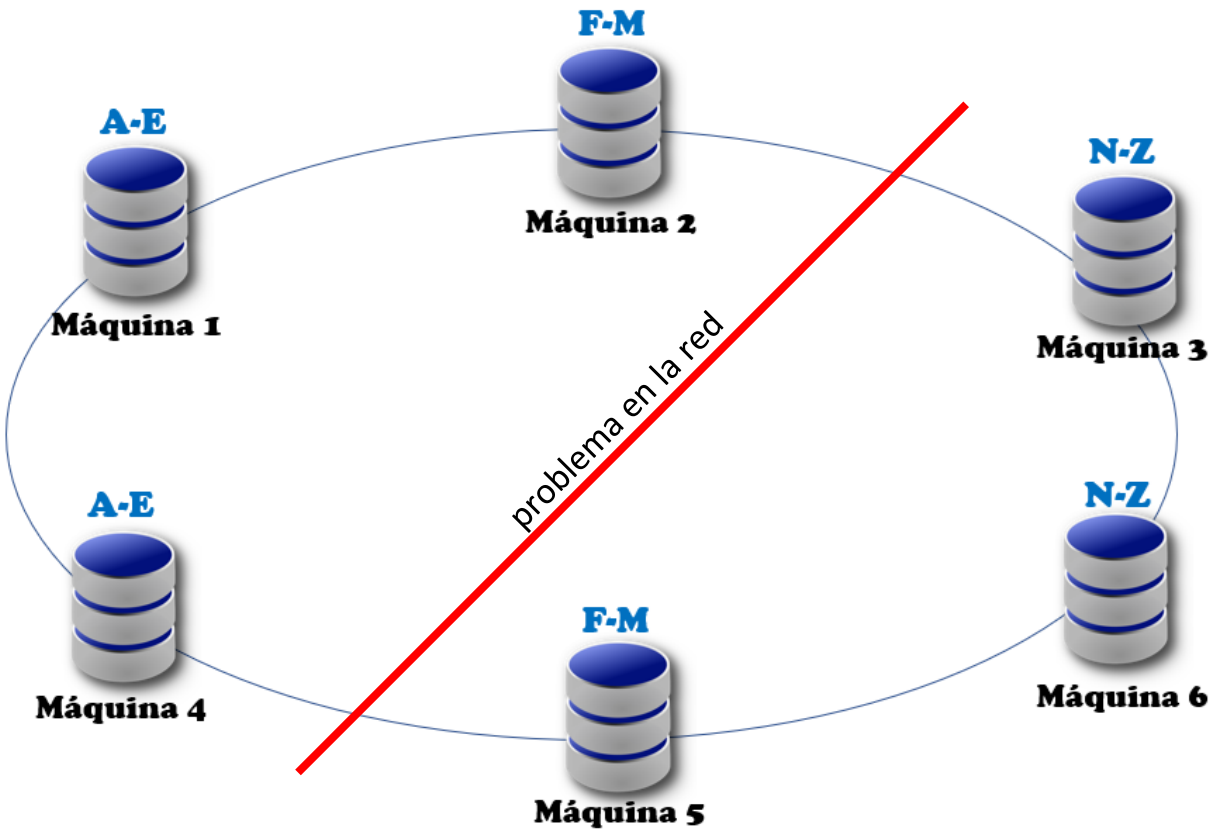
AP vs CP

¿Cuántos userID hay empezando con "Z"?



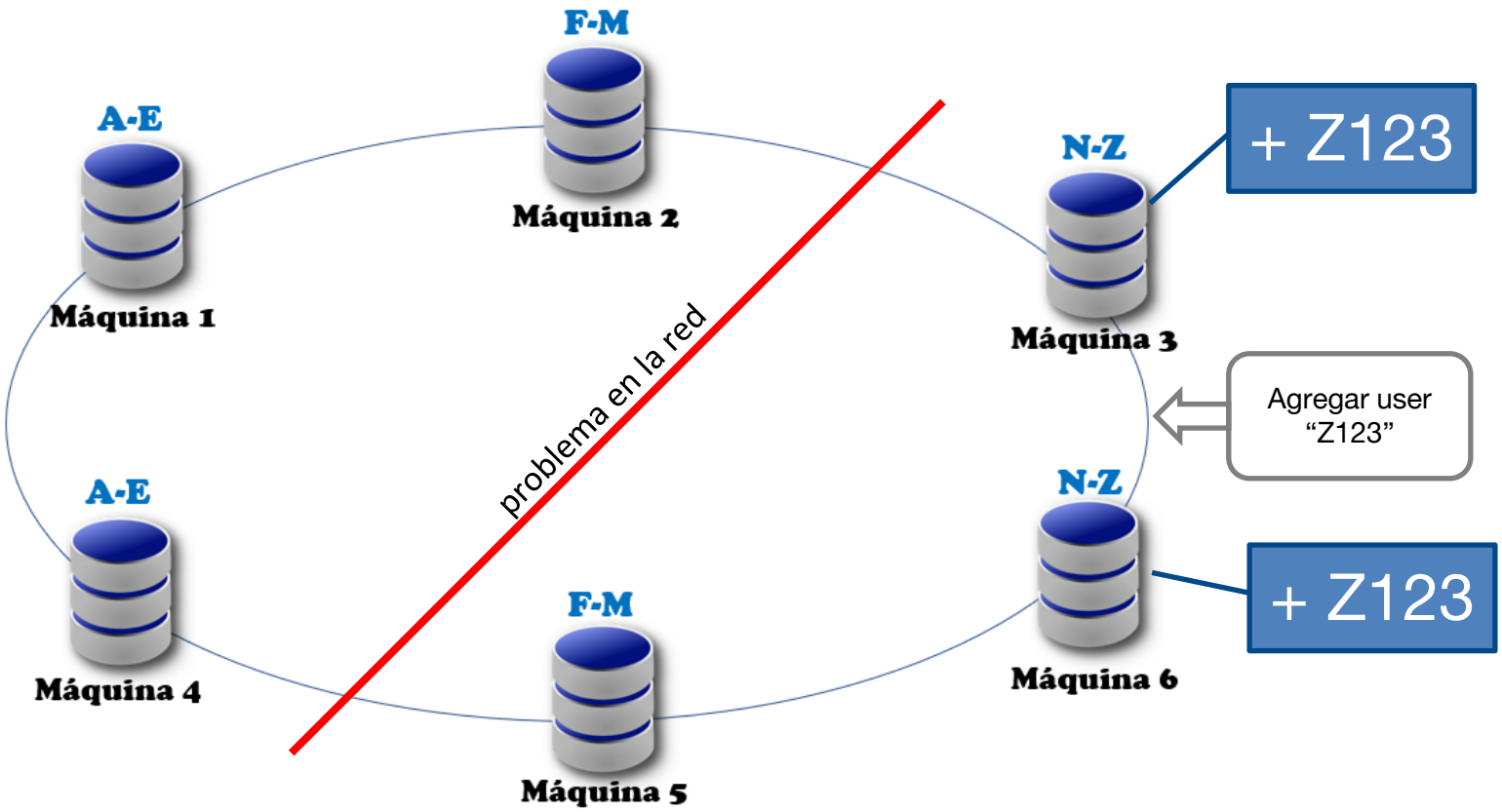
AP vs CP

¿Cuántos userID hay empezando con "Z"?



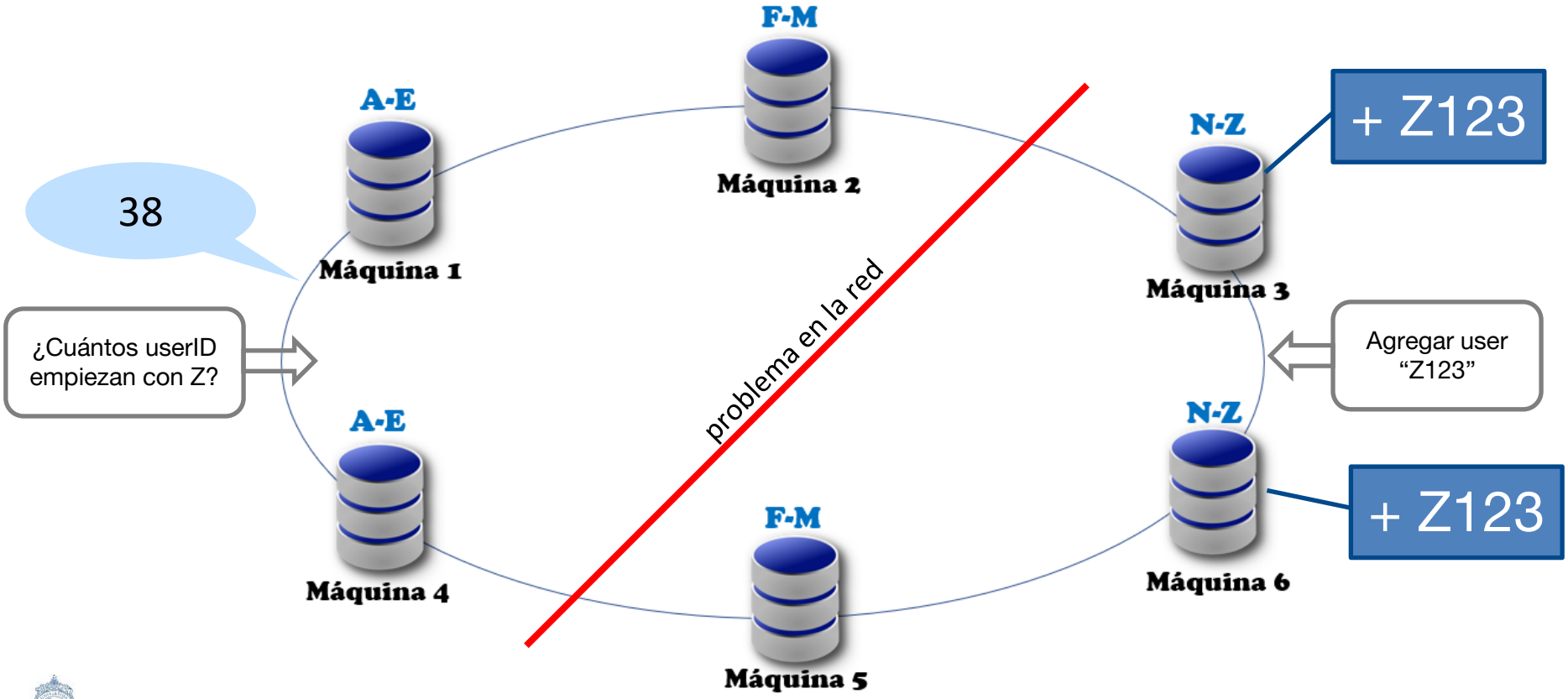
AP vs CP

¿Cuántos userID hay empezando con "Z"?



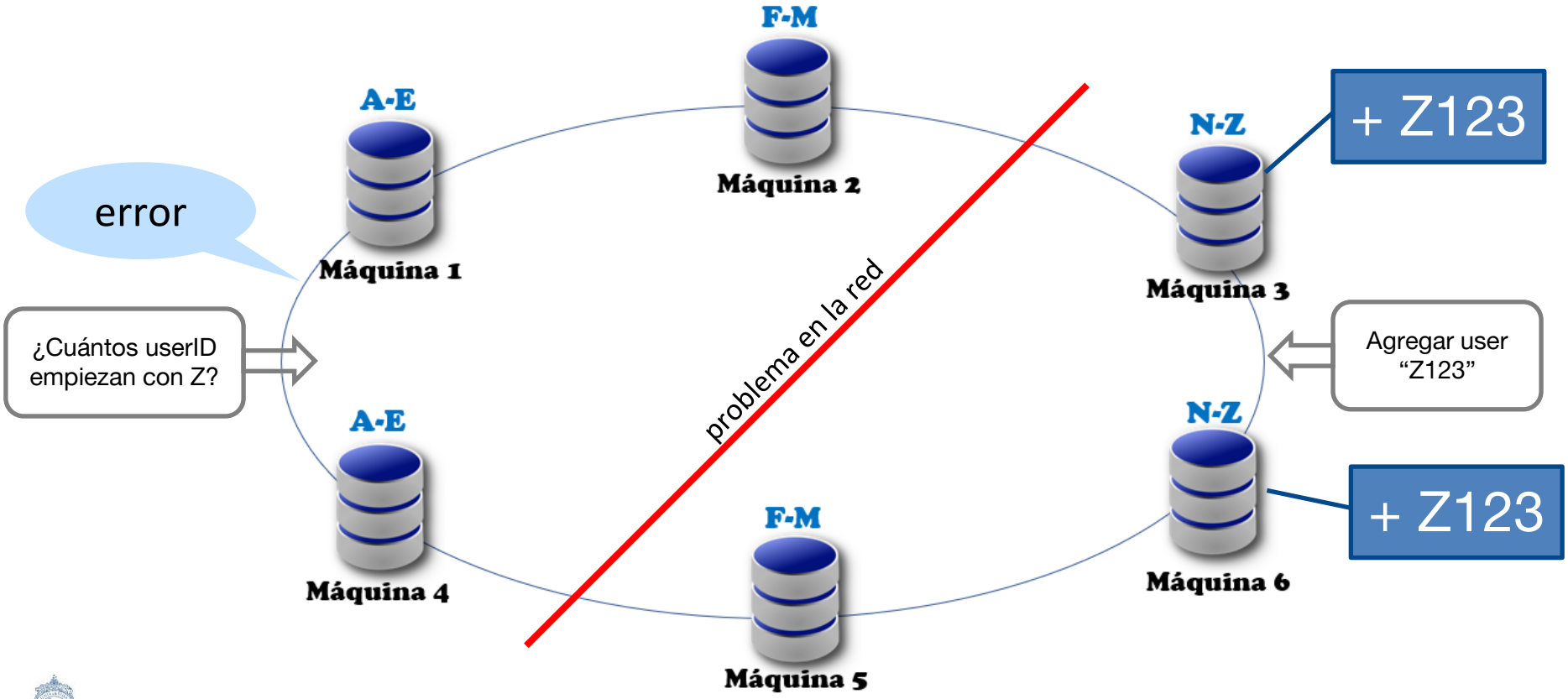
Un sistema AP

¿Cuántos userID hay empezando con "Z"?



Un sistema CP

¿Cuántos userID hay empezando con "Z"?



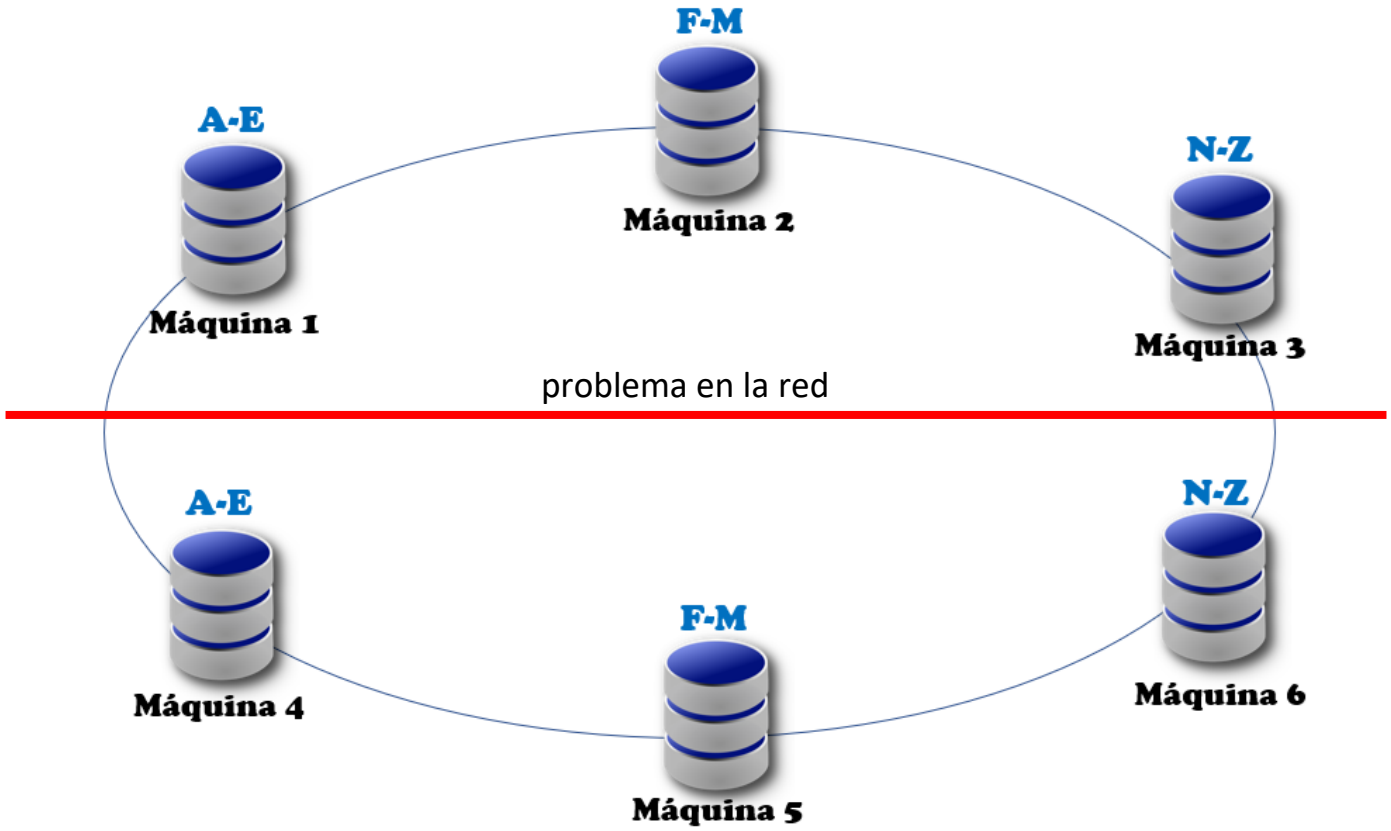
BASE

En la práctica, sistemas distribuidos fijan el P, y balancean entre C y A, sin elegir uno exclusivamente.

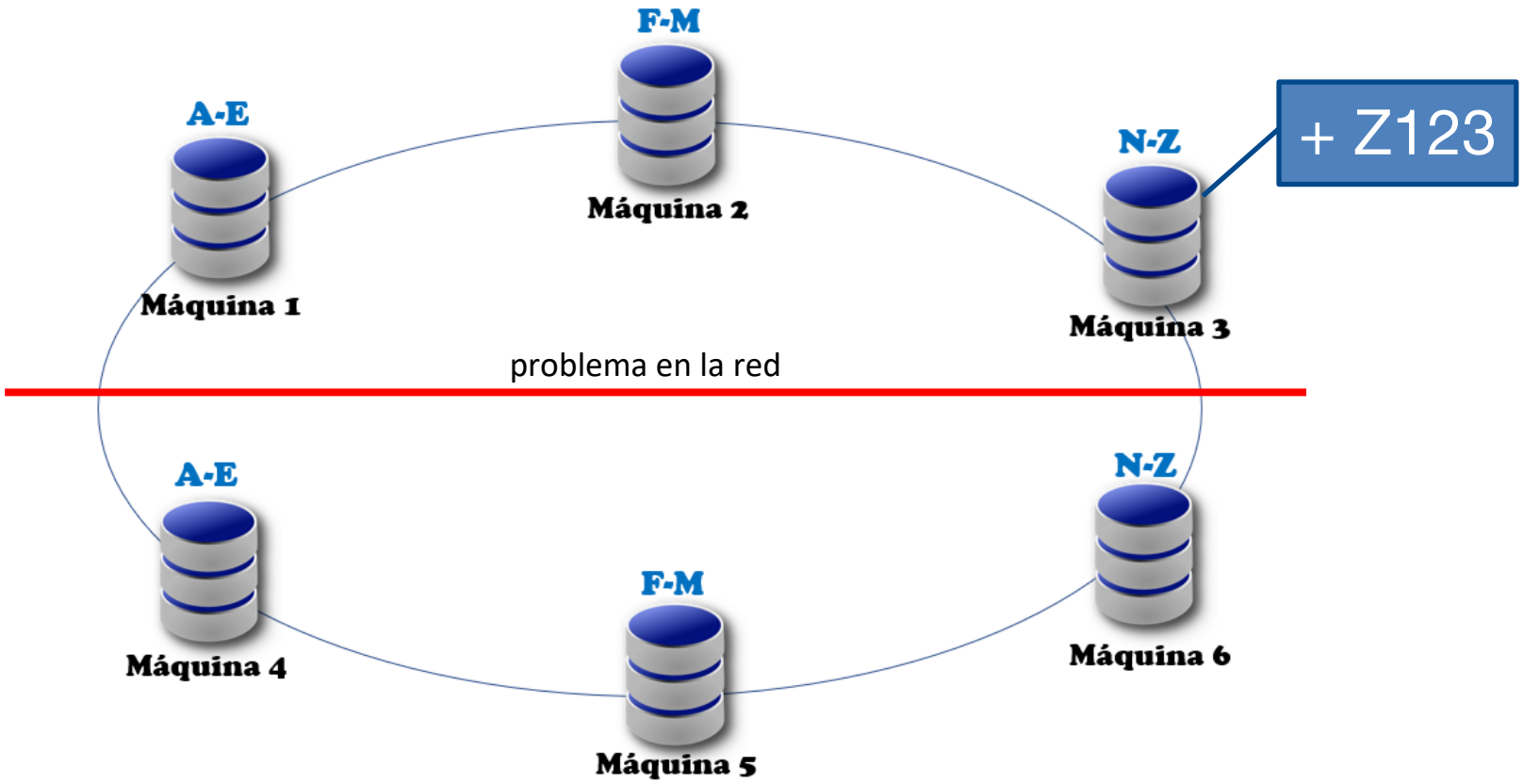
Paradigma BASE:

- Basically Available
- Soft state
- **Eventually consistent**

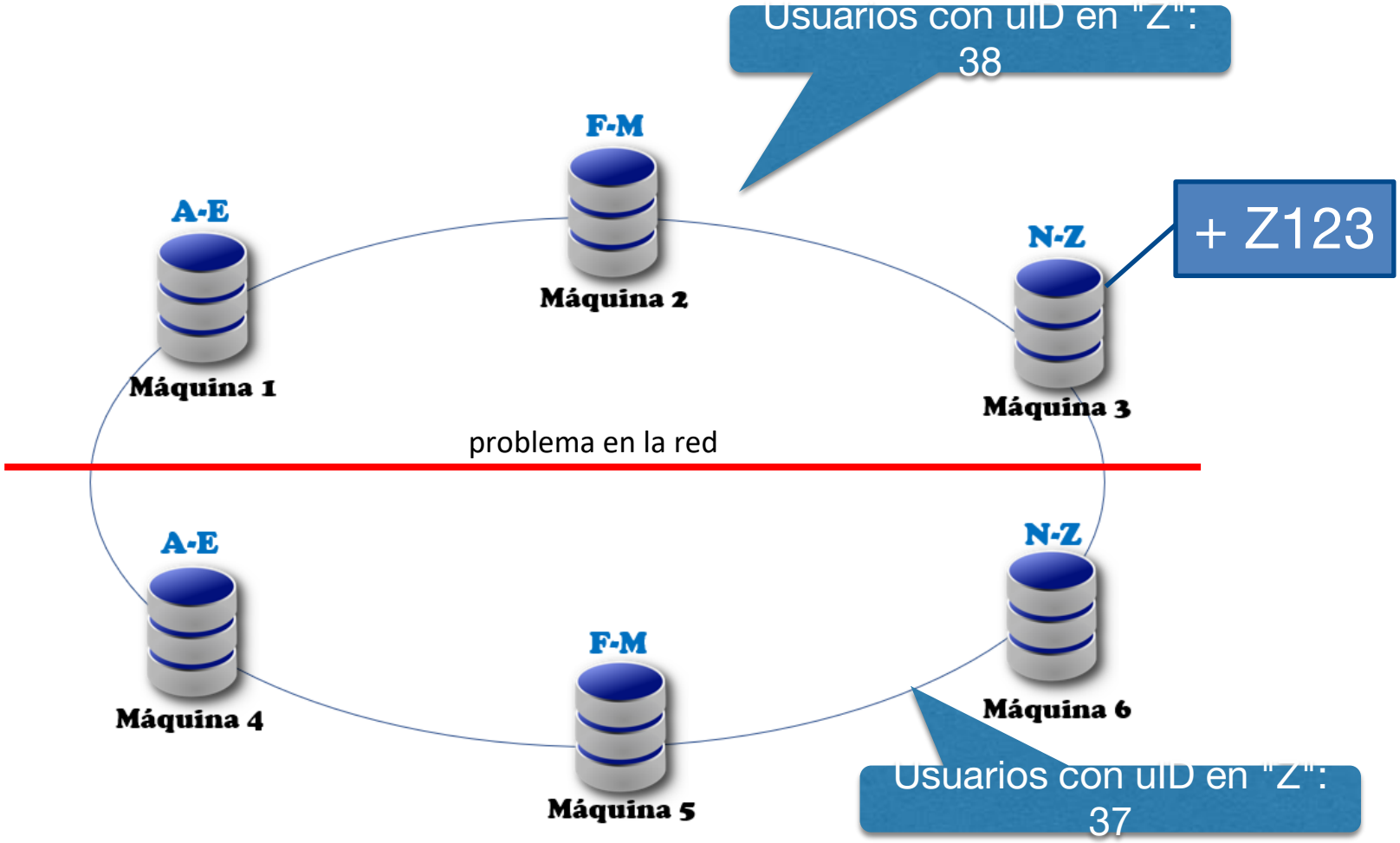
Consistencia eventual



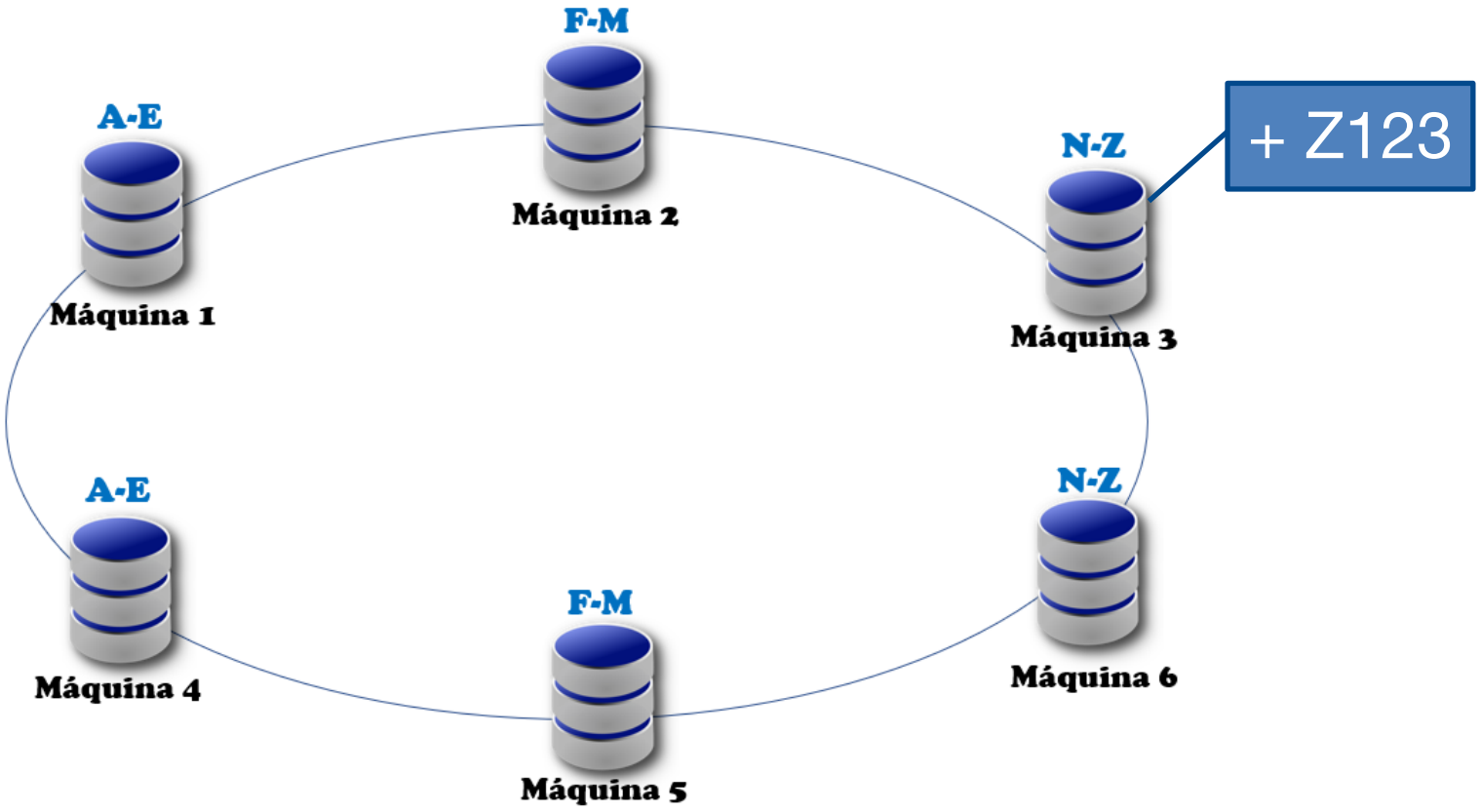
Consistencia eventual



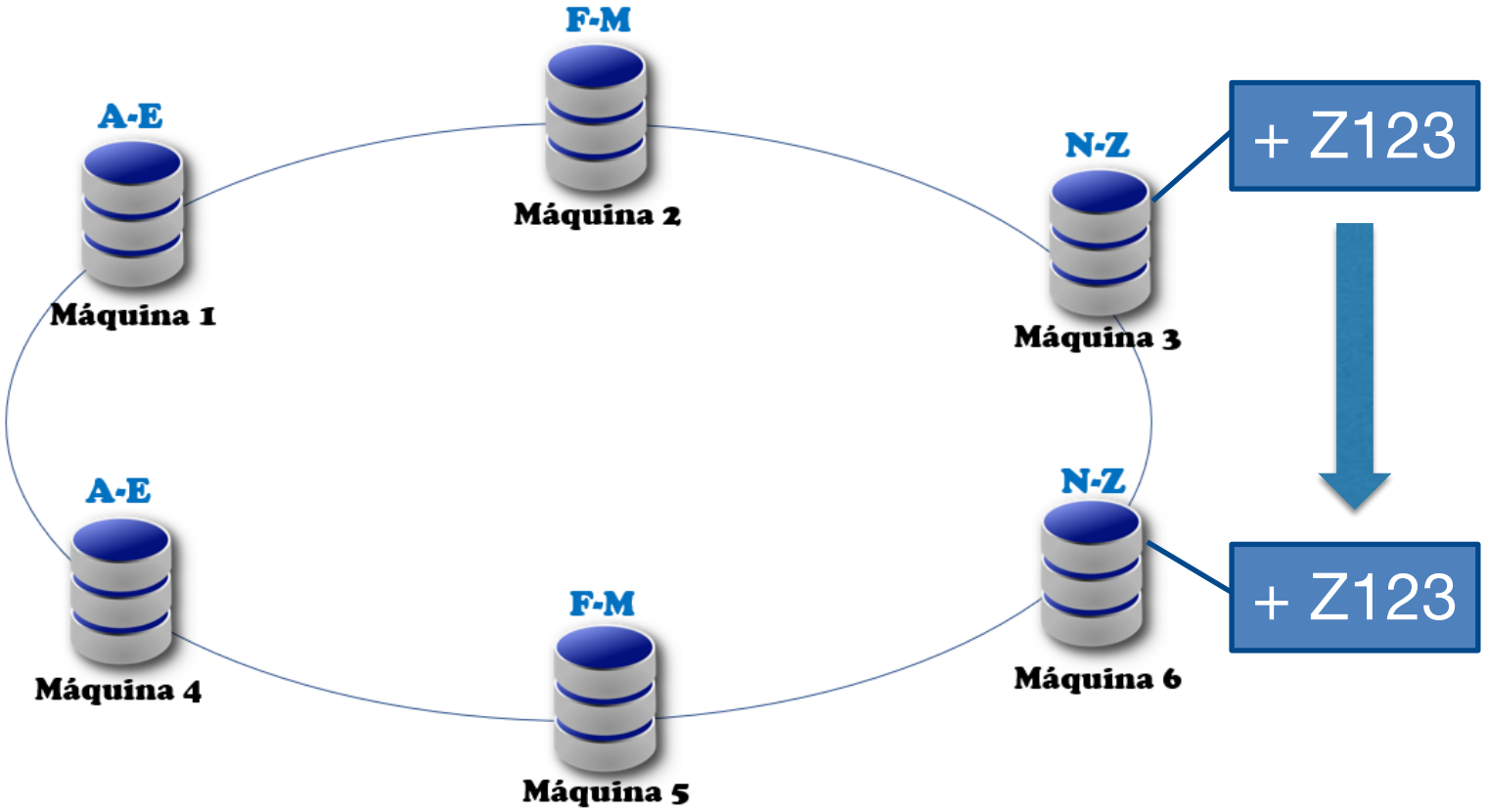
Consistencia eventual



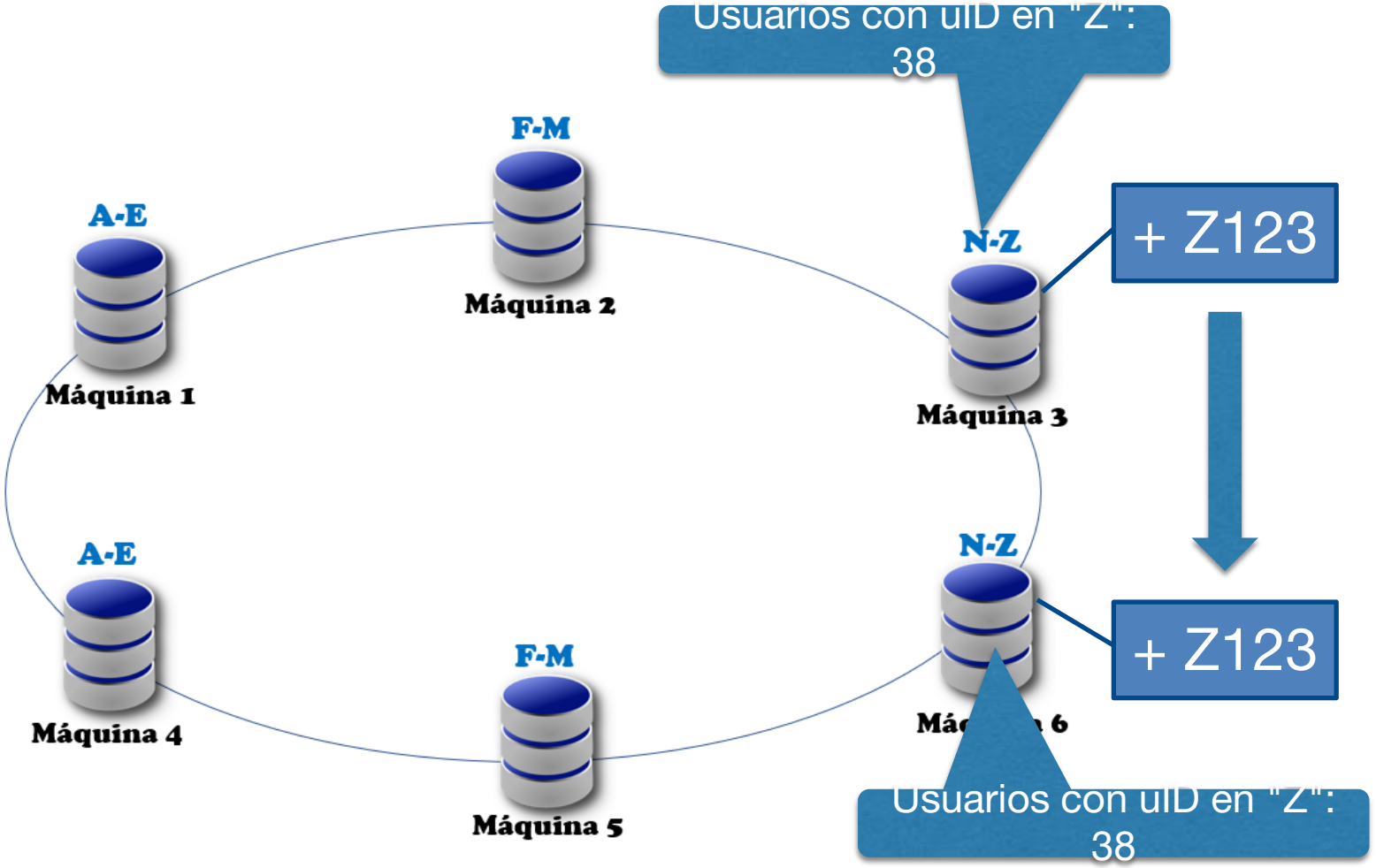
Consistencia eventual



Consistencia eventual



Consistencia eventual

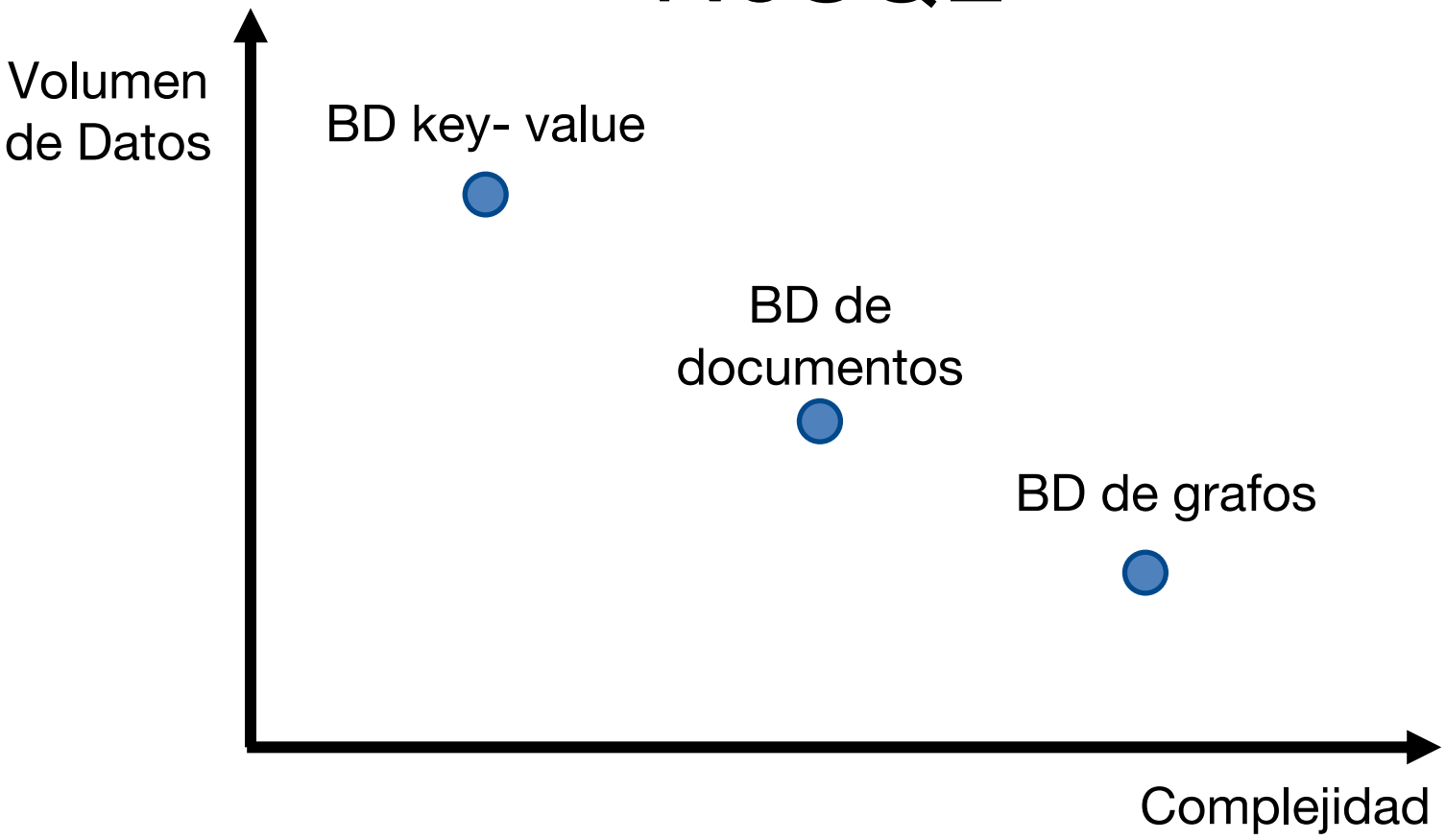


Sabores de NoSQL

Los más populares hoy en día:

- BD key-value
- BD de grafos
- BD de documentos

Sabores de NoSQL



BD Key - Value

Independientemente del esquema

- Arquitectura almacena información por medio de pares
- Cada par tiene una llave (identificador) y un valor

BD Key - Value

Operaciones cruciales:

- put(key,value)
- get(key)
- delete(key)

Key	Value
Chile	Santiago
Inglaterra	Londres
Escocia	Edinburgo
Francia	Paris
Alemania	Berlin
…	…

BD Key - Value

- Son grandes tablas de hash persistentes
- Esta categoría es difusa, pues muchas de las aplicaciones de otros tipos de BD usan key - value y hashing hasta cierto punto

Ejemplo más importante: Amazon Dynamo, otro es Redis

BD Key - Value

Puede representar cualquier valor

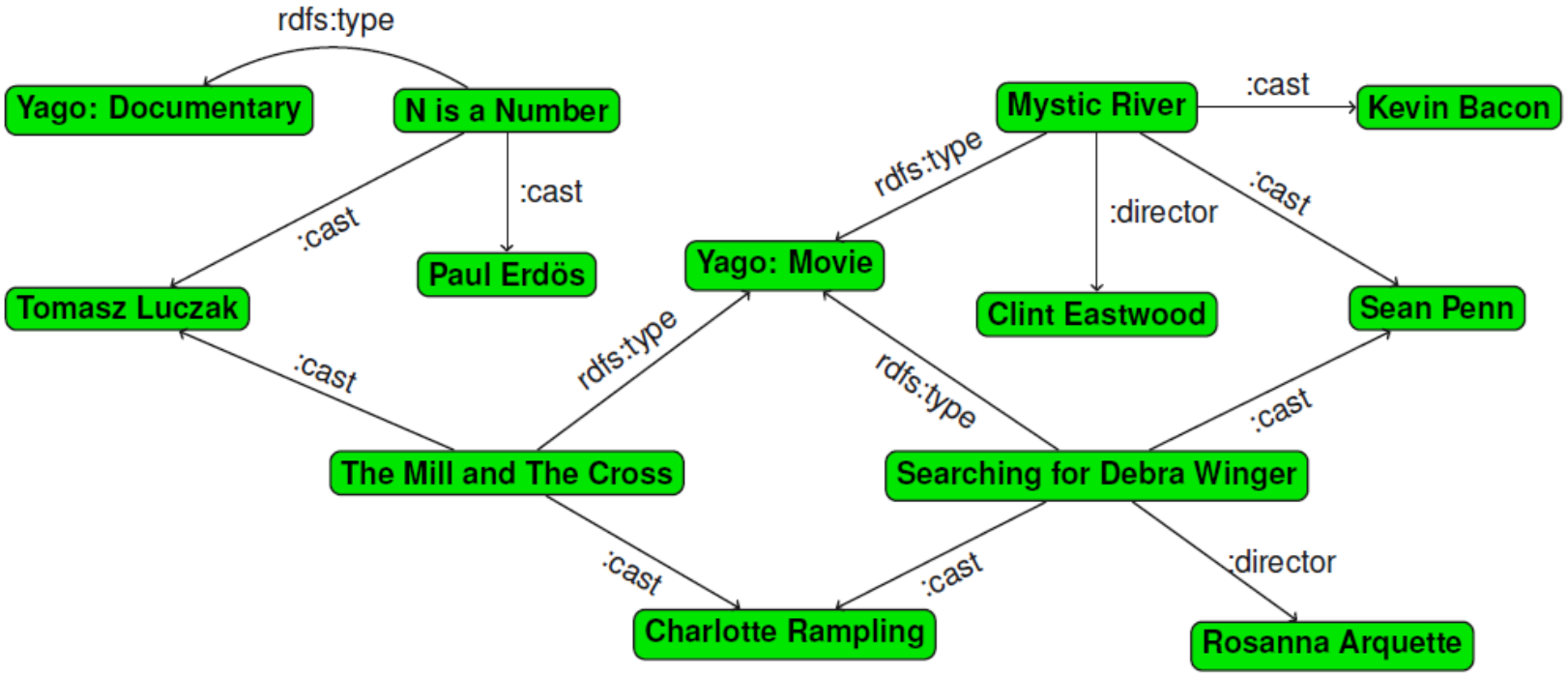
cartID	value
11789	usrID: "Juan", ítem: "Magic the Gathering Deck", value: …
12309	usrID: "Domagoj", item: "APEX XTX50 regulator set", value: …
…	…

BD de Grafos y RDF

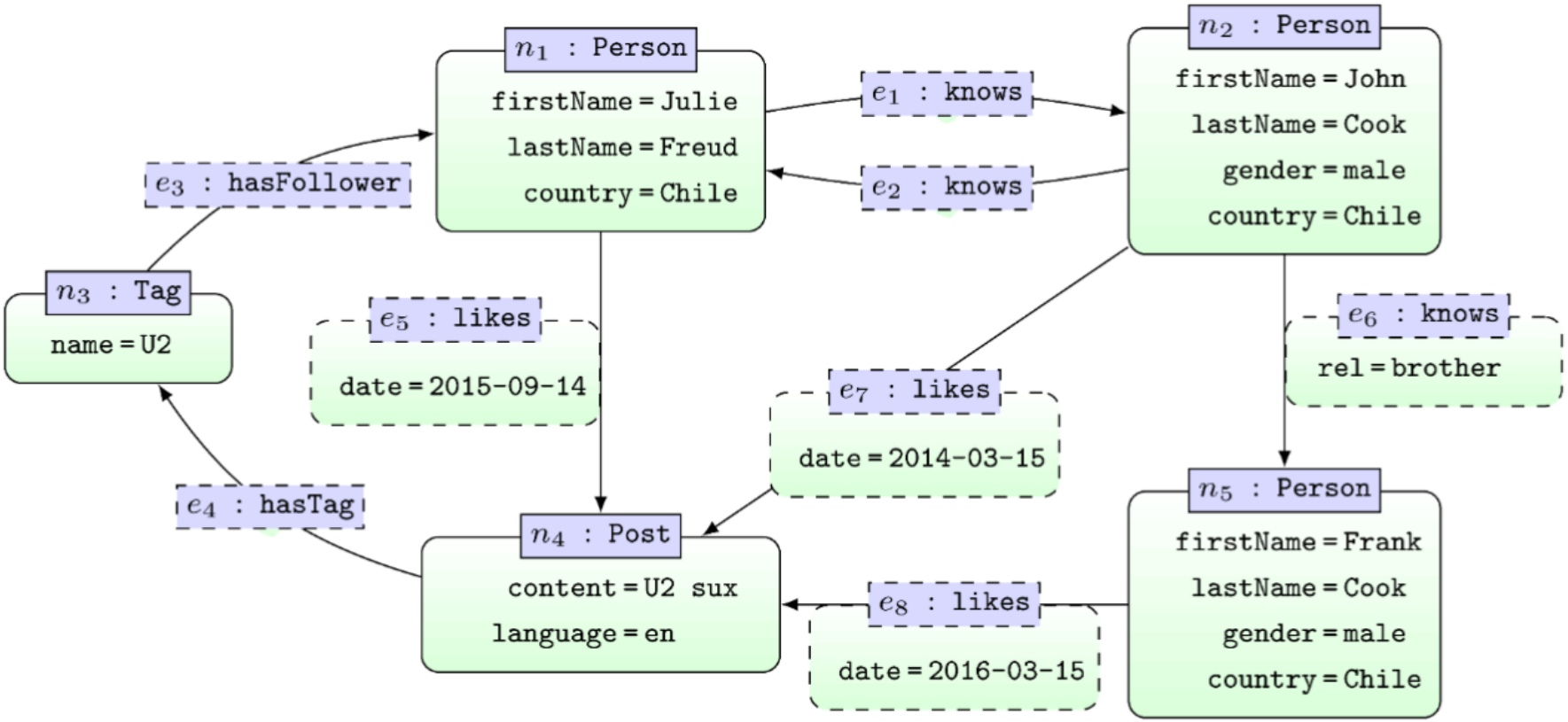
Especializadas para guardar relaciones

- En general, almacenan sus datos como property graphs
- Algunos ejemplos son Neo4J, Virtuoso, Jena, Blazegraph

BD de Grafos y RDF



BD de Grafos y RDF



BD Orientadas a Documentos

Especializadas en documentos

- CouchDB, MongoDB (estas y otras BD almacenan sus datos en documentos JSON)
- JSON no es el único estándar de documentos (por ejemplo, existe también XML)

BD Orientadas a Documentos

Parecidas a key-value stores:

- El valor es ahora un documento (JSON)
- Pueden agrupar documentos (colecciones)
- Lenguaje de consultas mucho más poderoso

BD Orientadas a Documentos

Usuarios	
Key	**JSON**
1	
2	…
…	…

JSON

Su nombre viene de JavaScript Object Notation

Estándar de intercambio de datos semiestructurados / datos en la Web

- JSON se acopla muy bien a los lenguajes de programación

JSON
Ejemplo


```
{
  "statuses": [
    {
      "id": 72545937362390630 4,
      "text": "@visitlondon: Have you been to any of these
               quirky London museums? https://t.co/tnrar8UttZ",
      "retweeted_status": {
        "metadata": {
          "result_type": "recent",
          "iso_language_code": "en"
        },
        "retweet_count": 239,
        "retweeted": false
      }
    }
  ]
}
```

JSON

La base son los pares key - value

```
{
  “nombre": “Matías”, “apellido": “Jünemann”
}
```

Valores pueden ser:

- Números
- Strings (entre comillas)
- Valores booleanos
- Arreglos (por definir)
- Objetos (por definir)
- null

JSON
Sintaxis

Los objetos se escriben entre {} y contienen una cantidad arbitraria de pares key - value

```
{
  “nombre": “Matías”, “apellido": “Jünemann”
}
```

JSON
Sintaxis

Los arreglos se escriben entre [] y contienen valores

```
{
  “profesores”: [
    {“nombre”: “Juan”, “apellido": “Reutter”},
    {“nombre”: “Cristian”, “apellido": “Riveros”},
    {“nombre”: “Marcelo”, “apellido": “Arenas”}
  ]
}
```

JSON vs SQL

SQL:

- Esquema de datos
- Lenguajes de consulta independientes del código

JSON:

- Más flexible, no hay que respetar necesariamente un esquema
- Más tipos de datos (como arreglos)
- Human - Readable

BD de documentos: ¿para qué?

Especializadas en documentos: almacenan muchos documentos JSON

- Si quiero libros: un documento JSON por libro
- Si quiero personas: un documento JSON por persona

Notar que esto es altamente jerárquico

BD de documentos

Qué hacen bien:

- Si quiero un libro o persona en particular
- Cruce de información **simple**

Muy útiles a la hora de desplegar información en la web

BD de documentos

Pueden verse como un caché de una BD relacional
¿Por qué?

Caché de BD
SQL

Students

StudentID	Nombre	Carrera
1	Alice Cooper	Computación
2	David Bowie	Todas
3	Charly García	Ingeniería Civil
…	…	…

Courses

courseID	name	year
IIC2413	Databases	2020
IMT3830	Game Theory	2020
…	…	…

Takes

courseID	StudentID
IIC2413	1
IIC2413	2
IMT3830	2
…	…

Caché de BD
SQL

Lista de alumnos por curso:

- SQL tiene que hacer un join
- En BD documentos prepararemos esta información

Caché de BD
SQL

Colección "**Courses**"

```
{
    "courseID": IIC2413,
    "name": "Databases",
    "year": 2020,
    "students":[
        {
            "studentID": 1,
            "name": "Alice Cooper"
        },
        {
            "studentID": 2,
            "name": "David Bowie"
        },
        ...
    ]
}
```

Caché de BD
SQL

Colección "**Courses**"

```
{
    "courseID": IIC2413,
    "name": "Databases",
    "year": 2020,
    "students":[
        {
            "studentID": 1,
            "name": "Alice Cooper"
        },
        {
            "studentID": 2,
            "name": "David Bowie"
        },
        ...
    ]
}
```

```
{
    "courseID": IMT3830,
    "name": "Game Theory",
    "year": 2020,
    "students":[
        {
            "studentID": 2,
            "name": "David Bowie"
        },
        {
            "studentID": 3,
            "name": "Charly García"
        },
        ...
    ]
}
```

BD de documentos

Qué hacen bien:

- Si quiero lista de alumnos de un curso
- Si quiero nombres de todos los cursos
- Si quiero todo los cursos tomados por David

Muy útiles a la hora de desplegar información en la web

BD de documentos

Qué hacen mal:

- Manejo de información que cambia mucho
- Cruce de información no trivial

BD de documentos

Colección "**Courses**"

- Todos los alumnos que toman IIC2413 y IMT3830

```
{
    "courseID": IIC2413,
    "name": "Databases",
    "year": 2020,
    "students":[
        {
            "studentID": 1,
            "name": "Alice Cooper"
        },
        {
            "studentID": 2,
            "name": "David Bowie"
        },
        ...
    ]
}
```

```
{
    "courseID": IMT3830,
    "name": "Game Theory",
    "year": 2020,
    "students":[
        {
            "studentID": 2,
            "name": "David Bowie"
        },
        {
            "studentID": 3,
            "name": "Charly García"
        },
        ...
    ]
}
```

BD de documentos

Colección "**Courses**"

- Todos los alumnos que toman IIC2413 y IMT3830

Efectivamente hay que hacer un nested loop join:

- Iterar por todos los alumnos de IMT3830
- Iterar por todos los alumnos de IIC2413
- Ver si hacen el join

MongoDB soporta JavaScript y Python

- Se puede hacer, pero no es elegante

En resumen

BD de documentos:

- Útiles para despliegue de información estática
- Búsquedas simples
- Cruces muy sencillos

BD SQL:

- Información cambia mucho
- Tengo que hacer cruces cada rato
- Necesito ACID

BD Documentos y BASE

- Distintas aplicaciones en una misma base de datos acceden a distintos documentos al mismo tiempo
- En general diseñadas para montar varias instancias que (en teoría) tienen la misma información
- Propagan updates en forma descoordinada

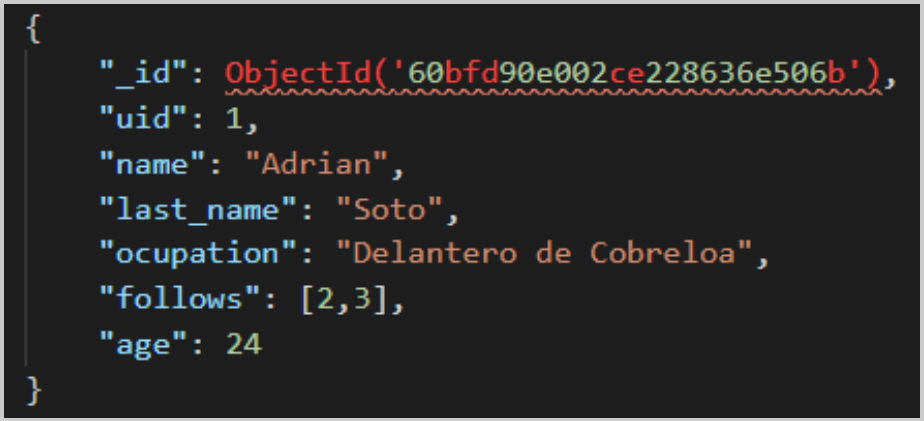
Proveen “**Consistencia Eventual**”

Consistencia Eventual

La consistencia eventual puede generar problemas

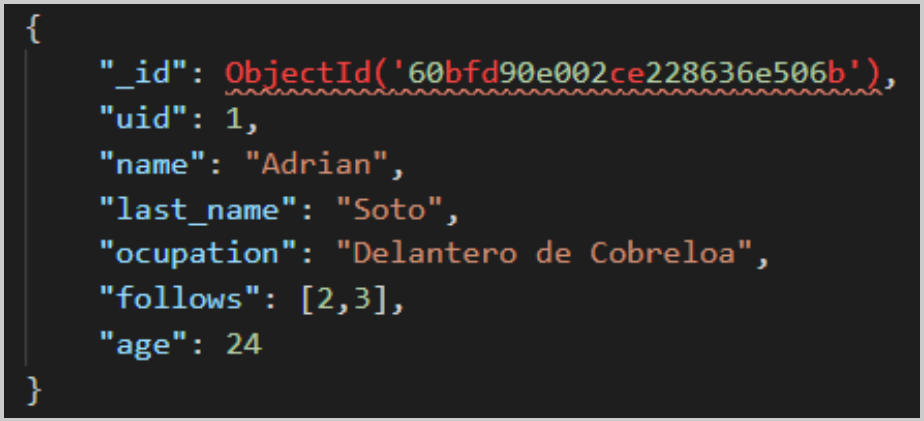
Si dos aplicaciones intentan acceder al mismo documento en MongoDB, estas pueden ser versiones diferentes del documento

MongoDB

Usuarios	
Key	**JSON**
60bfd90e002ce228636e506b	
60bfd90e002ce228636e5215	…
…	…

MongoDB

Colección: una agrupación de documentos similares

Usuarios	
Key	**JSON**
60bfd90e002ce228636e506b	
60bfd90e002ce228636e5215	…
…	…

MongoDB

Base de Datos: contienen colecciones relacionadas

Usuarios

Key	JSON
…	…

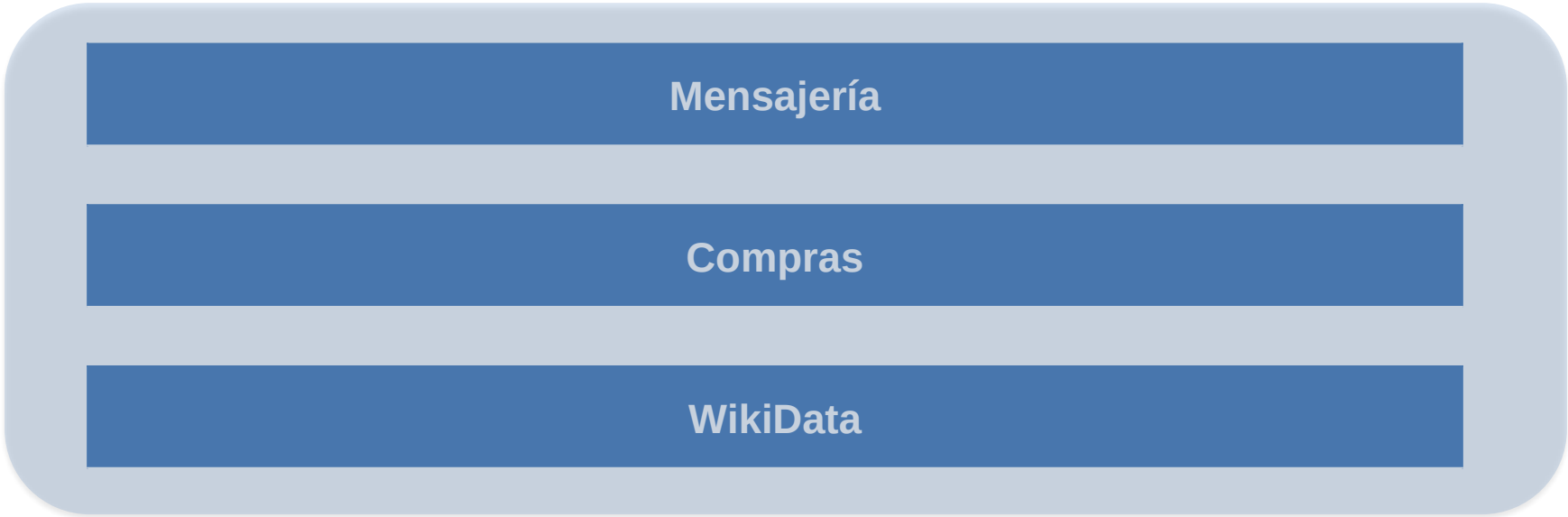
Mensajes

Key	JSON
…	…

Likes

Key	JSON
…	…

MongoDB



Un servidor contiene varias bases de datos

Consultando a MongoDB

show dbs … muestra bases de datos disponibles
use dbName … ahora usamos base de datos dbName
show collections … colecciones en nuestra base de datos

db.colName.find() … todos los documentos en la colección colName

db.colName.find().pretty() … pretty print

db.colName.find({"name": "Adrian"}) … selección

db.colName.find({"age": {$gte:23}}) … selección

db.colName.find({"age": {$gte:23}},{"name":1}) … proyección

Ver más en: http://aidanhogan.com/teaching/cc5212-1-2020/lectures/MDP2020-10.pdf

Text Search en MongoDB

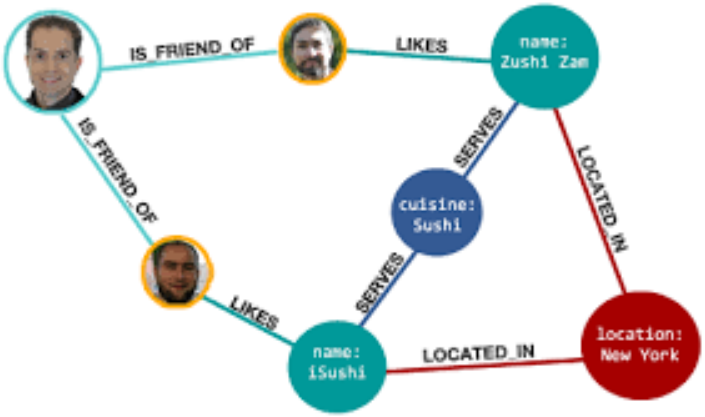
Un índice especial … permite búsqueda rápida de texto

db.colName.createIndex({"attributeName":"text"})

db.users.find({$text: {$search: "Delantero de Cobreloa"}})

Ver más en: https://www.youtube.com/watch?v=vR97-4UG7x0

Volviendo a los grafos…



WIKIDATA

elis silvestris catus

266 idiomas

ículo Discusión Leer Ver código fuente Ver historial Herramientas

edirigido desde «Gato»)

«Gato» y «Gata» redirigen aquí. Para otras acepciones, véanse Gato (desambiguación) y Gata (desambiguación).

gato doméstico[1] [2] (***Felis silvestris catus***) llamado más comúnmente **gato**, y de ma coloquial **minino**,[3] **michino**,[4] y algunos nombres más, es un mamífero rnívoro de la familia *Felidae*.

El nombre actual en muchas lenguas proviene del latín vulgar *catus*. Paradójicamente, *catus* aludía a los gatos salvajes, mientras que los gatos domésticos eran llamados *felis*.

de mutaciones genéticas, cruzamiento y selección artificial, hay s. Algunas, como la raza *sphynx* o la *peterbald* están desprovistas de en de cola, como los gatos de la raza *bobtail* o la *manx*, y algunas nes atípicas, como los llamados *gatos azules*.

nica a través de vocalizaciones. Las más populares son su aullido y el ronroneo, pero puede aullar, gemir, gruñir y bufar.[5] poses o expresiones que informan, a sus congéneres, sus enemigos s, de su ánimo o sus intenciones.

o, es el animal doméstico más popular como mascota, como ayuda a roedores o ambas cosas. En el 2017, la población mundial os estaba en seiscientos millones de felinos. En esta cifra se que son mascota, gatos callejeros (sin hogar) y gatos salvajes; s gatos silvestres alrededor de 100 millones. El país considerado que más felinos tiene como mascota es Estados Unidos. Rusia oximadamente 23 millones de gatos domésticos en 2021 n el país europeo con mayor población de este tipo de felinos.[6] [7] [8]

anico de presas potenciales, por su alta eficiencia como depredador éxito reproductivo —especialmente si se suministra artificialmente lonias sin tomar medidas adicionales para limitar su fertilidad— el está incluido en la lista de las cien especies exóticas invasoras más do[9] de la Unión Internacional para la Conservación de la Naturaleza.

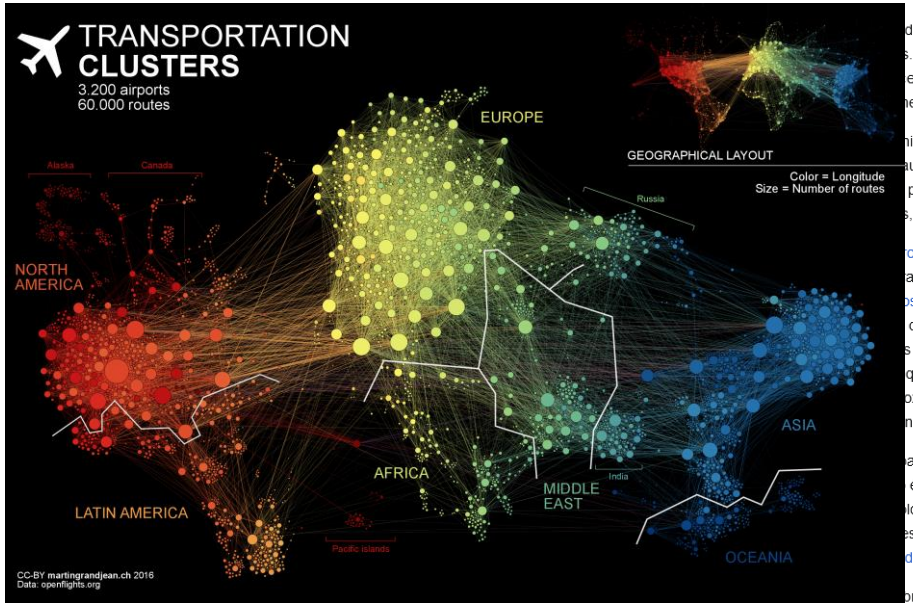
bres comunes, como michi,[10] [11] micho,[12] mizo,[13] miz,[14]

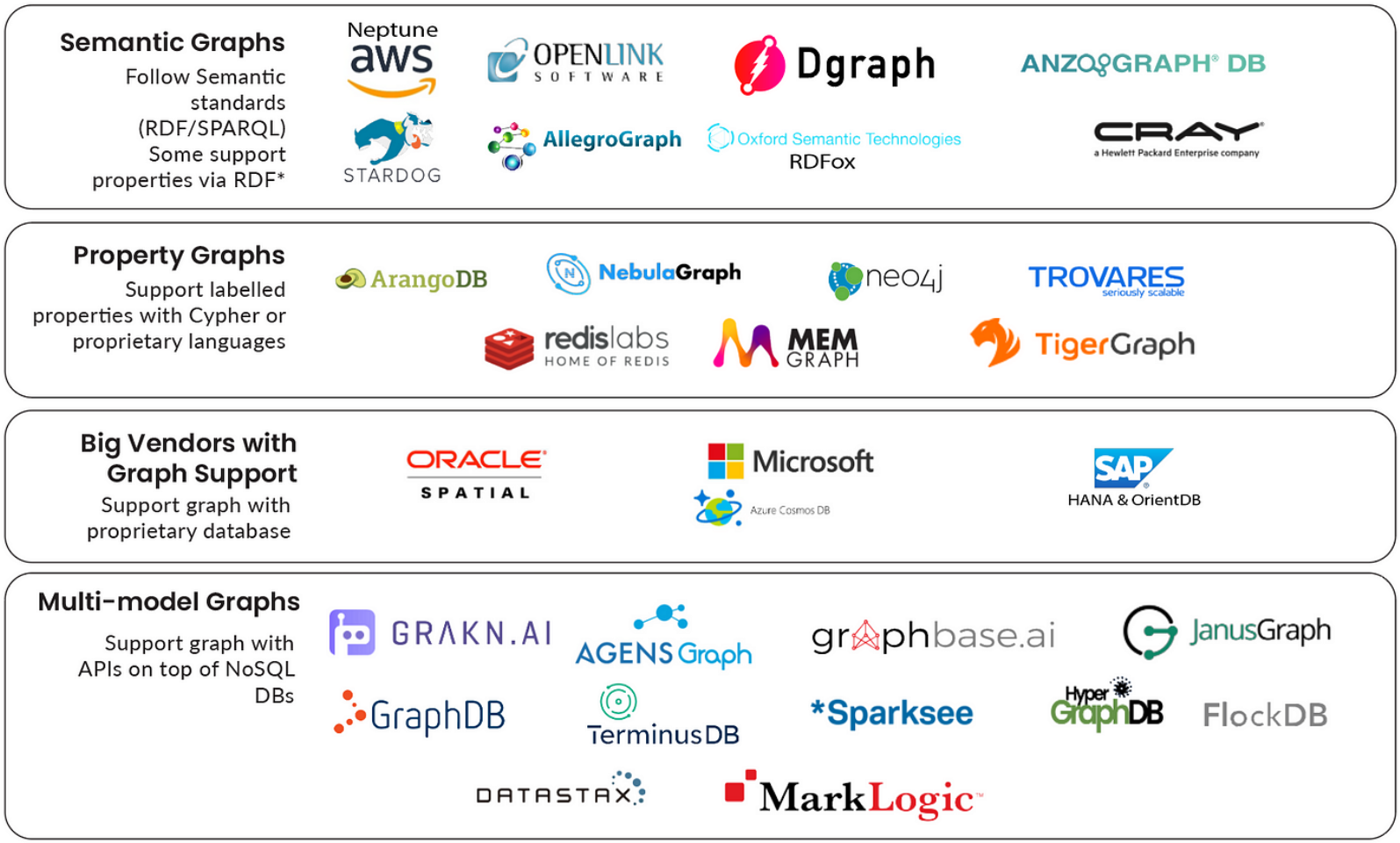
Gato doméstico

Gatos de diferentes razas

Estado de conservación	
Domesticado	
Taxonomía	
Reino:	Animalia
Filo:	Chordata
Subfilo:	Vertebrata
Clase:	Mammalia
Subclase:	Theria
Infraclase:	Placentalia
Orden:	Carnivora
Suborden:	Feliformia
Familia:	Felidae
Subfamilia:	Felinae
Género:	*Felis*
Especie:	*F. silvestris*
Subespecie:	***F. s. catus*** Schreber, 1775
Sinonimia	

- *Felis catus* Linnaeus, 1758
- *Felis silvestris domesticus*




Semantic Graphs
Follow Semantic standards (RDF/SPARQL) Some support properties via RDF*
Neptune
aws
OPENLINK SOFTWARE
Dgraph
ANZOGRAPH® DB
STARDOG
AllegroGraph
Oxford Semantic Technologies
RDFox
CRAY
a Hewlett Packard Enterprise company
Property Graphs
Support labelled properties with Cypher or proprietary languages
ArangoDB
NebulaGraph
neo4j
TROVARES
seriously scalable
redislabs
HOME OF REDIS
MEM GRAPH
TigerGraph
Big Vendors with Graph Support
Support graph with proprietary database
ORACLE
SPATIAL
Microsoft
Azure Cosmos DB
SAP
HANA & OrientDB
Multi-model Graphs
Support graph with APIs on top of NoSQL DBs
GRAKN.AI
AGENS Graph
graphbase.ai
JanusGraph
GraphDB
TerminusDB
*Sparksee
HyperGraphDB
FlockDB
DATASTAX
MarkLogic™

IMFD Chile

- Instituto Milenio Fundamentos de los Datos
  - Instituto de investigación interdisciplinario (Computación/Ciencias Sociales)
  - Enfocado en proyectos de gran escala
  - Uno de ellos: **“Construir una base de datos de grafos”**

MillenniumDB

- ¿Por qué?
  - Expertos en DB: M. Arenas, J. Reutter, C. Riveros, J. Pérez, D. Vrgoč
  - Semantic Web: A. Hogan, C. Gutierrez, R. Angles
  - Algorithms/compression: G. Navarro, D. Arroyuelo

MillenniumDB

- Open source
  - “Sandbox” de pruebas para algoritmos
  - Verificar hipótesis de investigación
  - Soportar Wikidata
  - Verificar si la teoría de computación vale la pena

- Involucrados
  - V. Calisto, C. Rojas (Chief Engineers)
  - B. Farías, G. Toro, …
  - T. Hehuer, K. Bosonney, J. Romero, …
  - D. Vrgoč, …

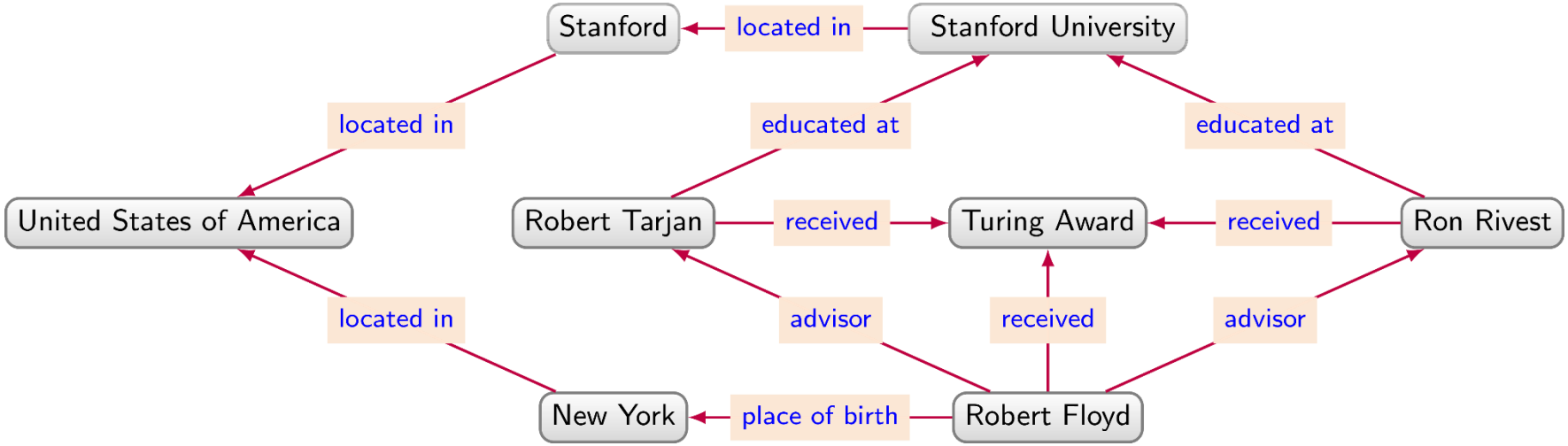
MillenniumDB: Highlights

- RDF/SPARQL y Property Graph
  - Multi modelo en el mismo motor de BD
  - SPARQL extendido con features de GQL

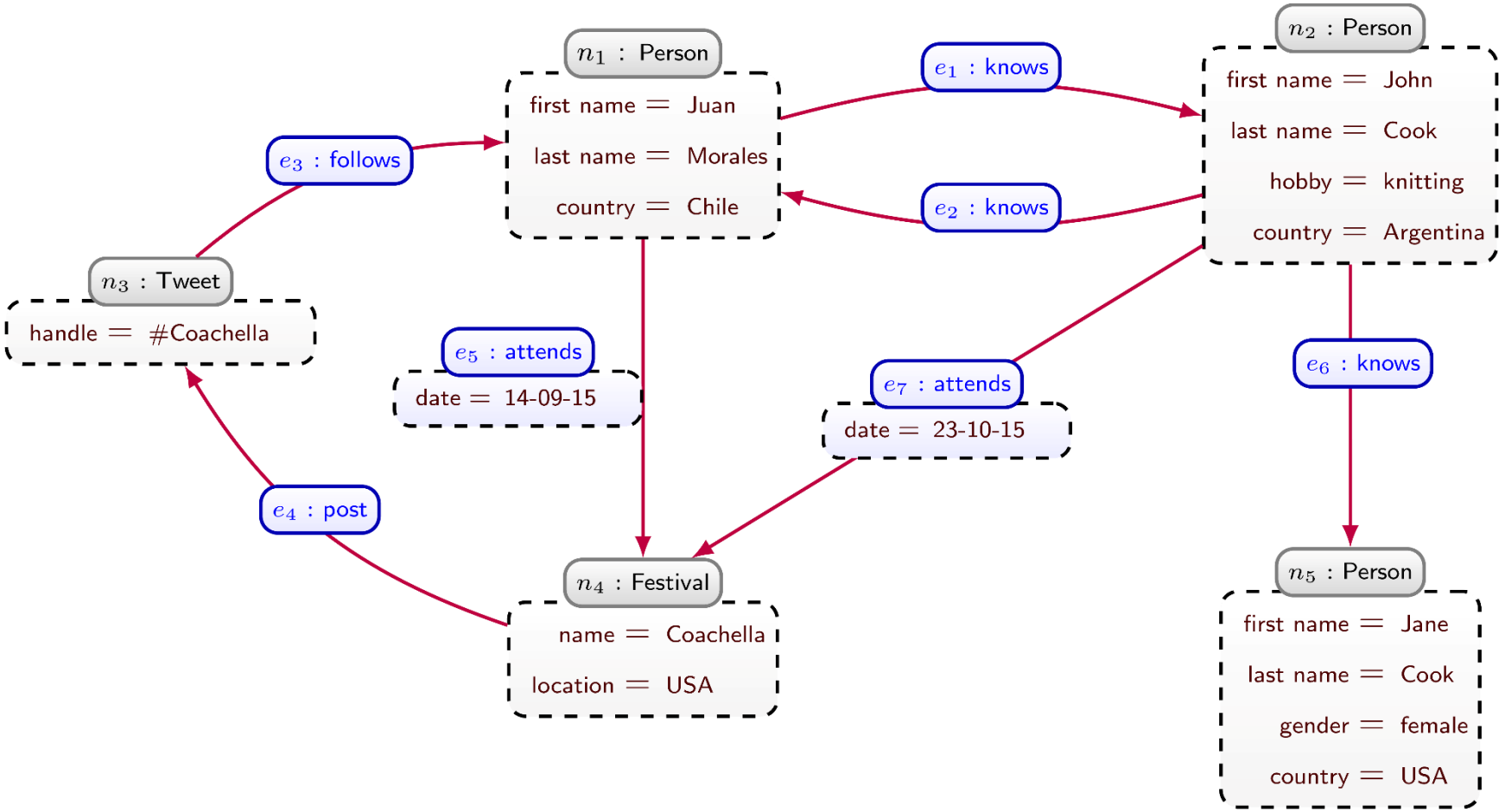
- Pipeline clásico de BD relacional
  - “Quasi-relacional”

- Arquitectura soporta endpoints públicos

MillenniumDB: RDF

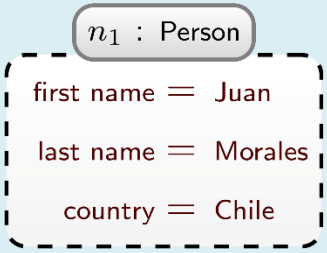


MillenniumDB: Property Graph

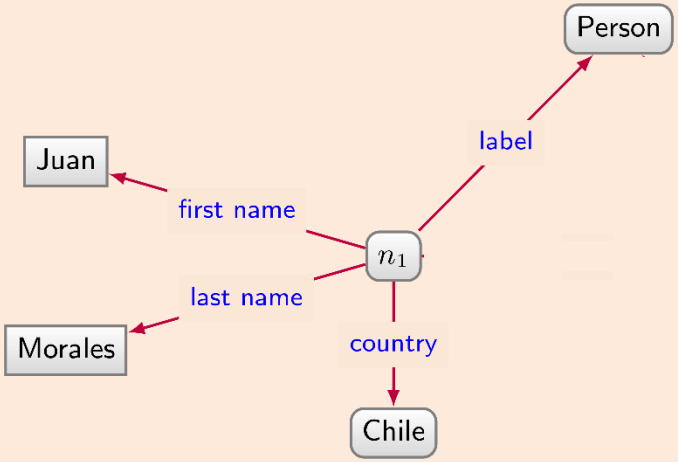


MillenniumDB: RDF vs Property Graph

Property Graphs

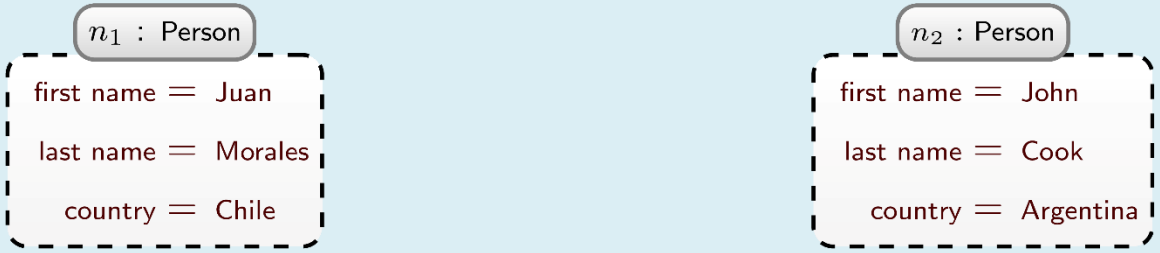


RDF

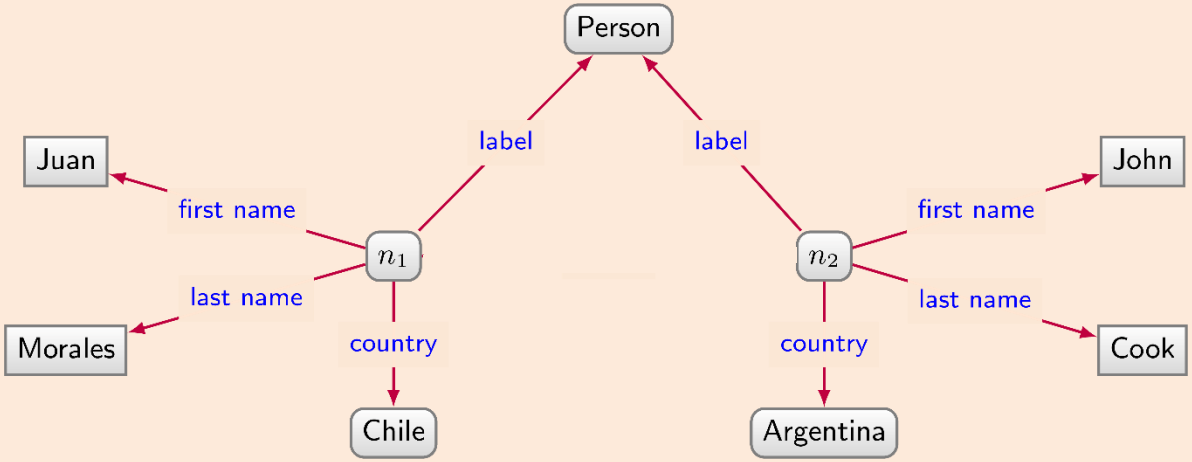


MillenniumDB: RDF vs Property Graph

Property Graphs

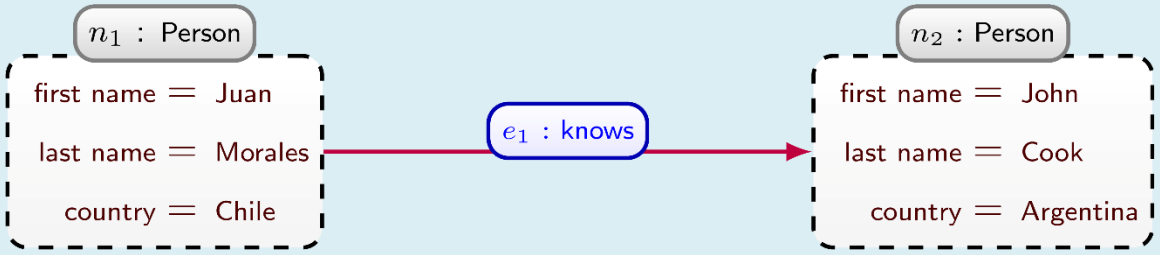


RDF

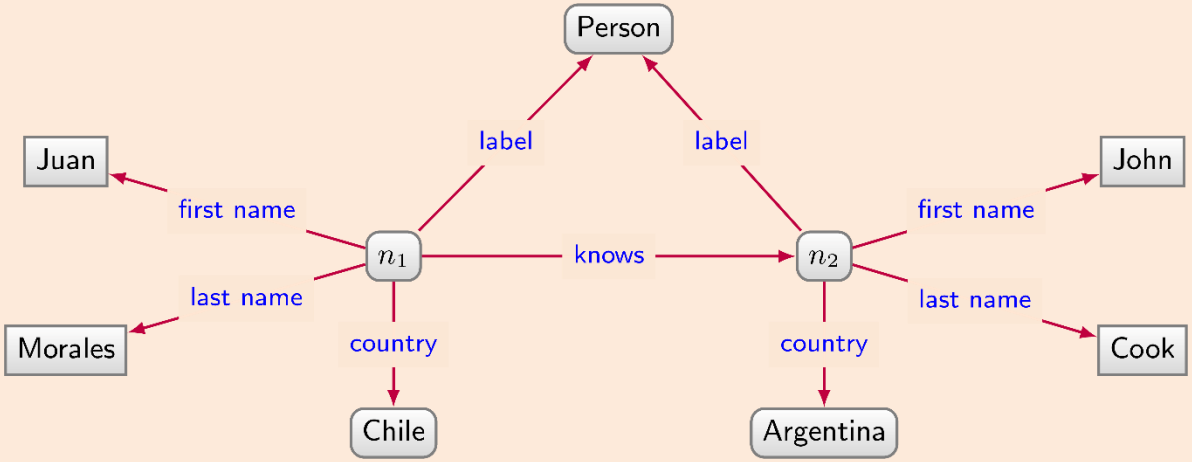


MillenniumDB: RDF vs Property Graph

Property Graphs

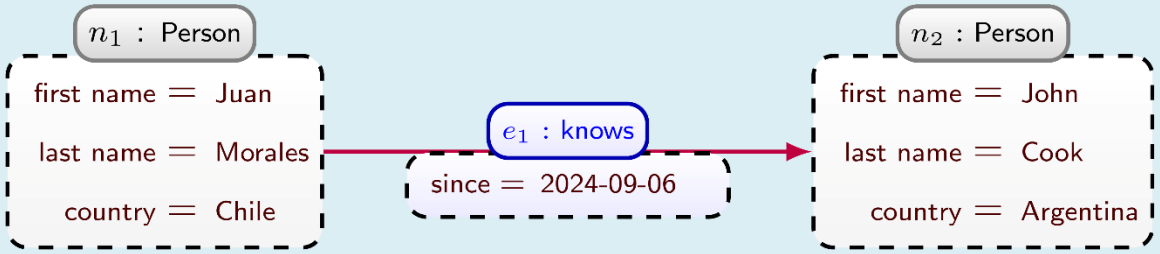


RDF

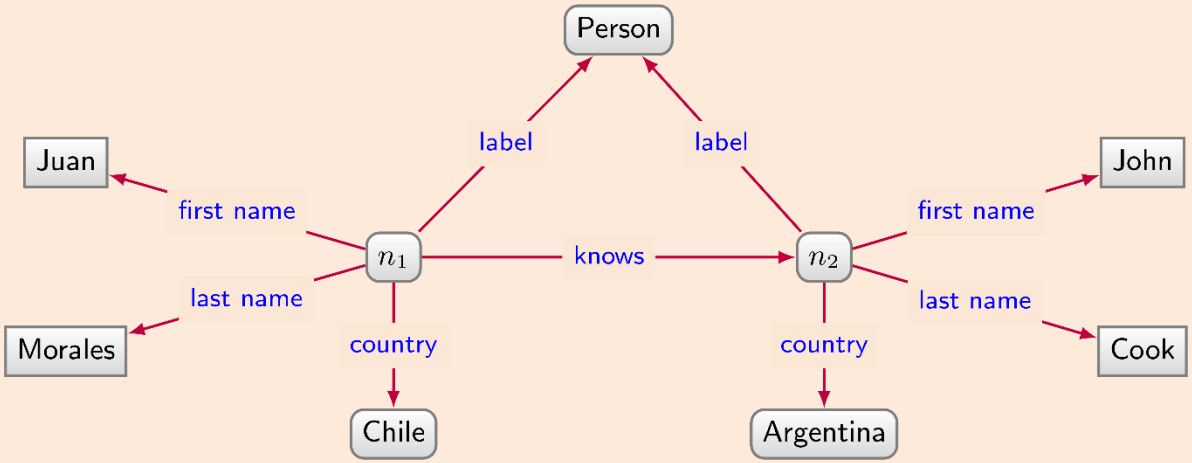


MillenniumDB: RDF vs Property Graph

Property Graphs

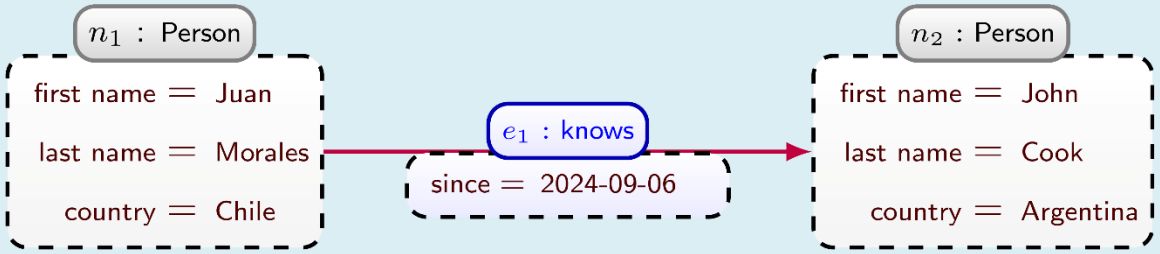


RDF

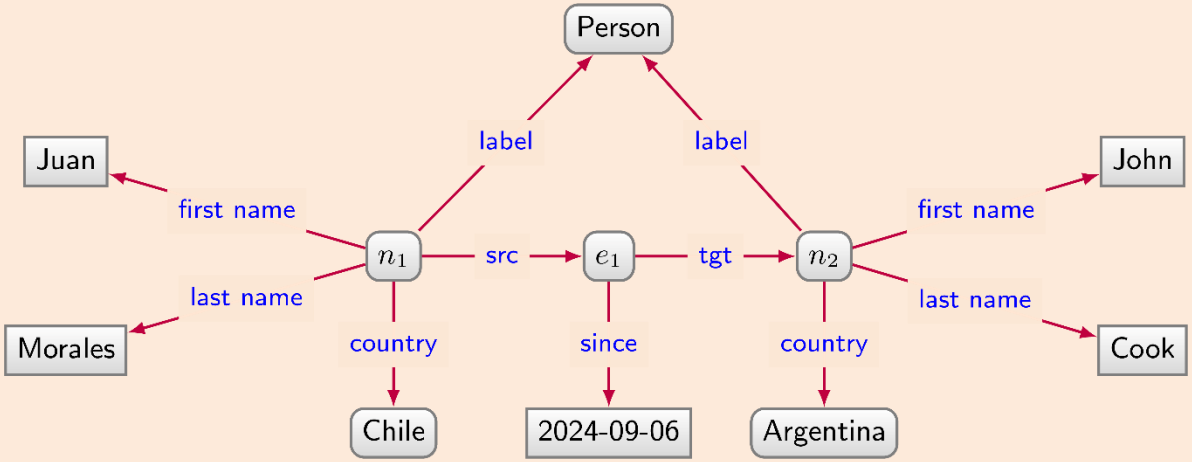


MillenniumDB: RDF vs Property Graph

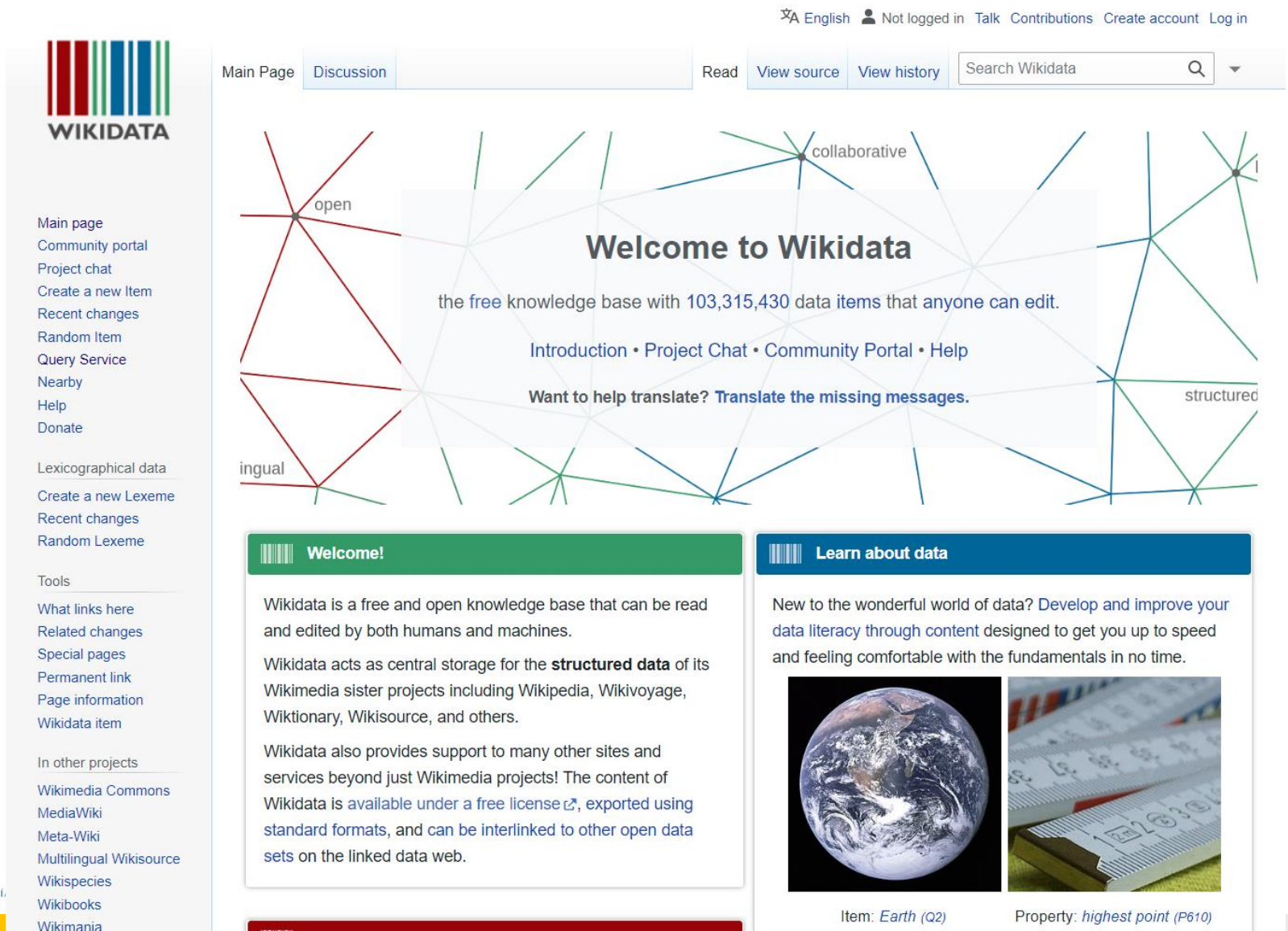
Property Graphs



RDF



Wikidata: Wikipedia, pero en grafos



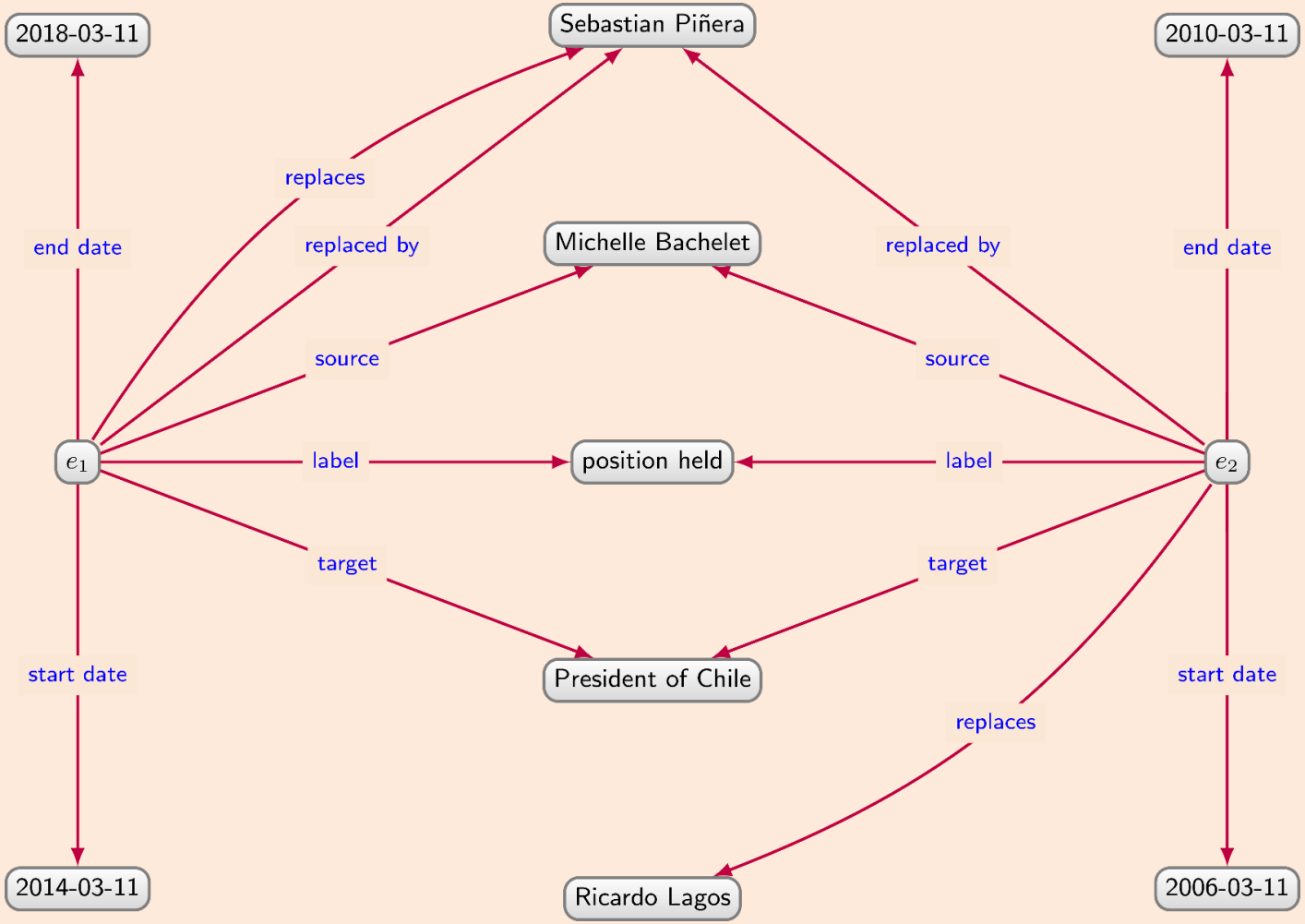
Wikidata statements

Michelle Bachelet [Q320]

position held [P39]	President of Chile [Q466956]	
	start date [P580]	2014-03-11
	end date [P582]	2018-03-11
	replaces [P155]	Sebastián Piñera [Q306]
	replaced by [P156]	Sebastián Piñera [Q306]
position held [P39]	President of Chile [Q466956]	
	start date [P580]	2006-03-11
	end date [P582]	2010-03-11
	replaces [P155]	Ricardo Lagos [Q331]
	replaced by P156	Sebastián Piñera Q306

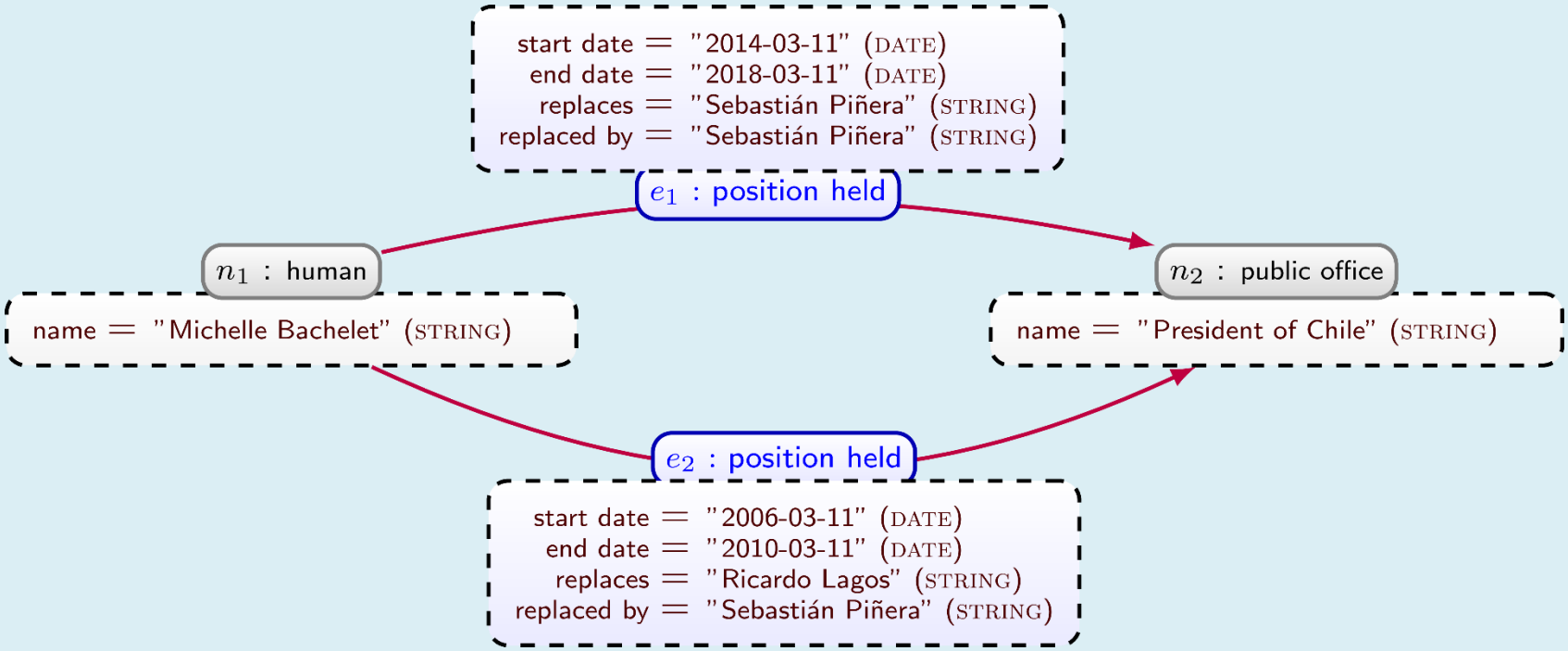
¿Puedo representarlo en RDF?

RDF



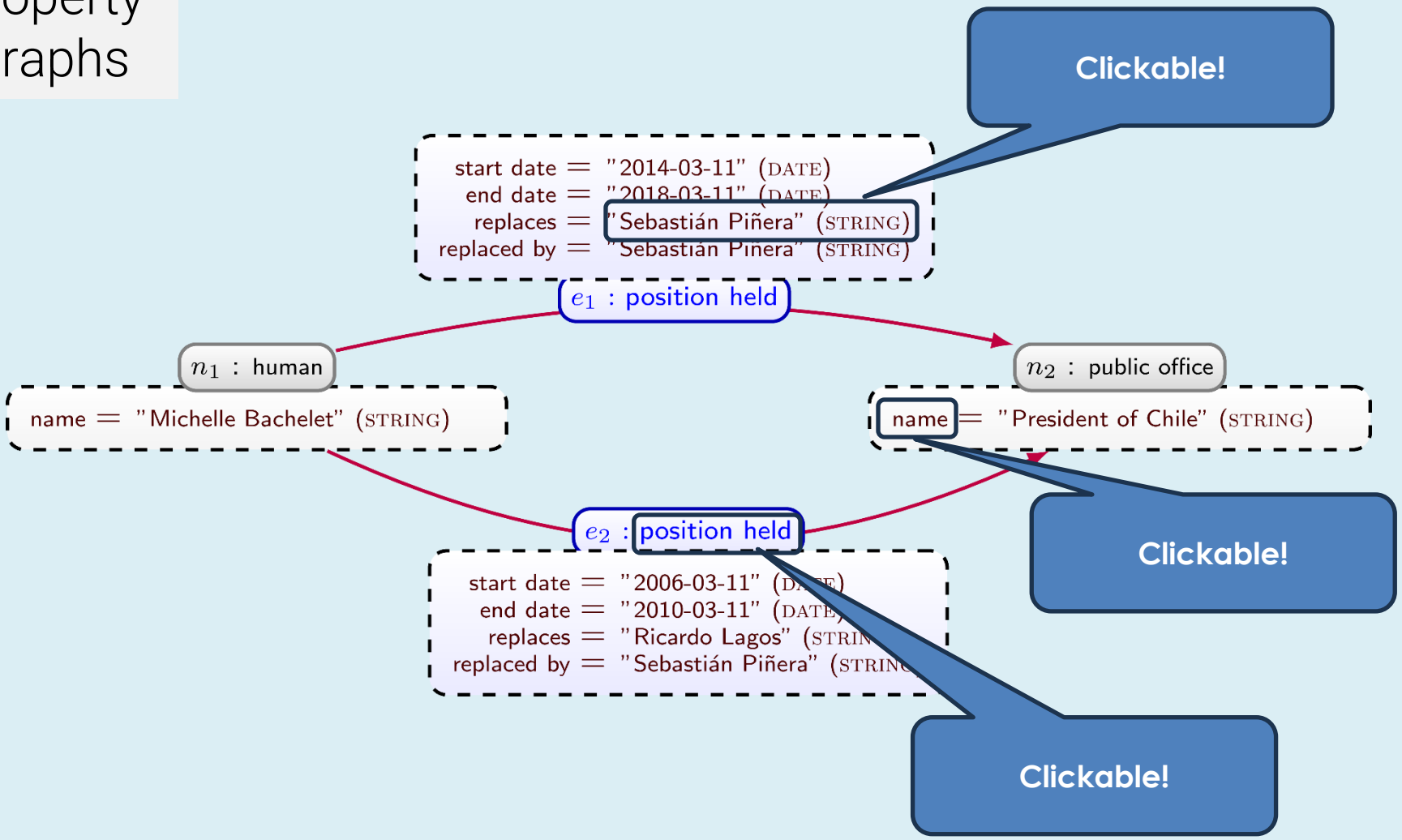
¿Y en Property Graph?

Property Graphs



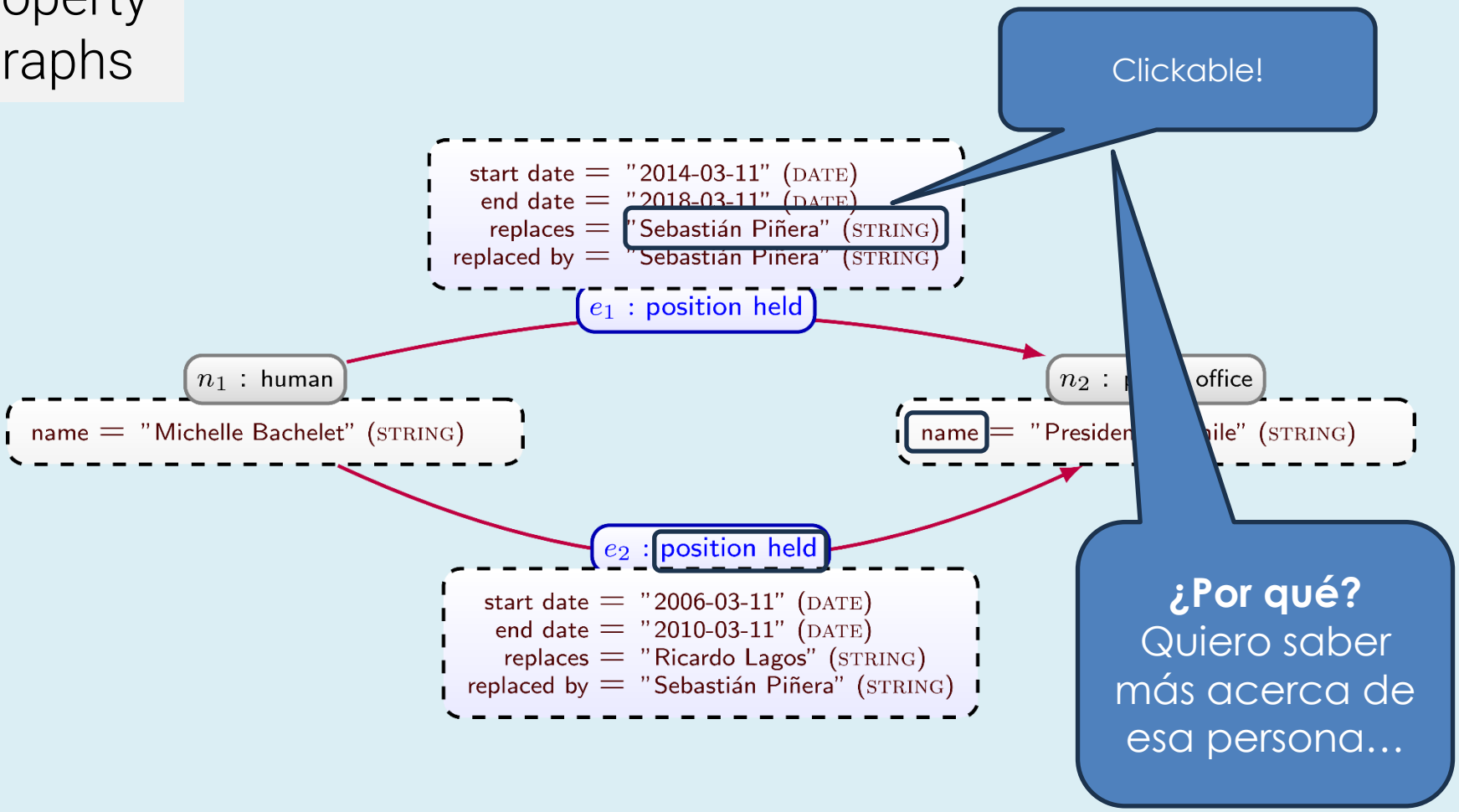
¿Será suficiente?

Property Graphs



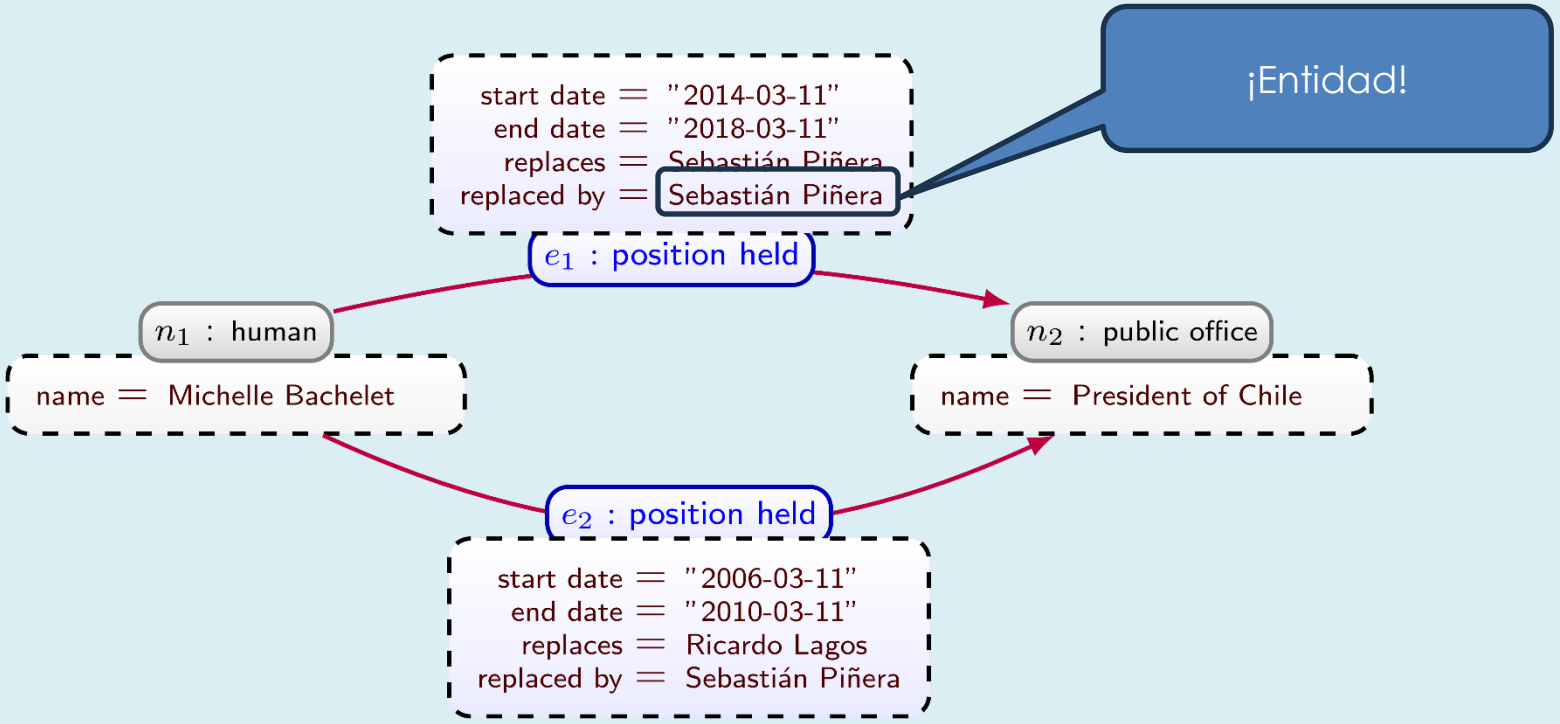
¿Será suficiente?

Property Graphs



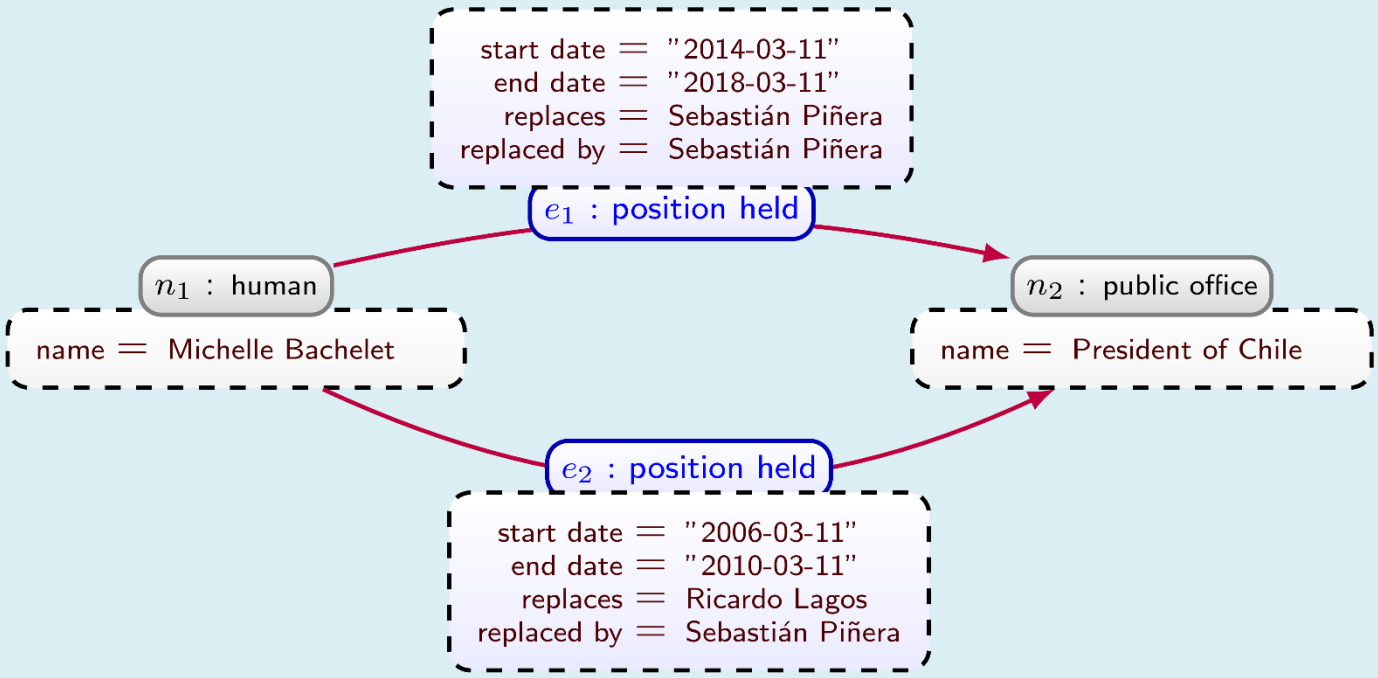
Solución: Grafos de Dominio

"In a nutshell": Todo es clickeable



Solución: Grafos de Dominio

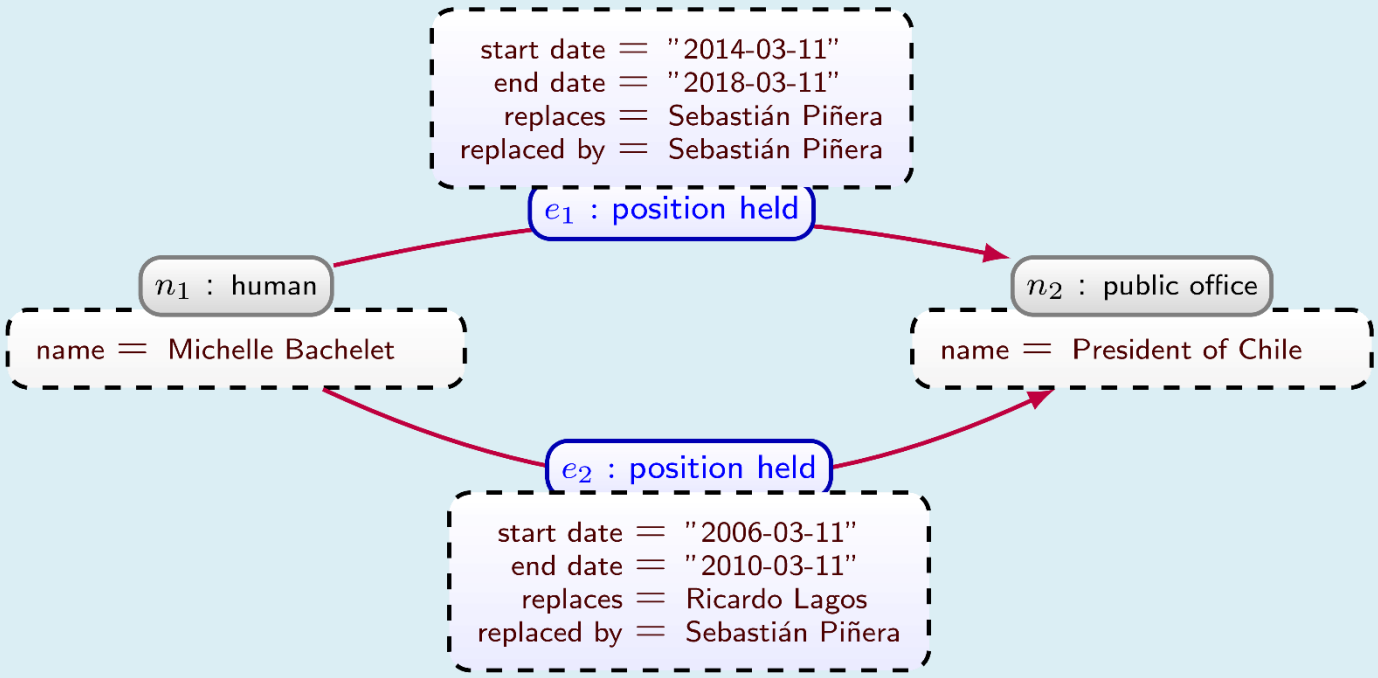
"In a nutshell": Todo es clickeable



DomainGraph(source, target, <u>eid</u>)
Labels(object, label)
Properties(<u>object, property</u>, value)

Solución: Grafos de Dominio (en realidad)

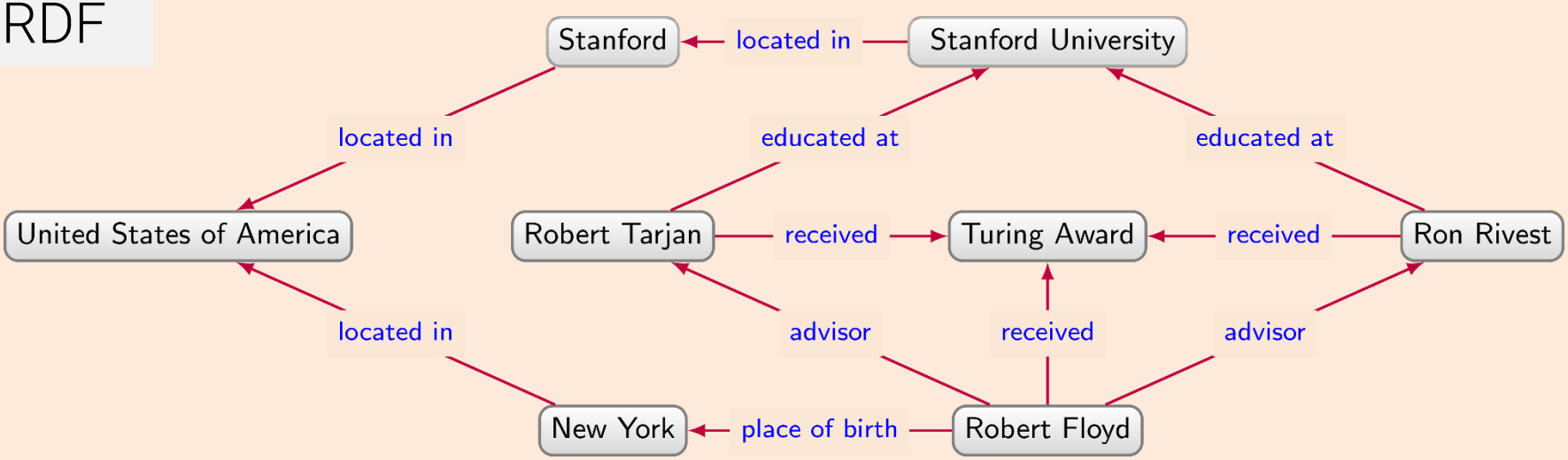
"In a nutshell": Todo es clickeable



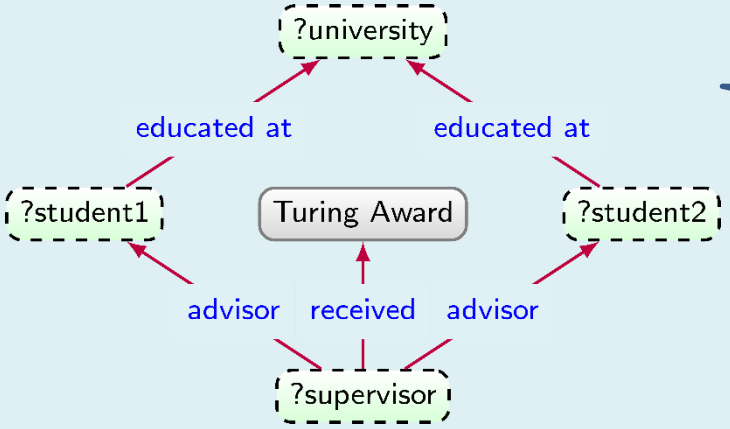
DomainGraph(source, type, target, <u>eid</u>)
Labels(object, label)
Properties(<u>object, property</u>, value)

Basic graph patterns

RDF



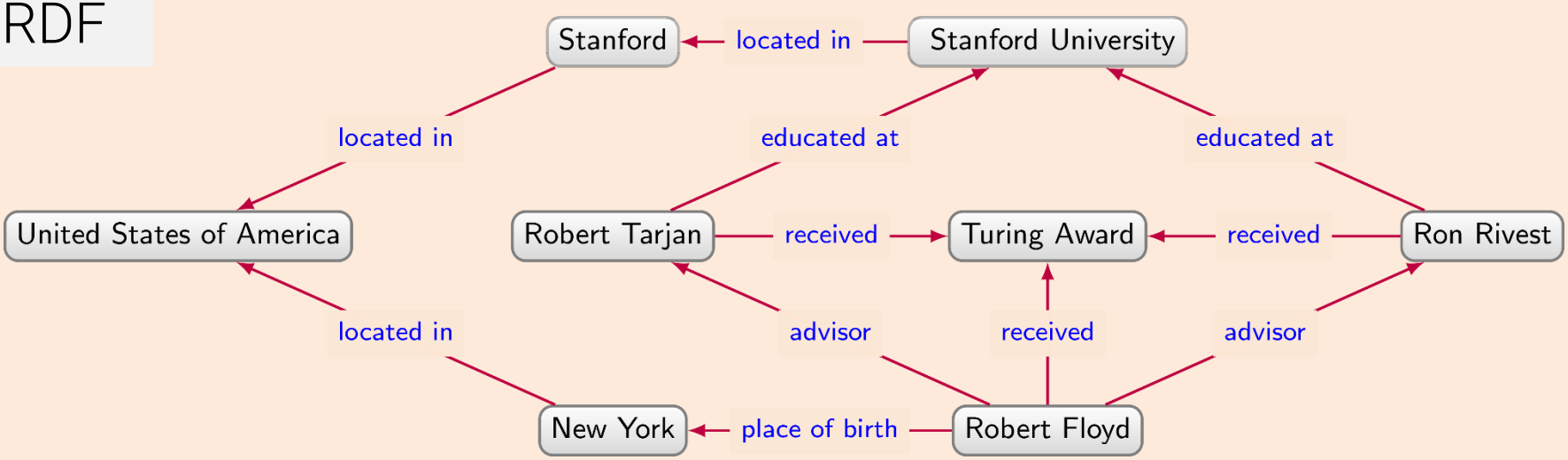
Académicos cuyo supervisor ganó el Turing Award



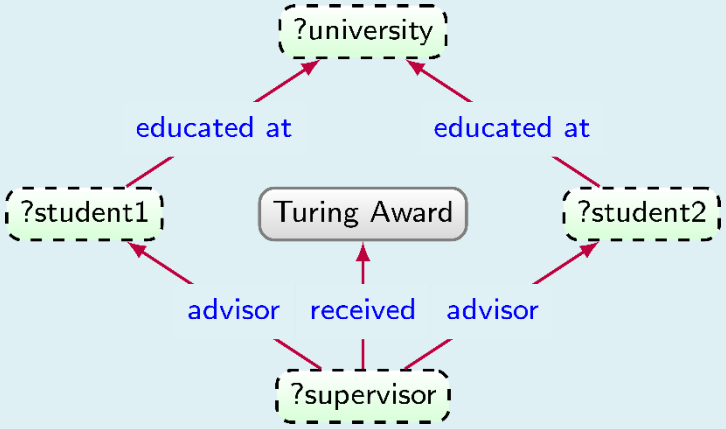
Idea:
Buscar esto en el grafo
(preservando constantes)

Basic graph patterns

RDF



Académicos cuyo supervisor ganó el Turing Award

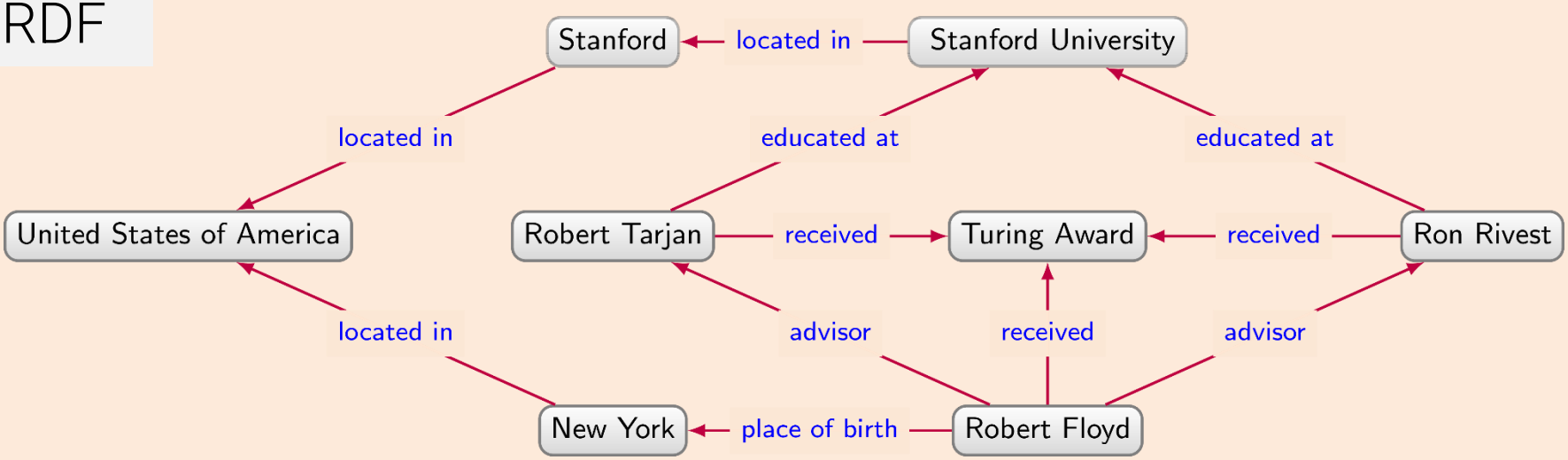


Semántica: Homomorfismo

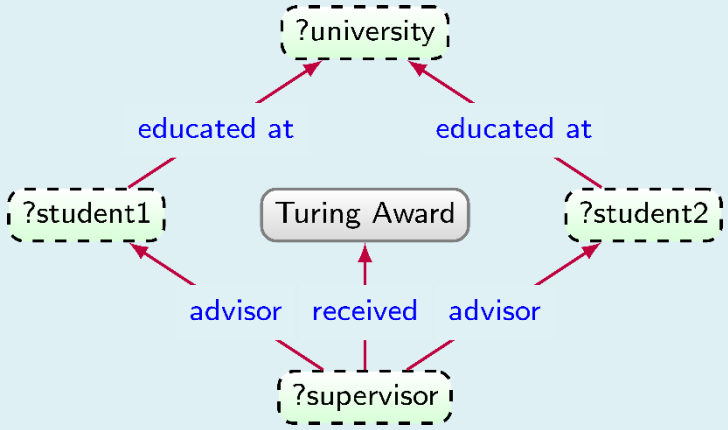
?supervisor	?student1	?student2	?univeristy
Robert Floyd	Robert Tarjan	Ron Rivest	Stanford Univeritsy
Robert Floyd	Ron Rivest	Robert Tarjan	Stanford Univeritsy
Robert Floyd	Robert Tarjan	Robert Tarjan	Stanford Univeritsy
Robert Floyd	Ron Rivest	Ron Rivest	Stanford Univeritsy

Basic graph patterns

RDF



Académicos cuyo supervisor ganó el Turing Award

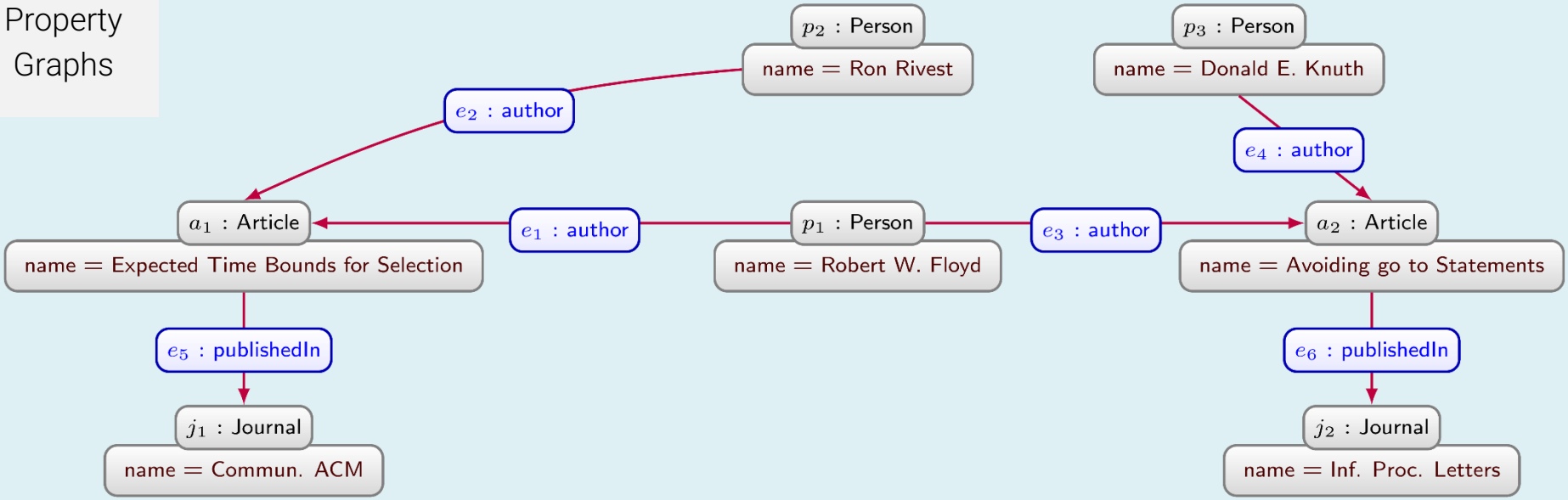


Semántica: Isomorfismo

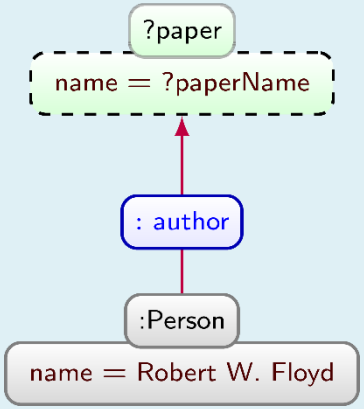
?supervisor	?student1	?student2	?univeristy
Robert Floyd	Robert Tarjan	Ron Rivest	Stanford Univeritsy
Robert Floyd	Ron Rivest	Robert Tarjan	Stanford Univeritsy
Robert Floyd	Robert Tarjan	Robert Tarjan	Stanford Univeritsy
Robert Floyd	Ron Rivest	Ron Rivest	Stanford Univeritsy

Basic graph patterns

Property Graphs



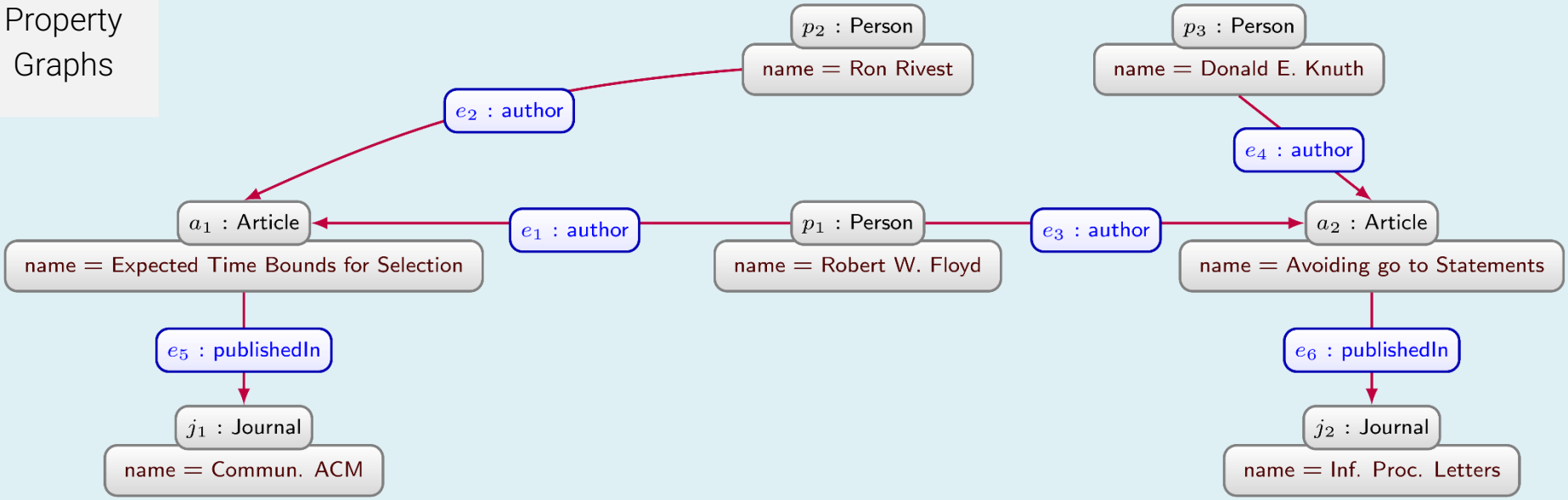
Papers escritos por Robert Floyd



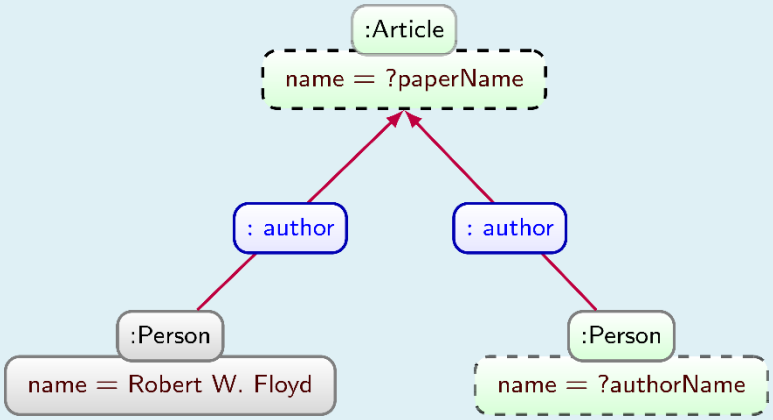
?paperName	?paper
Expected Time Bounds for Selection	a_1
Note on Avoiding go to Statements	a_2

Basic graph patterns

Property Graphs



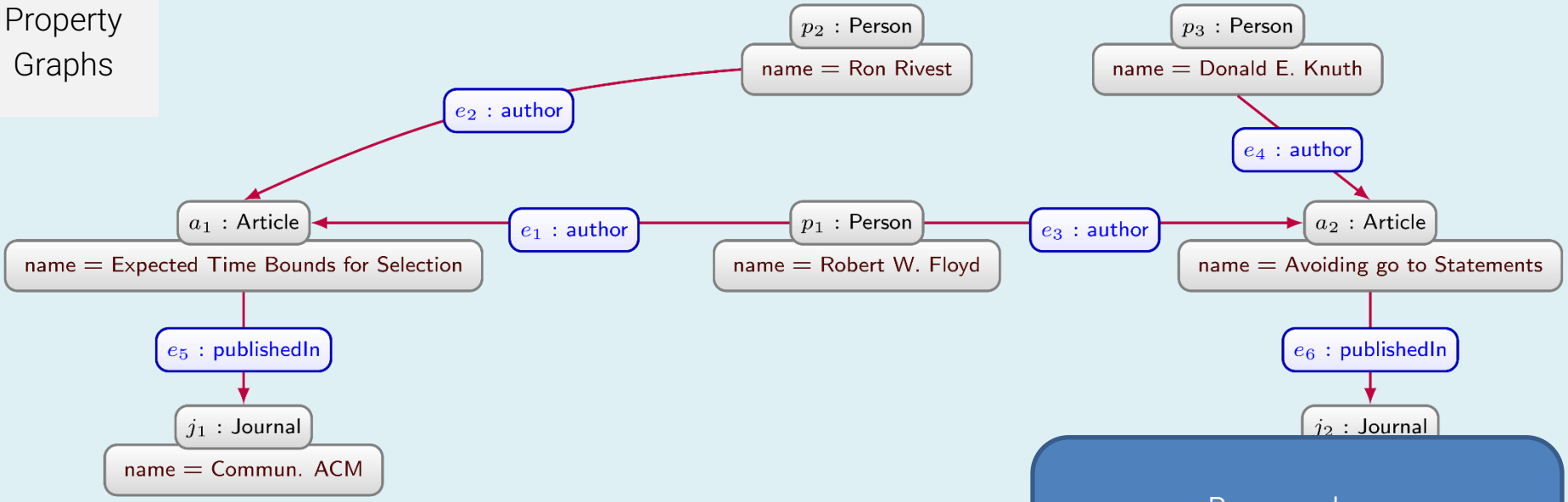
Co-autores de Robert Floyd



?authorName	?paperName
Ron Rivest	Expected Time Bounds for Selection
Donald E. Knuth	Note on Avoiding go to Statements
Robert W. Floyd	Expected Time Bounds for Selection
Robert W. Floyd	Note on Avoiding go to Statements

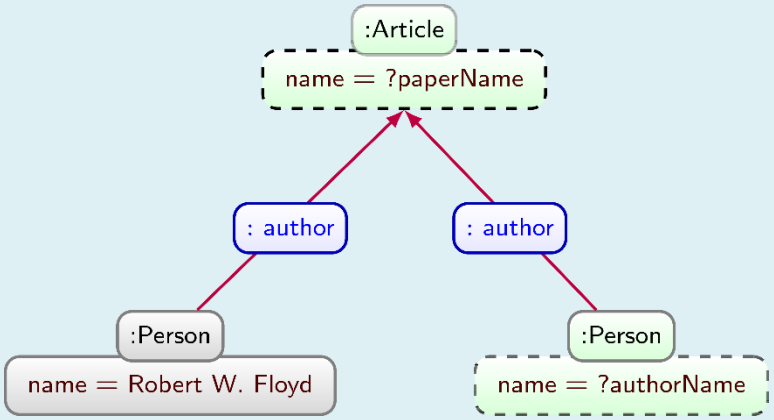
Basic graph patterns

Property Graphs



Recuerden:
Buscar el patron en el grafo,
¡nada más!

Co-autores de Robert Fl



?authorName	?[illegible]ame
Ron Rivest	[illegible]pected Time Bounds for Selection
Donald E. Knuth	Note on Avoiding go to Statements
Robert W. Floyd	Expected Time Bounds for Selection
Robert W. Floyd	Note on Avoiding go to Statements

¡Probemos MillenniumDB!

https://telarkg.imfd.cl/

https://bibkg.imfd.cl/

¿Qué hay disponible?

- Servidor de Base de Datos
- Interfaz Web (Alpha)
- Driver para Python (Alpha)
- Driver para JavaScript (Alpha)

MillenniumDB: Descarga y más información

Repositorio en GitHub:

https://github.com/MillenniumDB

¡Escríbanme!

vecalisto@uc.cl

¡Muchas gracias!