

CC3201-1

BASES DE DATOS

OTOÑO 2019

Clase 14: Conclusión

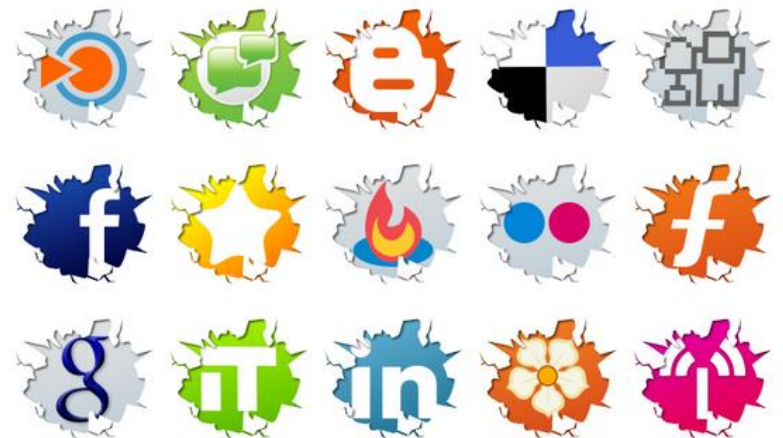
Aidan Hogan

[aidhog@gmail.com](mailto:aidhog@gmail.com)

¿ES IMPORTANTE EL CURSO?

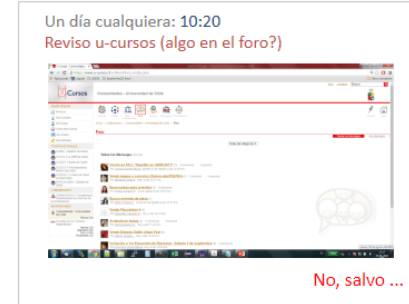
Interactuamos con bases de datos  
todo el tiempo, todos los días

- Especialmente con la Web:
  - Búsqueda (Google, Bing, Yahoo!, ...)
  - Tiendas (Amazon, eBay, ...)
  - Redes sociales (Facebook, Twitter, ...)
  - Enciclopedias (Wikipedia, IMDb, ...)
  - Bancos
  - Aerolíneas
  - U-cursos
  - ...



# Cómo podemos implementar consultas, actualizaciones, seguridad, etc., sobre estos datos

profesores.csv		
RUT	Nombre	Curso
...		
24.482.054-9	Aidan Hogan	CC3201
24.482.054-9	Aidan Hogan	CC5212
24.482.054-9	Aidan Hogan	CC6202
...		



alumnos.csv		
RUT	Nombre	Curso
...		
10.323.634-4	Pia García	CC3201
12.323.792-8	Juan Ramírez	CC6202
12.323.792-8	Juan Ramírez	CC5212
...		

auxiliares.csv		
RUT	Nombre	Curso
...		
12.412.412-4	Sebastián Ferrada	CC3201
12.412.412-4	Sebastián Ferrada	CC5208
13.123.024-9	Daniel Hernández	CC6202
...		

CURSOS.CSV	
Codigo	Nombre
...	
CC3201	Bases de Datos
CC5212	Proc. Masivo de Datos
...	

notas.csv			
RUT	Nombre	Eval	Nota
...			
10.323.634-4	CC3201	Lab1	6.0
10.323.634-4	CC3201	Lab2	4.5
12.323.792-8	CC6202	Examen	3.9
...			

# Estos son problemas generales que se encuentran en muchas aplicaciones

Un día cualquiera: 09:15  
Despierto



(Bostezo.)

Un día cualquiera: 10:15  
Me meto al banco (me pagaron?)



Sí. Me pagaron.

Un día cualquiera: 10:35  
Amazon (Mr. Robot ... ¿cuánto cuesta?)

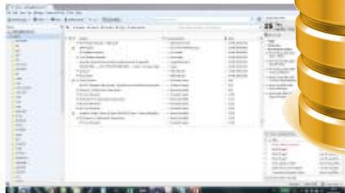


Demasiado.

Un día cualquiera: 10:55  
Al supermercado (esperando en la fila)

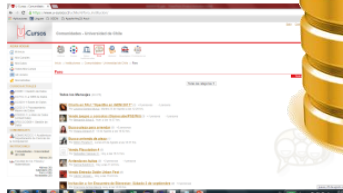


Un día cualquiera: 09:35  
Reviso el correo



Nada urgente, ¡uf!

Un día cualquiera: 10:20  
Reviso u-cursos (algo en el foro?)



No, salvo ...

Un día cualquiera: 10:36  
ThePirateBay (Mr. Robot ... cuesta dema)



Listo. Pero tengo hambre ...

Un día cualquiera: 10:57  
Al supermercado (uso mi tarjeta de fide)



Mucha gente.

Un día cualquiera: 09:50  
Café: pago con tarjeta



Mmm. Café.

Un día cualquiera: 10:30  
IMDb (Mr. Robot ... ¿es bueno?)



Sí.

Un día cualquiera: 10:52  
Al supermercado (¿cuánto cuesta?)



Barrato.

Un día cualquiera: 11:00  
Desayuno



HAN APRENDIDO SOBRE ...

# El Modelo Relacional

**Cervezas**(nombre:string, tipo:string, grados:float, ciudad-origen:string)  
**Vinos**(nombre:string, tipo:string, año:int, grados:float, ciudad-origen:string)  
**En-Stock**(nombre:string, cantidad:int, precio-unitario:int)

## Cervezas

nombre	tipo	grados	ciudad-origen
Austral Lager	Lager	4,6	Punta Arenas
Austral Yagan	Ale	5,0	Punta Arenas
Austral Pale Ale	Ale	5,0	Punta Arenas
Kuntsmann Torobayo	Ale	5,0	Valdivia
Kross 5	Ale	7,2	Curacaví
Kross Golden	Ale	5,3	Curacaví
Kross Pilsner	Pilsner	4,9	Curacaví

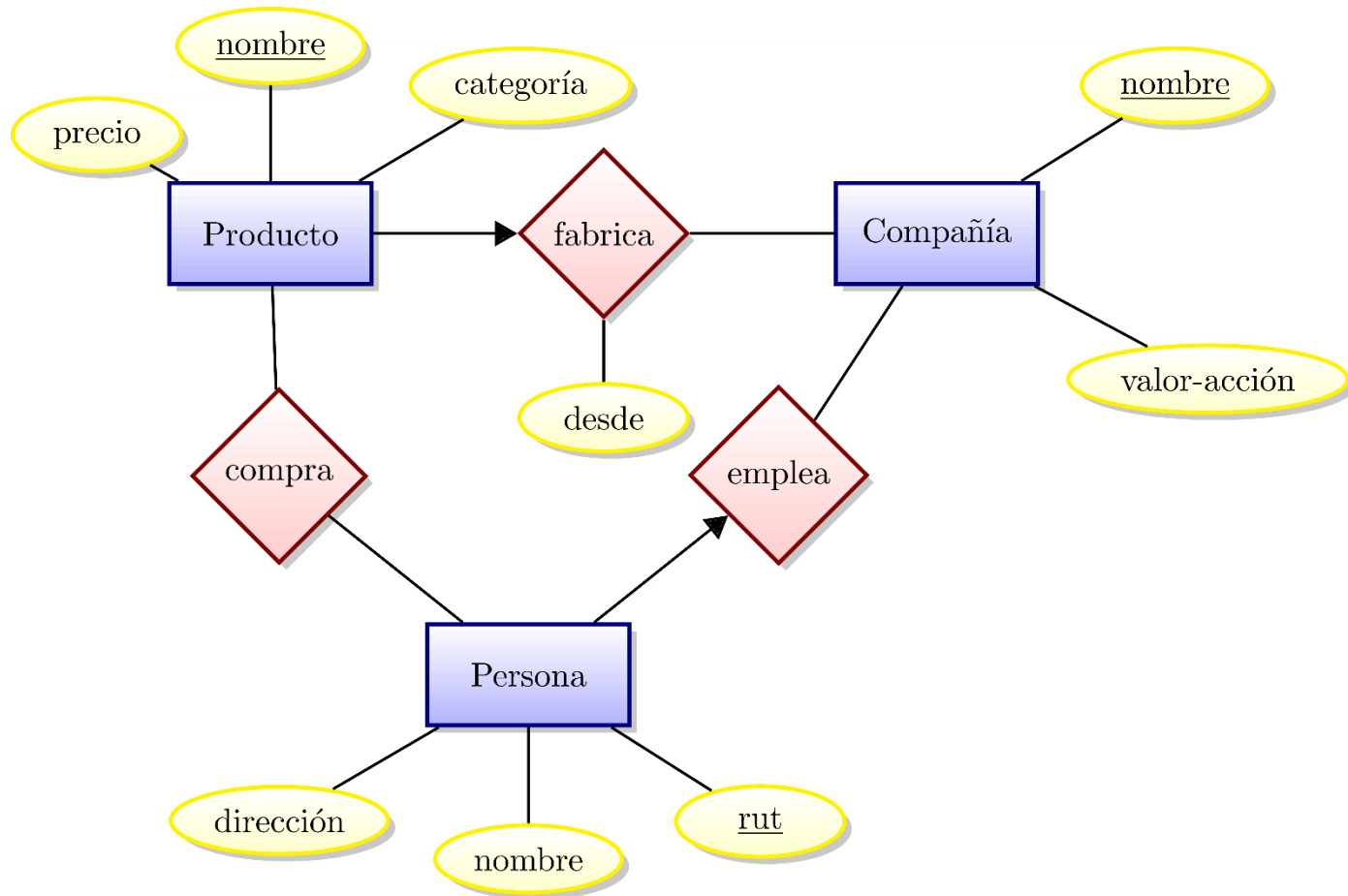
## Vino

nombre	tipo	año	grados	ciudad-origen
Tarapacá	Carménère	2014	13,5	Maipo
Tarapacá	Merlot	2014	13,5	Maipo
Gato	Merlot	2016	14,0	Maule

## En-Stock

nombre	cantidad	precio-unitario
--------	----------	-----------------

# Diagramas E-R:





# El Álgebra Relacional

$$\pi_{A_1, \dots, A_n}(\mathbf{R}) \quad \sigma_{\text{condición}}(\mathbf{R}) \quad \rho_{A_i/A_j}(\mathbf{R})$$

$$\mathbf{R}_1 \cup \mathbf{R}_2$$

$$\mathbf{R}_1 \times \mathbf{R}_2$$

$$\mathbf{R}_1 - \mathbf{R}_2$$

$$\mathbf{R}_1 \cap \mathbf{R}_2$$

$$\mathbf{R}_1 \bowtie_{\text{condición}} \mathbf{R}_2$$

# El Cálculo Relacional

- Fórmulas atómicas:

(sea  $c$  una constante,  $OP \in \{<, >, =, \leq, \geq, \neq\}$ )

$R$        $R.a \text{ OP } R'.a'$        $R.a \text{ OP } c$        $c \text{ OP } R.a$

- Una fórmula puede ser

- Una fórmula atómica o

- Sean (recursivamente)  $p$  y  $q$  formulas:

$\neg p, p \wedge q, p \vee q, p \Rightarrow q, \exists R(p), \forall R(p)$

# SQL: Consultas Básicas

Planeta							
nombre	dist	radio	grav	días	años	temp	anillo
Mercurio	0,39	0,38	2,8	58,646	0,241	440	false
Venus	0,72	0,95	8,9	-243,019	0,615	730	false
Tierra	1,00	1,00	9,8	0,997	1,000	288	false
Marte	1,52	0,53	3,7	1,026	1,880	186	false
Júpiter	5,20	10,97	22,9	0,414	11,862	152	true
Saturno	9,54	9,14	9,1	0,444	29,447	134	true
Urano	19,19	3,98	7,8	-0,719	84,017	76	true
Neptuno	30,07	3,86	11,0	0,671	164,791	53	true

Aterrizaje			
nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT nombre, año, nave
FROM Planeta, Aterrizaje
WHERE nombre = planeta
AND dist > 1.00
AND año >= 2000
```

nombre	año	nave
Marte	2003	Beagle 2
Júpiter	2003	Galileo

# SQL: Agregación

## Aterrizaje

<u>nave</u>	<u>planeta</u>	<u>país</u>	<u>año</u>
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT planeta, COUNT(*) AS conteo
FROM Aterrizaje
GROUP BY planeta
HAVING MAX(año)<2000
```

<u>planeta</u>	<u>conteo</u>
Venus	2

# SQL: Consultas Anidadas

## Planeta

nombre	dist	radio	grav	días	años	temp	anillo
Mercurio	0,39	0,38	2,8	58,646	0,241	440	false
Venus	0,72	0,95	8,9	-243,019	0,615	730	false
Tierra	1,00	1,00	9,8	0,997	1,000	288	false
Marte	1,52	0,53	3,7	1,026	1,880	186	false
Júpiter	5,20	10,97	22,9	0,414	11,862	152	true
Saturno	9,54	9,14	9,1	0,444	29,447	134	true
Urano	19,19	3,98	7,8	-0,719	84,017	76	true
Neptuno	30,07	3,86	11,0	0,671	164,791	53	true

## Aterrizaje

nave	planeta	país	año
Messenger	Mercurio	EEUU	2015
Venera 3	Venus	URRS	1966
Pioneer	Venus	EEUU	1978
Mars 2 lander	Marte	URRS	1971
Viking 1	Marte	EEUU	1976
Beagle 2	Marte	ESA	2003
Galileo	Júpiter	EEUU	2003

```
SELECT nave, planeta
FROM Aterrizaje
WHERE planeta NOT IN
( SELECT nombre
  FROM Planeta
  WHERE grav > 9.8 OR planeta IN
    ( SELECT planeta
      FROM Aterrizaje
      WHERE país = 'ESA'
    )
)
AND año > 2000
```

nave	planeta
Messenger	Mercurio



# Restricciones de integridad

Cuenta				
<u>número</u>	rut	tipo	saldo_clp	saldo_usd
7873698669	32.000.273-K	Estacional	225000	344,94

Banco de Chile

```
CREATE TABLE Cuenta (  
  número BIGINT PRIMARY KEY,  
  rut VARCHAR (12) NOT NULL,  
  tipo VARCHAR (12) NOT NULL,  
  saldo_clp BIGINT NOT NULL,  
  saldo_usd DOUBLE PRECISION NOT NULL,  
  CHECK ( ROUND ( (saldo_clp/saldo_usd)::NUMERIC - 652.275 , 1 ) = 0 )  
)
```

```
INSERT INTO Cuenta VALUES  
(7273697679, '28.923.123-7',  
  'Estacional', 100, 0.99)
```



# Formas Normales

## TransferenciaTotal

<u>cuenta_origen</u>	<u>cuenta_destina</u>	<u>total</u>
7873698669	1849123812	51920
8273697679	7873698669	529000
8273697679	1849123812	41920

## Destino

<u>cuenta</u>	<u>rut</u>
1849123812	12.491.671-K
7873698669	32.000.273-K



BCNF: Satisface 1NF y ...

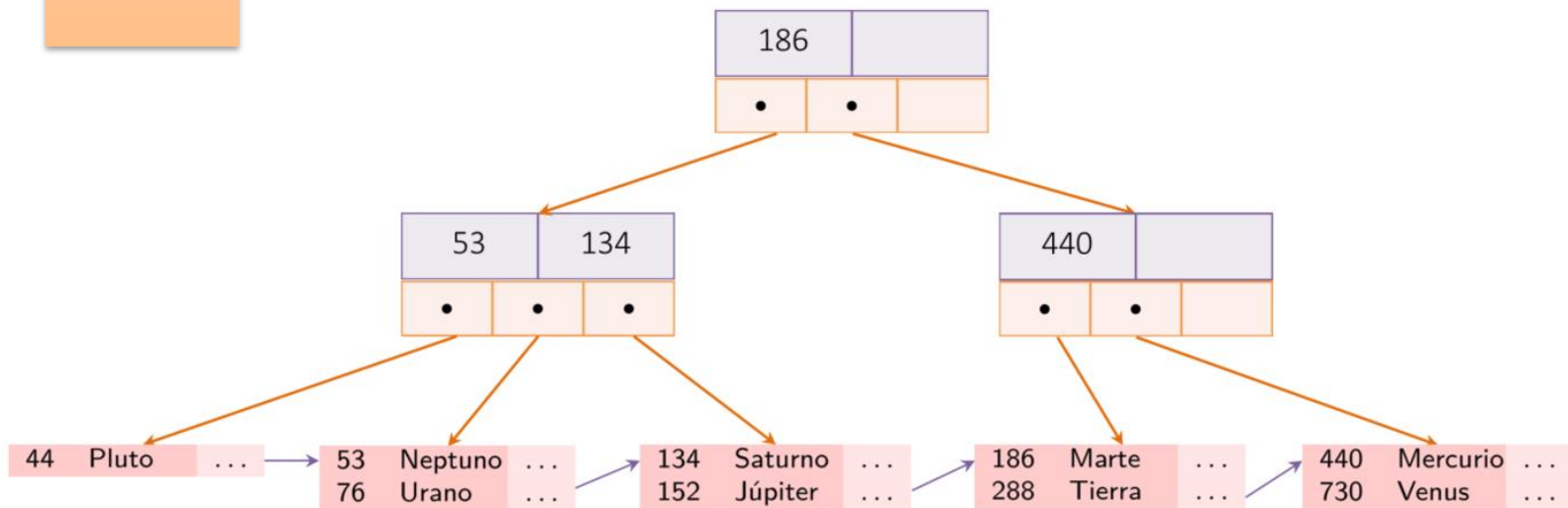
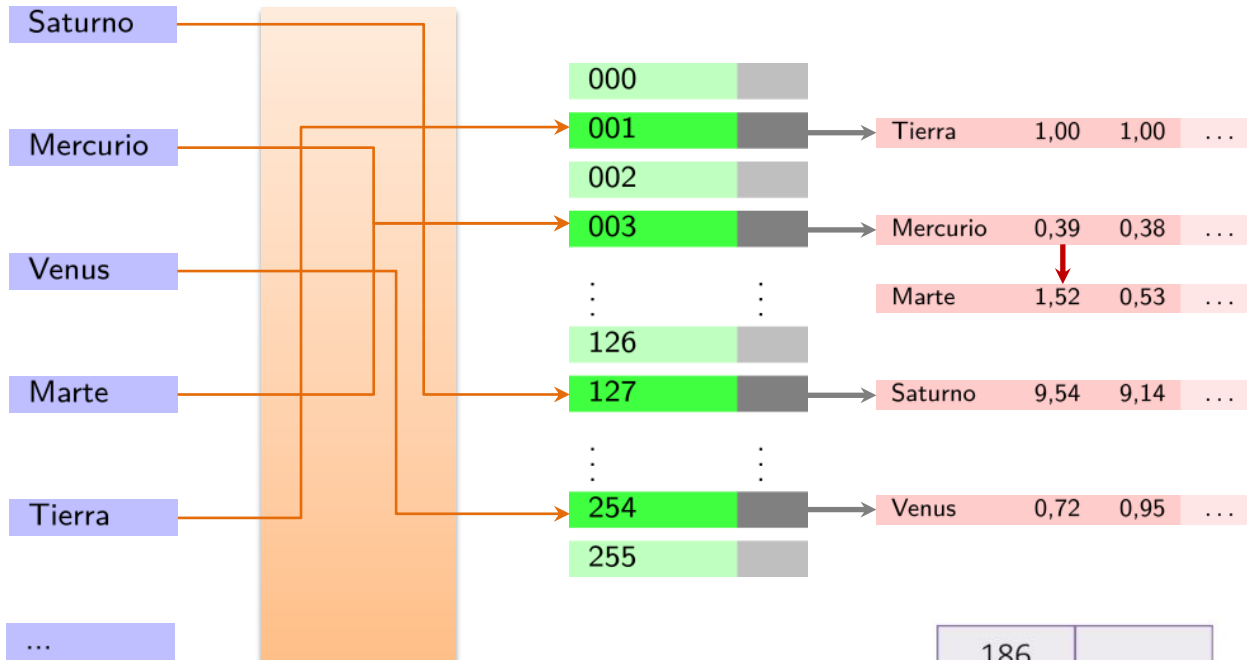
Para cada:

$$X \rightarrow Y$$

$X$  es una súper llave

$$\text{o } Y \subseteq X$$

# Indexación





# Implementación de Joins

$R \bowtie S$

- Para cada tupla  $r \in R$ 
  - Para cada tupla  $s \in S$ 
    - Si  $r$  y  $s$  satisfacen el join:  
escribir  $\{r\} \times \{s\}$

$R \bowtie S$

- Para cada tupla  $r \in R$ 
  - Buscar  $s \in S$  en el índice tal que  $r$  y  $s$  satisfagan el join:  
escribir  $\{r\} \times \{s\}$

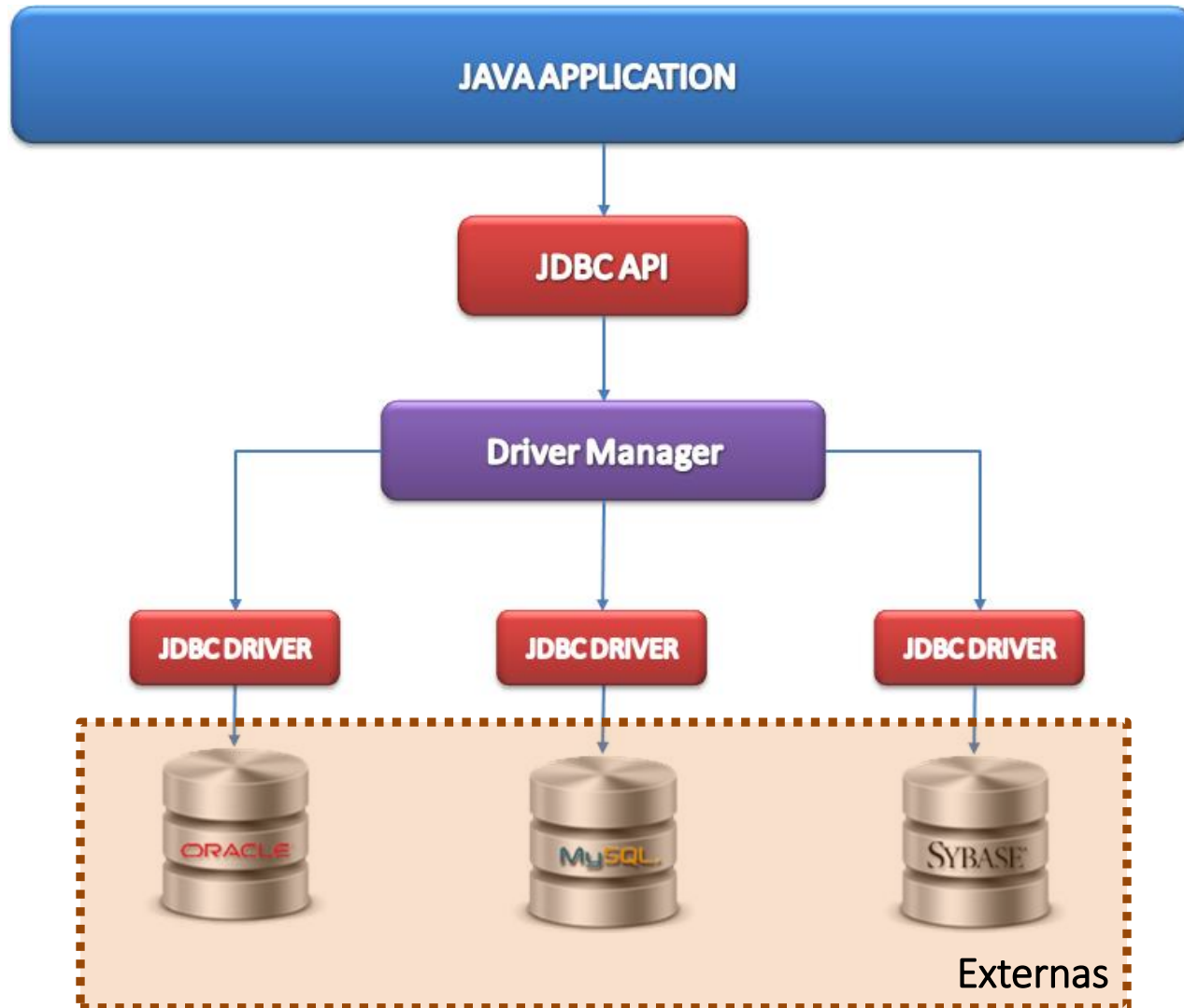
$R \bowtie S$

- Guardar  $R$  en memoria principal
- Para cada tupla  $s \in S$ 
  - Buscar  $r$  en memoria principal tal que  $r$  y  $s$  satisfagan el join:  
escribir  $\{r\} \times \{s\}$

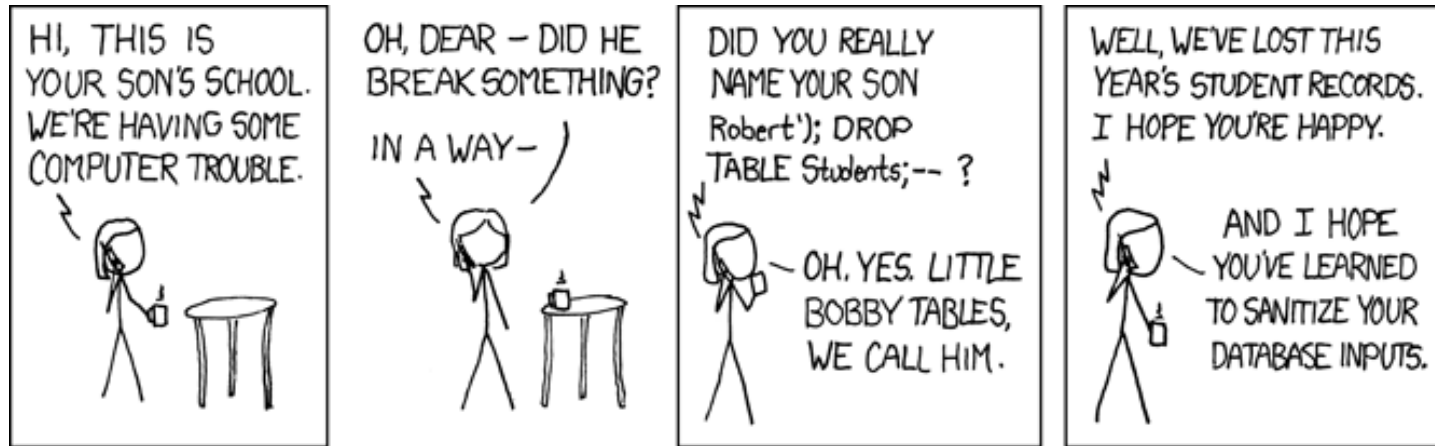
$R \bowtie S$

- Ordenar  $R$  y  $S$  por los atributos del join
- Aplicar un merge-sort y para cada tupla  $r$  y  $s$  que satisfagan el join:  
escribir  $\{r\} \times \{s\}$

# Acceso Programático



# Inyección SQL



```
"SELECT nota FROM Students WHERE name='"+input+'"
```

```
"SELECT nota FROM Students WHERE name='Robert' OR 1=1);'"
```

# Vistas

Artista		
nombre	país	retirado
Leonard Cohen	Canadá	false
Lady Gaga	EE.UU.	false
David Bowie	G.B.	true
Justin Bieber	Canadá	false
...	...	...

Álbum		
nombre	artista	año
Old Ideas	Leonard Cohen	2012
Dear Heather	Leonard Cohen	2004
You Want It Darker	Leonard Cohen	2016
Popular Problems	Leonard Cohen	2014
ARTPOP	Lady Gaga	2013
...	...	...

Evaluación					
álbum	artista	fuelle	eval		
Popular Problems	Leonard Cohen	The Guardian	80		
Popular Problems	Leonard Cohen	The Observer	80		
Popular Problems	Leonard Cohen	Uncut	90		
You Want It Darker	Leonard Cohen	The Observer	100		
You Want It Darker	Leonard Cohen	Uncut	90		
You Want It Darker	Leonard Cohen	Rolling Stone	80		
You Want It Darker	Leonard Cohen	The Guardian	100		
You Want It Darker	Leonard Cohen	Mojo	100		
Dear Heather	Leonard Cohen	The Guardian	60		
Dear Heather	Leonard Cohen	Uncut	100		
Dear Heather	Leonard Cohen	Rolling Stone	70		
Old Ideas	Leonard Cohen	Rolling Stone	90		
Old Ideas	Leonard Cohen	Uncut	80		
ARTPOP	Lady Gaga	Rolling Stone	60		
...	...	...	...		

## ÁlbumEval

álbum	artista	pm	num
Old Ideas	Leonard Cohen	85	2
Dear Heather	Leonard Cohen	76	3
You Want It Darker	Leonard Cohen	94	5
Popular Problems	Leonard Cohen	83	3
ARTPOP	Lady Gaga	64	3
...	...	...	...

```
CREATE VIEW ÁlbumEval AS
SELECT álbum, artista,
       FLOOR(AVG(eval)) AS pm,
       COUNT(eval) AS num
FROM Evaluación
GROUP BY álbum, artista
```

```
SELECT pm
FROM
  ( SELECT álbum, artista,
    FLOOR(AVG(eval)) AS pm,
    COUNT(eval) AS num
  FROM Evaluación
  GROUP BY álbum, artista ) ÁlbumEval
WHERE álbum='You Want It Darker'
AND artista='Leonard Cohen'
```

pm  
94

(2) Ejecutar la consulta extendida sobre las tablas bases

```
SELECT pm
FROM ÁlbumEval
WHERE álbum='You Want It Darker'
AND artista='Leonard Cohen'
```

(1) Extender la consulta de conformidad con la vista

# Disparadores

ÁlbumEval			
álbum	artista	pm	num
Old Ideas	Leonard Cohen	85	2
Dear Heather	Leonard Cohen	76	3
You Want It Darker	Leonard Cohen	92	5
Popular Problems	Leonard Cohen	83	3
ARTPOP	Lady Gaga	64	3
...	...	...	...

## Evaluación

álbum	artista	fuelle	eval
Popular Problems	Leonard Cohen	The Guardian	80
Popular Problems	Leonard Cohen	The Observer	80
Popular Problems	Leonard Cohen	Uncut	90
You Want It Darker	Leonard Cohen	The Observer	100
You Want It Darker	Leonard Cohen	Uncut	90
You Want It Darker	Leonard Cohen	Rolling Stone	80
You Want It Darker	Leonard Cohen	The Guardian	100
Dear Heather	Leonard Cohen	The Guardian	60
Dear Heather	Leonard Cohen	Uncut	100
Dear Heather	Leonard Cohen	Rolling Stone	70
Old Ideas	Leonard Cohen	Rolling Stone	90
Old Ideas	Leonard Cohen	Uncut	80
ARTPOP	Lady Gaga	Rolling Stone	60
...	...	...	...

```
CREATE TRIGGER NoOfenderANadie
AFTER UPDATE OF pm ON ÁlbumEval
REFERENCING
  OLD ROW AS TuplaAntigua
  NEW ROW AS TuplaNueva
FOR EACH ROW
WHEN (TuplaAntigua.pm > TuplaNueva.pm)
SET TuplaNueva.pm = TuplaAntigua.pm
```

```
UPDATE ÁlbumEval
SET pm = 50
WHERE álbum = 'ARTPOP'
AND artista = 'Lady Gaga'
```

# Garantías de ACID

## ACID: Atomicidad

```
CREATE TABLE Balance (  
  cuenta INTEGER PRIMARY KEY,  
  total_gasto BIGINT,  
  total_ingreso BIGINT,  
  saldo BIGINT,  
  CHECK (total_ingreso - total_gasto = saldo)  
)
```

```
START TRANSACTION  
UPDATE Balance SET saldo=saldo-10 WHERE Cuenta=7873698669 ;  
UPDATE Balance SET total_gasto=total_gasto+10 WHERE Cuenta=7873698669 ;  
COMMIT;
```

No se puede actualizar el saldo sin actualizar el gasto directamente después.  
(Si alguna actualización falla, ambas fallan.)

## ACID: Consistencia

```
CREATE TABLE Balance (  
  cuenta INTEGER PRIMARY KEY,  
  total_gasto BIGINT,  
  total_ingreso BIGINT,  
  saldo BIGINT,  
  CHECK (total_ingreso - total_gasto = saldo)  
)
```

```
START TRANSACTION  
UPDATE Balance SET saldo=saldo-100 WHERE Cuenta=7873698669 ;  
UPDATE Balance SET total_gasto=total_gasto+10 WHERE Cuenta=7873698669 ;  
COMMIT;
```



Si el resultado de la transacción no satisface todas las restricciones, fallará.

## ACID: Aislamiento (Isolation)

```
CREATE TABLE Balance (  
  cuenta INTEGER PRIMARY KEY,  
  total_gasto BIGINT,  
  total_ingreso BIGINT,  
  saldo BIGINT,  
  CHECK (total_ingreso - total_gasto = saldo)  
)
```

```
START TRANSACTION T1  
UPDATE Balance  
SET saldo=saldo-10  
WHERE Cuenta=7873698669 ; (1)  
UPDATE Balance  
SET total_gasto=total_gasto+100  
WHERE Cuenta=7873698669 ; (3) ✘  
COMMIT; (4) ROLLBACK;
```

```
START TRANSACTION T2  
UPDATE Balance  
SET saldo=saldo+100  
WHERE Cuenta=7873698669 ; (2)  
UPDATE Balance  
SET total_ingreso=total_ingreso+100  
WHERE Cuenta=7873698669 ; (5)  
COMMIT; (6)
```

Una transacción no puede interferir con otra transacción.  
En (4), hay que tener cuidado con el ROLLBACK: no se puede restaurar el valor de saldo antes del paso (1) porque el valor ya fue cambiado por (2).

## ACID: Durabilidad

```
CREATE TABLE Balance (  
  cuenta INTEGER PRIMARY KEY,  
  total_gasto BIGINT,  
  total_ingreso BIGINT,  
  saldo BIGINT,  
  CHECK (total_ingreso - total_gasto = saldo)  
)
```

```
START TRANSACTION  
UPDATE Balance SET saldo=saldo-10 WHERE Cuenta=7873698669 ;  
UPDATE Balance SET total_gasto=total_gasto+10 WHERE Cuenta=7873698669 ;  
COMMIT;
```



Una vez que haya un COMMIT exitoso, se persisten los cambios.  
(Normalmente la persistencia aquí significa en el disco duro. Sin persistencia, en el caso de que la máquina falla y toda la evidencia de los cambios está en memoria principal, el sistema de base de datos olvidará los cambios silenciosamente.)

# Transacciones

## Gasto

<u>cuenta</u>	<u>comentario</u>	<u>fecha</u>	<u>hora</u>	<u>monto</u>	<u>saldo</u>	<u>id</u>
7873698669	Electricidad	2020-02-02	20:00:01	8200	291800	TRCJASJDA9A
7873698669	Calefacción	2020-02-02	20:00:02	600	291200	TRC81KAQWAS
7873698669	Moviestar	2020-02-02	20:00:03	16200	275000	TRCK8J7JA8D
7873698669	Cajero	2020-02-08	16:05:02	100000	225000	TRCPM8A45AD

## Cuenta

<u>número</u>	<u>rut</u>	<u>tipo</u>	<u>saldo_clp</u>	<u>saldo_usd</u>
7873698669	32.000.273-K	Estacional	225000	344,94

## Divisa

<u>d1</u>	<u>d2</u>	<u>valor</u>
CLP	USD	0,0001533
USD	CLP	652,2750000

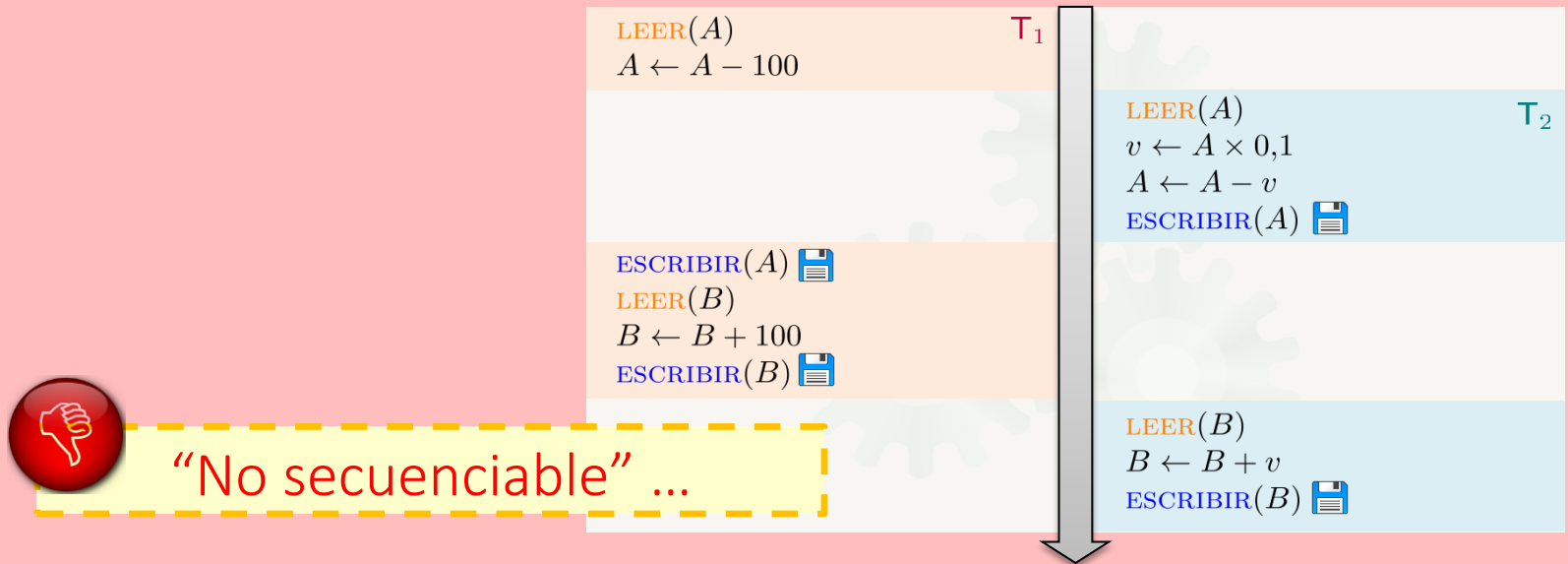
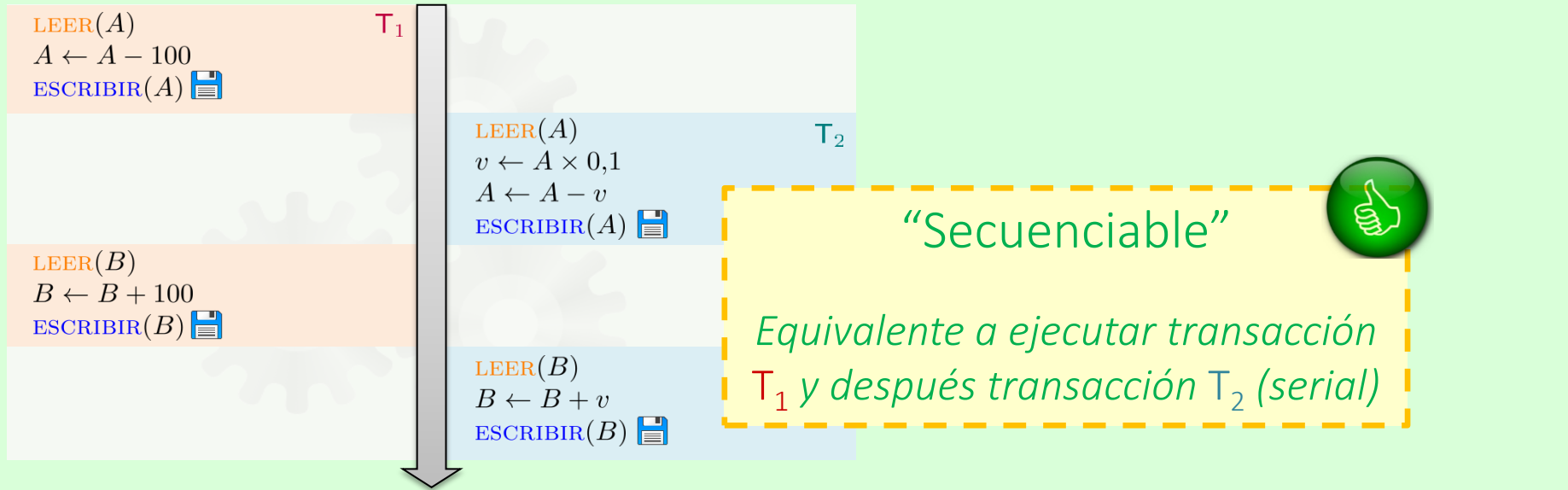
## Ciente

<u>rut</u>	<u>nombre</u>	<u>fono</u>	<u>dirección</u>
32.000.273-K	Kelvin	+56976698463	Campo de Hielo Sur, Depto 273

```
START TRANSACTION;
INSERT INTO Gasto VALUES
(7873698669, 'Noruega', '2020-02-12', '02:14:20', 400000, -175000, 'TRCLK9K24KS');
UPDATE Cuenta SET saldo_clp=saldo_actual
FROM ( SELECT (saldo_clp - 400000) AS saldo_actual
      FROM Cuenta WHERE número=7873698669 ) A;
UPDATE Cuenta SET saldo_clp=A.saldo_clp, saldo_usd=(A.saldo_clp/valor)
FROM ( SELECT valor FROM Divisa WHERE d1='USD' AND d2='CLP') T,
( SELECT saldo_clp FROM Cuenta WHERE número=7873698669 ) A;
COMMIT;
```



# Planificaciones Secuenciables vs. No Secuenciables





# El espectro de datos

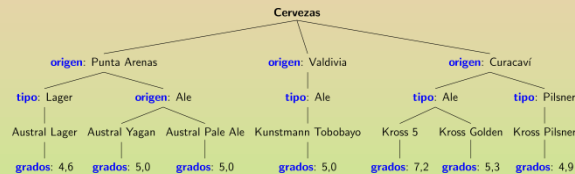
Texto Plano

Hay mucha variedad en las cervezas locales de Chile. La sede de la marca "Austral" se encuentra en Punta Arenas. Austral fabrica una amplia gama de cervezas, incluyendo Lager (4,6%), Yagan (un ale de 5%) y un Pale Ale (5%). La cerveza de marca "Kunstmann" también es popular, en particular su cerveza "Torobayo" (un ale de 5% elaborado en Valdivia). La marca Kross, basada en Curacaví, también tiene una gama de cervezas populares como, por ejemplo, Kross 5 (un ale fuerte de 7,2%) Kross Golden (un ale de 5,3%) y Kross Pilsner (4,9%).

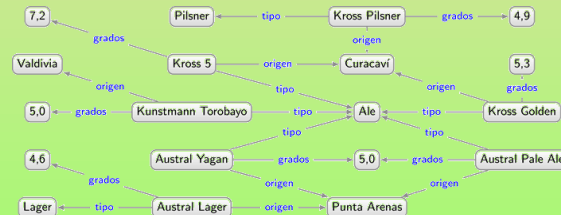
Texto Enriquecido  
(HTML, Word, ...)

```
<ul>
<li>Austral Lager [Lager] 4,6% de Punta Arenas</li>
<li>Austral Yagan [Ale] 5% de Punta Arenas</li>
<li>Austral Pale Ale [Ale] 5% de Punta Arenas</li>
<li>Kunstmann Torobayo [Ale] 5% de Valdivia</li>
<li>Kross 5 [Ale] 7,2% de Curacaví</li>
<li>Kross Golden [Ale] 5,3% de Curacaví</li>
<li>Kross Pilsner [Pilsner] 4,9% de Curacaví</li>
</ul>
```

Arboles  
(XML, JSON, ...)



Grafos  
(RDF, Prop. Gs, ...)



Relacional  
(SQL, CSV, ...)

nombre	tipo	grados	ciudad-origen
Austral Lager	Lager	4,6	Punta Arenas
Austral Yagan	Ale	5,0	Punta Arenas
Austral Pale Ale	Ale	5,0	Punta Arenas
Kuntsmann Torobayo	Ale	5,0	Valdivia
Kross 5	Ale	7,2	Curacaví
Kross Golden	Ale	5,3	Curacaví
Kross Pilsner	Pilsner	4,9	Curacaví

No estructurados

D

A

T

O

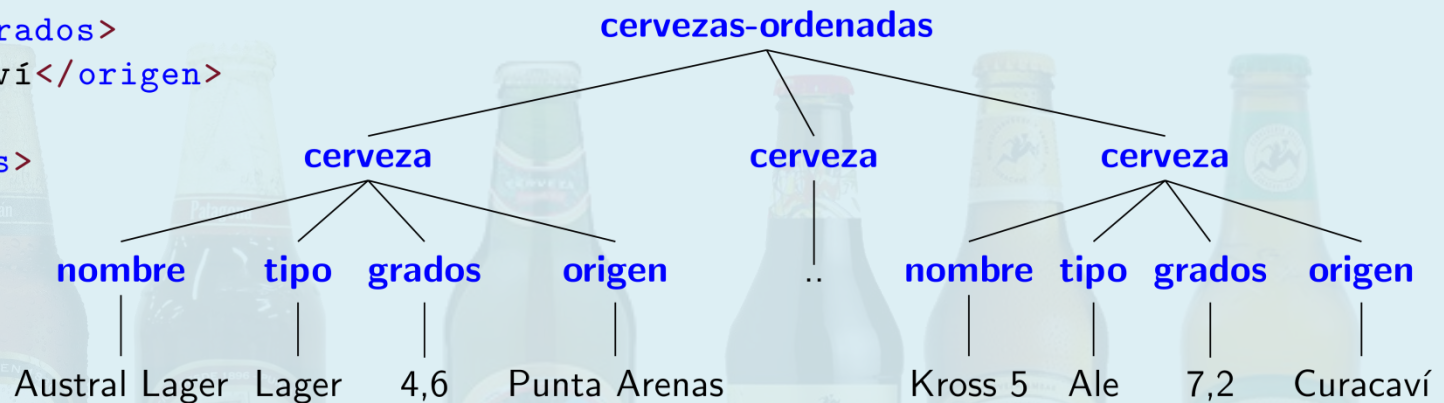
S

Semiestructurados

Estructurados

# XML

```
<cervezas-ordenadas>
  <cerveza>
    <nombre>Austral Lager</nombre>
    <tipo>Lager</tipo>
    <grados>4.6</grados>
    <origen>Punta Arenas</origen>
  <cerveza>
  <cerveza>
    <nombre>Kross Pilsner</nombre>
    <tipo>Pilsner</tipo>
    <grados>4.9</grados>
    <origen>Curacaví</origen>
  <cerveza>
  ...
  <cerveza>
    <nombre>Kross 5</nombre>
    <tipo>Ale</tipo>
    <grados>7.2</grados>
    <origen>Curacaví</origen>
  <cerveza>
</cervezas-ordenadas>
```



# XPath

```
<juego-de-tronos>
  <personaje uid="JDT01" genero="male" xml:lang="en">
    <nombre>Joffrey</nombre>
    <apellido>Baratheon</apellido>
    <actor>
      <nombre>Jack</nombre>
      <apellido>Gleeson</apellido>
      <pais xml:lang="es">Irlanda</pais>
    </actor>
    <debut>1</debut>
    <cultura>Andal</cultura>
    <casa>Baratheon</casa>
    <casa>Lannister</casa>
    <origen>King's Landing</origen>
  </personaje>
  <personaje uid="JDT02" genero="male" xml:lang="en">
    <nombre>Tommen</nombre>
    <apellido>Baratheon</apellido>
    <actor>...</actor>
    <actor>...</actor>
    <debut>1</debut>
    <cultura>Andal</cultura>
    <casa>Baratheon</casa>
    <casa>Lannister</casa>
    <origen>King's Landing</origen>
  </personaje>...
</juego-de-tronos>
```

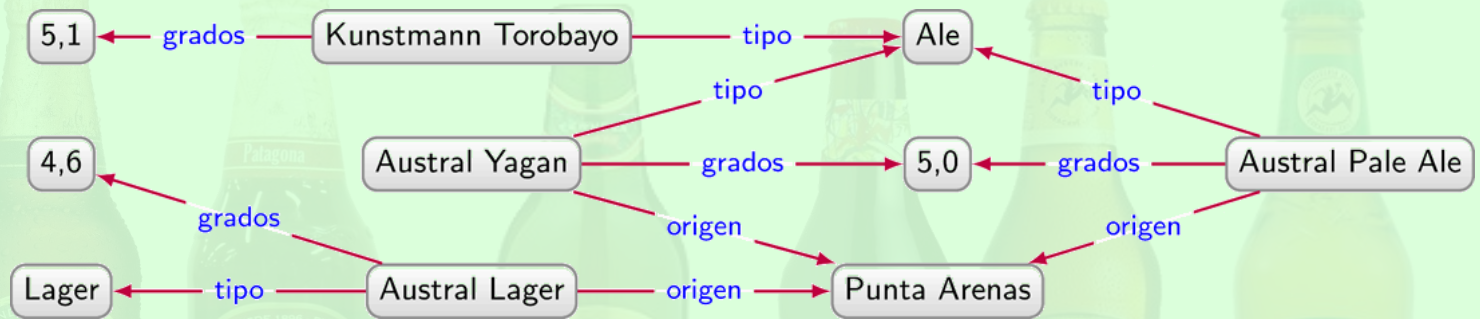
*¿Nombres de personajes actuados  
por un actor irlandés?*

```
//pais[text()='Irlanda']/ancestor::personaje/nombre
```

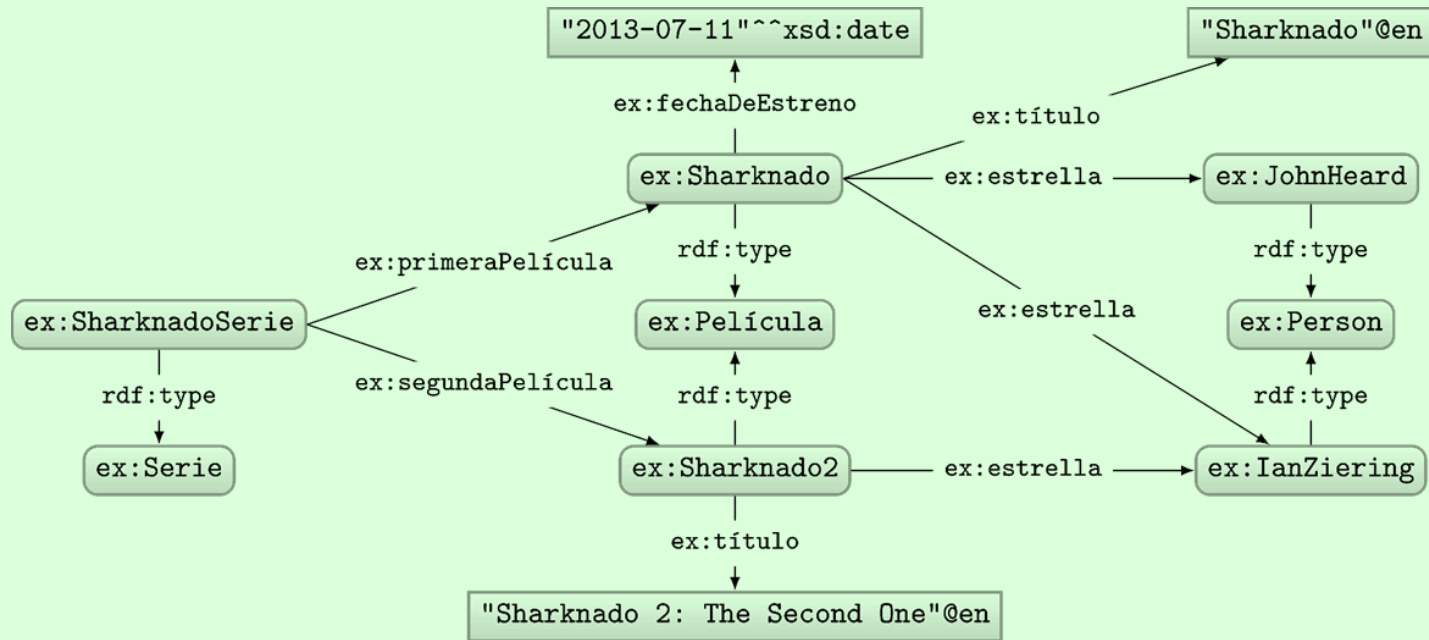
# Grafos

Kunstmann_Torobayo	tipo	Ale
Kunstmann_Torobayo	grados	5,1
Austral_Yagan	tipo	Ale
Austral_Yagan	grados	5,0
Austral_Yagan	origen	Punta_Arenas
Austral_Pale_Ale	tipo	Ale
Austral_Pale_Ale	grados	5,0
Austral_Pale_Ale	origen	Punta_Arenas
Austral_Lager	tipo	Lager
Austral_Lager	grados	4,6
Austral_Lager	origen	Punta_Arenas

*No hay orden.*



# RDF/SPARQL



```
PREFIX ex: <http://ex.org/voc#>
SELECT *
WHERE {
  { ex:SharknadoSerie ex:primeraPelícula ?p . }
  UNION
  { ex:SharknadoSerie ex:segundaPelícula ?p . }
  OPTIONAL
  { ?p ex:fechaDeEstreno ?f . }
  ?p ex:título ?t .
  FILTER(REGEX(STR(?t), "[0-9]*"))
}
```

?p	?t	?f
ex:Sharknado2	"Sharknado 2: The Second One"@en	

# Evaluación del curso

- 50% tareas:
  - 40% ejercicios/laboratorios
  - 10% proyecto
- 50% exámenes:
  - 2 controles, 1 examen final

# Proyectos

- Entregar las presentaciones y los informes:
  - El martes 18, 23:59 (estricto)
- Presentaciones de 4 minutos (estricto);
  - incluyendo (en breve):
    - el tema y la fuente de los datos
    - estadísticas de los datos
    - el esquema relacional
    - los índices y las vistas ocupados
    - las consultas
    - la implementación de la aplicación
    - las medidas de seguridad
    - una demostración de la aplicación
    - las lecciones aprendidas
- Todos los integrantes deberían participar

# Examen Final

- Nota de eximición:  $\geq 5,5$  promedio (controles)
- 1 de julio, 18:00 (Q10, S14)
- 3 horas
- 2 páginas de notas originales escritas a mano
- 4 preguntas (se cuentan las mejores 3)
  1. E/R, Álgebra Relacional
  2. SQL (consultas, crear tablas, vistas, inyección, etc.)
  3. Indexación, Formas Normales, Transacciones
  4. Datos semi-estructurados (árboles, grafos)



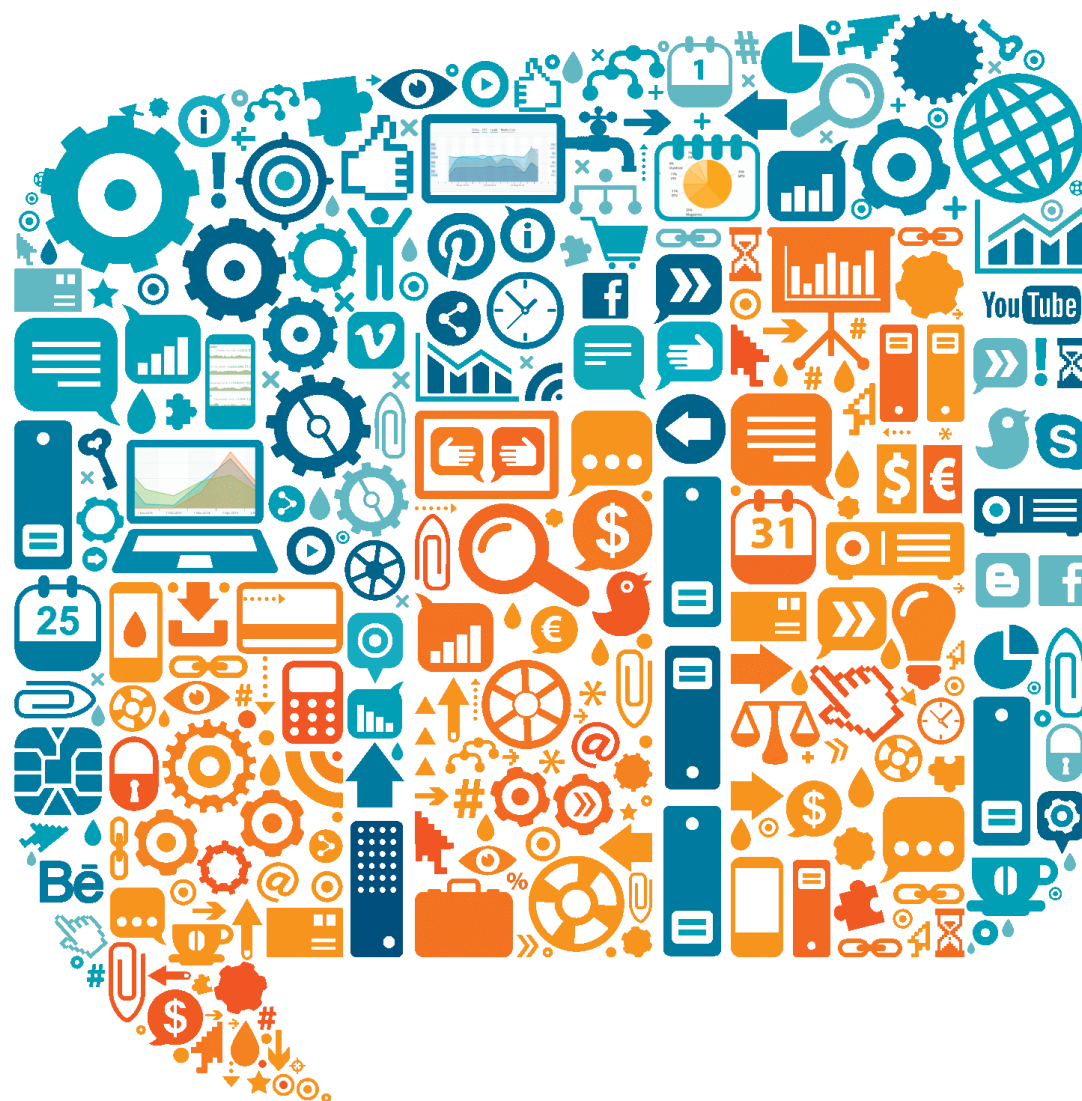
**THE END**



**THE END**



En muchos ámbitos de la computación,  
la gestión de datos es el desafío central



# Mucha demanda de “*data scientists*”

Harvard  
Business  
Review



The shortage of data scientists is becoming a serious constraint in some sectors.

DATA

## Data Scientist: The Sexiest Job of the 21st Century

by Thomas H. Davenport and D.J. Patil

FROM THE OCTOBER 2012 ISSUE

# “Data Scientist”: Ofertas de Trabajo (2016)

Here are the top 10 in-demand skills for data scientists:

Skills	Job skill appears in	% of jobs with skill
SQL	1987	56%
Hadoop	1713	49%
Python	1367	39%
Java	1287	36%
R	1120	32%
Hive	1099	31%
Mapreduce	768	22%
NoSQL	657	18%
Pig	561	16%
SAS	560	16%

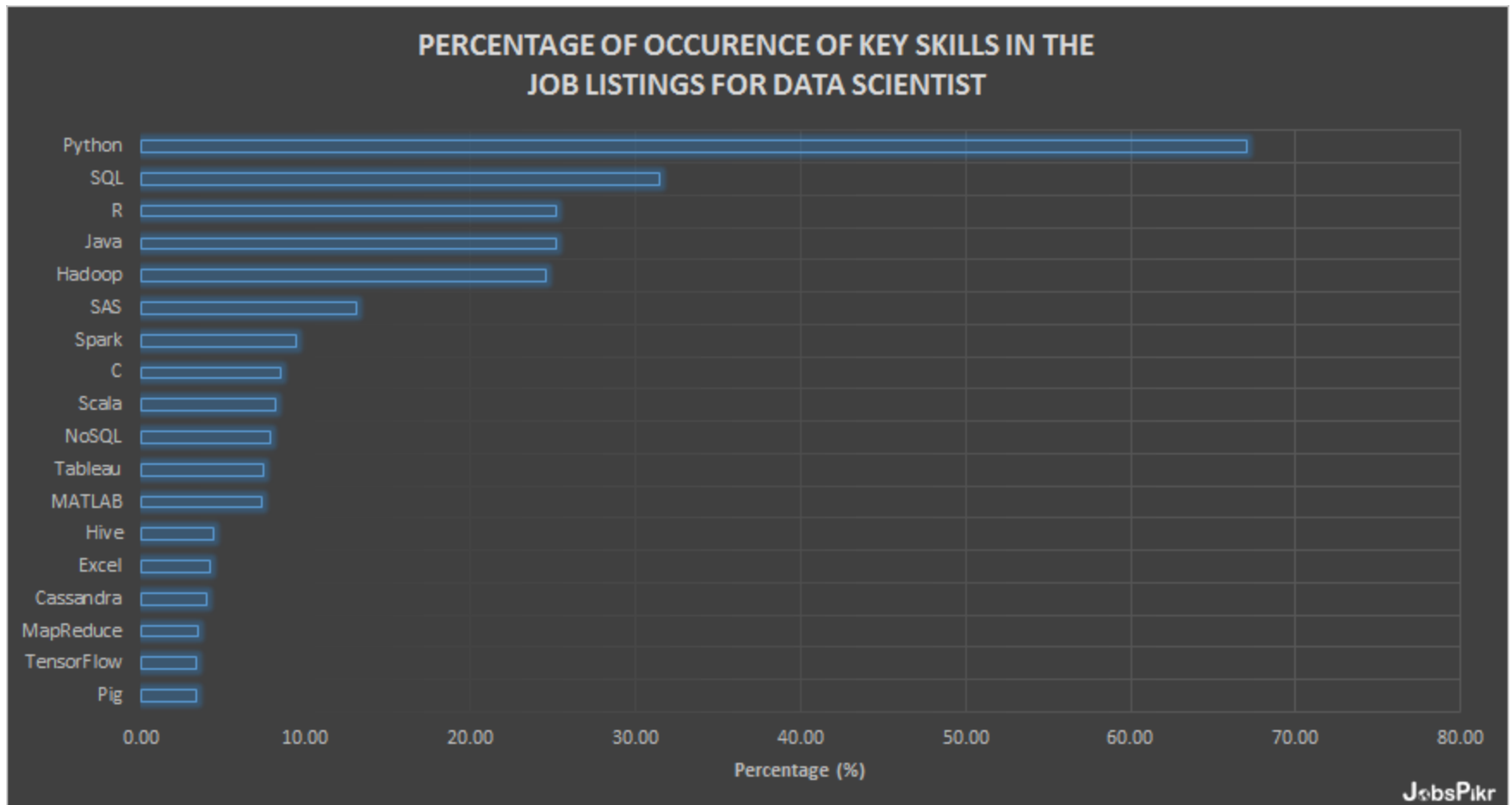
# “Data Scientist”: Ofertas de Trabajo (2017)

## Becoming A Data Scientist: The Skills That Can Make You The Most Money

To pinpoint the most common skills, Glassdoor took 10,000 data scientist job listings that appeared on its job search platform between January and July of this year. The skills required were noted, as were the salaries offered. The data coding skills were extrapolated and analysts searched for those that came up the most within listings. The ten skills that appeared most often as prerequisites for the job, and the percentage of job listings in which they appeared, were:

1. **Python** (72%)
2. **R** (64%)
3. **SQL** (51%)
4. **Hadoop** (39%)
5. **Java** (33%)
6. **SAS** (30%)
7. **Spark** (27%)
8. **Matlab** (20%)
9. **Hive** (17%)
10. **Tableau** (14%)

# “Data Scientist”: Ofertas de Trabajo (2018)





# Resumen de los cursos de datos



## WARNING

¡La información aquí no es oficial!

Puede ser que haya más cursos

Puede ser que los detalles hayan cambiado

No estoy promocionando nada

No sé qué se dicta en los cursos

No soy un experto en algunas áreas



APRENDIZAJE

# Minería de Datos

CC5206 [←CC3001 o FI2002]

Introducción a la Minería de Datos

Bárbara Poblete / Felipe Bravo

Primavera y Otoño



# Aprendizaje Computacional



CC5113 [←CC3301 y MA3403]

Aprendizaje Automático Bayesiano

Pablo Guerrero

Otoño

CC6204 [←CC3301]

Deep Learning

Juan Manuel Barrios, José M. Saavedra

Otoño

CC5509 [←CC3002 y CC3301]

Reconocimiento de Patrones

Mauricio Cerda

Otoño

MA5204 [←MA3401 o MA3403]

Aprendizaje de Máquinas

Felipe Tobar

Otoño

CC5112 [←CC3201]

Aprendizaje Computacional

Pablo Barceló

Primavera [2015]

CC5114 [←CC3002]

Redes Neuronales y

Programación Genética

Alexandre Bergel

Primavera

ESCALABILIDAD

# Procesamiento Masivo de Datos

CC5212 [←CC3201]

Procesamiento Masivo de Datos

Aidan Hogan

Otoño





# Sistemas Distribuidos

CC5303 [←(CC3201 o CC4303) y CC4302]

Sistemas Distribuidos

Javier Bustos Jiménez

Otoño [2018]

Computer 1

Computer 2

Computer 3

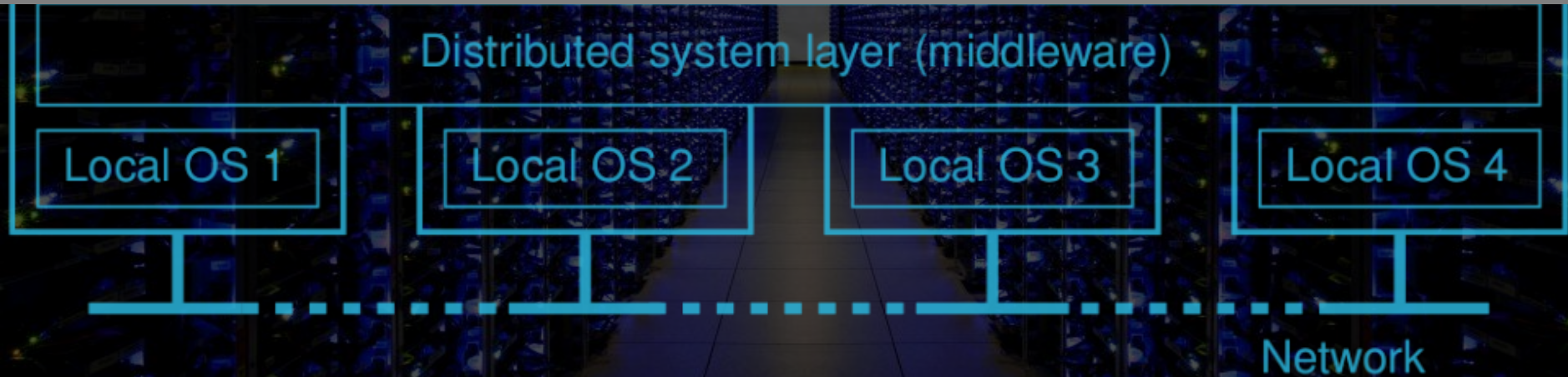
Computer 4

CC5304 [←(CC3201 o CC42A) y (CC4302 o CC41B)]

Arquitectura de Sistemas de Alta Disponibilidad

Cesar Guerrero

Otoño y Primavera



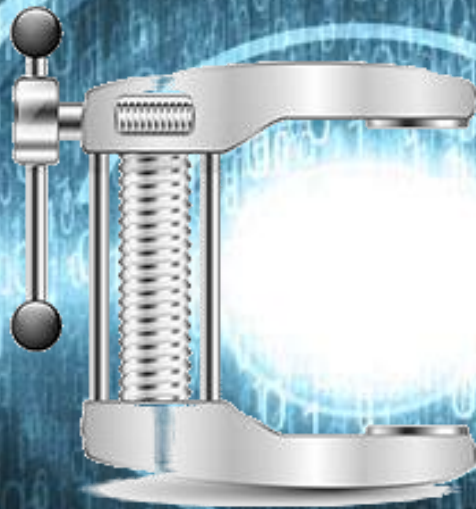
# Estructuras de Datos Comprimidas

CC7320

Estructuras de Datos Comprimidas

Gonzalo Navarro

Otoño



TEXTO/LENGUAJE NATURAL



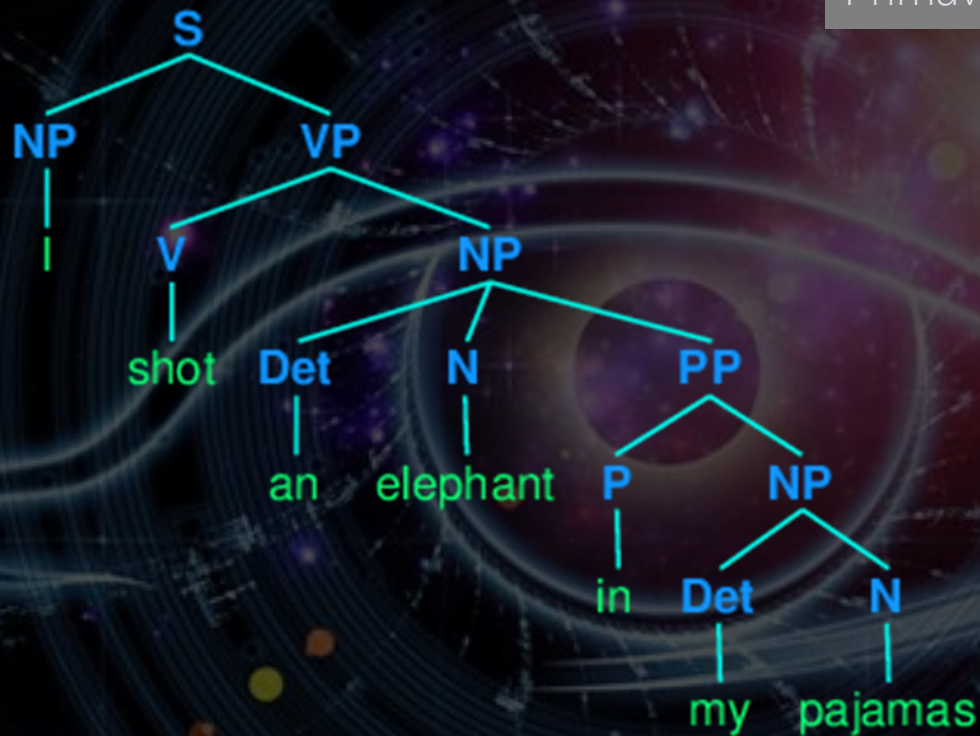
# Procesamiento de Lenguaje Natural

CC6205 [←CC3001 o MA3403]

Procesamiento de Lenguaje Natural

Felipe Bravo

Primavera



LÓGICA

# Métodos Lógicos

$L(a) \wedge L(b) \wedge L(c)$   
 $\text{¿}\forall x : L(x) \rightarrow Q(x)\text{?}$

CC5102 [ $\leftarrow$ CC3101]

Métodos Lógicos en  
Ciencias de la Computación

Pablo Barceló

Primavera [2017]



MULTIMEDIA

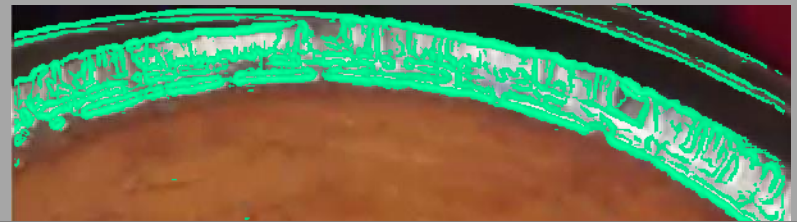
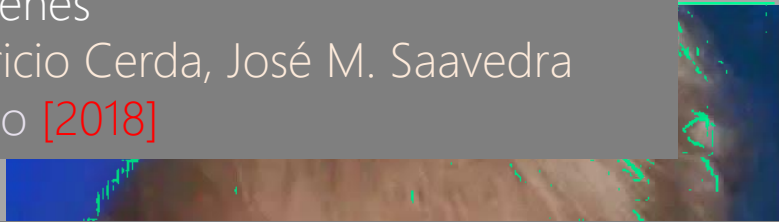
# Imágenes, Videos, Audio ...

CC5508 [←CC3001]

Procesamiento y Análisis de  
Imágenes

Mauricio Cerda, José M. Saavedra

Otoño [2018]

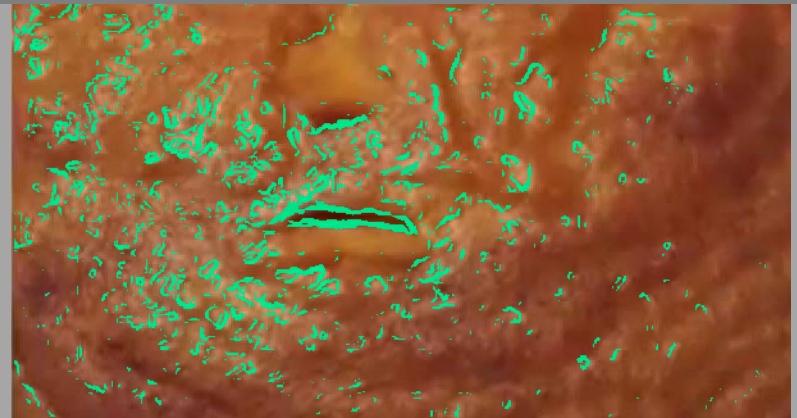
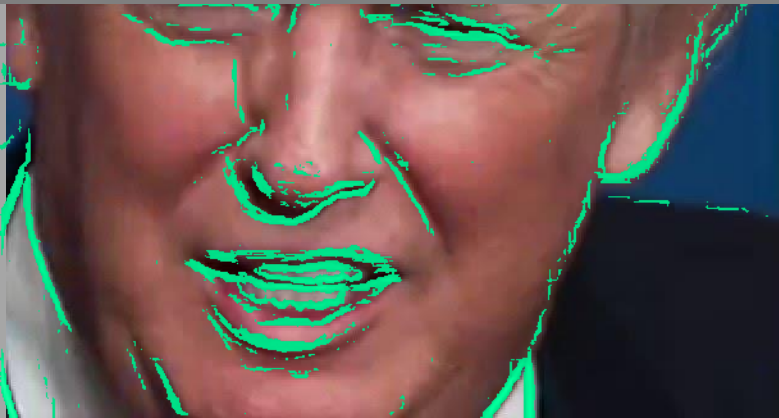


CC5213 [←CC4102 o CC5206 o EL4106]

Recuperación de Información Multimedia

Juan Manuel Barrios

Otoño y Primavera [2018]



VISUALIZACIÓN



# Visualización de Información

CC5208 [←CC3201]

Visualización de Información

Benjamín Bustos

Otoño



SEGURIDAD



# Seguridad de Datos

CC5301 [←CC3001 y CC3102 y MA3403]

Introducción a la Criptografía

Moderna

Alejandro Hevia

Otoño



LA WEB

# Desarrollo de Aplicaciones Web

CC5002 [←CC10A y CC30A y CC3001]  
Desarrollo de Aplicaciones Web  
José Urzúa  
Otoño





# La Web de Datos

CC7220 [←CC3201]

La Web de Datos

Aidan Hogan

Primavera



DOMINIOS ESPECÍFICOS

# Análisis de Negocio (*Business Analytics*)

CC5615 [←CC3201]

Business Analytics

Marcela Calderón, Hugo Mora

Primavera





# Astroinformática

AS4501 [←FI3104,MA3403,FI2003,FI2002]

Astroinformática

Francisco Förster

Juan Carlos Maureira

Patricio Rojo

Primavera



**THE END**





Preguntas?

