# CC3201-1

## Bases de Datos

## Otoño 2025

## Clase 11: SQL: Acceso Programático, Inyecciones, Respaldos

Aidan Hogan

aidhog@gmail.com

# Proyectos: Acceso Programático

# Tres opciones para la aplicación

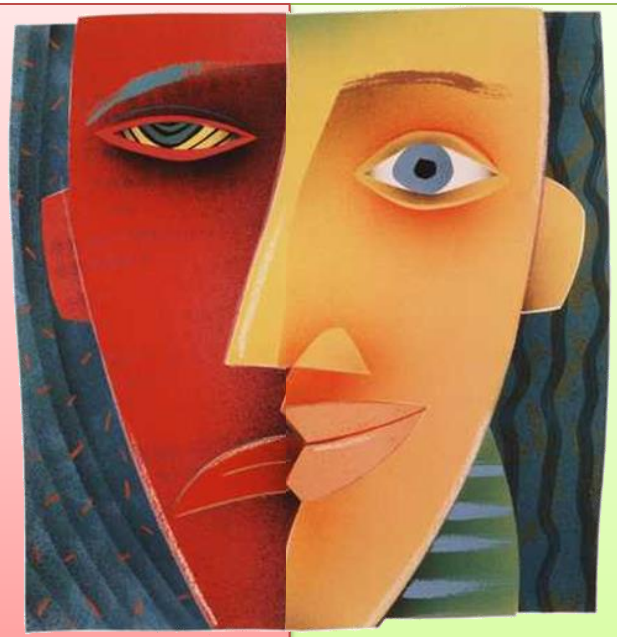
1. PHP
2. Django (Python)
3. Java Servlets

# (1) PHP


- Servidor Web: Apache2
- Aplicación: PHP
- Conector: PDO

Hay que programar en PHP.

Un poco desactualizado.



¡Se puede programar en PHP!

¡Liviano y fácil de instalar!

¡Todavía se usa bastante!

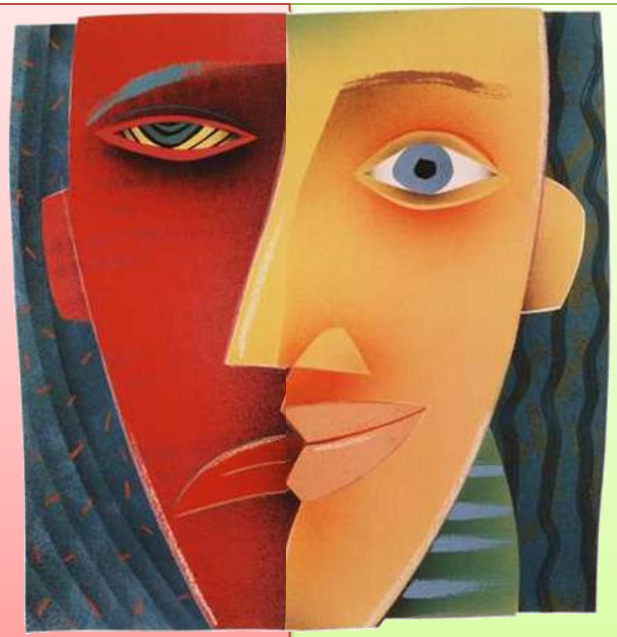
# (2) Django (Python)


- Servidor Web: Django
- Aplicación: Django/Python
- Conector: Django

Hay que programar en Python.

Difícil de instalar inicialmente.

Más indirecto.



¡Se puede programar en Python!

¡Un marco moderno!

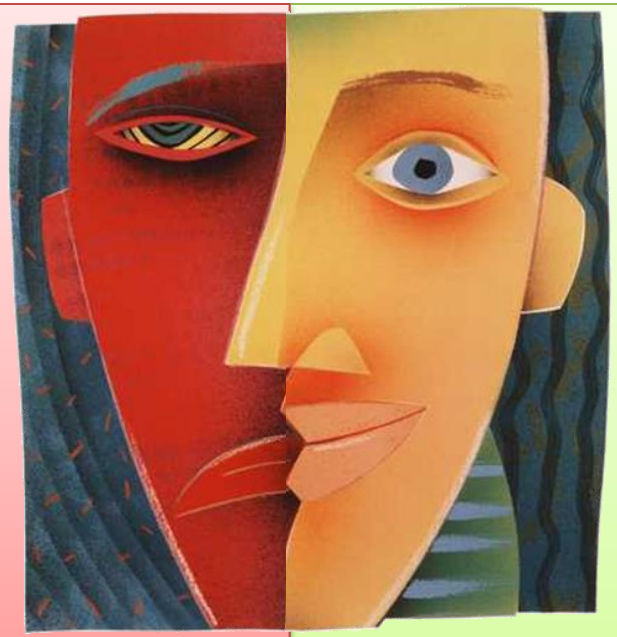
¡Da varias abstracciones!

# (3) Java Servlets


- Servidor Web: Apache Tomcat
- Aplicación: Java Servlets
- Conector: JDBC

Hay que programar en Java.

Un poco desactualizado.



¡Se puede programar en Java!

¡Se usa en muchos sistemas!

proyecto:armar_la_aplicacion_inicial

Tabla de Contenidos



# Armar la Aplicación

## Objetivo

La idea de la aplicación es tener una interfaz HTML donde el usuario puede ingresar algo (como en un "textbox" por ejemplo), y dada esta entrada, la aplicación crea una consulta SQL, ejecuta la consulta sobre la base de datos, y despliega los resultados en HTML para el usuario.

- Como fue mencionado antes, hay que tener al menos tres consultas demostrando una mezcla de rasgos de SQL, es decir, joins, consultas anidadas, agregación, etc.
- No es necesario tener todos los rasgos en todas las consultas. La idea es que se demuestren los rasgos en alguna consulta. Se puede empezar con una consulta simple.
- Es importante usar indices, vistas, etc., para optimizar las consultas.
- Si uno quiere usar una vista, y si la complejidad está en la vista (es decir, si la vista maneja la agregación o los joins, etc.), está bien tener una consulta más simple. O sea, si hay una vista con los rasgos mencionados, eso también cuenta.
- Se pueden armar varias interfaces (una página diferente para cada consulta) o una interfaz con varios formularios (una página para todas las consultas).
- Es importante usar métodos seguros, especialmente contra inyección.

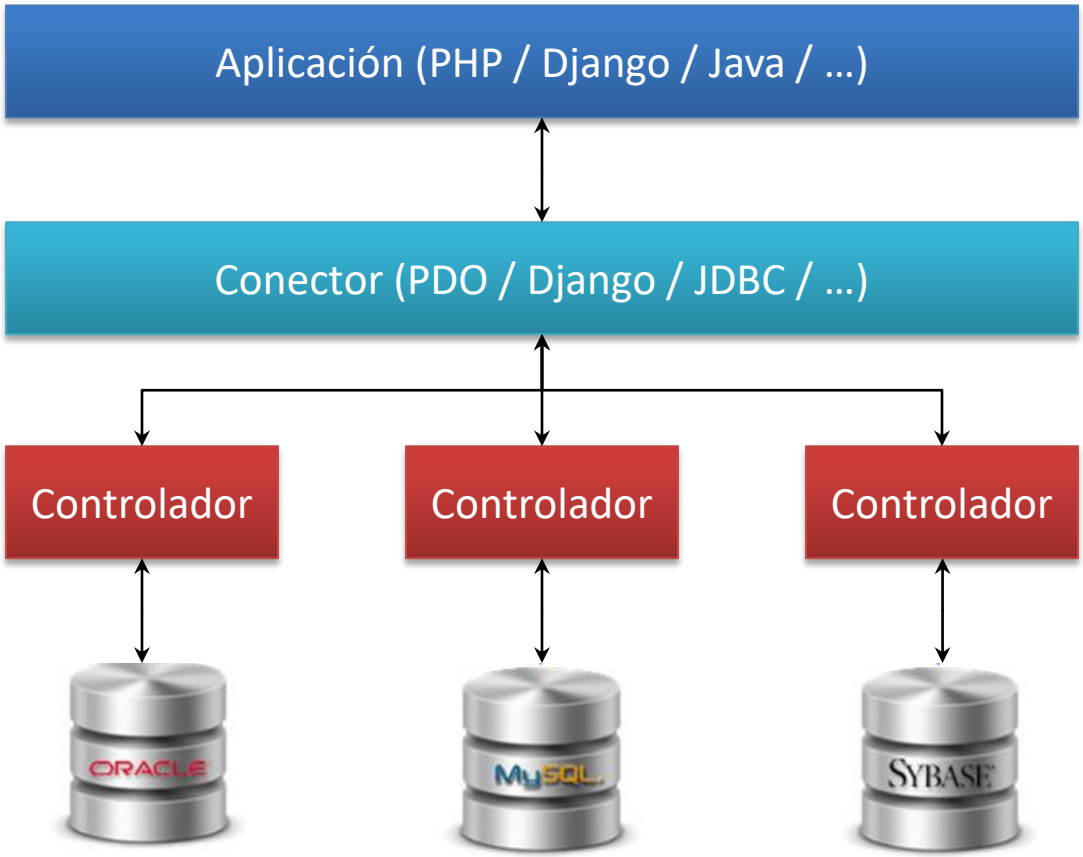
Este paso es más complejo que los otros, por lo que les aconsejamos que dejen bastante tiempo para completarlo en vez de hacerlo al último minuto. No importa qué software se use para armar la aplicación pero aquí daremos algunas opciones. Puede ser que haya mejores opciones y si ustedes quieren usar otra opción no hay problema (pero puede ser que no podamos ofrecerles apoyo … dependerá del software).

https://wiki.dcc.uchile.cl/cc3201/doku.php?id=proyecto:armar_la_aplicacion_inicial

# Acceso programático a bases de datos

Capítulo 6 | Ramakrishnan / Gehrke

# Acceso programático



# Definir conector y consultar

- PHP /PostgreSQL

```php
$conn = pg_connect("host = ... port = ... dbname = ... user = ... password = ...");
$result = pg_query($conn, "SELECT atr FROM tabla WHERE ...");
if($result){
    while ($row = pg_fetch_assoc($result)){
        echo "Result is ". $row["atr"];
    }
}
```

- PHP/PHP Data Objects (PDO)

```php
$conn = new PDO("pgsql:host=...;port=...;dbname=...;user=...;password=...");
$stmt = $conn->query("SELECT atr FROM tabla WHERE ...");
while ($row = $stmt->fetch()){
    echo "Result is ". $row["atr"];
}
```

PDO ofrece más flexibilidad para cambiar el sistema de bases de datos.

# Inyección SQL

# Un problema …

```
$input = ...;
$conn = pg_connect("host = ... port = ... dbname = ... user = ... password = ...");
$result = pg_query($conn, "SELECT atr FROM tabla WHERE id='". $input ."'");
if($result){
    while ($row = pg_fetch_assoc($result)){
        echo "Result is ". $row["atr"];
    }
}
```

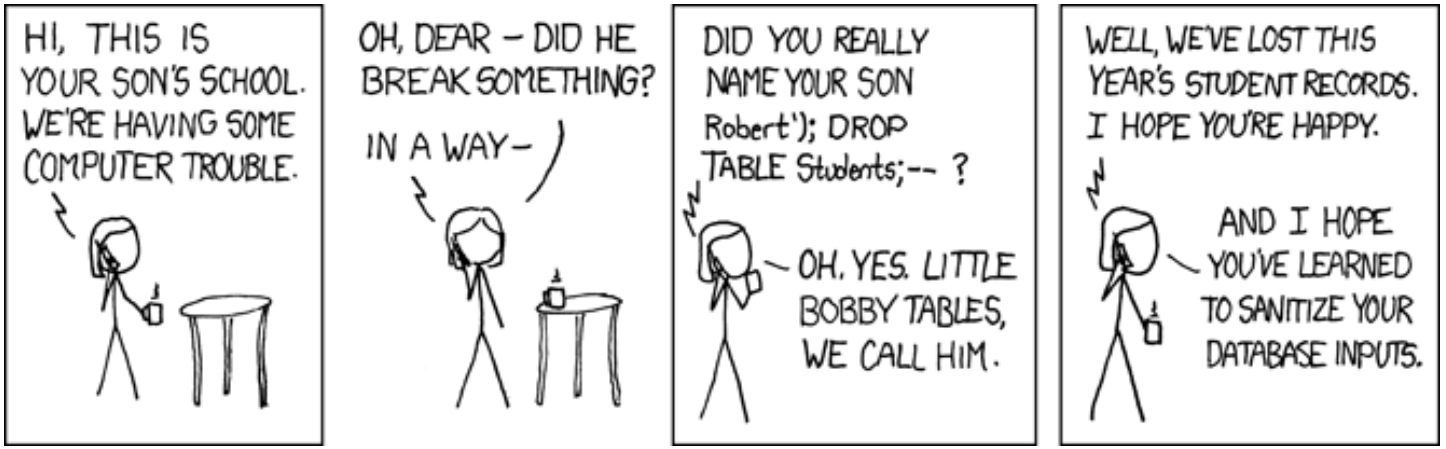
*¿Hay algún problema aquí?*

… no hemos “verificado” el `input`.


PC
imgflip.com

# Inyección SQL

- Un usuario malintencionado puede ingresar un string de entrada para hacer algo inesperado




https://xkcd.com/327/

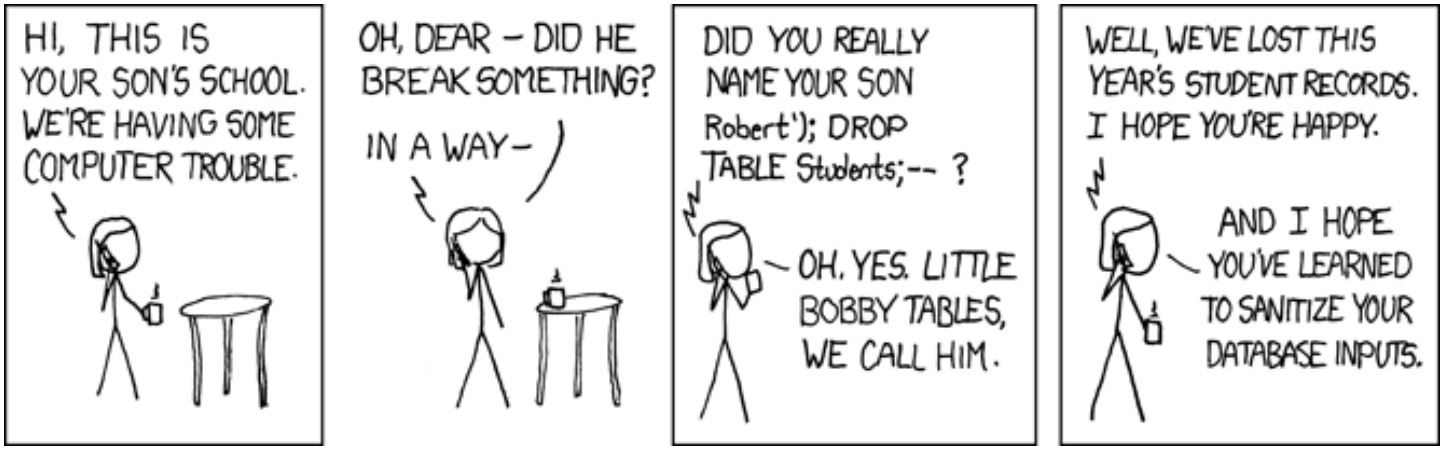
```
SELECT nota FROM Students WHERE name='".$input."'
```

```
SELECT nota FROM Students WHERE name='Robert'; DROP TABLE Students; -- '
```

('--' empieza un comentario)

# Inyección SQL

- Un usuario malintencionado puede ingresar un string de entrada para hacer algo inesperado




https://xkcd.com/327/

```
SELECT nota FROM Students WHERE name='".$input."'
SELECT nota FROM Students WHERE name='Robert' OR 1==1 -- '
```

*¿Qué hace el ejemplo?*

¡Devolverá toda la tabla!

# Parece estúpido pero …

WHITE HOUSE

## Mueller report: Russia hacked state databases and voting machine companies

### Russian intelligence officers injected malicious SQL code and then ran commands to extract information

The Russian intelligence officers at GRU exploited known vulnerabilities on websites of state and local election offices by injecting malicious SQL code on such websites that then ran commands on underlying databases to extract information.

Using those techniques in June 2016, "the GRU compromised the computer network of the Illinois State Board of Elections by exploiting a vulnerability in the SBOE's website," the report said. "The GRU then gained access to a database containing information on millions of registered Illinois voters, and extracted data related to thousands of U.S. voters before the malicious activity was identified."

# Parece estúpido pero ...

## The Top 10 OWASP vulnerabilities in 2020 are:

- Injection
- Broken Authentication
- Sensitive Data Exposure
- XML External Entities (XXE)
- Broken Access control
- Security misconfigurations
- Cross Site Scripting (XSS)
- Insecure Deserialization
- Using Components with known vulnerabilities
- Insufficient logging and monitoring

OWASP: Open Web Application Security Project

https://sucuri.net/guides/owasp-top-10-security-vulnerabilities-2020/

# Parece estúpido pero ...

## Injection

A code injection happens when an attacker sends invalid data to the web application with the intention to make it do something that the application was not designed/programmed to do.

Perhaps the most common example around this security vulnerability is the **SQL query consuming untrusted data**. You can see one of OWASP's examples below:

**String query = "SELECT * FROM accounts WHERE custID = '" + request.getParameter("id") + "'";**

This query can be exploited by calling up the web page executing it with the following URL: *http://example.com/app/accountView?id=' or '1'='1* causing the return of all the rows stored on the database table.

OWASP: Open Web Application Security Project

https://sucuri.net/guides/owasp-top-10-security-vulnerabilities-2020/

# Más ejemplos …

https://en.wikipedia.org/wiki/SQL_injection

## Examples [edit source]

- In February 2002, Jeremiah Jacks discovered that Guess.com was vulnerable to an SQL injection attack, permitting anyone able to construct a properly-crafted URL to pull down 200,000+ names, credit card numbers and expiration dates in the site's customer database.[23]
- On November 1, 2005, a teenage hacker used SQL injection to break into the site of a Taiwanese information security magazine from the Tech Target group and steal customers' information.[24]
- On January 13, 2006, Russian computer criminals broke into a Rhode Island government website and allegedly stole credit card data from individuals who have done business online with state agencies.[25]
- On March 29, 2006, a hacker discovered an SQL injection flaw in an official Indian government's tourism site.[26]
- On June 29, 2007, a computer criminal defaced the Microsoft UK website using SQL injection.[27][28] UK website *The Register* quoted a Microsoft spokesperson acknowledging the problem.
- In January 2008, tens of thousands of PCs were infected by an automated SQL injection attack that exploited a vulnerability in application code that uses Microsoft SQL Server as the database store.[29]
- In July 2008, Kaspersky's Malaysian site was hacked by a Turkish hacker going by the handle of "m0sted", who said to have used an SQL injection.
- In February 2013, a group of Maldivian hackers, hacked the website " UN-Maldives" using SQL Injection.
- In May 28, 2009 Anti-U.S. Hackers Infiltrate Army Servers Investigators believe the hackers used a technique called SQL injection to exploit a security vulnerability in Microsoft's SQL Server database to gain entry to the Web servers. "m0sted" is known to have carried out similar attacks on a number of other websites in the past—including against a site maintained by Internet security company Kaspersky Lab.
- On April 13, 2008, the Sexual and Violent Offender Registry of Oklahoma shut down its website for "routine maintenance" after being informed that 10,597 Social Security numbers belonging to sex offenders had been downloaded via an SQL injection attack[30]
- In May 2008, a server farm inside China used automated queries to Google's search engine to identify SQL server websites which were vulnerable to the attack of an automated SQL injection tool.[29][31]
- In 2008, at least April through August, a sweep of attacks began exploiting the SQL injection vulnerabilities of Microsoft's IIS web server and SQL Server database server. The attack does not require guessing the name of a table or column, and corrupts all text columns in all tables in a single request.[32] A HTML string that references a malware JavaScript file is appended to each value. When that database value is later displayed to a website visitor, the script attempts several approaches at gaining control over a visitor's system. The number of exploited web pages is estimated at 500,000.[33]
- On August 17, 2009, the United States Department of Justice charged an American citizen, Albert Gonzalez, and two unnamed Russians with the theft of 130 million credit card numbers using an SQL injection attack. In reportedly "the biggest case of identity theft in American history", the man stole cards from a number of corporate victims after researching their payment processing systems. Among the companies hit were credit card processor Heartland Payment Systems, convenience store chain 7-Eleven, and supermarket chain Hannaford Brothers.[34]
- In December 2009, an attacker breached a RockYou plaintext database containing the unencrypted usernames and passwords of about 32 million users using an SQL injection attack.[35]
- On July 2010, a South American security researcher who goes by the handle "Ch Russo" obtained sensitive user information from popular BitTorrent site The Pirate Bay. He gained access to the site's administrative control panel and exploited a SQL injection vulnerability that enabled him to collect user account information, including IP addresses, MD5 password hashes and records of which torrents individual users have uploaded.[36]
- From July 24 to 26, 2010, attackers from Japan and China used an SQL injection to gain access to customers' credit card data from Neo Beat, an Osaka-based company that runs a large online supermarket site. The attack also affected seven business partners including supermarket chains Izumiya Co, Maruetsu Inc, and Ryukyu Jusco Co. The theft of data affe… customers. As of August 14, 2010 it was reported that there have been more than 300 cases of credit card information being used by third parties to purchase goods and services in China.
- On September 19 during the 2010 Swedish general election a voter at… writing SQL commands as part of a write-in vote.[37]
- On November 8, 2010 the British Royal Navy website was compro… TinKode using SQL injection.[38][39]
- On February 5, 2011 HBGary, a technology security firm, was b… tion in their CMS-driven website[40]
- On March 27, 2011, mysql.com, the official homepage for M… ng SQL blind injection[41]
- On April 11, 2011, Barracuda Networks was compromised usi… s and usernames of employees were among the information obtained.[42]
- Over a period of 4 hours on April 27, 2011, an automated SQ… Reports website that was able to extract 8% of the username/password pairs: 8,000 random accounts of the 9,000 active and 90,000 old or inactive accounts.[43][44][45]
- On June 1, 2011, "hacktivists" of the group LulzSec were a… download keys, and passwords that were stored in plaintext on Sony's website, accessing the personal information of a million users.[46][47]
- In June 2011, PBS was hacked, mostly likely through use o… hackers to execute SQL injections was described in this Imperva blog.[48]
- In May 2012, the website for *Wurm Online*, a massively mu… m an SQL injection while the site was being updated.[49]
- In July 2012 a hacker group… he logins were stored in plain text and were allegedly taken from a Yahoo subdomain, Yahoo! Voices. The group breached Yahoo's security by using a "union-based SQL injection technique".[50][51]
- On October 1, 2012, a h… of students, faculty, employees, and alumni from 53 universities including Harvard, Princeton, Stanford, Cornell, Johns Hopkins, and the University of Zurich on pastebin.com. The hackers claimed that they were trying to "… changing education laws in Europe and increases in tuition in the United States.[52]
- On June 27, 201… they've been able to erase people's debts to water, gas, Internet, electricity, and telephone companies. Additionally, they published admin user name and password for other citizens to log in and clear …
- On … ing an SQL injection attack on the Chinese Chamber of International Commerce. The leaked data was posted publicly in cooperation with Anonymous.[55]

… hackers arrested for reporting the security vulnerability. 70,000 user details were exposed over this conflict.[56]

… me victim to an SQL injection attack carried out by an Anonymous hacker named "Hooky" and aligned with hacktivist group "RaptorSwag". The hackers compromised …

… information from nearly 420,000 websites through SQL injections.[60] *The New York Times* confirmed this finding by hiring a security expert to check the claim.[61]

… 2015, an … nications company Talk Talk's servers, exploiting a vulnerability in a legacy web portal[62]

# Más ejemplos …

https://en.wikipedia.org/wiki/SQL_injection

# La historia de HBGary y Anonymous

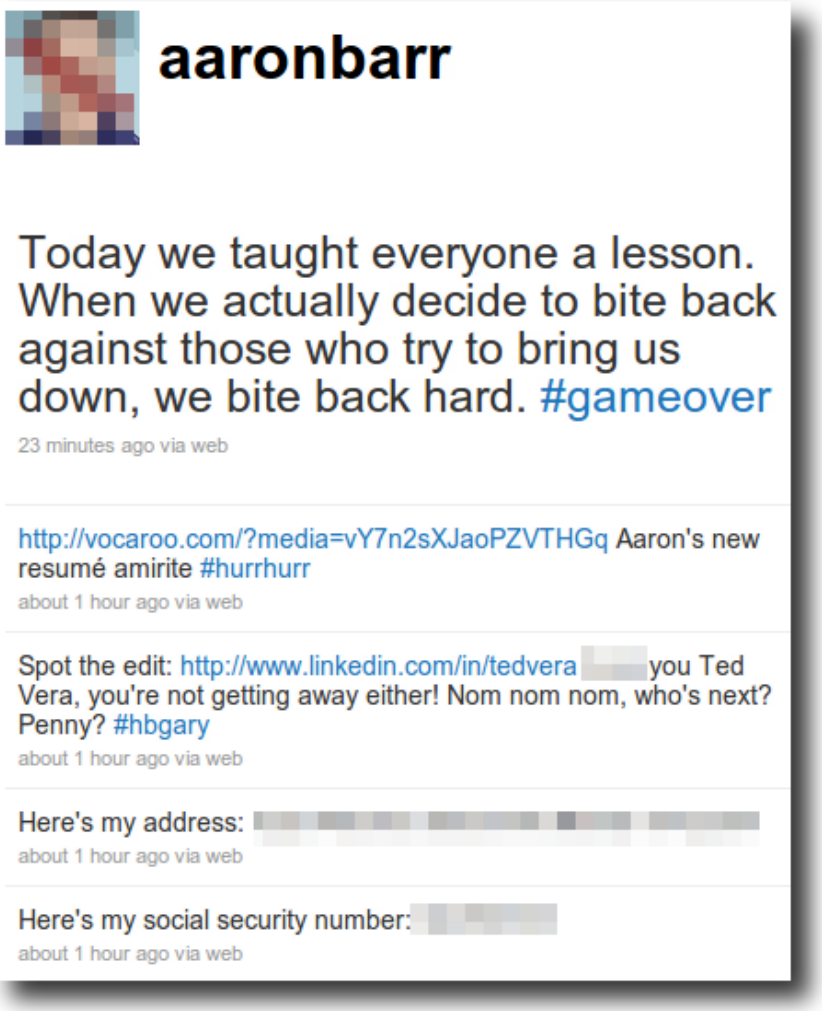
- 2010/12/08: Anonymous empiezan un “DDoS” (denegación de servicio distribuido) contra MasterCard, Visa, etc., por no aceptar donaciones a WikiLeaks

- 2011/02/05: Aaron Barr (CEO de HBGary Federal, una empresa de ciberseguridad) dice al Financial Times que ha logrado obtener información sobre las identidades de miembros de Anonymous

- 2011/02/05-06: Anonymous usa inyecciones SQL para obtener todos los datos de usuarios de HBGary, incluyendo hashes de contraseñas, con los cuales (y una tabla arcoíris) pueden hackear las cuentas sociales de Aaron Barr …

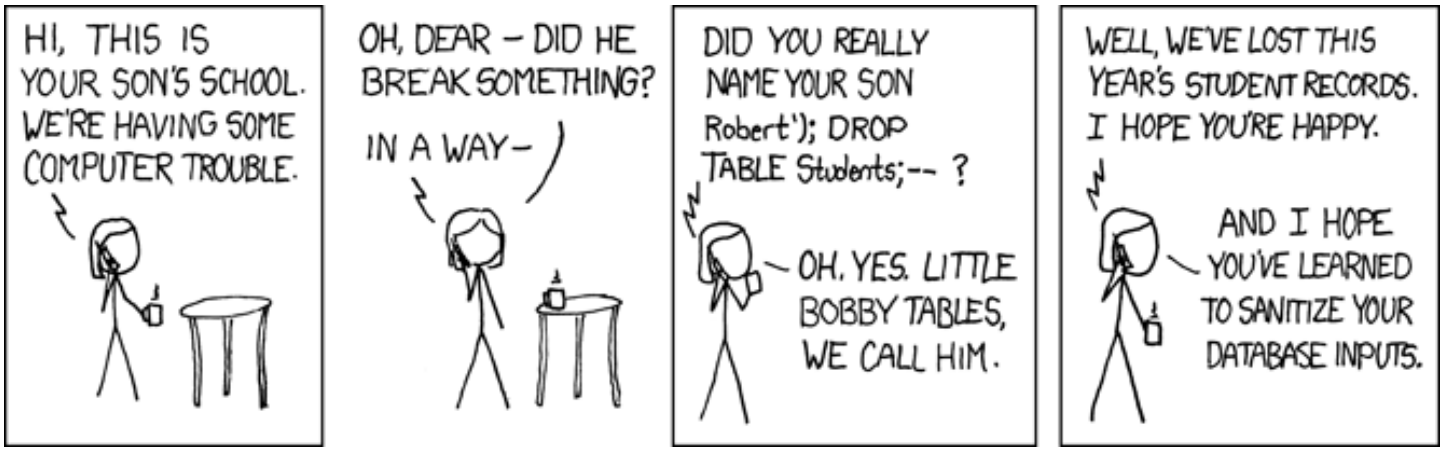
HBGary
Detecting Tomorrow's Threats Today
Part of ManTech International Corporation

# La historia de HBGary y Anonymous




HBGary
Detecting Tomorrow's Threats Today
Part of ManTech International Corporation

# Inyección SQL: *¿escapar los strings?*



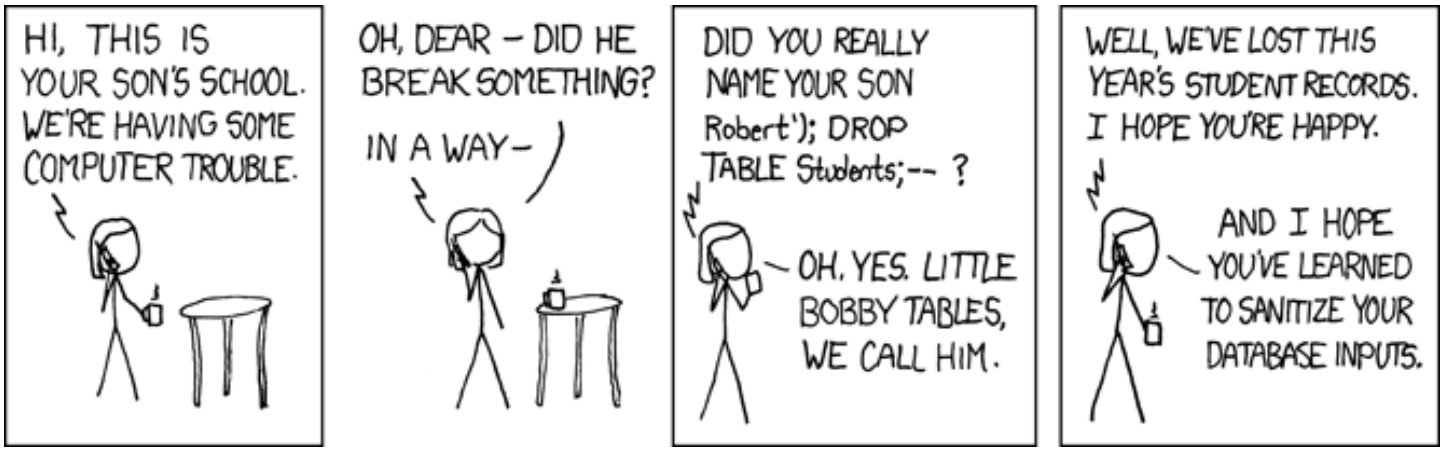

```
$stmt = $conn->query("SELECT nota FROM Students WHERE name='".$input."'");
```

*¿Cómo podemos resolver el problema?*

```
$einput = escape($input);
$stmt = $conn->query("SELECT nota FROM Students WHERE name='".$einput."'");
```

*Mejor*, pero sería complicado implementar la función escape en un lenguaje de programación general y garantizar que prevente cada tipo de inyección en cada versión (futura) de cada sistema de BdD dado cualquier tipo de consulta y entrada!

# Inyección SQL: *¡sentencias pre-compiladas!*




```
$stmt = $conn->prepare("SELECT nota FROM Students WHERE name=?");
$stmt->execute([$name]);
```

*Mandamos la consulta al sistema de bases de datos y después se reemplazarán los parámetros con la entrada del usuario*

# Inyección SQL: *¡sentencias pre-compiladas!*

```php
$stmt = $conn->prepare("SELECT nota FROM Students WHERE name=?"); // 1
$stmt->execute([$name]); // 2
```

```
// 1 :  El sistema de bases de datos compila la sentencia
SELECT nota FROM Students WHERE name=?
                                  QUERY PLAN
        ------------------------------------------------------------
         Seq Scan on Students  (cost=0.00..9654.67 rows=57 width=132)
           Filter: ((name)::text = ?::text)
```

El sistema de base de datos

*La consulta es compilada por el sistema **sin** la entrada*

# Inyección SQL: *¡sentencias pre-compiladas!*

```php
$stmt = $conn->prepare("SELECT nota FROM Students WHERE name=?"); // 1
$stmt->execute([$name]); // 2
```

```
// 2 :  Se reempleza el parametro en el plan y se ejecuta el plan
SELECT nota FROM Students WHERE name=?
```

| nota |
|---|
| 3,7 |

```
                                QUERY PLAN
-----------------------------------------------------------------------
 Seq Scan on Students  (cost=0.00..9654.67 rows=57 width=132)
   Filter: ((name)::text = 'Robert'::text)
```

El sistema de base de datos

*Se reemplaza el parámetro en la sentencia pre-compilada*

*(que es un plan en memoria, no un string)*

# Sentencias pre-compiladas

```
$stmt = $conn->prepare("SELECT nota FROM Students WHERE name=? AND year=?"); // 1
$stmt->execute([$name,$year]); // 2
```

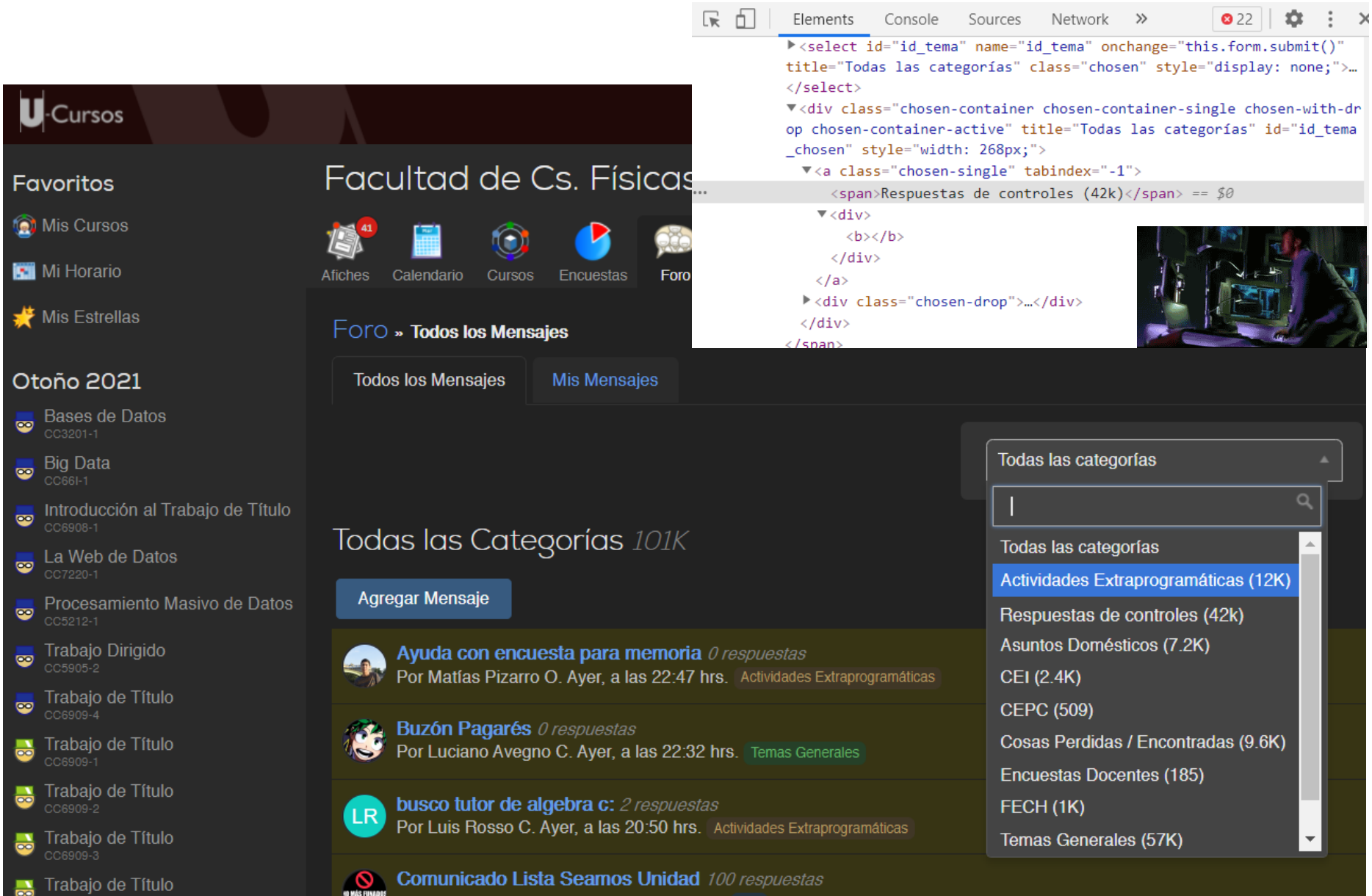
Se puede reutilizar el `PreparedStatement` varias veces

(es más eficiente también: se compila la sentencia solo una vez)

Se pueden tener varios parámetros

# ¡Interfaces restringidas de HTML no ayudan!

# ¡Interfaces restringidas de HTML no ayudan!



# ¡Interfaces restringidas de HTML no ayudan!

- Se puede editar la página HTML para cambiar la interfaz

- Se puede mandar cualquiera petición HTTP sin usar la interfaz HTML

# Respaldos

# Tipos de errores

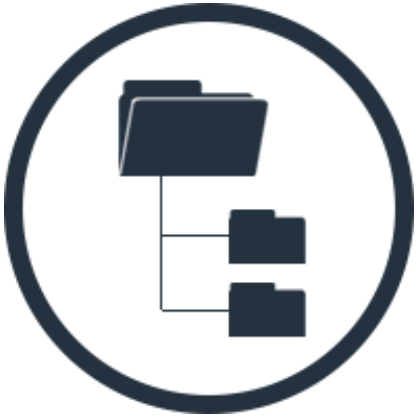
Humano

Hardware

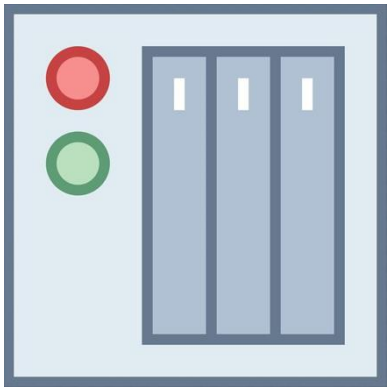
Software

# Tipos de respaldo

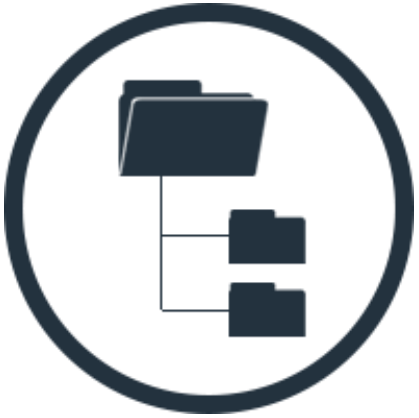
Nivel del
sistema de
base de datos

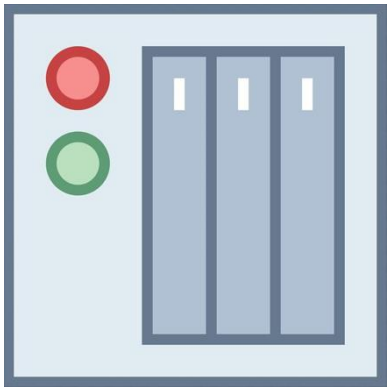


Nivel del
sistema de
archivos



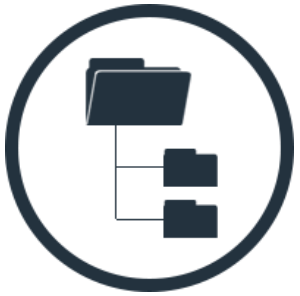
Nivel del
hardware





# RESPALDOS EXTERNOS

# Respaldos Externos:
## Sistema de Archivos



- Usar métodos estándares de respaldar archivos

✓ Algo simple

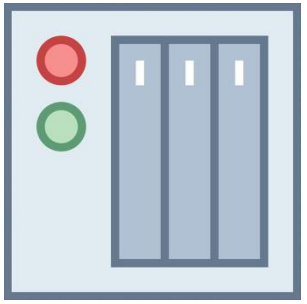
○ Solo hay protección ante errores de hardware si se usa otro disco para respaldar la información

× En sí, no provee protección ante errores humanos o de software (solo respaldará el estado actual)

× En general, habría que detener el sistema de base de datos para hacer un respaldo “coherente”

× No podemos consultar los respaldos

# Respaldos Externos: Hardware



- Replicar el disco (p.ej., RAID-1) o la máquina

✓ Protección ante errores de hardware

✓ Podemos consultar el respaldo (más lecturas)

✓ No es necesario desactivar el sistema de base de datos

× Mantener la réplica actualizada puede tener un costo (en particular en el caso de usar otra máquina)

× En sí, no provee protección ante errores humanos o de software (solo respaldará el estado actual)

× Existe el costo de comprar y mantener el hardware adicional

# RESPALDOS INTERNOS

# Respaldos Internos: Completos

- Respaldar todos los datos dentro del sistema de base de datos cada vez (por ejemplo, cada noche)


✓ No hay que parar el sistema de base de datos

✓ Más fácil de cargar de nuevo (que las opciones que siguen)

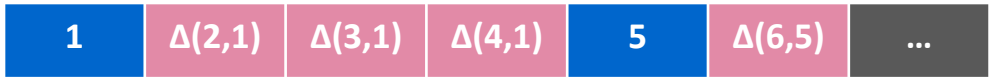
○ Solo hay protección ante errores de hardware si se usa otro disco para respaldar la información

× Mantener una historia de copias completas puede ocupar mucho espacio

× No podemos consultar los respaldos

# Respaldos Internos: Diferencial

- Respaldar todos los datos que hayan cambiado desde el último respaldo completo




✓ No hay que detener el sistema de base de datos

○ Puede mantener una historia de copias en menos espacio, pero el espacio usado aumenta en función del último respaldo completo

○ La carga del respaldo será un poco más costosa que en el caso del respaldo completo

○ Solo hay protección ante errores de hardware si se usa otro disco para respaldar la información

× No podemos consultar los respaldos

# Respaldos Internos: Incremental

- Respaldar todos los datos que han cambiado desde el último respaldo completo o incremental


- ✓ No hay que detener el sistema de base de datos
- ✓ Puede mantener una historia de copias en menos espacio, y el espacio no aumentará en función del último respaldo completo
- ○ Solo hay protección ante errores de hardware si se usa otro disco para respaldar la información
- × La carga del respaldo será un poco más costosa que en el caso del respaldo completo
- × No podemos consultar los respaldos

# Respaldos Internos: Registros

- Respaldar el registro de transacciones del sistema de base de datos (al nivel de sistema de archivos)


✓ No hay que detener el sistema de base de datos

✓ Se pueden repetir las transacciones desde un punto particular, o hasta un punto particular, o saltando una transacción

✓ No hay que mantener copias históricas (contiene la historia)

○ Solo hay protección ante errores de hardware si se usa otro disco para respaldar la información

× La carga del respaldo será significativamente más costosa que en los otros casos

× No podemos consultar los respaldos directamente

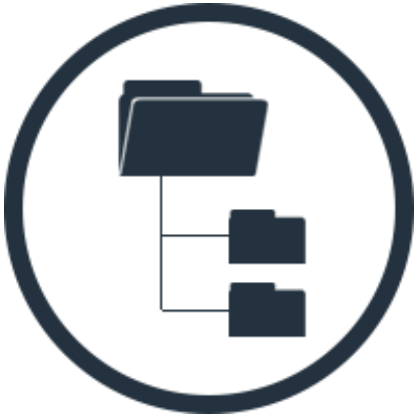
# Respaldos Internos: En la práctica

- No está en el estándar de SQL
  - Hay que revisar la documentación del sistema de base de datos particular

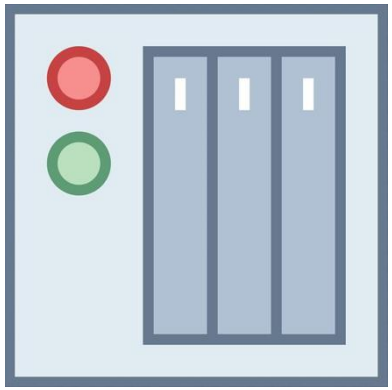
- Ejemplos de respaldos completos

MySQL:
```
shell> mysqldump --databases cc3201 > dump.msql
```
Postgres:
```
shell> pg_dump cc3201 > dump.psql
```
SQL Server
```
shell> BACKUP DATABASE cc3201 TO DISK = 'dump.sqls'
```

Nivel del
sistema de
base de datos



Nivel del
sistema de
archivos



Nivel del
hardware


Hemos terminado con bases de datos relacionales

# Preguntas?