CC3201-1

Bases de Datos

Otoño 2021

Clase 8: SQL: Acceso Programático, Inyecciones, Seguridad

Aidan Hogan

aidhog@gmail.com

Proyectos: Acceso Programático

Tres opciones para la aplicación

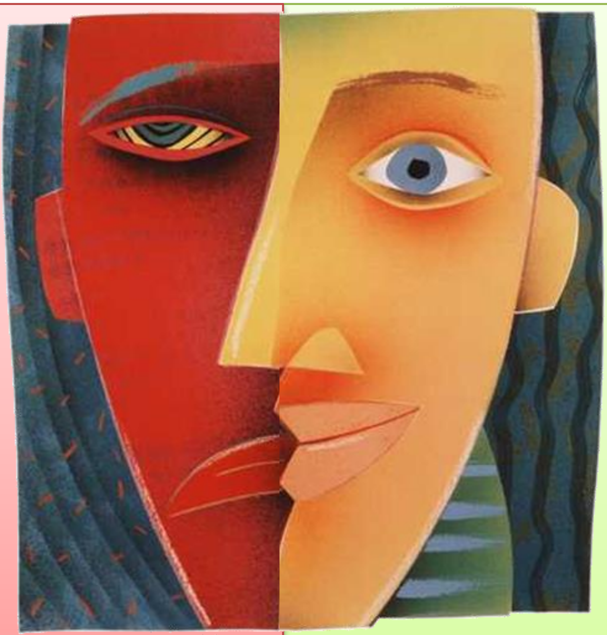
1. Java Servlets
2. PHP
3. Django (Python)

(1) Java Servlets

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

Hay que programar en Java.

Un poco desactualizado.



¡Se puede programar en Java!

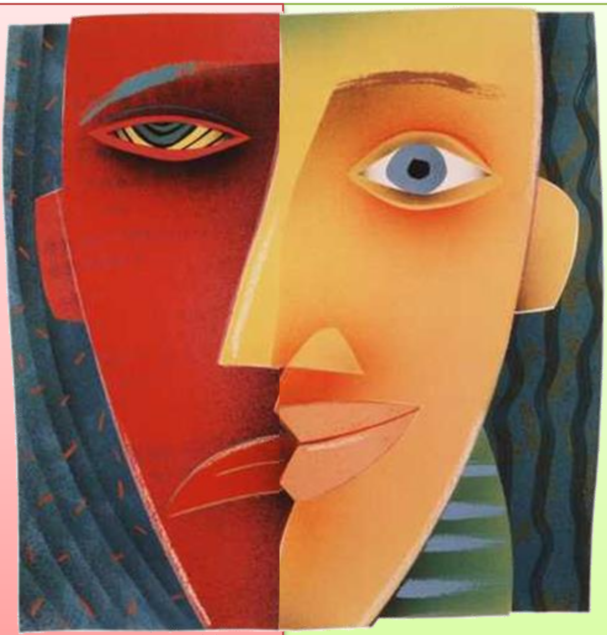
¡Se usa en muchos sistemas!

(2) 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!

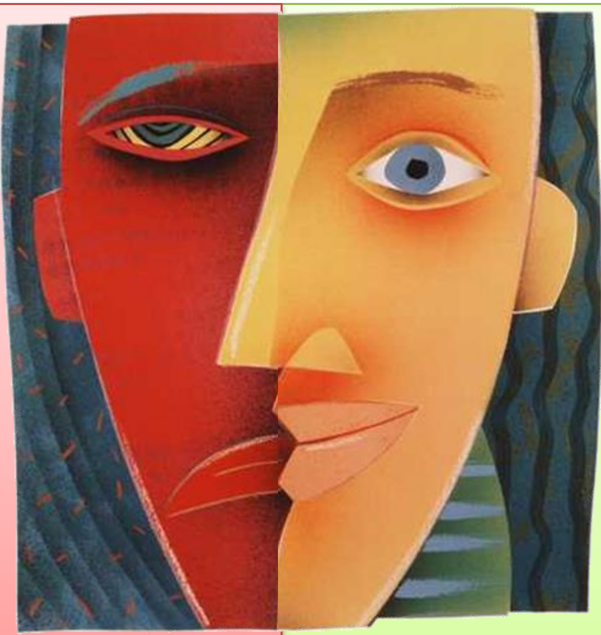
(3) Django (Python)

- 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!

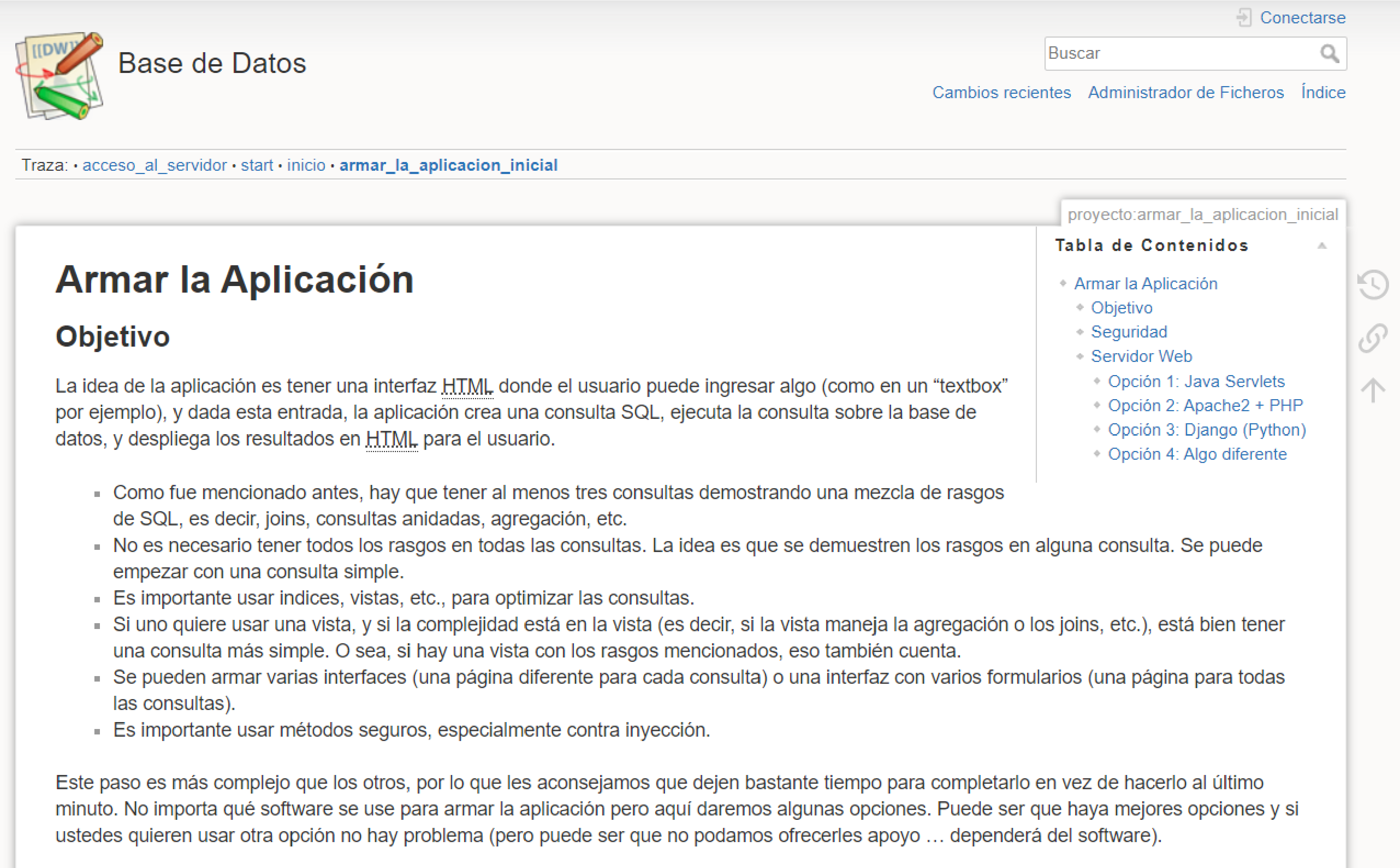

Conectarse
[[DW]]
Base de Datos
Buscar
Cambios recientes
Administrador de Ficheros
Índice
Traza: · acceso_al_servidor · start · inicio · armar_la_aplicacion_inicial
proyecto:armar_la_aplicacion_inicial

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.

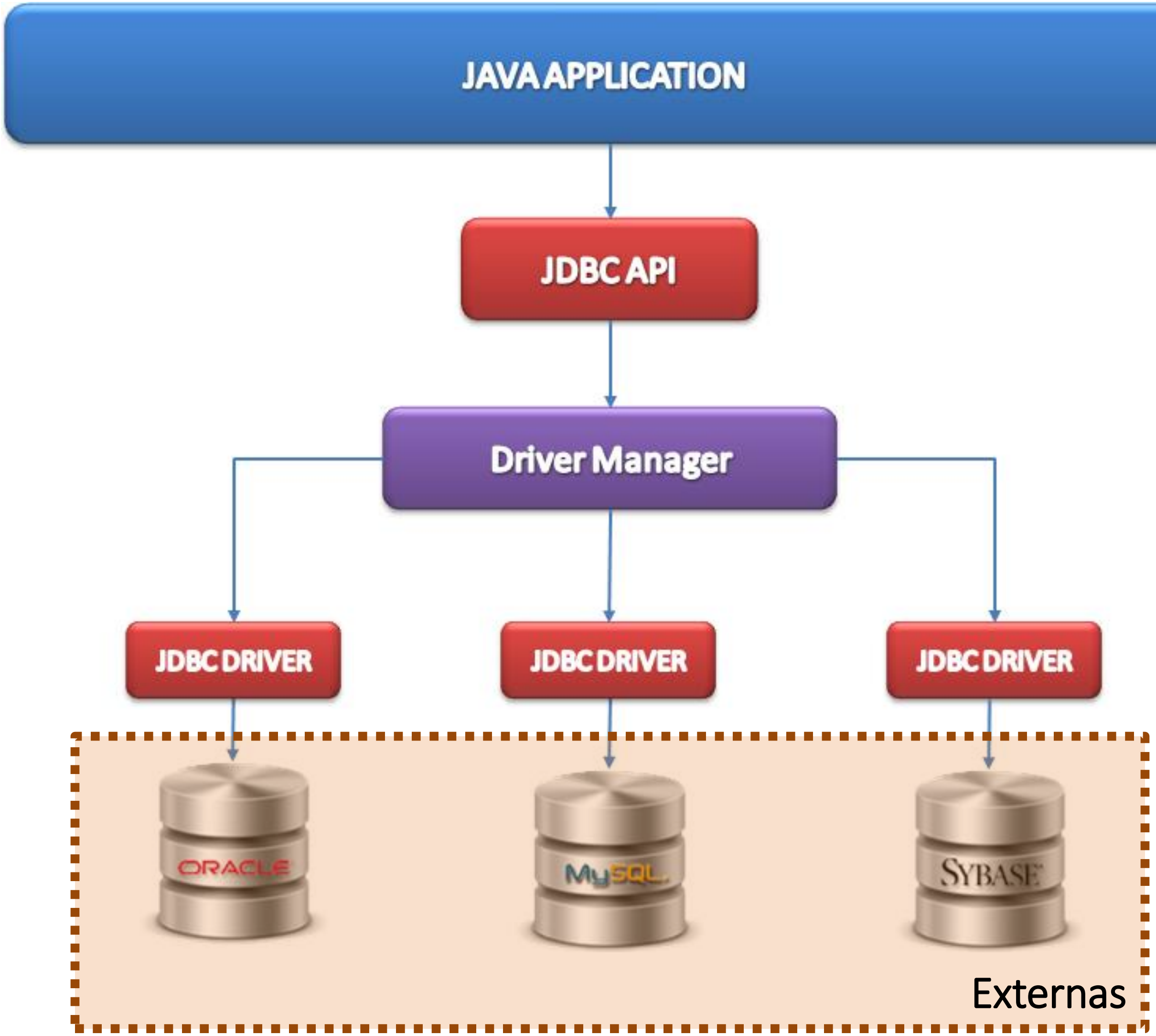
- Como fue mencionado antes, hay que tener al menos tres consultas demostrando una mezcla de rasgos de SQL, es decir, joins, consultas anidadas, agregación, etc.
- No es necesario tener todos los rasgos en todas las consultas. La idea es que se demuestren los rasgos en alguna consulta. Se puede empezar 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).

Acceso programático (Java): *Java Database Connectivity* (JDBC)

Capítulo 6 | Ramakrishnan / Gehrke

Java Database Connectivity (JDBC)



Consulta vs. Actualización

- Para hacer consultas (SELECT):

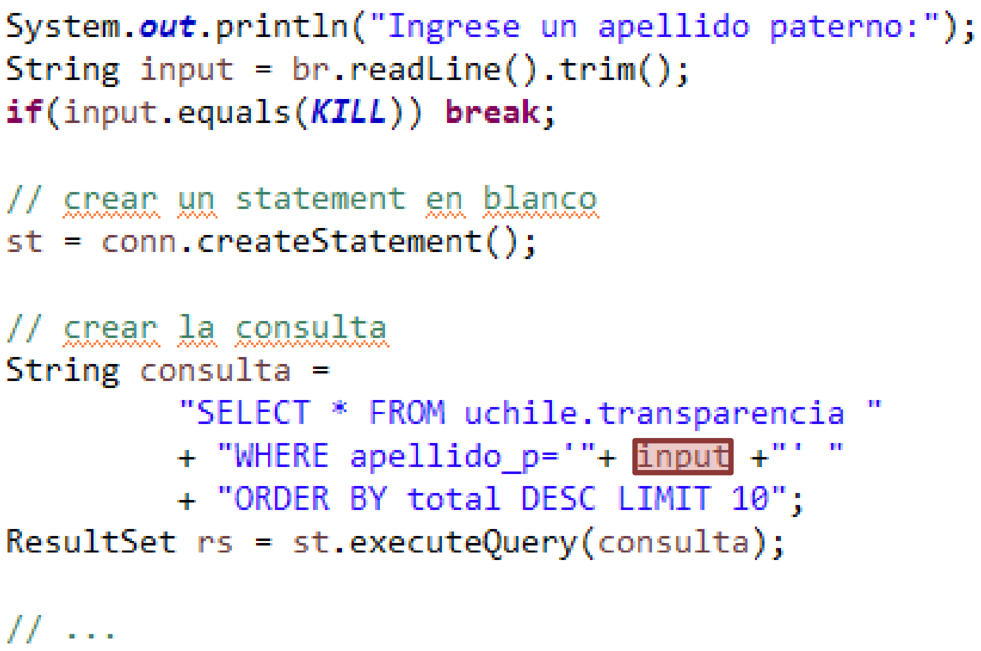
```
String consulta = "SELECT ...";
ResultSet rs = statement.executeQuery(consulta);
```

- Para hacer actualizaciones (INSERT; UPDATE, ...)

```
String actualizacion = "UPDATE ...";
int tuplasAfectadas = statement.executeUpdate(actualizacion);
```

INYECCIÓN SQL

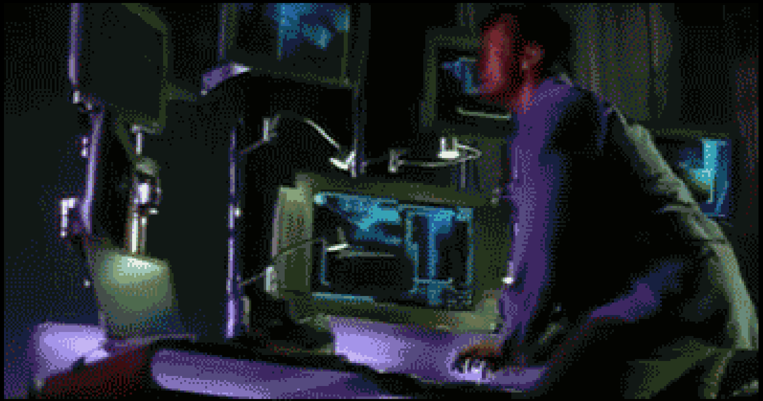
Un problema …



```
System.out.println("Ingrese un apellido paterno:");
String input = br.readLine().trim();
if(input.equals(KILL)) break;

// crear un statement en blanco
st = conn.createStatement();

// crear la consulta
String consulta =
         "SELECT * FROM uchile.transparencia "
       + "WHERE apellido_p='"+ input +"' "
       + "ORDER BY total DESC LIMIT 10";
ResultSet rs = st.executeQuery(consulta);

// ...
```

¿Hay algún problema aquí?

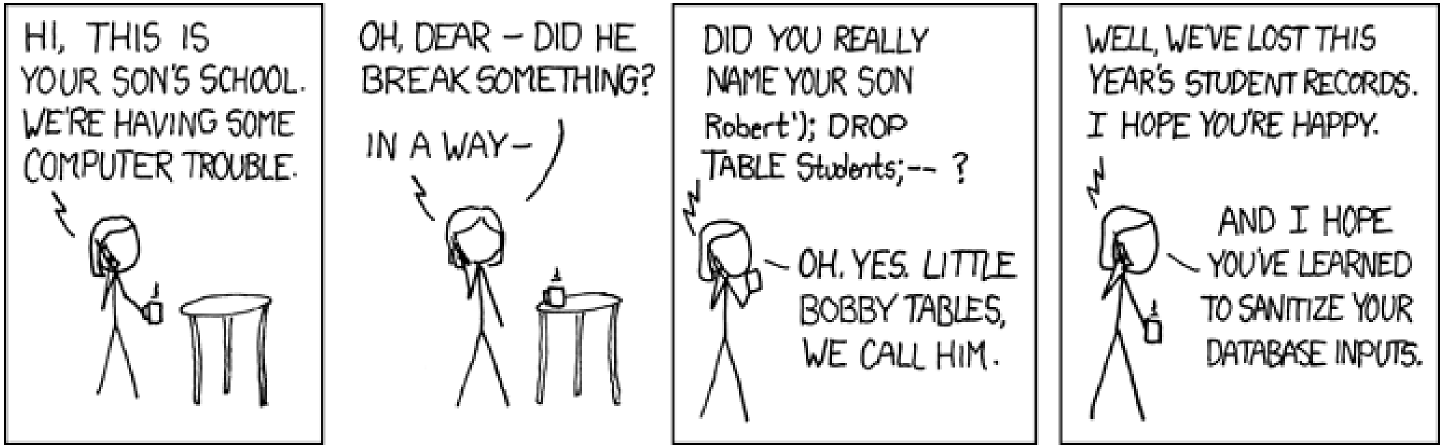
… no hemos “verificado” el input.



PC

Inyección SQL

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

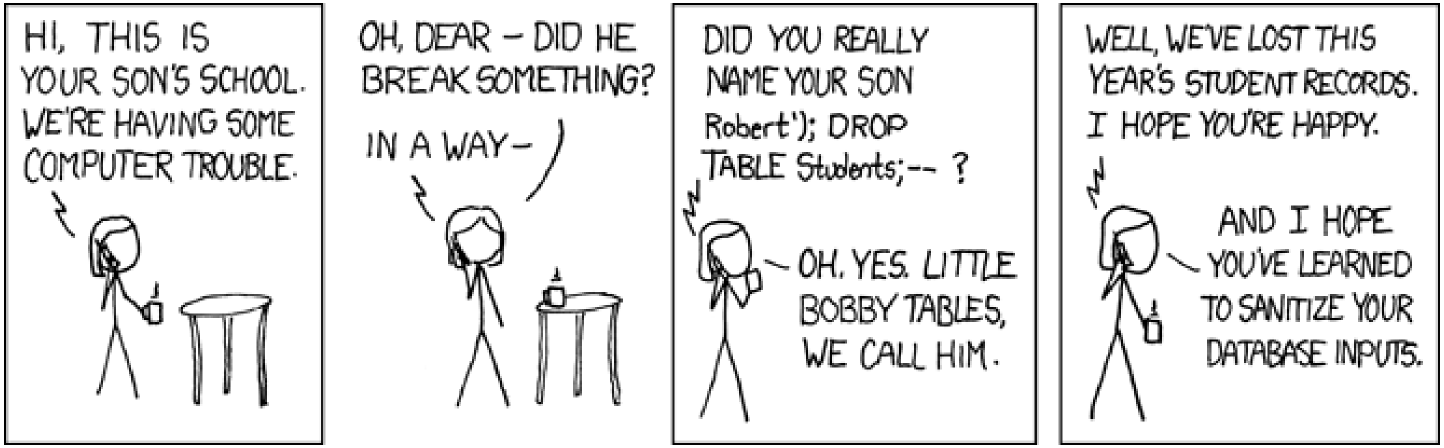

HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

```
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

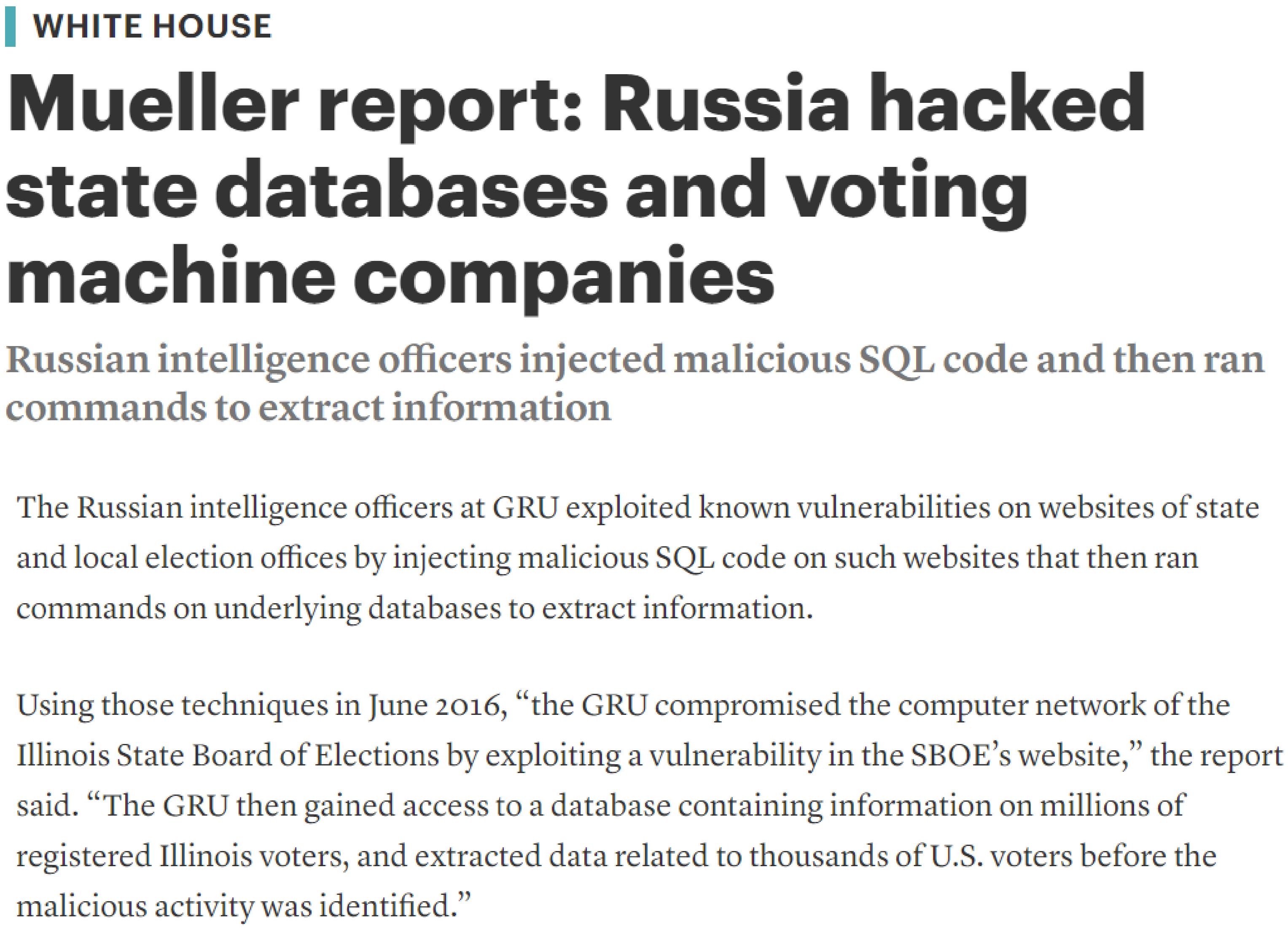

HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

```
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

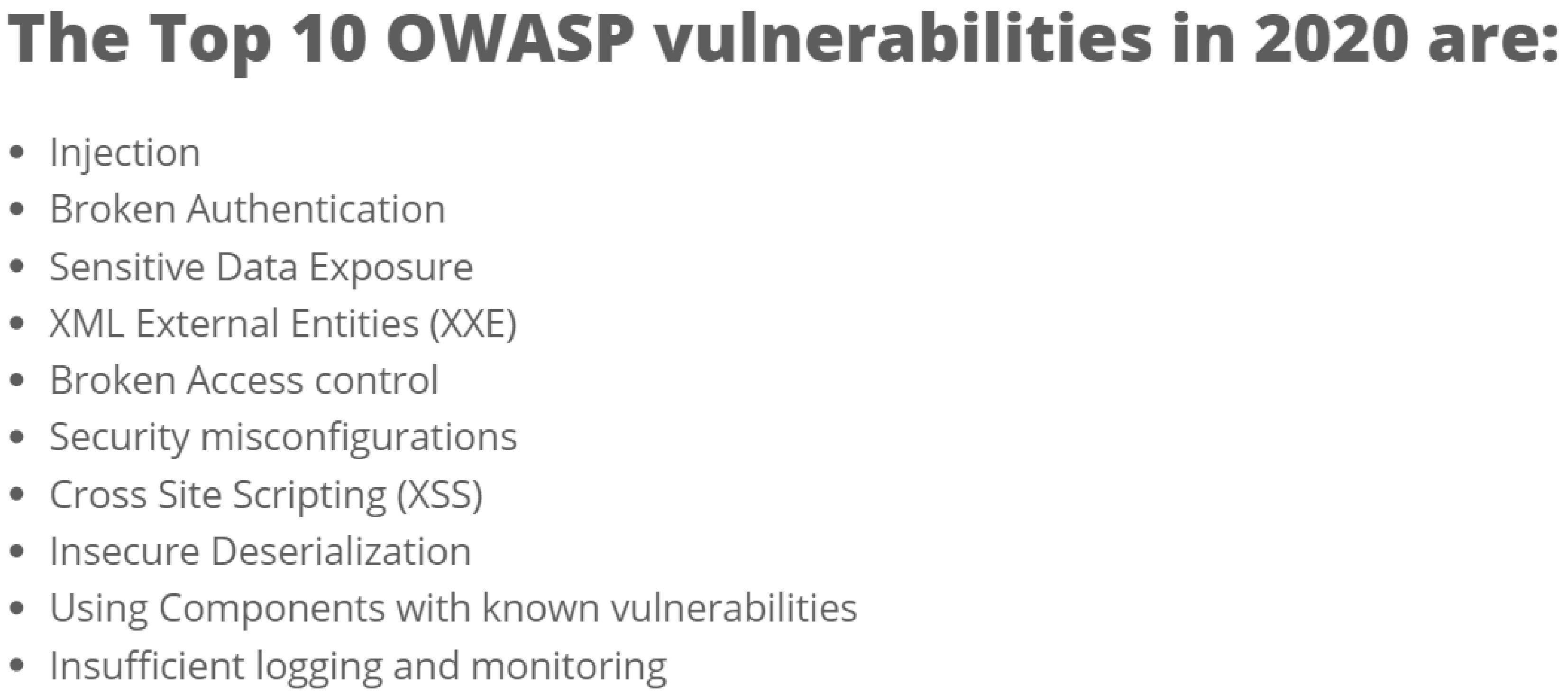
Mueller report: Russia hacked state databases and voting machine companies

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 ...



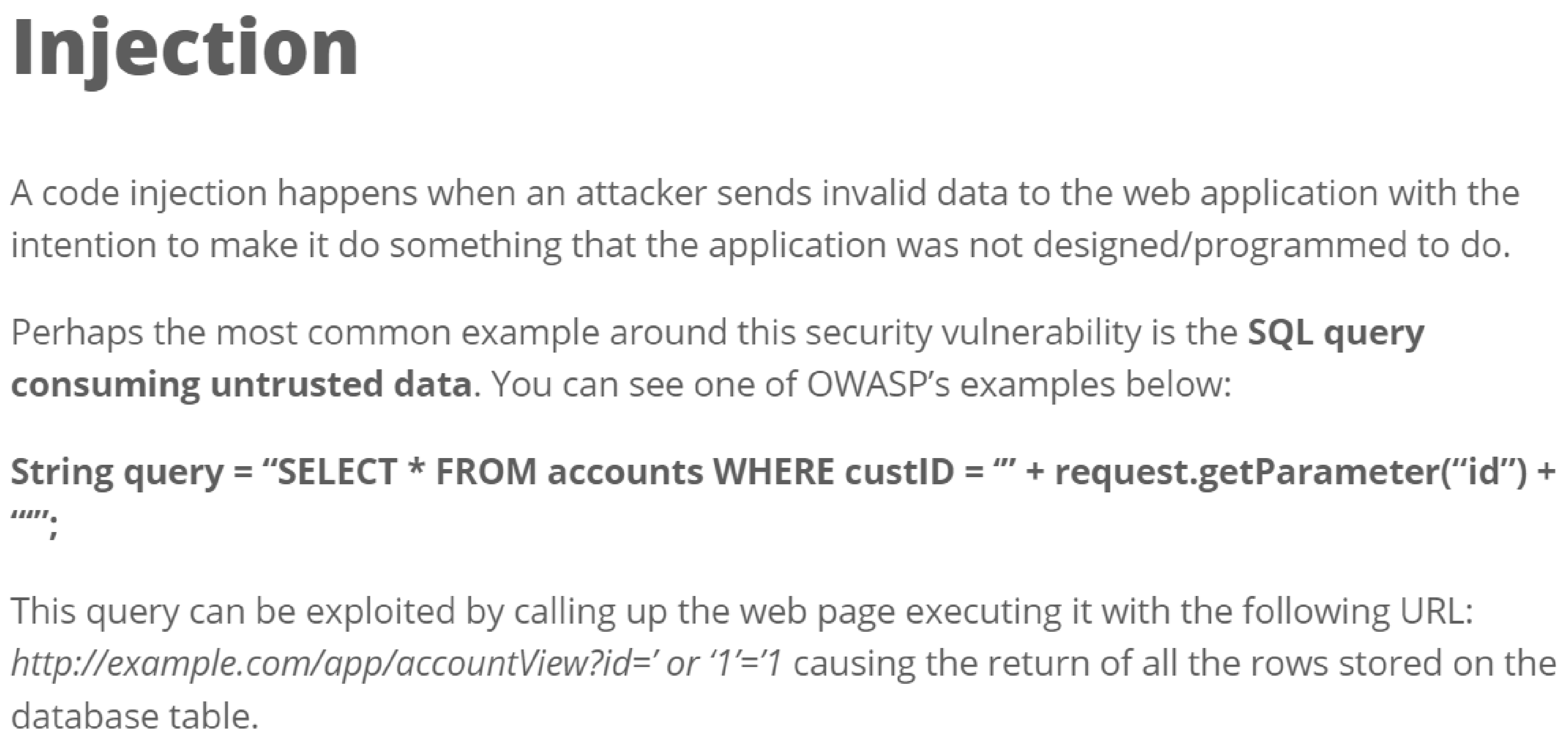
The Top 10 OWASP vulnerabilities in 2020 are:

- 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”) + “‘”;
```

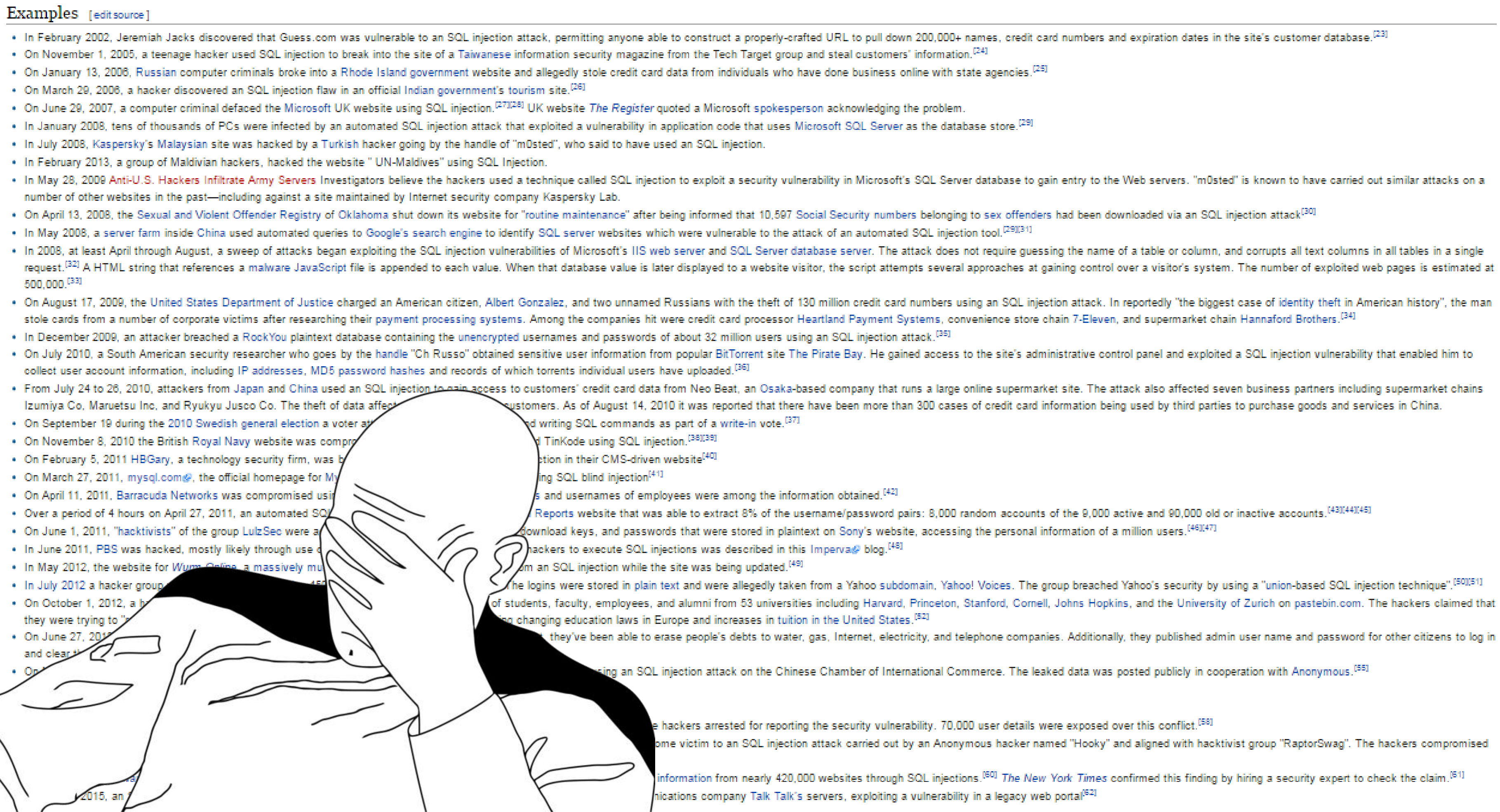
This query can be exploited by calling up the web page executing it with the following URL: *http://example.com/app/accountView?id=’ or ‘1’=’1* causing the return of all the rows stored on the database table.

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 affect 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 d writing SQL commands as part of a write-in vote.[37]
- On November 8, 2010 the British Royal Navy website was compro d TinKode using SQL injection.[38][39]
- On February 5, 2011 HBGary, a technology security firm, was b ction in their CMS-driven website[40]
- On March 27, 2011, mysql.com, the official homepage for My ing 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 om an SQL injection while the site was being updated.[49]
- In July 2012 a hacker group The 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 th
- On sing an SQL injection attack on the Chinese Chamber of International Commerce. The leaked data was posted publicly in cooperation with Anonymous.[55]

e hackers arrested for reporting the security vulnerability. 70,000 user details were exposed over this conflict.[58]

ome 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

Examples [edit source]

attack, permi mes, credit c
information se [24]
and allegedly agencies.[2
ourism site.[26]
[8] UK website
at exploited a e store.[29]
'm0sted", who
ection.
a technique c database to g
pany Kaspers
r "routine mai sex offende
ntify SQL serv tool.[29][31]

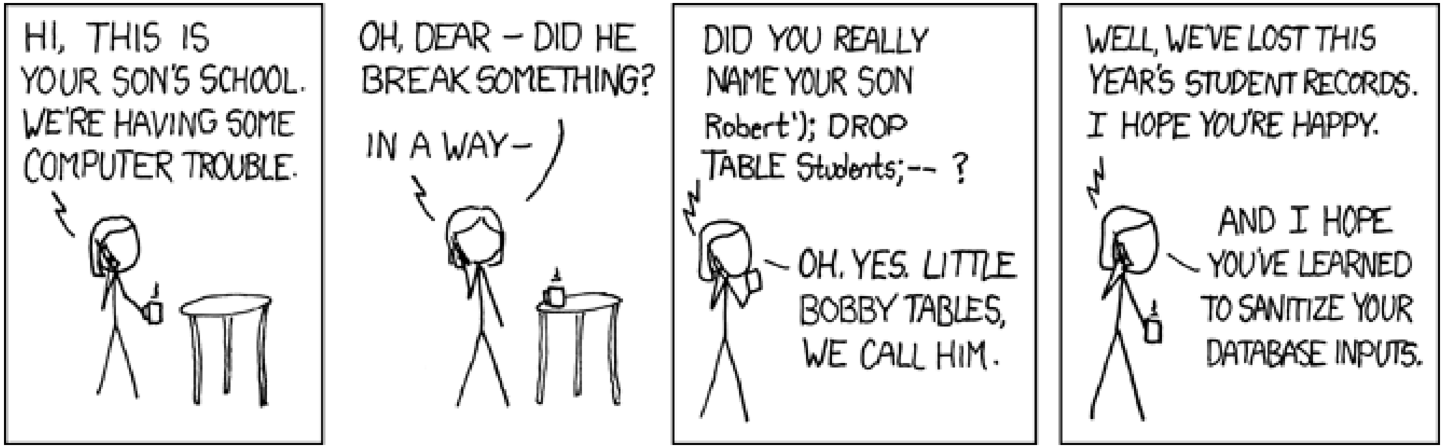
In 2008, at least April through August, a sweep of attacks began exploiting the SQL injection vulnerabilities of M require gues
request.[32] A HTML string that references a malware JavaScript file is appended to each value. When that data proaches at
500,000.[33]

rt Gonzalez, an g an SQL injection attack.
ems. Among the nience store chain 7-Eleven, and supermarket chain Hannaford Brothers.[34]
ed usernames
ed sensitive use the site's adm
which torrents i
to customers' line supermar
customers. As f credit card
y hand writing SQL commands as part of a write-in vote.[37]
named TinKode using SQL injection.[38][39]
QL injection in their CMS-driven website[40]
ker using SQL blind injection[41]
resses and usernames of employees were among the information obtained.[42]
band Reports website that was able to extract 8% of the username/password pairs: 8,000 random account

- On June 1, 2011, "hacktivists" of the group LulzSec were accused of using SQLI to steal coupons, download keys, and passwords that were stored in plaintext on Sony's website, accessing the personal i
- In June 2011, PBS was hack ; the full process use s described in this Imperva blog.[48]
- In May 2012, the website for game, was shut dow being updated.[49]
- In July 2012 a hacker group dentials from Yahoo! ere allegedly tak [50][51]
- On October 1, 2012, a hacke e personal records of from 53 univers med that
  they were trying to "raise awa education", bemoan increases in tui
- On June 27, 2013, hacker gr on Site.[53] They clai e's debts to wat to log in
  and clear their debts early mo r.[54]
- On November 4, 2013, hackt ed 71 Chinese gove ttack on the Chi
- On February 2, 2014, AVS TV oup called @deletes
- On February 21, 2014, United 15 account details le
- On February 21, 2014, Hacke l International after a rested for repo
- On March 7, 2014, officials a d that their Biomedic to an SQL inject omised
  personal details of 878 stude e leaked data on the
- In August 2014, Milwaukee-based computer security company Hold Security disclosed that it uncovered a theft of confidential information from nearly 420 [61]
- In October 2015, an SQL injection attack was used to steal the personal details of 156,959 customers from British telecommunications company Talk Talk's servers, exploiting a vulnerability in a legacy web portal[62]

SONY
YAHOO!
The Pirate Bay
Microsoft
MySQL®
7 ELEVEN®
ROYAL NAVY
HBGary
Detecting Tomorrow's Threats Today
Part of ManTech International Corporation

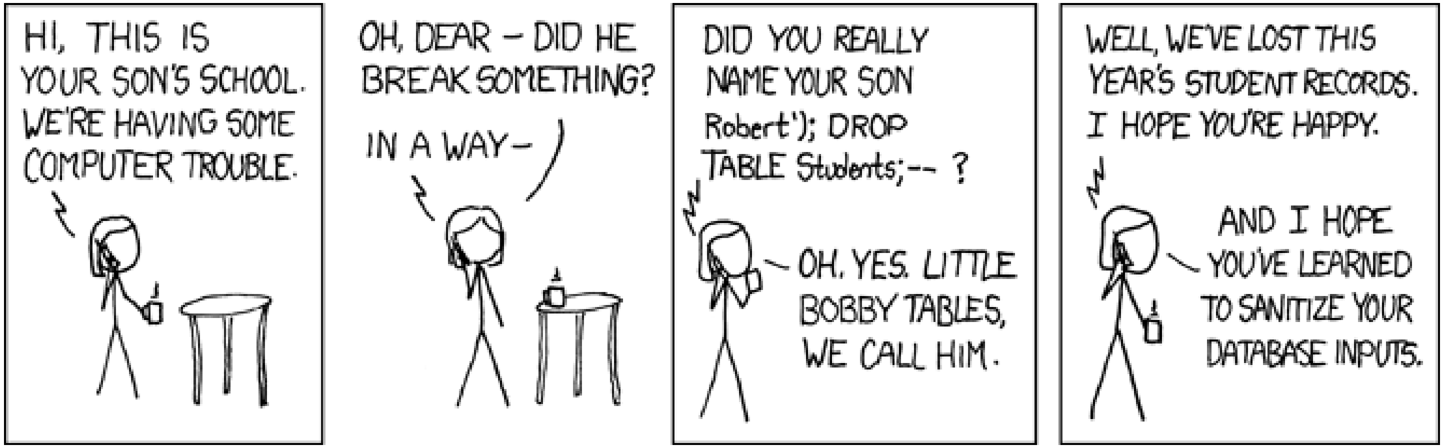
Inyección SQL


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

```
String consulta = "SELECT nota FROM Students WHERE name='"+input+"'";
ResultSet rs = statement.executeQuery(consulta);
```

¿Cómo podemos resolver el problema?

Inyección SQL: *¿escapar los strings?*


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

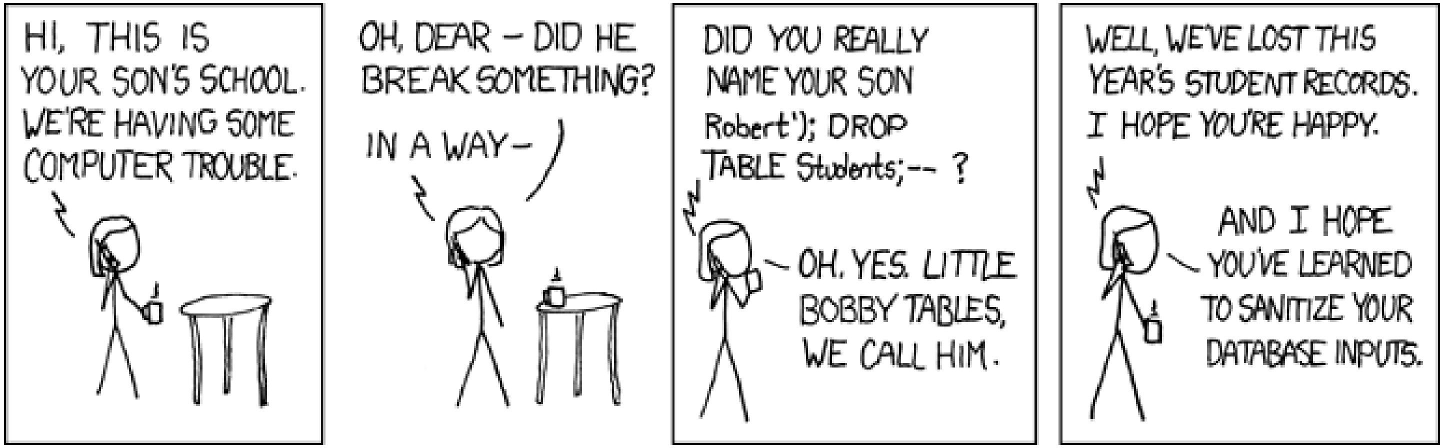
```
String consulta = "SELECT nota FROM Students WHERE name='"+input+"'";
ResultSet rs = statement.executeQuery(consulta);
```

¿Cómo podemos resolver el problema?

```
String consulta = "SELECT nota FROM Students WHERE name='"+escapar(input)+"'";
ResultSet rs = statement.executeQuery(consulta);
```

Mejor, pero sería complicado implementar la función escapar 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!*


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

```
String consulta = "SELECT nota FROM Students WHERE name='?'";
// donde ?  es un parámetro que reemplazaremos con la entrada del usuario
PreparedStatement ps = conn.prepareStatement(consulta);
ps.setString(1, input);
ResultSet rs = ps.executeQuery();
```

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!*

```
String consulta = "SELECT nota FROM Students WHERE name=?";

PreparedStatement ps = conn.prepareStatement(consulta);      // 1
ps.setString(1, input);                                      // 2
ResultSet rs = ps.executeQuery();                            // 3
```

```
// 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!*

```
String consulta = "SELECT nota FROM Students WHERE name=?";

PreparedStatement ps = conn.prepareStatement(consulta);      // 1
ps.setString(1, input);                                      // 2
ResultSet rs = ps.executeQuery();                            // 3
```

```
// 2 :  El sistema de bases de datos reemplaza el parametro en el plan
SELECT nota FROM Students WHERE name=?
                                    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)

Inyección SQL: *¡sentencias pre-compiladas!*

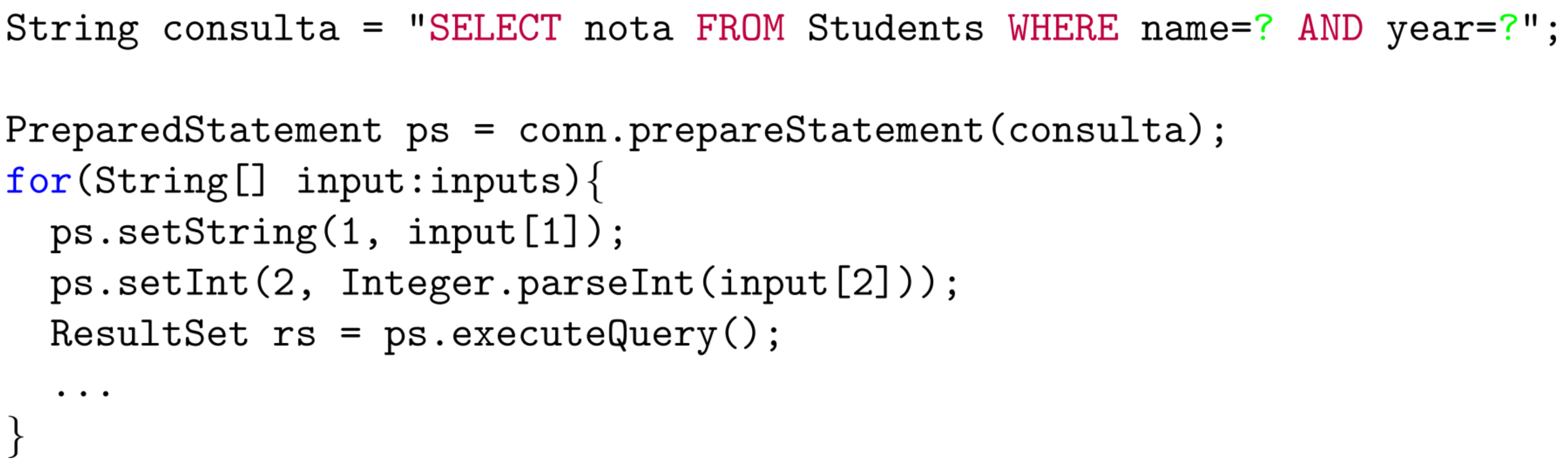
```
String consulta = "SELECT nota FROM Students WHERE name=?";

PreparedStatement ps = conn.prepareStatement(consulta);      // 1
ps.setString(1, input);                                       // 2
ResultSet rs = ps.executeQuery();                             // 3
```

```
// 3 :  El sistema de bases de datos ejecuta el plan
SELECT nota FROM Students WHERE name=?
                                     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

nota
3,7

Sentencias pre-compiladas

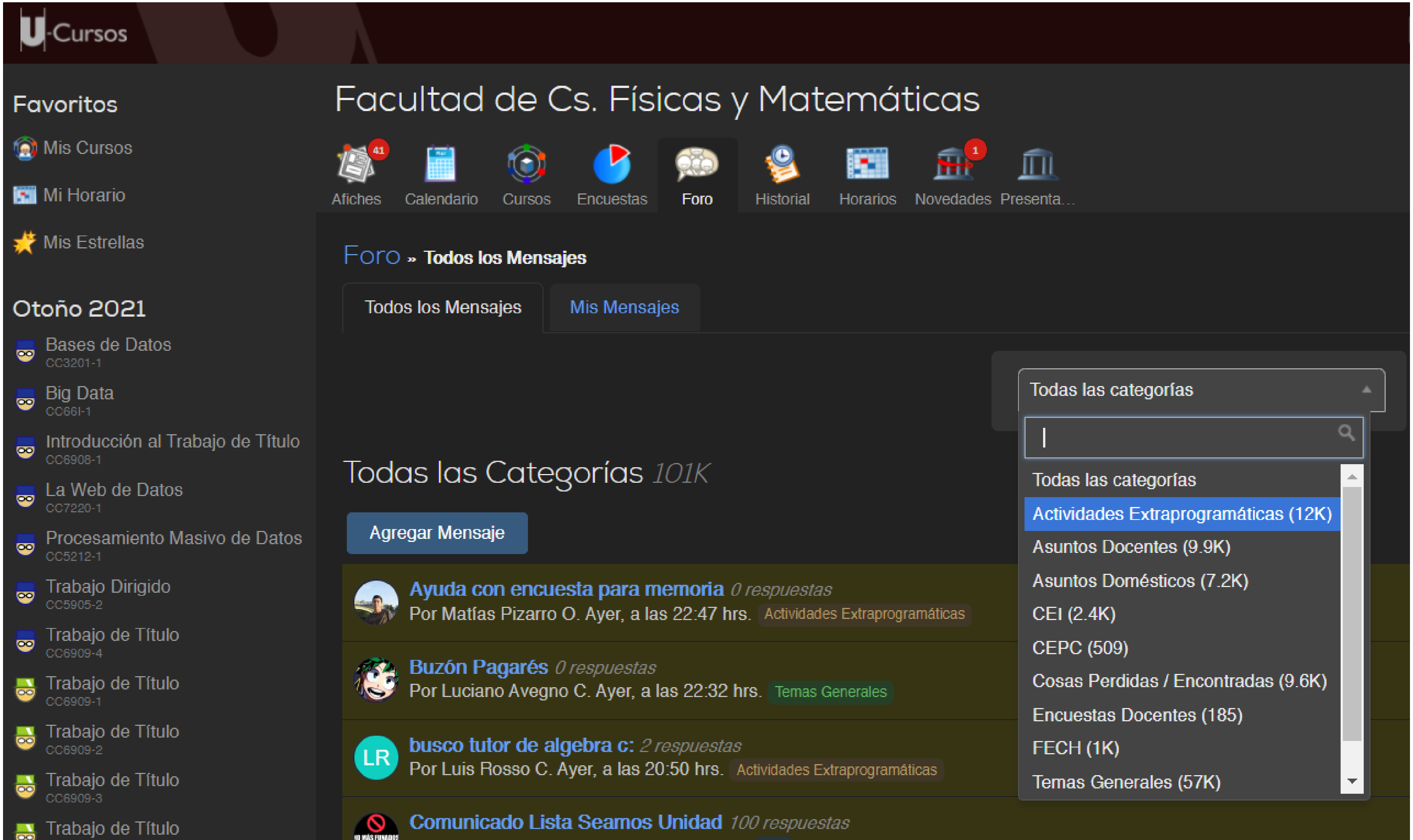


```
String consulta = "SELECT nota FROM Students WHERE name=? AND year=?";

PreparedStatement ps = conn.prepareStatement(consulta);
for(String[] input:inputs){
  ps.setString(1, input[1]);
  ps.setInt(2, Integer.parseInt(input[2]));
  ResultSet rs = ps.executeQuery();
  ...
}
```

Se puede reutilizar el `PreparedStatement` varias veces
(es más eficiente también: se compila la sentencia solo una vez)

Se puede tener varios parámetros con varios tipos

¡Interfaces restringidas de HTML no ayudan!



Bases de Datos CC3201-1
Big Data CC66I-1
Introducción al Trabajo de Título CC6908-1
La Web de Datos CC7220-1
Procesamiento Masivo de Datos CC5212-1
Trabajo Dirigido CC5905-2
Trabajo de Título CC6909-4
Trabajo de Título CC6909-1
Trabajo de Título CC6909-2
Trabajo de Título CC6909-3
Trabajo de Título

Facultad de Cs. Físicas y Matemáticas

Foro » **Todos los Mensajes**

Todas las Categorías *101K*

Ayuda con encuesta para memoria *0 respuestas*
Por Matías Pizarro O. Ayer, a las 22:47 hrs. Actividades Extraprogramáticas

Buzón Pagarés *0 respuestas*
Por Luciano Avegno C. Ayer, a las 22:32 hrs. Temas Generales

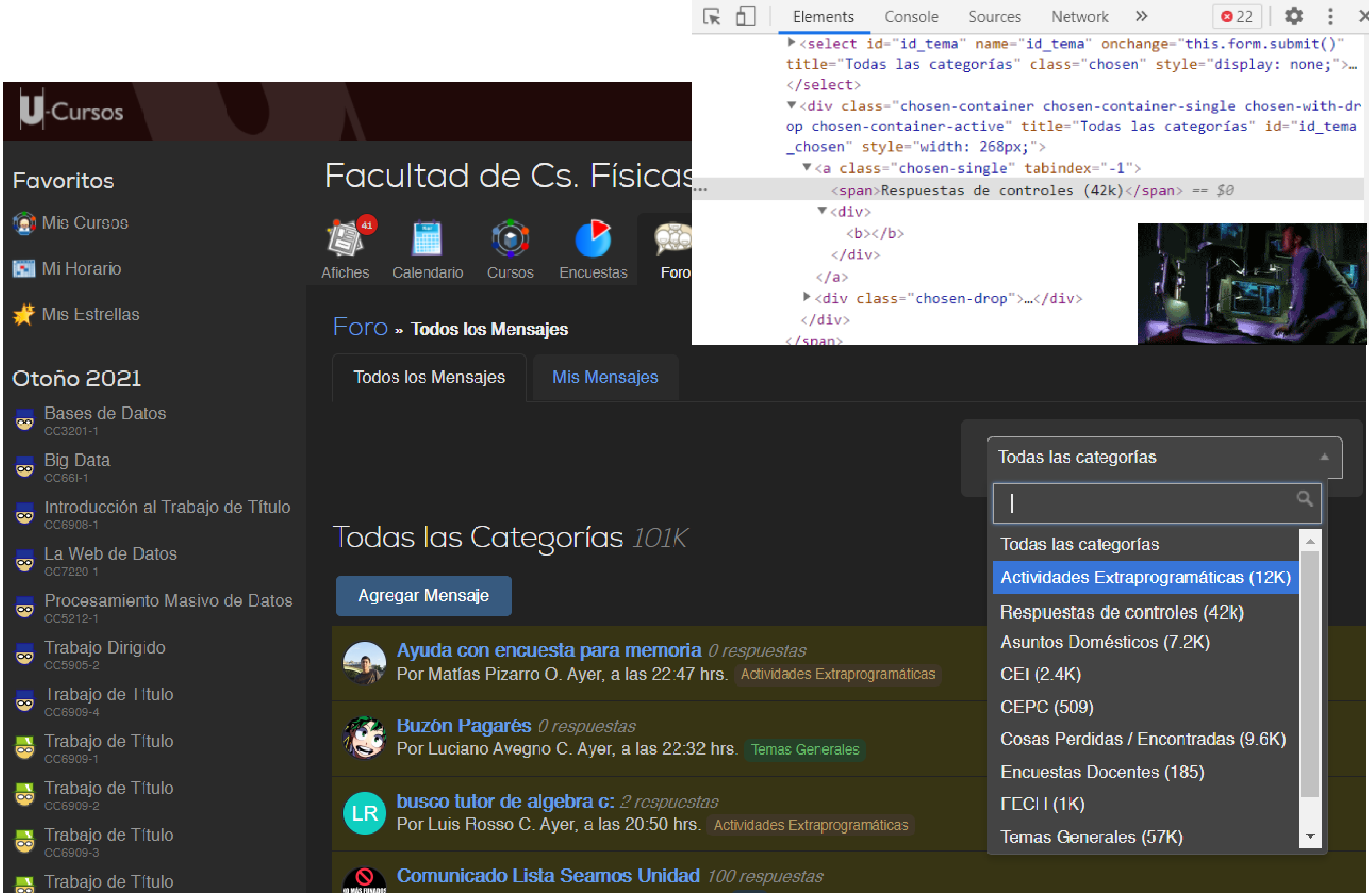
busco tutor de algebra c: *2 respuestas*
Por Luis Rosso C. Ayer, a las 20:50 hrs. Actividades Extraprogramáticas

Comunicado Lista Seamos Unidad *100 respuestas*

Todas las categorías

- Todas las categorías
- Actividades Extraprogramáticas (12K)
- Asuntos Docentes (9.9K)
- Asuntos Domésticos (7.2K)
- CEI (2.4K)
- CEPC (509)
- Cosas Perdidas / Encontradas (9.6K)
- Encuestas Docentes (185)
- FECH (1K)
- Temas Generales (57K)

¡Interfaces restringidas de HTML no ayudan!



```
▶<select id="id_tema" name="id_tema" onchange="this.form.submit()"
title="Todas las categorías" class="chosen" style="display: none;">…
</select>
▼<div class="chosen-container chosen-container-single chosen-with-dr
op chosen-container-active" title="Todas las categorías" id="id_tema
_chosen" style="width: 268px;">
  ▼<a class="chosen-single" tabindex="-1">
      <span>Respuestas de controles (42k)</span> == $0
    ▼<div>
        <b></b>
      </div>
    </a>
  ▶<div class="chosen-drop">…</div>
  </div>
</span>
```

Ayuda con encuesta para memoria *0 respuestas*
Por Matías Pizarro O. Ayer, a las 22:47 hrs. Actividades Extraprogramáticas

Buzón Pagarés *0 respuestas*
Por Luciano Avegno C. Ayer, a las 22:32 hrs. Temas Generales

busco tutor de algebra c: *2 respuestas*
Por Luis Rosso C. Ayer, a las 20:50 hrs. Actividades Extraprogramáticas

Comunicado Lista Seamos Unidad *100 respuestas*

Todas las categorías

- Todas las categorías
- Actividades Extraprogramáticas (12K)
- Respuestas de controles (42k)
- Asuntos Domésticos (7.2K)
- CEI (2.4K)
- CEPC (509)
- Cosas Perdidas / Encontradas (9.6K)
- Encuestas Docentes (185)
- FECH (1K)
- Temas Generales (57K)

¡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

Preguntas?

CATS: ALL YOUR BASE ARE BELONG
TO US.