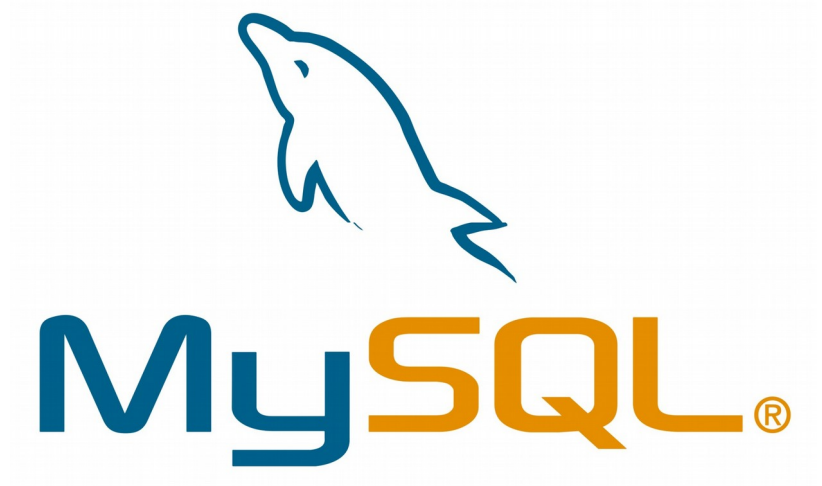


# Gestió eficient d'una Perruqueria Documentació



Itiel Luque Díaz  
EPSEVG 2018-2019 (Primavera)  
Disseny i Administració de Bases de Dades (Jordi Esteve)

# Index

• <b>Introducció</b>	
• Enunciat del problema.....	3
• Glossari.....	4
• <b>Esquema relacional</b>	
• Escrit.....	5
• Visual.....	9
• <b>UML</b>	
• Diagrama.....	10
• Restriccions semàntiques.....	11
• <b>Optimització de consultes</b>	
• SELECT.....	12
• INSERT.....	13
• UPDATE.....	14
• DELETE.....	15
• <b>Optimitzacions SGBD</b>	
• MySQL.....	16
• Connexions simultànies (optimització).....	19
• <b>Tecnologies emprades</b>	
• Frontend.....	22
• Scripts.....	23
• <b>Conclusions</b>	
• Què he après.....	29

## Introducció

### Enunciat del problema

Una perruqueria es vol modernitzar i vol llençar a la brossa la típica agenda on s'anoten per a una jornada laboral, en un dia concret els clients, les franges horàries i el dia de la jornada. Ens ha demanat que vol saber en tot moment quins clients té en el dia d'avui, és a dir, els clients que tenen cita.

Per fer-ho possible, el perruquer ens ha comunicat que treballa en diferents intervals, per exemple, comença ben d'hora, fa un descans d'una hora per esmorçar i després torna a treballar fins l'hora de dinar. Després fa un altre descans i torna a la tarda fent un altre interval.

El perruquer estima que els intervals es poden dividir en slots de 15 minuts tenint en compte els marges de preparació del client, aclaracions del servei i pagament més generació del ticket del client.

Les franges es determinaran segons la quantitat de serveis que demana el client. Cada servei requereix de diferents slots. Els serveis ens els dona el mateix perruquer. Per exemple, pot venir una persona i demanar un tall només i una d'altre que en vol un tall de barba, tenyir-se'n el cabell i a més un pentinat, en qualsevol cas, la cita tindrà com a duració des de l'hora que especifica el client, sempre que es pugui reservar la quantitat d'slots necessaris per a tots els serveis plegats, fins:  $\text{quantitat d'slots} * \text{temps d'slot}$ .

El client pot indicar a quin hora vol reservar una cita. Una cita tindrà slots consecutius i no pot creuar-se amb una d'altre, són seqüencials i no concurrents.

Es vol contemplar les persones que no coneixen les tecnologies i que es passen per l'establiment a demanar hora, en aquest cas, el mateix perruquer enregistrarà quin dia i hora els hi dona als clients.

Hi ha clients que sovint no venen o arriben tard, els quals perdran la reserva i els slots que tenia reservats quedaran marcats lliure, en aquests casos, es vol aprofitar les franges mortes, per a això, s'oferirà el primer slot a persones que necessiten un servei exprés tenint un descompte. De tant en tant es vol recompensar als clients fidels, volen muntar un programa de punts que guanyarà el client per haver demanat serveis. S'oferiran diferents punts per a cada servei i que aquests poden variar en el temps, per exemple cada mes.

Els punts del client es descomptaran en el pròxim tiquet, i, per a cada servei on hagi acumulat la quantitat suficient de punts en cites anteriors, aquests tindran cost zero en euros.

## Glossari

- **CITA:** Dia i hora en que el client s'ha de presentar cinc minuts abans per a que el perruquer li faci els serveis sol·licitats.
- **SERVEI:** Acció que pot demanar el client per a que el perruquer li realitzi en l'establiment en la cita assignada.
- **PUNTS:** Nombre natural que pot guanyar un client al pagar un servei i hi ha un d'altre que té un servei, especificat com a cost, i per tant, se li descompta al client si té tants punts com punts de cost requereix el servei, llavors el servei passa a tenir cost zero en euros.
- **TICKET:** Paper on s'especifica el preu total del serveis, els punts acumulats, els punts utilitzats, hora de la generació del ticket i data i hora de la cita.
- **INTERVAL:** Rang d'hores seqüencial i no commutable amb un altre rang vigent en el quals el perruquer dona servei a les cites.
- **SLOT:** Temps en el que el perruquer realitzarà els serveis repartints dins de l'interval. Els serveis demanats d'un client en una mateixa cita es fan seqüencialment.
- **T\_SLOT:** Quantitat de minuts que vol dedicar el perruquer als slots.
- **N\_SLOT:** Quantitat d'slots que necessita un servei per poder realitzar-se.
- **SLOTS EN UN INTERVAL:**  $\frac{(hf - hi) * 60}{ts}$ 
  - => hf: hora final interval
  - => hi: hora inici interval
  - => ts: temps de l'slots en minuts
  - => 60 minuts una hora.
- **COPS\_FALLIDA:** Nombre natural que indica la quantitat de vegades que el client no ha assistit a una cita.

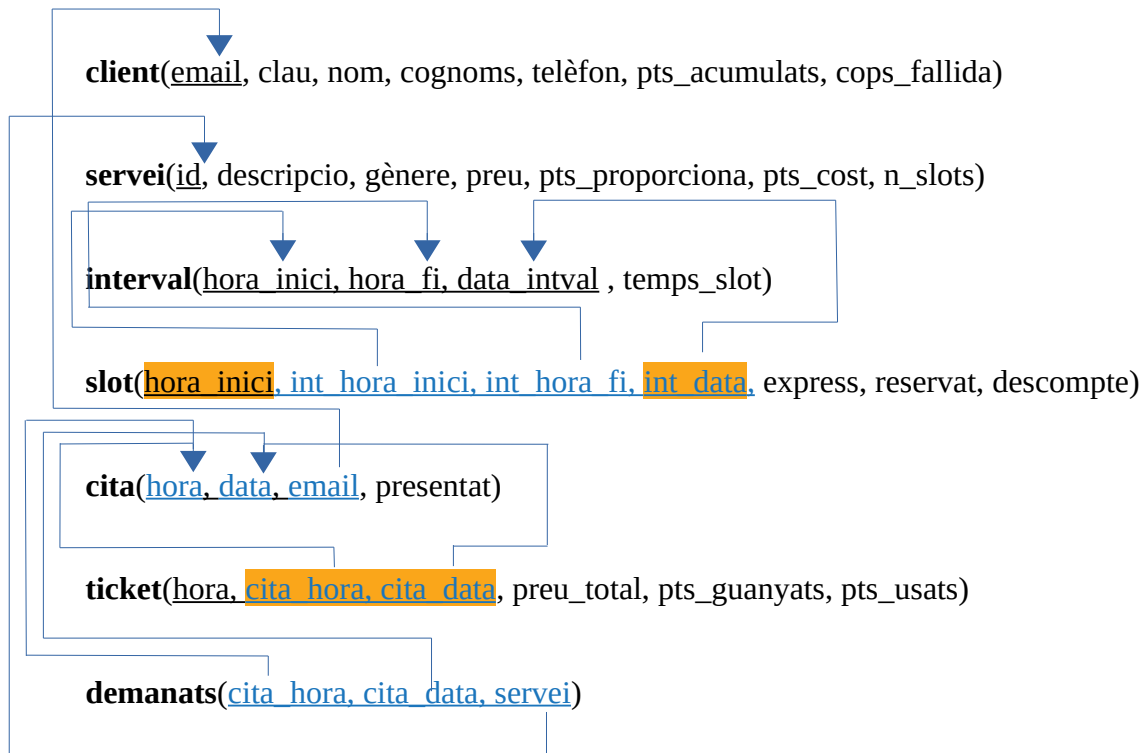
# Esquema relacional

## Escrit

PRIMARY KEY

FOREIGN KEY

UNIQUE



Les polítiques per defecte per a les claus de les taules **slot** i **cita** són **ON UPDATE CASCADE** i **ON DELETE CASCADE**, perquè;

- Els slots depenen dels intervals que són molt específics (en temps), si es modifica un interval, l'slot ha de saber el canvi i si s'elimina, l'slot perd el sentit.
- La clau de l'usuari, això implica que aquest no vol estar enregistrat i per tant probablement no vindrà a la cita.

Les polítiques per defecte per a les claus de les taules **demanats**, **ticket** i **cita** són **ON UPDATE CASCADE** i **ON DELETE NO ACTION**, perquè;

- Si un servei es modifica, la taula de demanats queda cuixa i si s'esborra i un client té una cita amb aquests serveis, se'ls hi farà.
- En el cas dels tickets, es volen mantenir tot i que s'esborrin les cites per poder fer estadístiques a nivell de negoci.
- En els cas de les claus que apunten a l'slot, si s'esborra i un usuari té cita o va tindre, es vol mantenir per poder saber quan ha de venir o si va venir i es va presentar per analitzar el model de negoci.

```
CREATE DATABASE IF NOT EXISTS perruqueria DEFAULT CHARACTER SET latin1
COLLATE latin1_swedish_ci;
USE perruqueria;
```

```
CREATE TABLE IF NOT EXISTS cita (
  c_hora time NOT NULL,
  c_data date NOT NULL,
  email varchar(200) NOT NULL,
  presentat tinyint(1) NOT NULL,
  PRIMARY KEY (c_hora,c_data,email),
  KEY cita_ibfk_3 (c_data),
  KEY email (email),
  KEY c_hora (c_hora,c_data)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `client` (
  email varchar(200) NOT NULL,
  clau varchar(128) NOT NULL,
  nom varchar(20) NOT NULL,
  cognoms varchar(50) NOT NULL,
  telefon varchar(30) NOT NULL,
  pts_acumulats int(6) NOT NULL DEFAULT '0',
  cops_fallida int(2) NOT NULL DEFAULT '0',
  PRIMARY KEY (email)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS demanats (
  cita_hora time NOT NULL,
  cita_data date NOT NULL,
  servei int(11) NOT NULL,
  PRIMARY KEY (cita_hora,cita_data,servei),
  KEY demanats_ibfk_1 (servei)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS intval (  
    hora_inici time NOT NULL,  
    hora_fi time NOT NULL,  
    data_intval date NOT NULL,  
    temps_slot int(2) NOT NULL,  
    PRIMARY KEY (hora_inici,hora_fi,data_intval)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS servei (  
    id int(11) NOT NULL,  
    descripcio varchar(50) NOT NULL,  
    genere varchar(1) NOT NULL,  
    tipus varchar(30) NOT NULL,  
    preu float(5,2) NOT NULL DEFAULT '0.00',  
    pts_proporciona int(3) NOT NULL,  
    pts_cost int(3) NOT NULL,  
    n_slots int(2) NOT NULL,  
    PRIMARY KEY (id)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS slot (  
    hora_inici time NOT NULL,  
    int_hora_inici time NOT NULL,  
    int_hora_fi time NOT NULL,  
    int_data date NOT NULL,  
    express tinyint(1) NOT NULL DEFAULT '0',  
    reservat tinyint(1) NOT NULL DEFAULT '0',  
    descompte int(3) NOT NULL DEFAULT '0',  
    PRIMARY KEY (hora_inici,int_hora_inici,int_hora_fi,int_data) USING BTREE,  
    UNIQUE KEY hora_inici (hora_inici,int_data),  
    KEY int_hora_inici (int_hora_inici,int_hora_fi,int_data) USING BTREE  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS ticket (  
    hora time NOT NULL,  
    cita_hora time NOT NULL,  
    cita_data date NOT NULL,  
    preu_total int(4) NOT NULL,  
    pts_guanyats int(4) NOT NULL,  
    pts_usats int(4) NOT NULL,  
    PRIMARY KEY (hora,cita_hora,cita_data),  
    UNIQUE KEY cita_hora (cita_hora,cita_data)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
ALTER TABLE cita
```

```
    ADD CONSTRAINT cita_ibfk_2 FOREIGN KEY (c_hora,c_data) REFERENCES slot  
(hora_inici, int_data) ON DELETE NO ACTION ON UPDATE CASCADE,
```

```
    ADD CONSTRAINT cita_ibfk_3 FOREIGN KEY (email) REFERENCES client (email) ON  
DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE demanats
```

```
    ADD CONSTRAINT demanats_ibfk_1 FOREIGN KEY (servei) REFERENCES servei (id) ON  
DELETE NO ACTION ON UPDATE CASCADE,
```

```
    ADD CONSTRAINT demanats_ibfk_2 FOREIGN KEY (cita_hora,cita_data) REFERENCES  
cita (c_hora, c_data) ON DELETE NO ACTION ON UPDATE CASCADE;
```

```
ALTER TABLE slot
```

```
    ADD CONSTRAINT slot_ibfk_2 FOREIGN KEY (int_hora_inici) REFERENCES intval  
(hora_inici) ON DELETE CASCADE ON UPDATE CASCADE,
```

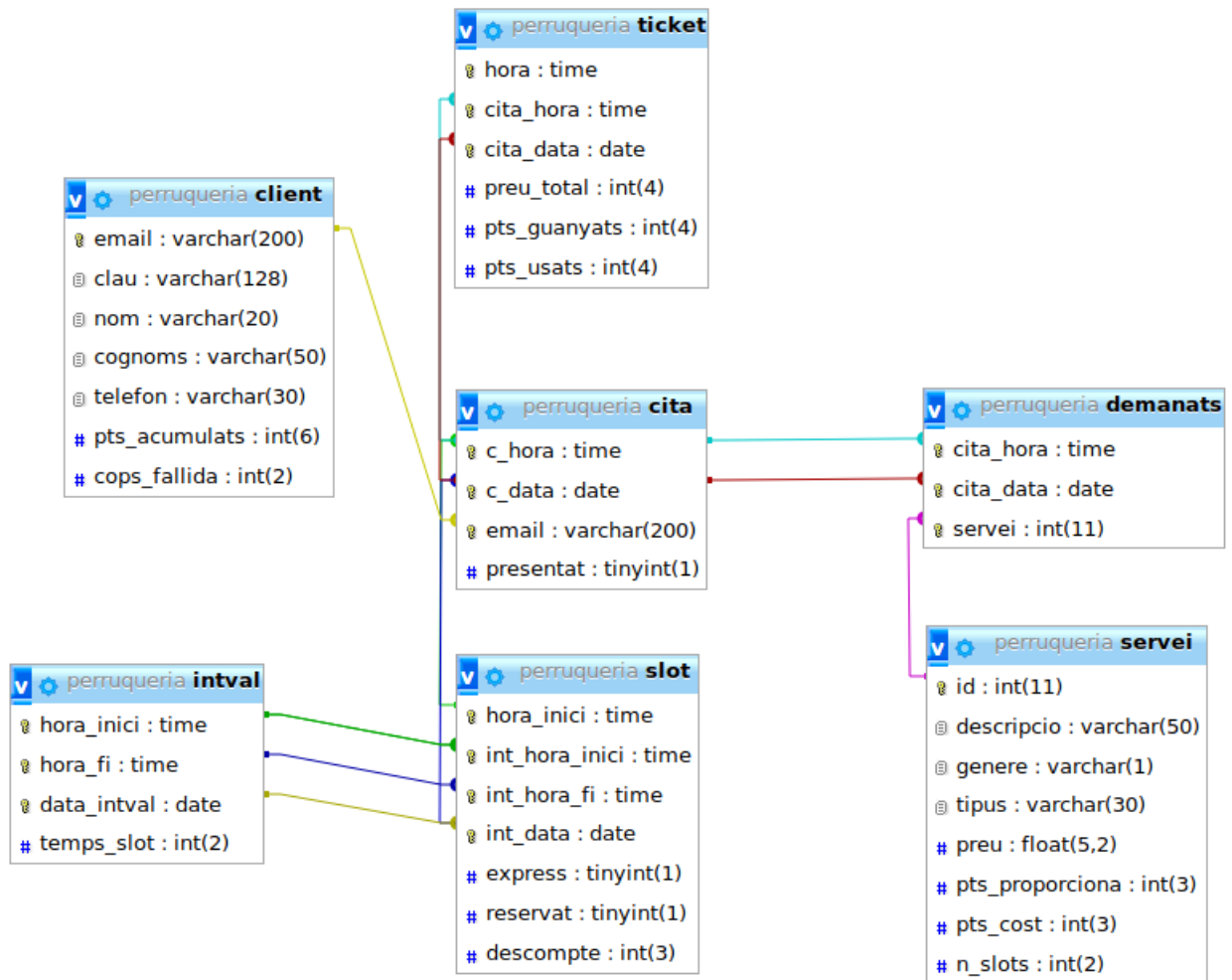
```
    ADD CONSTRAINT slot_ibfk_3 FOREIGN KEY (int_hora_inici,int_hora_fi,int_data)  
REFERENCES intval (hora_inici, hora_fi, data_intval);
```

```
ALTER TABLE ticket
```

```
    ADD CONSTRAINT ticket_ibfk_1 FOREIGN KEY (cita_hora,cita_data) REFERENCES cita  
(c_hora, c_data) ON DELETE NO ACTION ON UPDATE CASCADE;
```

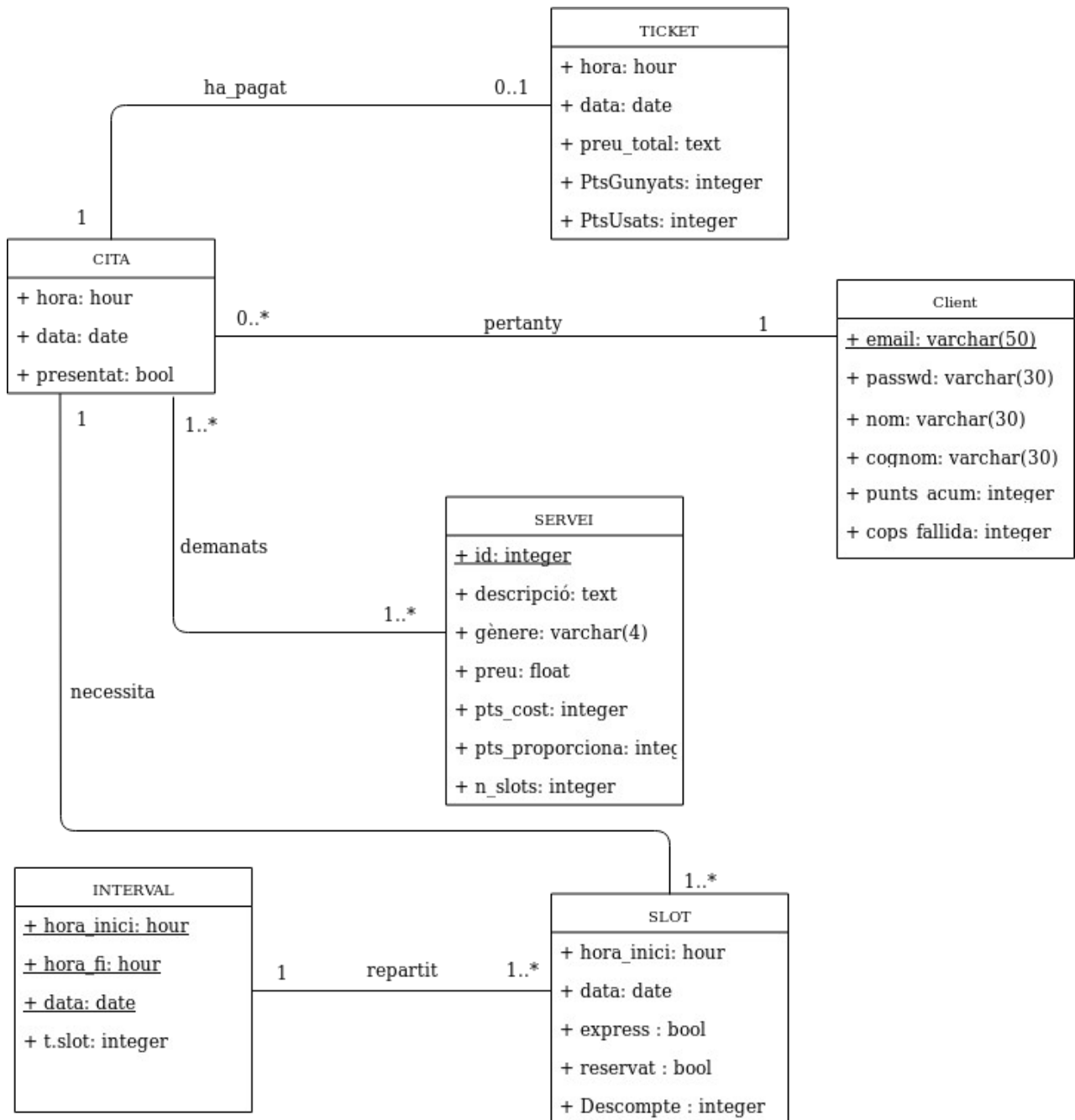


## Visual



# UML

## Diagrama



## Restriccions semàntiques

**RS:** Un tall express només es crearà si un client amb cita no s'ha presentat abans de l'hora indicada. És crearà un tall express per al primer slot de la cita.

**RS:** El bool express s'activarà quan es cancel·li una cita per manca d'assistència o quan el següent slot de l'actual estigui sense cita. Llavors se li aplicarà el descompte en cas que algun client ho reservi en la cita.

**RS:** Un ticket no pot estar lligat amb una cita si no està lligada a un o més slots.

**RS:** Un client conegut no pot obtenir a tenir premis si  $\text{punts\_acumulats} < \text{servei.punts\_regal}$ .

**RS:** Una cita ha de tenir tants slots consecutius com la suma de slots de tots els serveis demanats.

**RS:** La cita només necessita saber el primer slot a la qual està relacionada, la resta es sap sumant la quantitat d'slots dels serveis demanats en aquella cita.

# Optimització de consultes

## SELECT

Su consulta se ejecutó con éxito.

```
EXPLAIN EXTENDED SELECT * FROM servei
```

+ Opciones

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	servei	NULL	ALL	NULL	NULL	NULL	NULL	17	100.00	NULL

Su consulta se ejecutó con éxito.

```
EXPLAIN EXTENDED SELECT email FROM client WHERE email = 'julia75@yahoo.com'
```

+ Opciones

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	client	NULL	const	PRIMARY	PRIMARY	202	const	1	100.00	Using index

⚠ La selección actual no contiene una columna única. La edición de la grilla y los enlaces de copiado, eliminación y edición no están dis

Su consulta se ejecutó con éxito.

```
EXPLAIN SELECT TIME_FORMAT(hora_inicio,"%H:%i") as temps FROM slot WHERE int_data = CURRENT_DATE AND NOT reservat
```

+ Opciones

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	slot	NULL	ALL	NULL	NULL	NULL	NULL	15344	1.00	Using where

Su consulta se ejecutó con éxito.

```
EXPLAIN SELECT cl.nom, cl.cognoms, cl.email, TIME_FORMAT(ci.c_hora,"%H:%i") as hora FROM client cl, cita ci WHERE cl.email = ci.email AND ci.c_data = CURRENT_DATE AND ci.c_hora >= CURRENT_TIME AND NOT ci.  
[ Editar en línea ] [ Editar ] [ Omitir la explicación ]
```

+ Opciones

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	ci	NULL	range	PRIMARY,cita_ibfk_3,email,c_hora	cita_ibfk_3	6	NULL	1	10.00	Using index condition; Using where
1	SIMPLE	cl	NULL	eq_ref	PRIMARY	PRIMARY	202	perruqueria.ci.email	1	100.00	NULL

En aquesta join veiem que es podria optimitzar una mica. La resta fan servir l'índex adientment.

# INSERT

Su consulta se ejecutó con éxito.

```
EXPLAIN INSERT INTO intval VALUES ('10:30','14:40', '2025-08-20', 15)
```

+ Opciones

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	INSERT	intval	NULL	ALL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Su consulta se ejecutó con éxito.

```
EXPLAIN INSERT INTO client VALUES ('itiel@hotmail.com', 'asdasd', 'Itiel', 'Luque Díaz', '23423423', 0, 0)
```

+ Opciones

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	INSERT	client	NULL	ALL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Su consulta se ejecutó con éxito.

```
EXPLAIN INSERT INTO slot VALUES ('10:30', '12:30', '14:30', '2019-09-20', 0, 0, 0)
```

+ Opciones

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	INSERT	slot	NULL	ALL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Amb els inserts no té cap mena de problema, ja que sap gestionar-se correctament fent servir les claus primàries que són els indexos en B+.

# UPDATE

Su consulta se ejecutó con éxito.

```
EXPLAIN UPDATE client SET email = 'testing' WHERE email = 'zabalaxavier@gmail.com'
```

+ Opciones

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	UPDATE	client		NULL range	PRIMARY	PRIMARY	202	const	1	100.00	Using where

Su consulta se ejecutó con éxito.

```
EXPLAIN UPDATE slot SET express=1 WHERE hora_inici = '12:15:00'
```

+ Opciones

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	UPDATE	slot		NULL range	PRIMARY,hora_inici	PRIMARY	3	const	312	100.00	Using where

En aquest cas no s'ha fet servir cap clau i l'analitzador aconsella una.

Su consulta se ejecutó con éxito.

```
EXPLAIN UPDATE slot SET express=1 WHERE hora_inici = '12:15:00' AND int_data = '2020-02-13'
```

+ Opciones

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	UPDATE	slot		NULL index_merge	PRIMARY,hora_inici	hora_inici,PRIMARY	6,3	NULL	1	100.00	Using intersect(hora_inici,PRIMARY); Using where

Mateix exemple, però, fent servir un índex.

# DELETE

En aquest cas, l'únic que es tractarà d'analitzar és d'esborrar intervals, ja que no s'esborrar res més i només s'esborren quan es produeix un error a l'introduir dades.

Su consulta se ejecutó con éxito.

```
EXPLAIN DELETE FROM intval WHERE hora_inici = '15:00:00'
```

+ Opciones

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	DELETE	intval	NULL	range	PRIMARY	PRIMARY	3	const	168	100.00	Using where

En aquesta consulta no s'ha especificat cap clau.

Su consulta se ejecutó con éxito.

```
EXPLAIN DELETE FROM intval WHERE hora_inici = '15:00:00' AND data_intval = '2021-06-01'
```

+ Opciones

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	DELETE	intval	NULL	range	PRIMARY	PRIMARY	3	const	168	100.00	Using where

En aquesta passa igual tot i afegir una dada més, no es concissa.

Su consulta se ejecutó con éxito.

```
EXPLAIN DELETE FROM intval WHERE hora_inici = '15:00:00' AND hora-fi = '19:00:00' AND data_intval = '2021-06-01'
```

+ Opciones

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	DELETE	intval	NULL	range	PRIMARY	PRIMARY	9	const,const,const	1	100.00	Using where

Aquesta ultima ha estat molt més eficient, ja que se li ha indicat l'índex complet, a més, que el resultat esperar no era el mateix que les anteriors, ja que no estava ben formulada la query.

# Optimitzacions SGBD

## MySQL

Per poder optimitzar el sistema gestor, cal instal·lar el paquet mysqltuner executant la comanda `sudo apt-get install mysqltuner`.

Després cal executar-lo i afegir les recomendacions en el fitxer `/etc/mysql/mysql.conf.d/mysqld.conf`

```
itiel@X550JX:~$ mysqltuner
>> MySQLTuner 1.7.2 - Major Hayden <major@mhtx.net>
>> Bug reports, feature requests, and downloads at http://mysqltuner.com/
>> Run with '--help' for additional options and output filtering

[+] Skipped version check for MySQLTuner script
Please enter your MySQL administrative login: itiel
Please enter your MySQL administrative password: [OK] Currently running supported MySQL version 5.7.26-0ubuntu0.18.04.1
[OK] Operating on 64-bit architecture

----- Log file Recommendations -----
[+] Log file: /var/log/mysql/error.log(104B)
[OK] Log file /var/log/mysql/error.log exists
[OK] Log file /var/log/mysql/error.log is readable.
[OK] Log file /var/log/mysql/error.log is not empty
[OK] Log file /var/log/mysql/error.log is smaller than 32 Mb
[OK] /var/log/mysql/error.log doesn't contain any warning.
[OK] /var/log/mysql/error.log doesn't contain any error.
[+] 0 start(s) detected in /var/log/mysql/error.log
[+] 0 shutdown(s) detected in /var/log/mysql/error.log

----- Storage Engine Statistics -----
[+] Status: +ARCHIVE +BLACKHOLE +CSV -FEDERATED +InnoDB +MEMORY +MRG_MYISAM +MyISAM +PERFORMANCE_SCHEMA
[+] Data in InnoDB tables: 5M (Tables: 27)
[OK] Total fragmented tables: 0

----- Security Recommendations -----
[OK] There are no anonymous accounts for any database users
[!] User 'root@localhost' has no password set.
[+] There are 612 basic passwords in the list.

----- CVE Security Recommendations -----
[OK] NO SECURITY CVE FOUND FOR YOUR VERSION

----- Performance Metrics -----
[+] Up for: 3h 6m 9s (27K q [2.431 qps], 2K conn, TX: 2M, RX: 2M)
[+] Reads / Writes: 23% / 77%
[+] Binary logging is disabled
[+] Physical Memory : 7.7G
[+] Max MySQL Memory : 352.4M
[+] Other process memory: 2.0G
[+] Total buffers: 192.0M global + 1.1M per thread (151 max threads)
[+] P_S Max memory usage: 72B
[+] Galera GCache Max memory usage: 0B
[OK] Maximum reached memory usage: 195.2M (2.48% of installed RAM)
[OK] Maximum possible memory usage: 352.4M (4.48% of installed RAM)
[OK] Overall possible memory usage with other process is compatible with memory available
[OK] Slow queries: 0% (0/27K)
[OK] Highest usage of available connections: 1% (3/151)
[OK] Aborted connections: 0.05% (1/2064)
[!] name resolution is active : a reverse name resolution is made for each new connection and can reduce performance
[!] Query cache may be disabled by default due to mutex contention.
```



```

[!] Query cache efficiency: 0.0% (0 cached / 4K selects)
[OK] Query cache prunes per day: 0
[OK] Sorts requiring temporary tables: 0% (2 temp sorts / 1K sorts)
[OK] No joins without indexes
[OK] Temporary tables created on disk: 17% (618 on disk / 3K total)
[OK] Thread cache hit rate: 99% (3 created / 2K connections)
[OK] Table cache hit rate: 82% (410 open / 499 opened)
[OK] Open file limit used: 0% (37/5K)
[OK] Table locks acquired immediately: 100% (376 immediate / 376 locks)

----- Performance schema -----
[+] Memory used by P_S: 72B
[+] Sys schema is installed.

----- ThreadPool Metrics -----
[+] ThreadPool stat is disabled.

----- MyISAM Metrics -----
[!] Key buffer used: 18.3% (3M used / 16M cache)
[OK] Key buffer size / total MyISAM indexes: 16.0M/43.0K
[!] Read Key buffer hit rate: 93.5% (108 cached / 7 reads)

----- InnoDB Metrics -----
[+] InnoDB is enabled.
[+] InnoDB Thread Concurrency: 0
[OK] InnoDB File per table is activated
[OK] InnoDB buffer pool / data size: 128.0M/5.4M
[!] Ratio InnoDB log file size / InnoDB Buffer pool size (75 %): 48.0M * 2/128.0M should be equal 25%
[OK] InnoDB buffer pool instances: 1
[+] Number of InnoDB Buffer Pool Chunk : 1 for 1 Buffer Pool Instance(s)
[OK] InnoDB buffer pool size aligned with InnoDB buffer pool chunk size & InnoDB buffer pool instances
[OK] InnoDB Read buffer efficiency: 99.95% (1127339 hits/ 1127888 total)
[!] InnoDB Write Log efficiency: 85.04% (7691 hits/ 9044 total)
[OK] InnoDB log waits: 0.00% (0 waits / 16735 writes)

----- AriaDB Metrics -----
[+] AriaDB is disabled.

----- TokuDB Metrics -----
[+] TokuDB is disabled.

----- XtraDB Metrics -----
[+] XtraDB is disabled.

----- RocksDB Metrics -----
[+] RocksDB is disabled.

----- Spider Metrics -----
[+] Spider is disabled.

----- Connect Metrics -----
[+] Connect is disabled.

----- Galera Metrics -----
[+] Galera is disabled.

----- Replication Metrics -----
[+] Galera Synchronous replication: NO
[+] No replication slave(s) for this server.
[+] This is a standalone server.

----- Recommendations -----
General recommendations:
  Set up a Password for user with the following SQL statement ( SET PASSWORD FOR 'user'@'SpecificDNSorIp' = PASSWORD('secure_password'); )
  MySQL started within last 24 hours - recommendations may be inaccurate
  Configure your accounts with ip or subnets only, then update your configuration with skip-name-resolve=1
Variables to adjust:
  query_cache_size (=0)
  query_cache_type (=0)
  query_cache_limit (> 1M, or use smaller result sets)
  innodb_log_file_size * innodb_log_files_in_group should be equal to 1/4 of buffer pool size (=64M) if possible.
titiel@X550JX:~$

```

No necessita massa canvis, ja que està suportant bastant bé la càrrega de les querys tot i que recomana no cachejar les querys, no és del tot cert.

També cal dir, que en aquesta prova, s'ha fet servir només una connexió.

Caldrà veure més endavant si afegint més connexions és capaç de millorar.

```
itiel@X550JX:~$ cat /etc/mysql/mysql.conf.d/mysqld.cnf | grep -v "#"

[mysqld_safe]
socket      = /var/run/mysqld/mysqld.sock
nice        = 0

[mysqld]
user        = mysql
pid-file    = /var/run/mysqld/mysqld.pid
socket      = /var/run/mysqld/mysqld.sock
port        = 3306
basedir     = /usr
datadir     = /var/lib/mysql
tmpdir      = /tmp
lc-messages-dir = /usr/share/mysql
skip-external-locking
bind-address      = 127.0.0.1
key_buffer_size  = 16M
max_allowed_packet = 16M
thread_stack     = 192K
thread_cache_size = 8
myisam-recover-options = BACKUP
query_cache_limit = 1M
query_cache_size  = 16M
log_error         = /var/log/mysql/error.log
expire_logs_days  = 10
max_binlog_size   = 100M
itiel@X550JX:~$
```

Aquesta és la configuració òptima que necessita el SGBD per poder fer funcionar l'aplicació per a un usuari.

## Connexions simultànies (optimització)

```
#!/usr/bin/env python

import MySQLdb

conn = MySQLdb.connect(user = "itiel",
                        passwd = "asd123",
                        host = "localhost",
                        port = 3306,
                        db = "perruqueria")
c = conn.cursor()

for i in range(1000):
    c.execute("SELECT hora_inici FROM slot WHERE hora_inici > CURRENT_TIME AND int_data = CURRENT_DATE")

# print all the first cell of all the rows
#for row in c.fetchall():
#    print row[0]

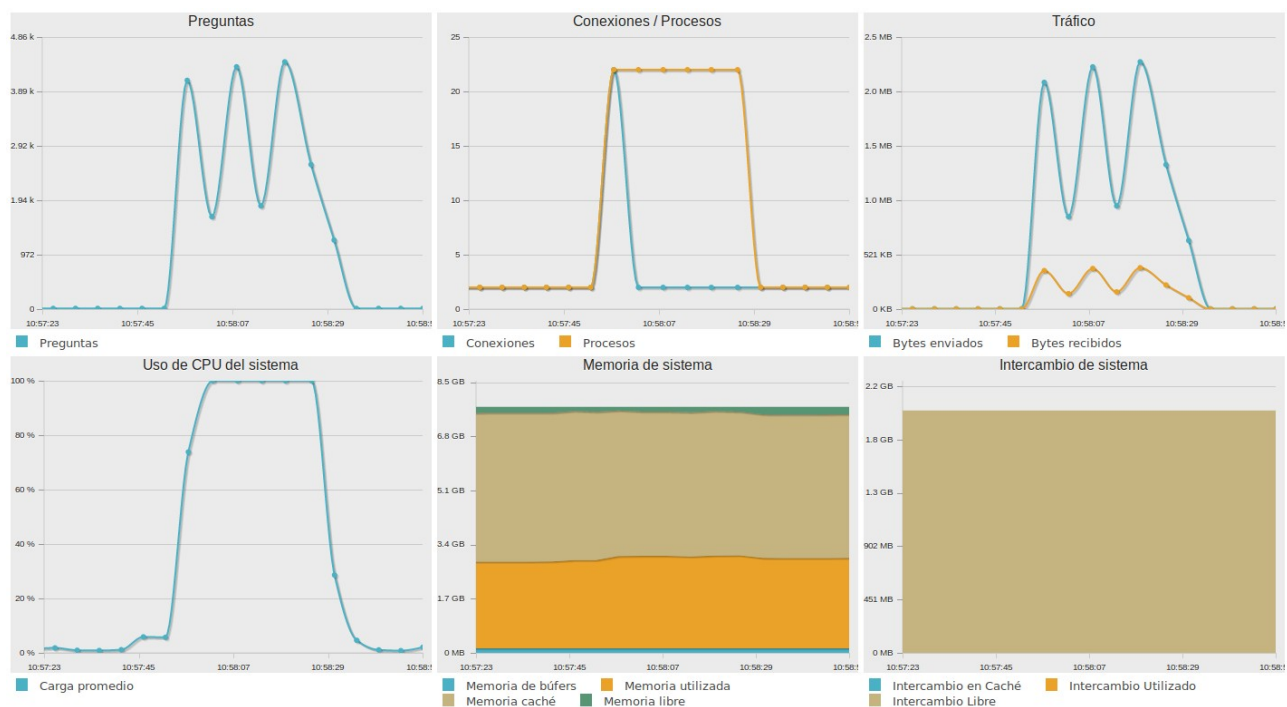
c.close()
```

Codi python que simularà connexions de usuaris demanant la disponibilitat que hi ha en el dia en que s'executi la query.

```
#!/bin/bash

for i in {1..20}
do
    python select.py &
done
```

Script en bash que crida a l'anterior 20 cops simulant que es connecten 20 usuaris simultanis.



Es pot observar que fent l'estrès arriba a 20 connexions de pic en el gràfic que són les esperades. Probablement això es podria optimitzar una mica demanant-li suggeriments de canvi a mysqltuner.

Filtros

Que contengan la palabra:
☐ Mostrar sólo valores de alerta
☐ Mostrar valores sin formato

Actualizar

Enlaces relacionados: 1

Variable	Valor	Descripción
Slow launch threads	0	El número de procesos que han tomado más de los segundos registrados en <code>slow_launch_time</code> para crear.
Threads cached	6	El número de procesos en el cache de procesos. La tasa de aciertos al cache puede calcularse como <code>Threads_created/Connections</code> . Si este valor es rojo, debe incrementar su <code>thread_cache_size</code> .
Threads connected	2	El número de conexiones abiertas actualmente.
Threads created	22	El número de procesos creados para manejar las conexiones. Si <code>Threads_created</code> es grande, querrá incrementar el valor <code>thread_cache_size</code> . (Normalmente esto no aporta una mejora notable en el rendimiento si usted tiene una buena implementación de procesos.)
Threads running	1	El número de procesos que no están en reposo.
Threads cache hitrate %	95.22 %	Porcentaje de aciertos del caché de hilos (valor calculado)

S'observa la quantitat de fils que utilitza el gestor i que tots els paràmetres tenen valors positius.

Variable	Valor	Descripción
<b>Handler commit</b> ⓘ	42.5.k	El número de sentencias COMMIT internas.
<b>Handler delete</b> ⓘ	0	El número de veces que una fila fue eliminada de una tabla.
<b>Handler discover</b> ⓘ	0	El servidor MySQL puede solicitar al motor de almacenamiento NDB Cluster si conoce acerca de una tabla con un nombre dado. Esto se llama descubrimiento. Handler_discover indica el número ocasiones que las tablas han sido descubiertas.
<b>Handler external lock</b> ⓘ	86.5.k	
<b>Handler mrr init</b> ⓘ	0	
<b>Handler prepare</b> ⓘ	0	
<b>Handler read first</b> ⓘ	65	El número de veces en que el primer registro ha sido leído desde un índice. Si este valor es alto, sugiere que el servidor está haciendo gran cantidad de escaneos completos del índice; por ejemplo, SELECT col1 FROM foo, asumiendo que col1 está indizado.
<b>Handler read key</b> ⓘ	42.5.k	El número de solicitudes hechas para leer una fila basado en una clave. Si este valor es alto, es una buena indicación de que sus consultas y tablas están indexadas apropiadamente.
<b>Handler read last</b> ⓘ	0	
<b>Handler read next</b> ⓘ	466.M	El número de solicitudes hechas para leer la siguiente fila en un orden de clave. Este se incrementa si usted está consultando una columna índice con un limitante de rango o si usted está haciendo un escaneo del índice.
<b>Handler read prev</b> ⓘ	0	El número de solicitudes hechas para leer la fila previa en un orden de clave. Este método de lectura se usa principalmente para optimizar a ORDER BY ... DESC.
<b>Handler read rnd</b> ⓘ	109	El número de solicitudes hechas para leer una fila basado en una posición fija. Este es alto si usted está haciendo muchas consultas que requieren el reordenamiento del resultado. Usted probablemente tiene muchas consultas que requieren que MySQL escanee tablas enteras o usted debe tener vínculos (joins) que no usan las claves de manera apropiada.
<b>Handler read rnd next</b> ⓘ	163.4.k	El número de solicitudes hechas para leer la siguiente fila en el archivo de datos. Este es alto si usted está haciendo muchos escaneos de tablas. Generalmente, esto sugiere que sus tablas no están indizadas apropiadamente o que sus consultas no están escritas para tomar ventaja de los índices que tiene.
<b>Handler rollback</b> ⓘ	42	El número de sentencias ROLLBACK internas.
<b>Handler savepoint</b> ⓘ	0	
<b>Handler savepoint rollback</b> ⓘ	0	
<b>Handler update</b> ⓘ	0	El número de solicitudes hechas para actualizar una fila en una tabla.
<b>Handler write</b> ⓘ	83.k	El número de solicitudes hechas para insertar una fila en una tabla.

Fent un anàlisi del gestor, es pot observar que està fent bé la feina tot i ser incongruent el handler read rnd next, això són consultes d'accés aleatori, és a dir, això és culpa de l'script que insereix dades, ja que obté clients de forma aleatoria.

```

----- Recommendations -----
General recommendations:
  Control warning line(s) into /var/log/mysql/error.log file
  Control error line(s) into /var/log/mysql/error.log file
  Set up a Password for user with the following SQL statement ( SET PASSWORD FOR 'user'@'SpecificDNSorIp' = PASSWORD('secure_password'); )
  MySQL started within last 24 hours - recommendations may be inaccurate
  Configure your accounts with ip or subnets only, then update your configuration with skip-name-resolve=1
  Adjust your join queries to always utilize indexes
Variables to adjust:
  query_cache_size (=0)
  query_cache_type (=0)
  query_cache_limit (> 1M, or use smaller result sets)
  join_buffer_size (> 256.0K, or always use indexes with joins)
  innodb_log_file_size * innodb_log_files_in_group should be equal to 1/4 of buffer pool size (=64M) if possible.
ttlel@X550JX:--$

```

Finalment, després d'executar mysqltuner, les seves recomanacions són les que esperàvem i a més ens ha indicat que hem fet alguna join desaprofitant l'índex i que aconsella incrementar el búffer o bé, fer servir l'índex.

Es pot concloure que amb 20 connexions és capaç de donar servei sense cap mena de problemes, ja que és la quantitat mitjana, més que mitjana la quantitat pic d'usuaris que podria connectar-se a l'aplicació, tal com està pensada, en un altre cas, potser caldria fer un anàlisi més exhaustiu, però, partint de la base que és una perruqueria amb un sol treballador i que la destinació de la pròpia aplicació és un poble molt petit, funciona correctament i és capaç d'executar les consultes.

La raó principalment és que l'escenari on s'està executant, és a dir, l'ordinador, és molt potent i gairebé no necessita millores amb aquest tipus de casuística.

# Tecnologies emprades

## Frontend

Primerament, cal esmentar les característiques de l'escenari on s'han executat les proves:

- **Sistema operatiu:** Ubuntu 18.04 LTS.
- **Processador:** i7-4750 HQ.
- **RAM:** 8 GB.
- **Disc dur:** Samsung SSD 850 EVO.

Per continuar, es nomenaran les tecnologies que s'han fet servir en l'aplicació i una petita explicació indicant el perquè:

- **PHP:** És un llenguatge ràpid, robust i molt fàcil, amb poques línies de codi es poden fer meravelles i aconseguir un resultat molt professional. A més és un llenguatge que ja coneixia i en tenia molta pràctica.

Totes les validacions s'han fet en l'àmbit de servidor, és a dir, amb php, perquè l'aplicació sigui més robusta i no pugui ser fàcilment enganyada.

- **JavaScript:** Ofereix un punt extra molt professional a la pàgina web, sobretot si es fa servir AJAX que és el meu cas. Gràcies a aquesta eina, he pogut actualitzar de forma asíncrona la pàgina web sense que l'usuari s'adoni de l'ho que està succeint.
- **CSS:** Ajuda a donar-li l'estil final a la pàgina web. Tot i no saber massa, m'ha ajudat a donar-li un millor aspecte del qual podria haver-hi aconseguit sense, però, al cap i la fi, és un aspecte que no és valorava en la pràctica.
- **MySQL:** És el sistema gestor amb el qual em sento més còmode, tot i no ser igual de robust que PostgreSQL, és un sistema que està a l'altura per posar-se'n en producció en moltes empreses. Per exemple, **UPCConnect** i **RakutenTV** el fan servir i donen servei a milers de persones.
- **PhpMyAdmin:** Client gestor del SGBD MySQL per poder crear la bases de dades, taules i executar queries de forma més simple i poder navegar en entorn web per tota la informació emmagatzemada.

# Scripts

Per un costat he fet un script en python que genera clients falsos fent servir la llibreria Faker.

```
#!/usr/bin/env python

import MySQLdb
from faker import Faker
import random
import string
from datetime import *

def id_generator(size=20, chars=string.ascii_uppercase + string.digits):
    return ''.join(random.choice(chars) for _ in range(size))

def do_query(c, q, n=''):
    if opt.show:
        print("Query is: " + q + " with values " + str(n))
    c.execute(q, n)

conn = MySQLdb.connect(user = "itiel",
                        passwd = "asd123",
                        host = "localhost",
                        port = 3306,
                        db = "perruqueria")
c = conn.cursor()

# Prova per llista usuaris
#c.execute("SELECT * FROM client")

# print all the first cell of all the rows
#for row in c.fetchall():
#    print row[0]

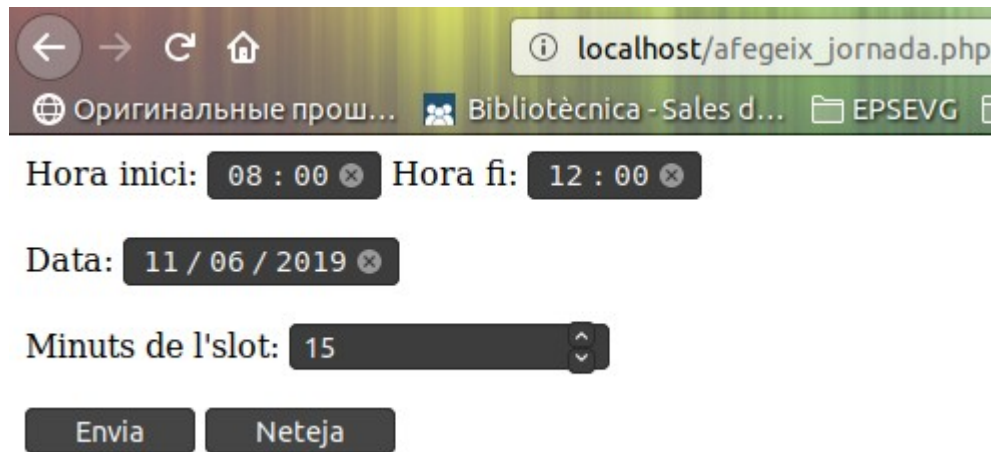
fake = Faker('es_ES')

# INSERTA USUARIS
q = "INSERT INTO client VALUES (%s, %s, %s, %s, %s, 0, 0);"
for x in range(10000):
    try:
        do_query(c, q, (fake.free_email(), id_generator(), fake.first_name(), fake.last_name(), fake.phone_number()))
        conn.commit()
    except:
        print("Duplicated entry!!")

c.close()
```



D'altra banda he fet que introduïnt un nou interval, afegeixi l'interval especificat més els slots necessaris per a aquest fent servir una transacció, que, si falla, fa un rollback dels slots i elimina l'interval afegit.



localhost/afegeix\_jornada.php

Оригинальные прош... Bibliotècnica - Sales d... EPSEVG

Hora inici: 08 : 00 Hora fi: 12 : 00

Data: 11 / 06 / 2019

Minuts de l'slot: 15

Envia Neteja

```
<?php

// $sql = 'SELECT date_format('data_intval', '%m/%d/%Y') as mydatefield FROM intval';

ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);

$missatge = "";

if (isset($_POST['hora_inici'])) {

    include "funcions/database.php";
    $inici = new DateTime($_POST['hora_inici']);
    $fi = new DateTime($_POST['hora_inici']);

    // Comprova que l'interval és correcte (hora inici és menor a hora fi).
    if ($inici > $fi)
    {

        $sql = $conn->prepare("SELECT MAX(hora_inici) AS fi FROM intval WHERE data_intval = :d");
        $sql->bindParam(':d', $_POST['data']);
        $sql->execute();

        $result = $sql->fetch(PDO::FETCH_ASSOC);

        // Comprova que l'interval i data inserits sigui darrer a l'últim existent.
        if (empty($result['fi']) == true || (empty($result['fi']) == false && new DateTime($_POST['hora_inici']) > new DateTime($result['fi']))) {

            $sql = "INSERT INTO intval VALUES (:hi, :hf, :d, :s)";

            $sql = $conn->prepare($sql);
            $sql->bindParam(':hi', $_POST['hora_inici']);
            $sql->bindParam(':hf', $_POST['hora_inici']);
            $sql->bindParam(':d', $_POST['data']);
            $sql->bindParam(':s', $_POST['slot']);

            if ($sql->execute()) {

                // Genera els slots necessaris
                $diff_temps=$fi->diff($inici)->h;
                $slots = ($diff_temps*60) / $_POST['slot'];
                $slot = $_POST['slot'] * 60; // Conversió a segons
                $hora_inici = $_POST['hora_inici'];
                $incrementa_slot = date("H:i",strtotime($hora_inici) + $slot);
```



```

try {
    // First of all, let's begin a transaction
    $conn->beginTransaction();

    // A set of queries; if one fails, an exception should be thrown

    $sql = "INSERT INTO slot VALUES (:hi, :ihi, :ihf, :id, 0, 0, 0)";

    $qq = $conn->prepare($sql);

    for ($i=0; $i<$slots; $i++) {

        $qq->bindParam(':hi',$hora_inici);
        $qq->bindParam(':ihi',$_POST['hora_inici']);
        $qq->bindParam(':ihf',$_POST['hora_fi']);
        $qq->bindParam(':id',$_POST['data']);

        $qq->execute();

        $hora_inici = $incrementa_slot;
        $incrementa_slot = date("H:i",strtotime($hora_inici) + $t_slot);
    }

    // If we arrive here, it means that no exception was thrown
    // i.e. no query has failed, and we can commit the transaction
    $conn->commit();
    $missatge = "Jornada registrada correctament.";
} catch (Exception $e) {
    // An exception has been thrown
    // We must rollback the transaction
    $conn->rollback();
    $missatge = "Error al registrar els slots per al nou interval. ".$e->getMessage();

    $sql = "DELETE FROM intval WHERE hora_inici = :ihi AND hora_fi = :ihf AND data_intval = :id";

    $qq = $conn->prepare($sql);

    $qq->bindParam(':ihi',$_POST['hora_inici']);
    $qq->bindParam(':ihf',$_POST['hora_fi']);
    $qq->bindParam(':id',$_POST['data']);

    $qq->execute();

    $missatge.="Eliminat interval";
}

} else {
    $missatge = "Ha ocorregut un error. Probablement ja està registrat l'interval.";
}

}else{
    $missatge = "No es pot afegir un interval dins d'un ja vigent.";
}

}else{
    $missatge = "L'hora fi de l'interval és menor a la de l'inici!!.";
}

}

?>

<form action="afegeix_jornada.php" method="post">
    Hora inici: <input type="time" name="hora_inici" value="08:00">
    Hora fi: <input type="time" name="hora_fi" value="12:00"> <br/><br/>
    Data: <input type="date" name="data" min="<?= date("Y-m-d") ?>" value="<?= date("Y-m-d") ?>"> <br/><br/>
    Minuts de l'slot: <input type="number" name="slot" min="1" max="60" value="15"> <br/><br/>
    <input type="submit" value="Envia">
    <input type="reset" value="Neteja">
</form>

<?= $missatge ?>

```

I per acabar amb els scripts, també he fet un altre que assigna el calendari del perruquer per als dos proxims anys de forma aleatoria, sent dissabtes, diumenges i mes d'agost festius. Crea els slots per a els intervals i assigna cites a clients també al atzar.

L'ordre d'execució és, primer cal afegir clients i després, assignar intervals, slots i cites.

```
#!/usr/bin/php
ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);

function add_slots($hora_in, $hora_fi, $data) {
    include "funcions/database.php";

    $inici = new DateTime($hora_in);
    $fi = new DateTime($hora_fi);
    // Genera els slots necessaris
    $diff_temps=$fi->diff($inici)->h;
    $slots = ($diff_temps*60) / 15;
    $t_slot = 15 * 60; // Conversió a segons
    $hora_inici = $hora_in;
    $incrementa_slot = date("H:i",strtotime($hora_inici) + $t_slot);

    $sql = "INSERT INTO slot VALUES (:hi, :ihi, :ihf, :id, 0, 0, 0)";

    $qq = $conn->prepare($sql);

    for ($i=0; $i<$slots; $i++) {
        $qq->bindParam(':hi',$hora_inici);
        $qq->bindParam(':ihi',$hora_in);
        $qq->bindParam(':ihf',$hora_fi);
        $qq->bindParam(':id',$data);

        $qq->execute();

        //echo $qq->debugDumpParams();

        $hora_inici = $incrementa_slot;
        $incrementa_slot = date("H:i",strtotime($hora_inici) + $t_slot);
    }
}

include "funcions/database.php";

$data = new DateTime(date("Y-m-d"));

$interval = $conn->prepare("INSERT INTO intval VALUES (:hi, :hf, :di, 15)");

// INSERTA ELS PRÓXIMS 2 ANYS
```

```

for ($i=0; $i<730; $i++) {

    $date=$data->format('Y-m-d');

    // AGOST I CAPS DE SETMANA SÓN FESTIUS!!
    if ($data->format('m') != 8 && $data->format('N') != 6 && $data->format('N') != 7) {

        $jornada = rand(1,3);
        $slots = 0;

        if ($jornada == 1) {          // Treballa de 7:30 a 15:30
            $hora_inici = "07:30";
            $hora_fi = "15:30";
            $interval->bindParam(':hi',$hora_inici);
            $interval->bindParam(':hf',$hora_fi);
            $interval->bindParam(':di',$date);
            $interval->execute();
            add_slots($hora_inici, $hora_fi, $date);

        }elseif($jornada == 2) {     // Treballa de 9:00 a 12:00, de 13:00 a 14:00 i de 15:00 a 19:00
            $hora_inici = "09:00";
            $hora_fi = "12:00";
            $interval->bindParam(':hi',$hora_inici);
            $interval->bindParam(':hf',$hora_fi);
            $interval->bindParam(':di',$date);
            $interval->execute();
            add_slots($hora_inici, $hora_fi, $date);

            $hora_inici = "13:00";
            $hora_fi = "14:00";
            $interval->bindParam(':hi',$hora_inici);
            $interval->bindParam(':hf',$hora_fi);
            $interval->bindParam(':di',$date);
            $interval->execute();
            add_slots($hora_inici, $hora_fi, $date);

            $hora_inici = "15:00";
            $hora_fi = "19:00";
            $interval->bindParam(':hi',$hora_inici);
            $interval->bindParam(':hf',$hora_fi);
            $interval->bindParam(':di',$date);
            $interval->execute();
            add_slots($hora_inici, $hora_fi, $date);

        }elseif($jornada == 3) {     // Treballa de 10:00 a 14:00, de 17:00 a 21:00
            $hora_inici = "10:00";
            $hora_fi = "14:00";
            $interval->bindParam(':hi',$hora_inici);
            $interval->bindParam(':hf',$hora_fi);
            $interval->bindParam(':di',$date);
            $interval->execute();
            add_slots($hora_inici, $hora_fi, $date);

            $hora_inici = "17:00";
            $hora_fi = "21:00";
            $interval->bindParam(':hi',$hora_inici);
            $interval->bindParam(':hf',$hora_fi);
            $interval->bindParam(':di',$date);
            $interval->execute();
            add_slots($hora_inici, $hora_fi, $date);

        }

        // ASSIGNA CITES ALEATORIES ALS USUARIS

        $serveis_rnd = rand(1,3);

        $q_client = $conn->prepare("SELECT email FROM client ORDER BY RAND() LIMIT 1");
        $q_client->execute();
        $mail = $q_client->fetch(PDO::FETCH_ASSOC)['email'];

        if ($serveis_rnd == 1){
            $qq = $conn->prepare("SELECT id, n_slots FROM servei ORDER BY RAND() LIMIT 1");
            $qq->execute();
        }elseif ($serveis_rnd == 2){
            $qq = $conn->prepare("SELECT id, n_slots FROM servei ORDER BY RAND() LIMIT 2");
            $qq->execute();
        }else{
            $qq = $conn->prepare("SELECT id, n_slots FROM servei ORDER BY RAND() LIMIT 3");
            $qq->execute();
        }

        echo "<br/>". $qq->debugDumpParams();

        $n_slots=0;
        $serveis_arr = array();

        while ($res = $qq->fetch(PDO::FETCH_ASSOC)) {

            $n_slots+=$res['n_slots'];

```

```

        array_push($serveis_arr, $res['id']);
    }

    if ($n_slots > 8) $msg = "No pot reservar més de dos hores seguides.";
    else{

        $t_slot = 15 * 60;

        // Comprova si es poden fer seguits els serveis
        $hora_inici = $hora_inici;
        $hora_fi = date("H:i",strtotime($hora_inici) + ($t_slot * ($n_slots-1)));

        $sq = $conn->prepare("SELECT count(hora_inici) as total FROM slot WHERE (hora_inici BETWEEN :hi AND :hf) AND int_data = :d AND reservat=0");
        $sq->bindParam(':hi',$hora_inici);
        $sq->bindParam(':hf',$hora_inici);
        $sq->bindParam(':d',$date);
        $sq->execute();

        //echo "<br/>". $sq->debugDumpParams();

        $result = $sq->fetch(PDO::FETCH_ASSOC);

        // Es pot fer la reserva
        if ($result['total'] >= $n_slots) {

            $q_slot = $conn->prepare("UPDATE slot SET reservat=1 WHERE (hora_inici BETWEEN :hi AND :hf) AND int_data = :d AND reservat=0 AND expr");
            $q_slot->bindParam(':hi',$hora_inici);
            $q_slot->bindParam(':hf',$hora_inici);
            $q_slot->bindParam(':d',$date);
            $q_slot->execute();

            echo "<br/>". $q_slot->debugDumpParams();

            // Afegim la cita
            $sq = $conn->prepare("INSERT INTO cita VALUES (:h, :d, :e, 0)");
            $sq->bindParam(':h',$hora_fi);
            $sq->bindParam(':d',$date);
            $sq->bindParam(':e',$mail);
            if ($sq->execute()) {

                // Afegir serveis que vol l'usuari a la cita

                $q_demanats = $conn->prepare("INSERT INTO demanats VALUES (:h, :d, :s)");

                foreach($serveis_arr as $value) {
                    $q_demanats->bindParam(':h',$hora_fi);
                    $q_demanats->bindParam(':d',$date);
                    $q_demanats->bindParam(':s',$value);
                    $q_demanats->execute();
                }

                echo $q_demanats->debugDumpParams() . "<br/>";

                $msg = "Vostè té una cita per el dia " . $date . " a les " . $hora_fi . ".<br/>Recordi arribar 5 minuts abans i validar la cita s";
            }
            else{
                $msg = "Error al registrar la cita.";
            }
        }
        else{
            $msg = "No hi ha slots consecutius per poder prestar els serveis sol·licitats.<br/>Consideri en fer dos cites o fer una reserva en un";
        }

        echo "$msg<br>";
    }
}

}

$data->add(new DateInterval('P1D'));

echo "OK!! <br/>";

?>

```

## Conclusions

### Què he après

És una pràctica que requereix molt de temps i que tot i haver implementat tot el problema sense ser un requeriment i haver dedicat més de 60h a tota la pràctica en conjunt, he après a què no és bo tractar amb hores i dates, sobretot si es vol tenir un bon control i es vol evitar crear rangs (intervals) de temps.

Probablement m'he esforçat massa a fer que tot funcioni correctament i cenyir-me als requeriments fent servir diferents operacions diverses operacions CRUD que van estar una necessitat a més, per poder generar el frontend, com el calendari, les franges lliures, els serveis, etc.

També, haver utilitzat aquest tipus de tecnologies que t'obliguen a fer-te tu mateix tot el frontend, ha requerit el doble de temps, però, tot i així, la meva conclusió és que estic satisfet amb l'ho que he fet. Tenint en compte que té petits errors i que el problema està pensat per a una casuística específica, es podria posar gairebé en producció dedicant-li unes hores més.

Quant a llenguatges no he après res de nou, ja que he fet servir tecnologies que ja coneixia. A nivell de llenguatge, no el llenguatge com a tal, sino, fent servir-lo per implementar sql, he après a fer transaccions que no havia fet mai cap i a evitar l'sql injection. També he après a introduir correctament índexs i fixar-me com milloren les consultes i empitjoren segons el cas concret.

Després de haver realitzat la pràctica, la meva sensació és bona, però, potser canviaria de llenguatges i provaria d'utilitzar un altre SGBD, que tal i com he programat l'aplicació, canviant en la connexió el tipus de gestor de bases de dades que es vol fer servir i canviant els tokens específics de cada SGBD, el codi és "universal".

Com a contrapunt i com a opinió personal, potser, en la presentació amb el professor s'hauria de valorar més l'aprenentatge de les bases de dades, és a dir, la quantitat de consultes que s'han fet, com s'han fet i el perquè. He trobat que és valorava més la quantitat de pantalles en el frontend que haver afegit la funcionalitat cenyida als requeriments/enunciat i ben implementada. Sóc conscient que no es culpa del professor, ja que jo vaig triar la forma de presentar la pràctica i que el temps és el que és i valorar això en quinze minuts és molt poc i difícil, però, com a estudiant, el meu sabor de boca és que m'ha faltat poder ensenyar realment què m'ha fet trigar tant de temps i perquè li he dedicat més temps a la part de les bases de dades que és on volia aprendre i d'això es tracta l'assignatura, en tot cas, adjuntaré els codis en l'entrega.

De totes maneres, insisteixo en què el **feedback** del **professor** ha estat **molt positiu** i que és una conclusió/crítica constructiva per ajudar als estudiants del següent quadrimestre, però, realment, gràcies a l'aprenentatge que hem adquirit durant el quadrimestre, ha estat més que suficient per poder ser capaços de realitzar aquesta i qualsevol pràctica.