



Institut Puig Castellar  
Santa Coloma de Gramenet

# OFICINA VIRTUAL

CFGS DAM

Maria Junoy Gomez

Curso:2018-2019



Esta obra esta sujeta a una licencia de  
[Reconocimiento-NoComercial-SinObraDerivada 3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

# Índice

1-Introducción.....	3
1.1-Resumen del proyecto.....	3
1.2-Motivaciones.....	3
1.3-Tareas y planificación .....	4
2-Desarrollo del proyecto.....	5
2.1-Explicación de las tareas.....	5
2.2-Imprevistos surgidos.....	15
3-Conclusiones.....	17
3.1-La experiencia.....	17
3.2-Posibles mejoras del proyecto.....	17
4-Bibliografía.....	18

# **1-Introducción**

## **1.1-Resumen del proyecto**

Visto el problema de acumulación de gente que se genera cuando llega el periodo de inscripciones en el instituto se decide generar una aplicación que permita agilizar el proceso de las gestiones.

En las primeras reuniones el proyecto se enfocaba unicamente en solucionar el problema de las matriculaciones, pero viendo el potencial que tenia el proyecto y que este podía implementarse en diferentes centros se decidió crear una aplicación que fuera genérica y pudiera adaptarse tanto a los centros como las gestiones que estos pueden ofrecer.

## **1.2-Motivaciones**

Las motivaciones que han impulsado a crear este proyecto a sido el poder crear desde cero una aplicación que sea funcional y que ademas pueda ayudar a los centros académicos a gestionar los tramites administrativos que los alumnos solicitan (matriculas , convalidaciones,etc).

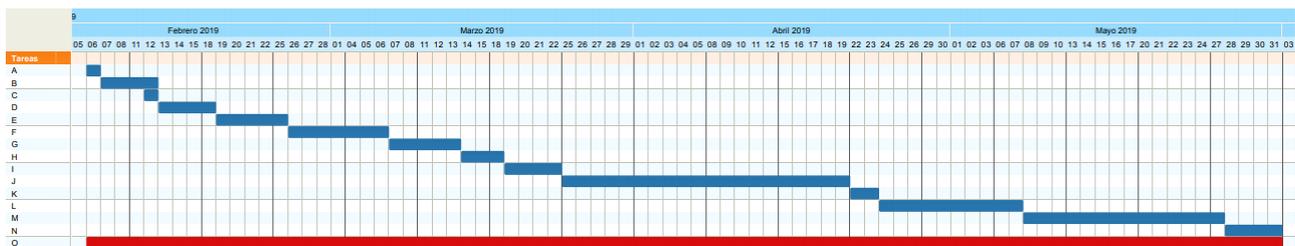
También el poner en practica los conocimientos adquiridos durante el curso en base de datos, programación, etc.

### 1.3-Tareas y planificación .

Las tareas a realizar en este proyecto han sido las siguientes:

- a) Reunión inicial con el jefe de proyecto.
- b) Presentación de las ideas iniciales.
- c) Selección de la idea a desarrollar.
- d) Selección del lenguaje de programación.
- e) Selección del framework a utilizar.
- f) Aprendizaje del uso del framework seleccionado.
- g) Preparación del entorno de trabajo a utilizar.
- h) Testeo de la estabilidad del entorno de trabajo.
- i) Preparación de la base de datos de la aplicación.
- j) Programación de la primera parte de la aplicación.
- k) Testeo de la primera parte de la aplicación.
- l) Programación de la segunda parte de la aplicación.
- m) Testeo de la segunda parte de la aplicación.
- n) Testeo de la aplicación en global una vez finalizada.
- o) Creación de la memoria del proyecto.

La planificación de las tareas a quedado de la siguiente manera:



## **2-Desarrollo del proyecto.**

### 2.1-Explicación de las tareas.

#### **Reunión inicial con el jefe de proyecto:**

En la primera reunión se abordó la problemática a solucionar y se plantearon posibles ideas sobre como solucionar ese problema.

#### **Presentación de las ideas iniciales y selección de la idea a desarrollar:**

En la segunda reunión se exponen diferentes ideas para afrontar la problemática. En esta reunión surgen dos ideas firmes las cuales son:

- Una aplicación que solo se encarga unicamente de las matriculas.
- Una aplicación que se puede encargar de cualquier tipo de gestión que pueda ofrecer el centro.

Una vez presentadas las ideas se opta por la segunda opción ya que esta permite mayor alcance y soluciona tanto el problema actual como los futuros problemas. También permitiendo agilizar mas el trabajo de las administrativas.

#### **Selección del lenguaje de programación:**

Una vez seleccionado el abarque de la aplicación se empieza una búsqueda exhaustiva de los posibles lenguajes de programación a utilizar, encontrando los siguientes candidatos.

- JavaScript
- PHP
- Python
- Ruby

De estos cuatro lenguajes tanto Python como Ruby se descartan por no tener conocimientos de estos lenguajes, quedando la decisión final entre JavaScript y PHP.

Entre estos dos lenguajes se decide utilizar PHP por dos grandes motivos el primero es que se esta mas familiarizado con este lenguaje y la segunda es que este se interpreta en el lado del servidor quedando el código oculto al usuario. También por la gran comunidad que tiene en la red pudiendo encontrar mas soporte a la hora de corregir errores de la aplicación.

### **Selección del framework a utilizar:**

Una vez seleccionado el lenguaje se decide utilizar un framework para poder programar con mas comodidad la aplicación y ofrecer mas seguridad .Se realiza una búsqueda de posibles candidatos a utilizar quedando las siguientes opciones.

- Laravel.
- Symfony
- Yii 2

De estos tres frameworks se opta por utilizar laravel en vez de los otros dos ya que es el que esta mas extendido actualmente teniendo una gran comunidad en la red. También es el mas utilizado en las empresas y el poder aprender a utilizar este framework resulta mas interesante de cara a futuros proyectos con empresas.

### **Aprendizaje del uso del framework seleccionado:**

Se utilizan tres semanas en aprender las nociones básicas sobre su instalación, la estructura de los archivos, la programación de los códigos y la estructura de las vistas.

### **Preparación del entorno de trabajo a utilizar:**

Una vez se tienen nociones sobre el framework a utilizar se comienza a preparar un entorno de trabajo teniendo que tomar diferentes decisiones importantes.

Se utilizara una maquina virtual con un Ubuntu server 18.04 LTS , en esta se instalara apache2 , PHP , MySQL y laravel.

Para la base de datos se a decidido utilizar MySQL por que mariaDB da problemas a la hora de implementar por que no tiene soporte para emojis y laravel necesita este tipo de soporte.

Para programar se decide utilizar Visual Studio code ya que es un entorno conocido y se prefiere ir a lo seguro.

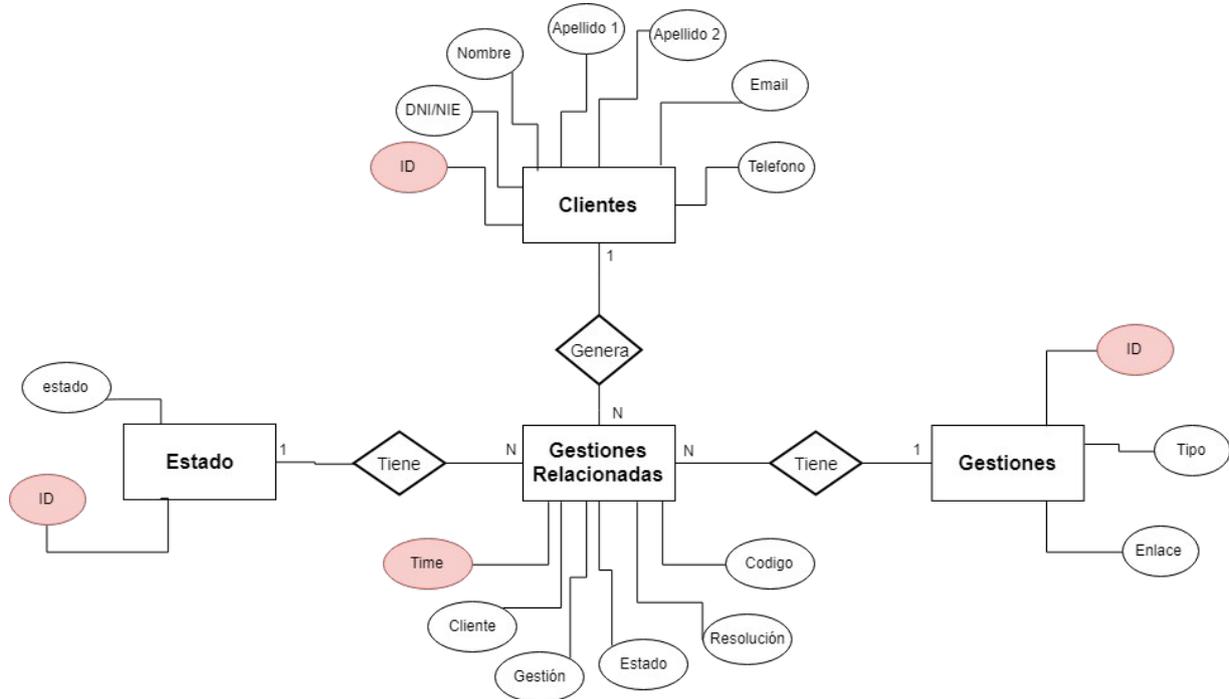
En el tema visual se decide utilizar Bootstrap ya que su uso es muy extendido y se adapta perfectamente a las necesidades del proyecto.

### **Testeo de la estabilidad del entorno de trabajo:**

Se comienza una fase que dura dos días en la cual se testea que el entorno sea estable y no de problemas de conexión y navegación encontrando errores a la hora de la redirigir las rutas en apache haciendo que la fase de testeo se alargue a 3 días.

## Preparación de la base de datos de la aplicación.

Una vez el entorno de trabajo es estable se comienza a plantear como sera la estructura necesaria para la base de datos, se decide que información sera necesaria almacenar y cual no quedando al fina tal y como se ve en la imagen.

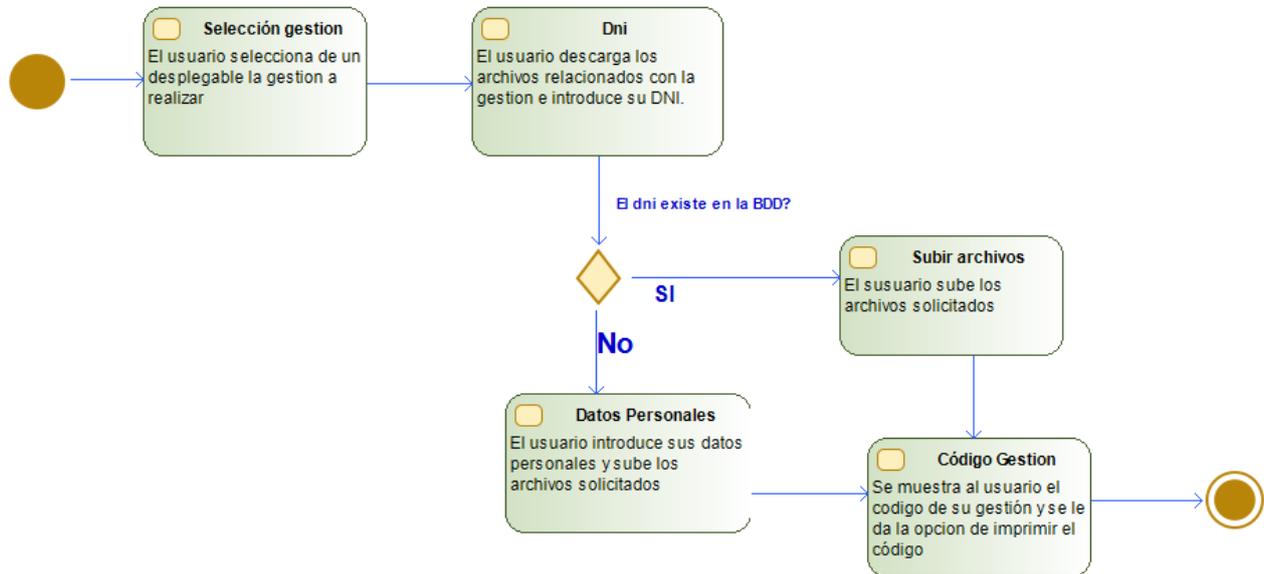


Una vez decidida la estructura de la base de datos se comienzan a preparar la implementación de esta a traves de las migraciones de laravel , estas nos permiten programar la estructura y relaciones de nuestra base de datos y crearla e implementarla con tan solo un comando, permitiendo actualizar la base de datos si la aplicación lo requiere simplemente agregando nuevas migraciones.

## Programación de la primera parte de la aplicación:

Una vez tenemos creada nuestra base de datos se procede a programar la parte que será utilizada por el usuario .

Los pasos que tendrá que hacer un usuario para que su gestión quede finalizada serán los siguientes



Para que la aplicación sea dinámica y pueda ir adaptándose sin necesidad de re-programar las opciones que el usuario puede seleccionar en el formulario se rellenan con la información que se recoge en la base de datos. De esta manera cuando una gestión nueva sea introducida en la base de datos esta aparecerá de forma inmediata para ser seleccionada en el formulario del usuario.

Cuando se finaliza la gestión esta muestra una pantalla en la cual también hay un código QR que permitirá que en administración localicen más rápidamente la gestión quedando como en la imagen.

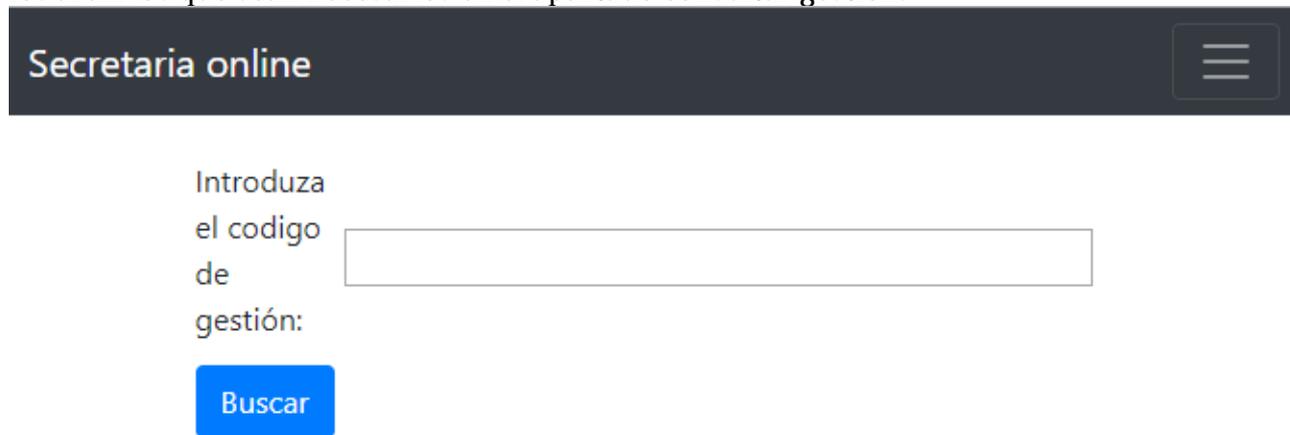
El identificador de su gestión es:

**20190523\_172435\_53331624j\_Inscripción**



Para poder generar fácilmente este código QR se recurre al paquete de composer llamado simple QR code que nos permite generar el código . Se recurre a esta paquete ya que aunque con php se pueden generar códigos QR es mucho mas complicado y se requiere procesar mas código haciendo que la opción de usar este paquete para sea mucho mas simple de utilizar y hará la aplicación mas rápida de cara al usuario.

Una vez el usuario consigue el código de la gestión este puede consultar el estado y subir los archivos que sean necesarios en el apartado consultar gestión.



Secretaria online

Introduza  
el código  
de  
gestión:

Buscar

### **Testeo de la primera parte de la aplicación:**

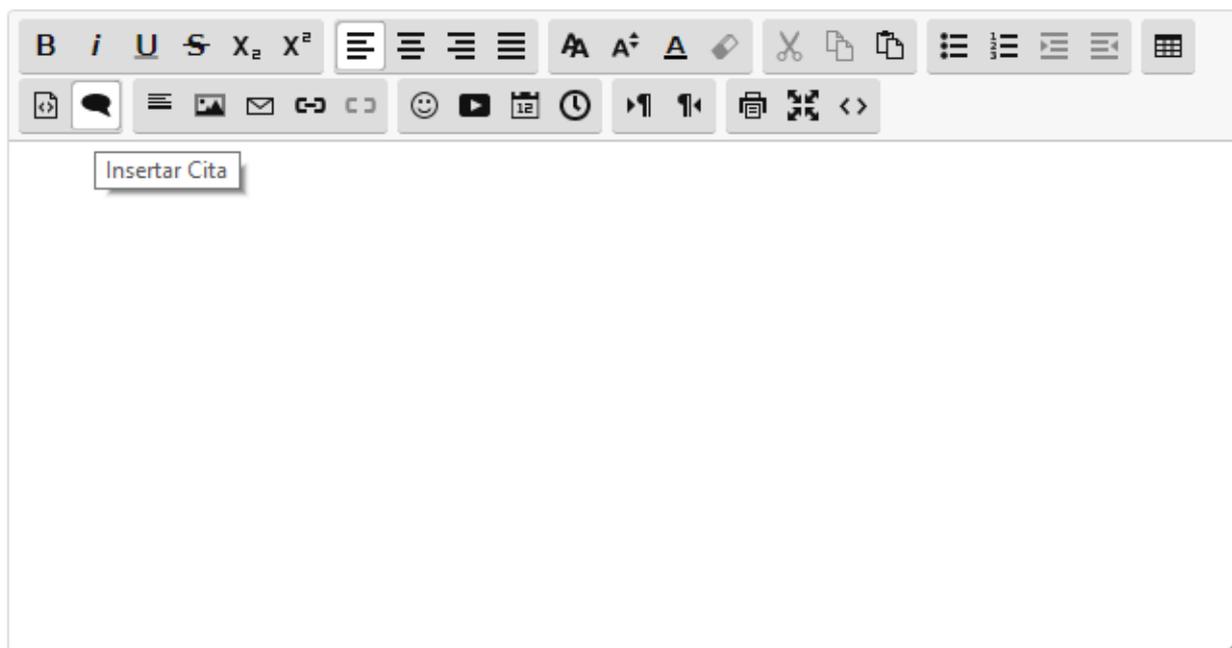
Una vez terminada la programación de la parte del usuario se comienzan a realizar una serie de pruebas individuales en busca de fallos y errores en la aplicación. Una vez solucionado errores menores se procederá a pasar al siguiente paso.

### **Programación de la segunda parte de la aplicación:**

Una vez que la parte del usuario funciona correctamente y se generan y almacenan gestiones se puede proceder a programar la parte de las administrativas. Estas solo podrán consultar y modificar las gestiones generadas.

Cuando se comienza a programar la pantalla de modificación de la gestión se da la necesidad de proporcionar a las administrativas un editor de texto que les permita dar formato al texto sin la necesidad de utilizar código especial . Se revisan varias opciones y se opta por utilizar javascript para proporcionar esta herramienta ya que es mas viable que no programarlo desde PHP.

Se investiga y se encuentran dos opciones compatibles con laravel una llamada Ckeditor y otra llamada SCEditor. Una vez investigado y testeado su uso de opta por utilizar SCEditor por ser mas agradable de utilizar y de implementar, también incluye la opción de poner los textos de ayuda que salen en castellano. Quedando tal y como vemos a continuación una vez implantado.



En esta parte de la aplicación surge la necesidad de informar al usuario de que el estado de su gestión se a modificado, se plantean las siguientes opciones:

- Llamada al teléfono solicitado para informar.
- Mensaje de texto enviado al teléfono solicitado.
- Envío de un correo electrónico al correo solicitado.

De estas tres opciones se utiliza el envío de correo electrónico por dos motivos el primero es que es menos invasivo con la privacidad del usuario y el segundo es que se puede programar la tarea para que sea mas cómoda de utilizar la aplicación por parte de las administrativas.

Para poder generar el correo de forma automática se comienza una búsqueda de opciones. Se considera usar la función mail() que viene incorporada en PHP combinada con la librería PHPMailer pero esto se descarta en el momento en que se comprueba que el propio framework tiene opción de enviar email con una librería que lleva implementada.

Una vez se decide utilizar la librería que viene por defecto en el framework se decide utilizar una cuenta de correo de Google ya que es el mas sencillo de configurar. Después de solucionar unos errores a la hora de procesar el texto enviado en los correos se logra que este se envíe de forma correcta. Siendo un correo como el que se ve en la imagen el que reciba el usuario



Una vez la búsqueda y modificación de las gestiones esta terminada se procede a pensar en como se mostraran todas las gestiones y como se filtrara la información. Se evalúan dos opciones una que filtra en función de un formulario que generara una consulta especifica a la base de dato y otra que es utilizar JavaScript para mostrar toda la información por pantalla y filtrarla de forma dinámica. Después de sopesar las dos opciones la escogida es la segunda ya que es mas sencilla y rápida de implementar y en este momento del proyecto falta poco tiempo para la fecha de entrega.

Para esta opción se escoge utilizar un script llamado DataTables que permite realizar búsquedas en una tabla con datos y generar un paginado. Una vez implementado el script la sección de consulta queda tal y como se ve en la imagen permitiendo realizar una búsqueda de forma rápida y sencilla.

Secretaría online Inicio

Show  entries Search:

Código	Fecha	DNI/NIE	Gestión	Estado	Link
20190516_172613_53332414z_Inscripción	2019-05-16 17:26:13	53332414z	Inscripción	en proceso	Ver Gestión
20190516_172646_54447169J_Convalidación FCT	2019-05-16 17:26:46	54447169J	Convalidación FCT	En proceso	Ver Gestión
20190523_172435_53331624j_Inscripción	2019-05-23 17:24:35	53331624j	Inscripción	En proceso	Ver Gestión
20190523_173910_53331624j_Inscripción	2019-05-23 17:39:10	53331624j	Inscripción	En proceso	Ver Gestión
20190523_174158_53331624j_Inscripción	2019-05-23 17:41:58	53331624j	Inscripción	En proceso	Ver Gestión
20190523_174212_53331624j_Inscripción	2019-05-23 17:42:12	53331624j	Inscripción	En proceso	Ver Gestión
20190529_170556_53332414z_Convalidación	2019-05-29 17:05:56	53332414z	Convalidación	denegada	Ver Gestión
20190530_172325_123456789i_Inscripción	2019-05-30 17:23:25	123456789i	Inscripción	En proceso	Ver Gestión

Showing 1 to 8 of 8 entries Previous  Next

### Testeo de la segunda parte de la aplicación:

Una vez terminada la programación de la parte del administración se comienzan a realizar una serie de pruebas individuales en busca de fallos y errores en la aplicación. Durante este proceso se produce una actualización de la base de datos que genera un error grave que provoca que la fase de testeo de alargue durante dos semanas hasta que se logra corregir el error.

### Testeo de la aplicación en global una vez finalizada:

Para esta parte del testeo se seleccionan diferentes personas a las que se le asignan diferentes roles ( usuario, administradora) y se comienza una fase de pruebas en las que estos generan gestiones. Se corrigen pequeños errores durante todo este proceso logrando llegar a que las dos partes de la aplicación funcionen correctamente.

## Creación de la memoria del proyecto:

Durante todo este proceso se a ido elaborando la memoria para así poder documentar de forma correcta los errores graves surgidos durante el desarrollo.

## 2.2-Imprevistos surgidos.

### 1-error en las migraciones.

Se encuentra un error al intentar utilizar las migraciones de laravel en el servidor final encontrándose el siguiente error:

```
oficinavirtual@oficinavirtual:/var/www/html/MyProject$ php artisan migrate

Illuminate\Database\QueryException : SQLSTATE[HY000] [1045] Access denied for user 'laravel'@'localhost' (using password: YES) (SQL: select * from information_schema.tables where table_schema = pruebas and table_name = migrations)

at /var/www/html/MyProject/vendor/laravel/framework/src/Illuminate/Database/Connection.php:664
660|         // If an exception occurs when attempting to run a query, we'll format the error
661|         // message to include the bindings with SQL, which will make this exception a
662|         // lot more helpful to the developer instead of just the database's errors.
663|         catch (Exception $e) {
> 664|             throw new QueryException(
665|                 $query, $this->prepareBindings($bindings), $e
666|             );
667|         }
668|

Exception trace:
 1  PDOException::__construct("SQLSTATE[HY000] [1045] Access denied for user 'laravel'@'localhost' (using password: YES)")
   /var/www/html/MyProject/vendor/laravel/framework/src/Illuminate/Database/Connectors/Connector.php:70
 2  PDO::__construct("mysql:host=127.0.0.1;port=3306;dbname=pruebas", "laravel", "laravel", [])
   /var/www/html/MyProject/vendor/laravel/framework/src/Illuminate/Database/Connectors/Connector.php:70

Please use the argument -v to see more details.
```

se soluciona cambiando la ip por localhost en los siguientes archivos:

.env

DB\_HOST → localhost

database.php

mysql  
host → localhost.

Una vez echo esto se recarga la cache con los siguientes comandos.

php artisan cache:clear  
php artisan config:cache

web donde se encuentra la [solucion](#).

Tiempo empleado en solucionar el error : 4 horas.

## **2-Error redirección PHP**

Para solucionar el error de redirección que ocurre cuando intentas navegar por la aplicación se necesita activar el modulo rewrite de apache con el siguiente comando.

```
sudo a2enmod rewrite
```

Una vez activado este modulo reiniciaremos el servicio y comprobaremos que podemos navegar perfectamente por laravel.

```
systemctl restart apache2
```

```
systemctl status apache2
```

Tiempo empleado en solucionar el error : 2 horas.

## **3-Error de permisos en las vistas.**

Link de la solución [aquí](#).

Si una vista se bloquea y salta un error de que esta no tiene permisos lo primero es comprobar el permiso de ese archivo.

1. Se cambia el permiso si no esta bien.
2. Si el problema persiste se utiliza el siguiente comando `php artisan view:clear` para refrescar la cache de las vistas .

Tiempo empleado en solucionar el error : 1 horas.

## **4-Error del cifrado de contraseña del usuario de MySQL**

Este error a sido el mas difícil de solventar y ocurrió durante una actualización de la versión de la base de datos.

Al actualizar Mysql a la versión 8 todas las contraseñas de los usuarios de la BDD se cifraron con SHA-2 pero laravel no tiene la opción de cifrar las contraseñas.

Se investiga como se puede solucionar el error ya que la opción de volver a la versión anterior no es posible sin perder la base de datos.

Después de investigar mucho y leer la pagina oficial se descubre el que cifrado anterior no a desaparecido y se puede seguir utilizando. Se intenta configurar la el cifrado por defecto de las contraseñas en los archivos de configuración pero no se logra el resultado deseado.

Se intenta cambiar la contraseña del usuario actual cambiando el tipo de cifrado al anterior pero no se consigue . Después de platearse la opción de sacrificar la BDD y tenerla que volver a programar se intenta crear un nuevo usuario forzando la contraseña al cifrado anterior con el siguiente comando.

```
CREATE USER 'usuario'@'localhost' IDENTIFIED WITH mysql_native_password BY 'password';
```

Una vez creado el nuevo usuario se comprueba a traves de PHPMyadmin que se puede iniciar sesión correctamente se configura el nuevo usuario en laravel , se reinicia la cache para asegurarse que cargara la nueva configuración y se comprueba que el error se a solucionado y las consultas se generan correctamente a la base de datos.

Tiempo empleado en solucionar el error : 2 semanas.

## 3-Conclusiones.

### 3.1-La experiencia.

Esta experiencia a sido enriquecedora ya que he podido comprobar como seria el trabajar en un proyecto con un cliente desde el inicio con las reuniones las demostraciones de código y los plazos de entrega de estos .

También me a permitido familiarizarme con una de los frameworks mas utilizados actualmente en el desarrollo de aplicaciones web , y ampliar aun mas los conocimientos de PHP que tenia actualmente. También me a permitido tener nociones de JavaScript ya que a sido necesario implementarlo en el proyecto en puntos muy en concreto .

Aunque en algunos puntos del desarrollo y con ciertos errores ha sido un poco agobiante la experiencia en si del proyecto y el ver que al final funciona tal y como quería a sido gratificante en general .

### 3.2-Posibles mejoras del proyecto.

Las posibles mejoras del proyecto serian :

1. Creación de usuarios: el poder crear usuarios y asignarles roles para que estos al iniciar sesión les lleve a la sección correspondiente de la aplicación.
2. Inserción automática de gestiones y estados en la BDD a traves de un archivo xml desde la propia aplicación.
3. Posibilidad de descarga de la información de la BDD seleccionando criterios de filtrado desde la aplicación( por ejemplo que genere un documento con todas las inscripciones generadas o las modificaciones de matricula).

También cambios mínimos a nivel estético para poderla hacer aun mas atractiva e intuitiva la aplicación.

## 4-Bibliografía

Las webs consultadas durante el proyecto son las siguientes:

<https://www.neoguias.com/mejores-frameworks-gratuitos-de-php/>

<https://docs.laraveles.com/docs/5.5>

<https://www.digitalocean.com/community/tutorials/how-to-install-and-use-composer-on-ubuntu-18-04>

<https://styde.net/laravel-5/>

<https://stackoverflow.com/questions/35036547/mysql-server-has-gone-away-error-during-installing-migration-laravel>

<https://programacionymas.com/blog/como-enviar-mails-correos-desde-laravel>

<https://rimorsoft.com/enviar-correos-a-gmail-desde-laravel-56-2018-solucionado>

[http://www.java2s.com/Tutorial/Java/0180\\_File/](http://www.java2s.com/Tutorial/Java/0180_File/)

<ReadingobjectsfromfileusingObjectInputStream.htm>

<https://styde.net/componentes-dinamicos-para-formularios-con-blade-y-laravel/>

<https://datatables.net/>

<https://getbootstrap.com/>

<https://gist.github.com/FezVrasta/16c5d5e5ff1211922ddcf090c8454d74>

<https://www.tutsmake.com/laravel-php-simple-qr-codes-generate/>

<https://www.sceditor.com/>

## 5-Código fuente.

Enlace al para descargar el código fuente [aquí](#).