



UNIVERSIDAD CATÓLICA
"NUESTRA SEÑORA DE LA ASUNCIÓN"
CAMPUS UNIVERSITARIO DE ITAPÚA
Facultad de Ciencias y Tecnología - Ingeniería Informática

Calificaciones estudiantes

Alumno: Fabio Gamaliel Saracho Franco.

Curso: Cuarto.

Profesor: Javier Acosta.

Año: 2018

Nombre del Proyecto

Calificaciones estudiantes.

Resumen Ejecutivo

Sistema para la administración de las calificaciones de los alumnos de la carrera de ingeniería informática, será una web en donde los mismos docentes o alumnos podrán registrarse y poder acceder a las funcionalidades del sistema, el alumno podrá acceder y ver todas sus calificaciones obtenidas que serán cargadas por el docente, lo que lo hace diferente a este sistema que los alumnos podrán hacer uso de sus teléfonos inteligentes, ya que serán capaces de instalar la aplicación como si fuera una aplicaciones nativa en sus teléfonos y también, todo esto se podrá lograr gracias a que la construcción del sistema será basado en progressive web app (aplicaciones web progresivas).

Fundamentación

Se pretende desarrollar este sistema ya que han notados inconvenientes que tienen los alumnos en cuanto a la entrega de sus calificaciones, en muchos casos los alumnos tienen que llegar a la facultad para ingresar al sistema con el que cuenta la universidad, esto es muy tedioso para cualquier alumno que quiera ver su calificación de alguna materia en específica, pero es aún muchísimo más tedioso para aquellos alumnos que viven lejos de la universidad, ya que tiene que viajar, lo cual pierden tiempo y dinero, una alternativa de solución es llamar a secretaria para ver si hay nota, pero por motivos de que los funcionarios de secretaría están muy ocupados con otra actividad dejan pasar por alto la solicitud que desean saber los alumnos.

Naturaleza del Proyecto

Sistema para la administración de las calificaciones para los alumnos de la carrera de ingeniería en informática, esta idea nació gracias a que se ha notado una gran dificultad de los alumnos a la hora de saber sus calificaciones, ya que en muchos casos las fechas de las oportunidades para rendir tienen un límite para pagar 72 horas antes, y los alumnos se encontraban en casos en que las notas aún no estaban dentro del sistema haciéndoles entrar en gran duda si es que pasaron o no la materia, y en que si tienen que pagar o no la a las oportunidades siguientes a rendir, esta investigación se ha realizado y llegado a una conclusión por medio de encuestas a los alumnos de primer curso hasta quinto curso de la carrera de ingeniería en informática.

Visión

Lograr que los alumnos de la carrera en ingeniería informática puedan cursar sus estudios con mayor agilidad en cuanto a la información y notificación de sus calificaciones obtenidas durante los semestres, logrando así un mayor desempeño y confianza de los alumnos hacia la universidad.

Objetivos

- **Generales**

Crear un sistema para la distribución de calificaciones a los alumnos que responda con las necesidades encontradas dentro de la carrera de ingeniería en informática.

- **Específicos**

a) **Información:** Calificaciones según su desempeño durante los exámenes finales.

b) **Interactividad:** Los usuarios podrán acceder a la aplicación web y registrarse con sus datos personales, y podrán acceder al mismo para ver la lista de calificaciones que han sido cargados por los docentes de alguna materia en específica.

c) **Servicios:** Se hará uso del sistema de base de datos mongodb para recolectar, almacenar, modificar y recuperar toda la información generada de la organización.

El Producto o Servicio

Dentro de carrera de ingeniería en informática los alumnos se han encontrados con muchas dificultades a la hora de saber sus calificaciones obtenidas de sus exámenes finales, ya que tiene que hacer uso del sistema con el que cuenta la universidad, en muchos casos no es de gran dificultad para los alumnos revisar el sistema si ellos están dentro de por algunas actividades comunes, pero el problema comienza cuando algunos alumnos viven lejos de la universidad, y tienen que viajar únicamente para saber si aprobaron o no algunas materias en específico, lo cual los hace perder tiempo y dinero. Una solución a este problema consiste en crear una aplicación web para que los alumnos puedan acceder al mismo para mirar sus calificaciones obtenidas, desde cualquier lugar en donde se encuentren por medio de una computadora o teléfono, dentro del sistema los alumnos y los docentes se podrán registrar de manera libre y gratuita, también los

usuarios de dividirán en tres roles diferentes, docente que será el encargado de cargar las calificaciones de los alumnos según la materia que le corresponda dictar durante el semestre, administrador que será el encargado como su nombre lo indica en administrar los usuarios y todo el sistema en general, tendrá la responsabilidad de asignar como administradores del sistema todos los usuarios que quiera, y como último rol, el de alumno pero no por eso el menos importante, este rol se asigna por defecto a los usuarios cuando se registren dentro del sistema, los alumnos podrán acceder a sus datos personales, como también a sus calificaciones obtenidas. Lo que tendrá de diferente este sistema del resto es que los alumnos o docentes podrán instalar la aplicación en teléfonos como si fuese una aplicación nativa, teniendo la posibilidad de acceder al sistema por medio de un acceso directo que se mostrará en la pantalla de sus teléfonos, esto es realmente ágil a la hora de acceder al sistema ya que es muy tedioso tener que entrar al navegador cada tanto para acceder al sistema de notas, otro punto muy importante a tener en cuenta es que al no ser una aplicación móvil nativa los usuarios no tienen que hacer uso del almacenamiento de sus teléfonos, ya que en muchos casos los usuarios cuentan con muchas aplicaciones nativas instaladas haciendo que tengan memoria insuficiente de almacenamiento.

Ambiente de Hardware y Software

En cuanto hardware la aplicación podrá ser desplegada en cualquier servidor que soporte nodejs y mongodb, una posibilidad sería hacer uso la plataforma heroku que es un servicio en la nube para la administración de aplicaciones. En cuanto a software se hará uso de la arquitectura **cliente-servidor**, del lado del servidor se hará uso del lenguaje de programación JavaScript por medio de la plataforma nodejs lo cual permitirá correr javascript del lado del servidor, también junto de la mano de mongodb para almacenar los datos mediante vayan interactuando los usuarios con la aplicación. Luego del lado del cliente se hará uso de reactjs y reduxjs para mostrar las vistas a los usuarios.

Fuentes Referenciales

<https://developers.google.com/web/progressive-web-apps/>

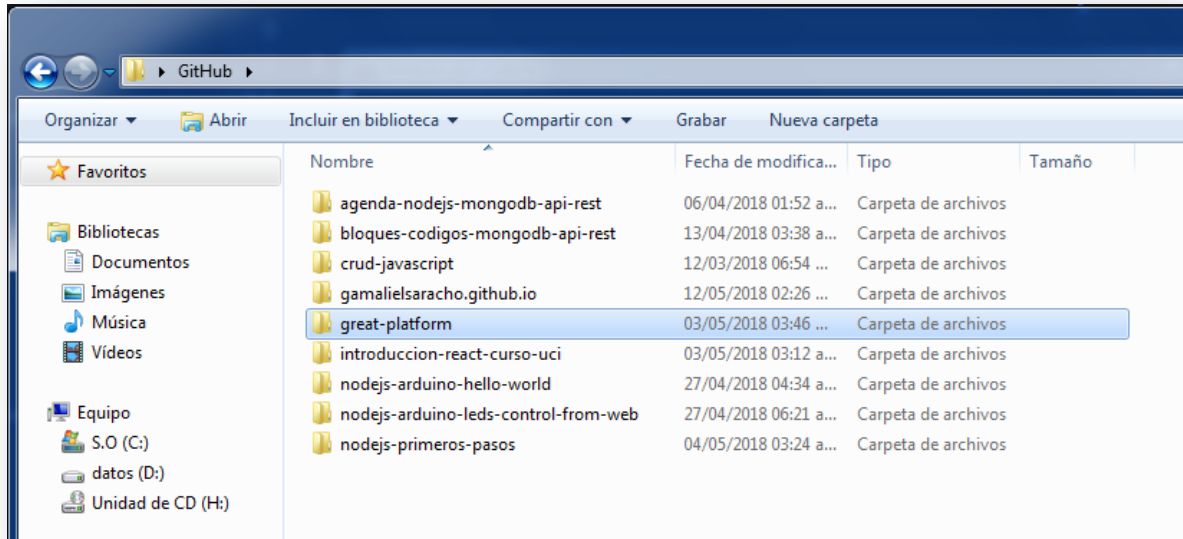
https://developer.mozilla.org/es/docs/Web/API/Service_Worker_API

<https://docs.mongodb.com/manual/crud/>

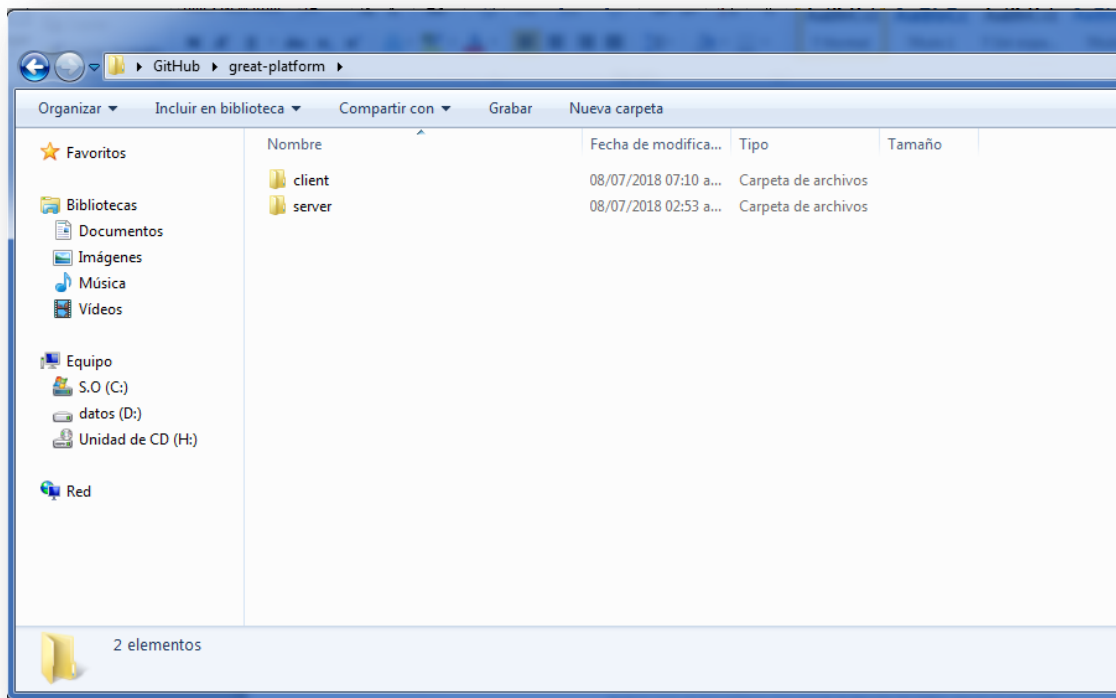
<https://www.tutorialspoint.com/nodejs/index.htm>

Funcionamiento del sistema

Entramos dentro de la carpeta great-platform donde se encuentre el proyecto.



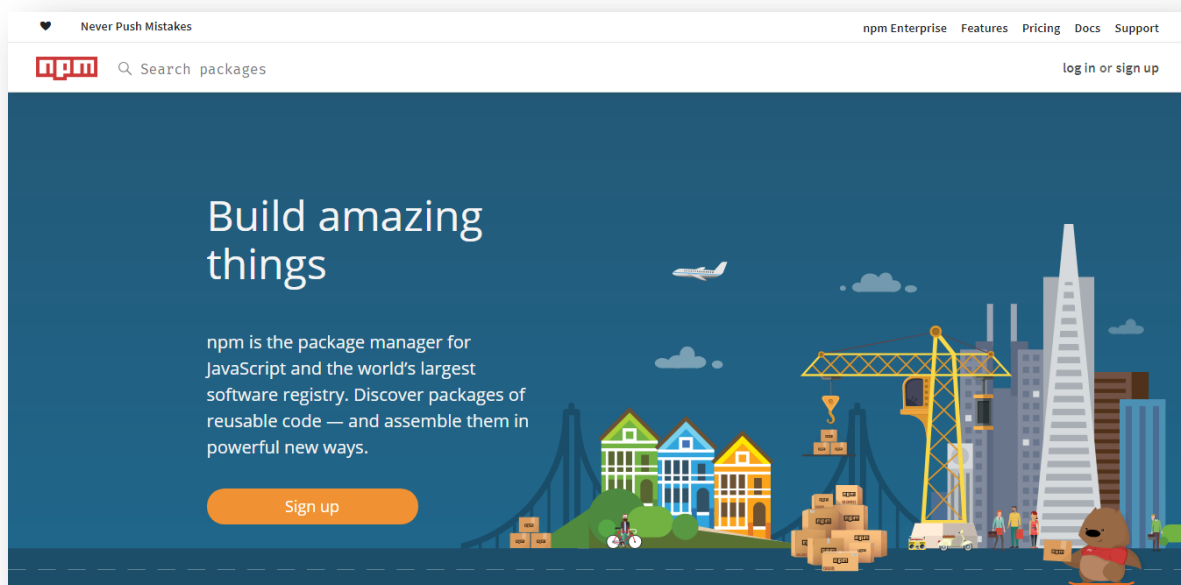
En la carpeta server se encuentra la programación de nuestro backend. Mientras que es client la programación de nuestro frontend.



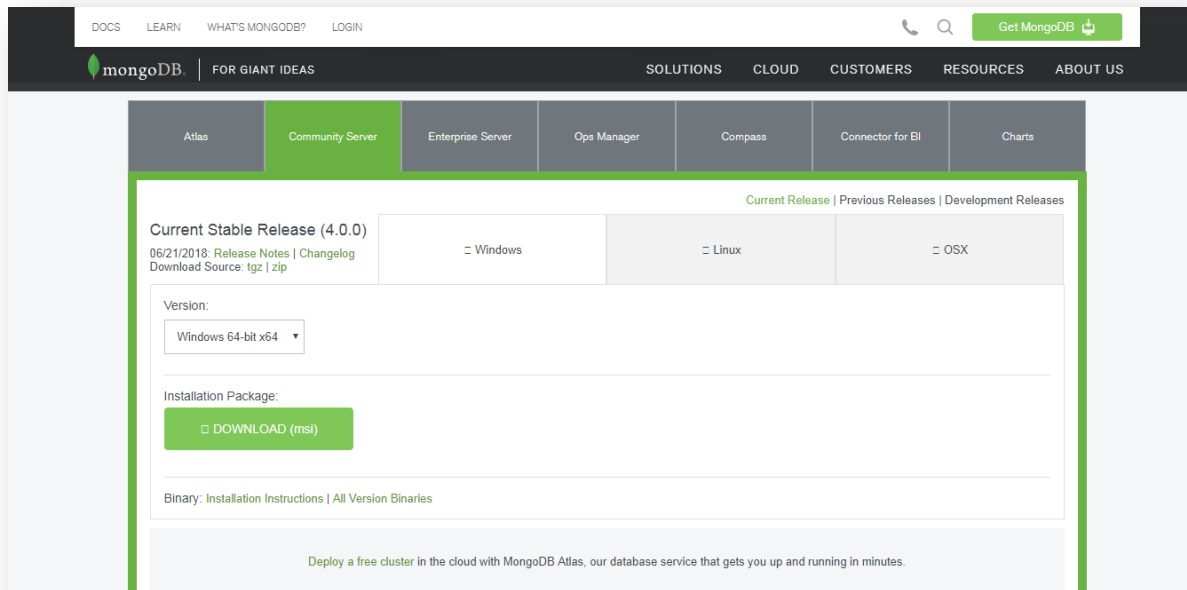
Para que el proyecto pueda ejecutarse y probarse sin problemas es necesario descargar nodejs y como también instalarlo.



Una vez que tenemos instalado nodejs vendrá también el gestor de paquetes de node, en donde podremos encontrar diferentes librerías.



Usaremos mongodb para almacenar los datos, lo cual tendremos que descargar e instalar dentro de nuestra computadora, se puede descargar desde su página web oficial.



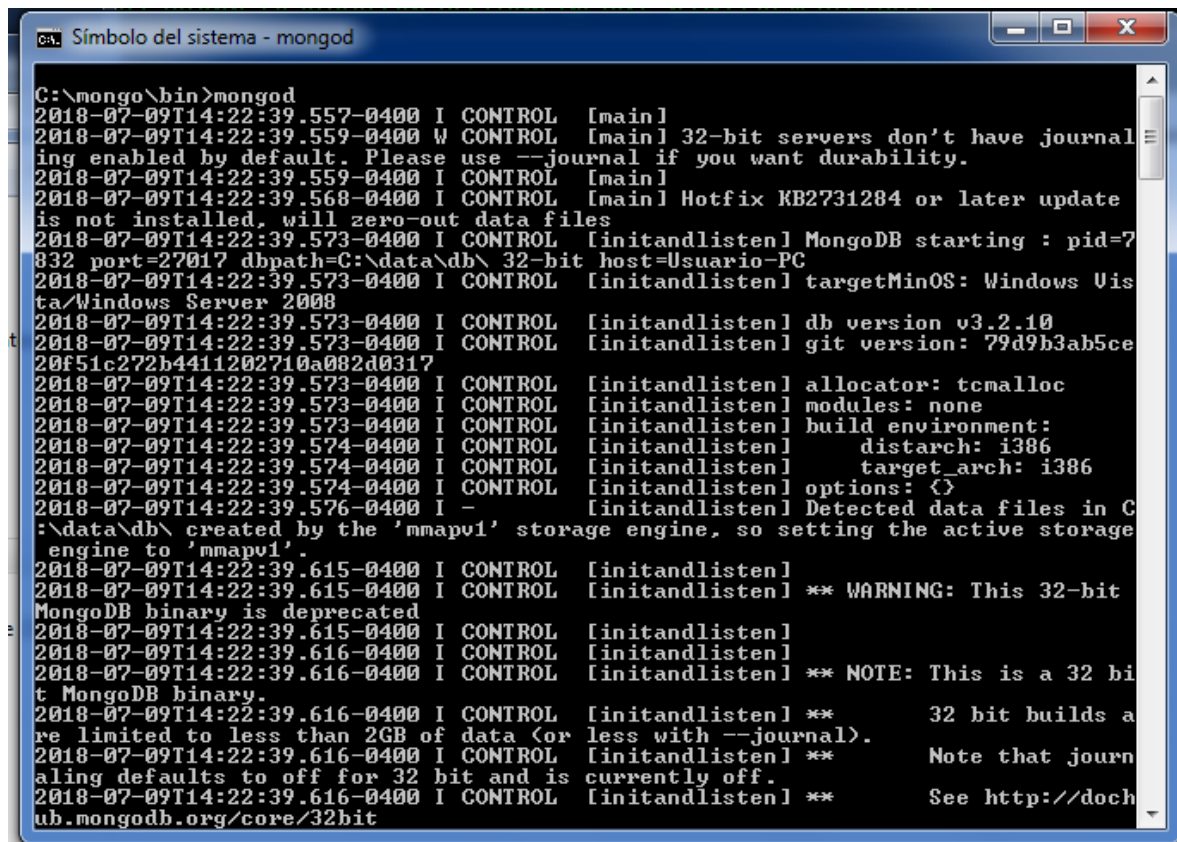
Antes de comenzar es necesario instalar las dependencias tanto para nuestro servidor como nuestro cliente, y lo hacemos de la siguiente manera:

Entramos a `\great-platform\client` y a `\great-platform\server` desde la línea de comandos y ejecutamos.

npm install

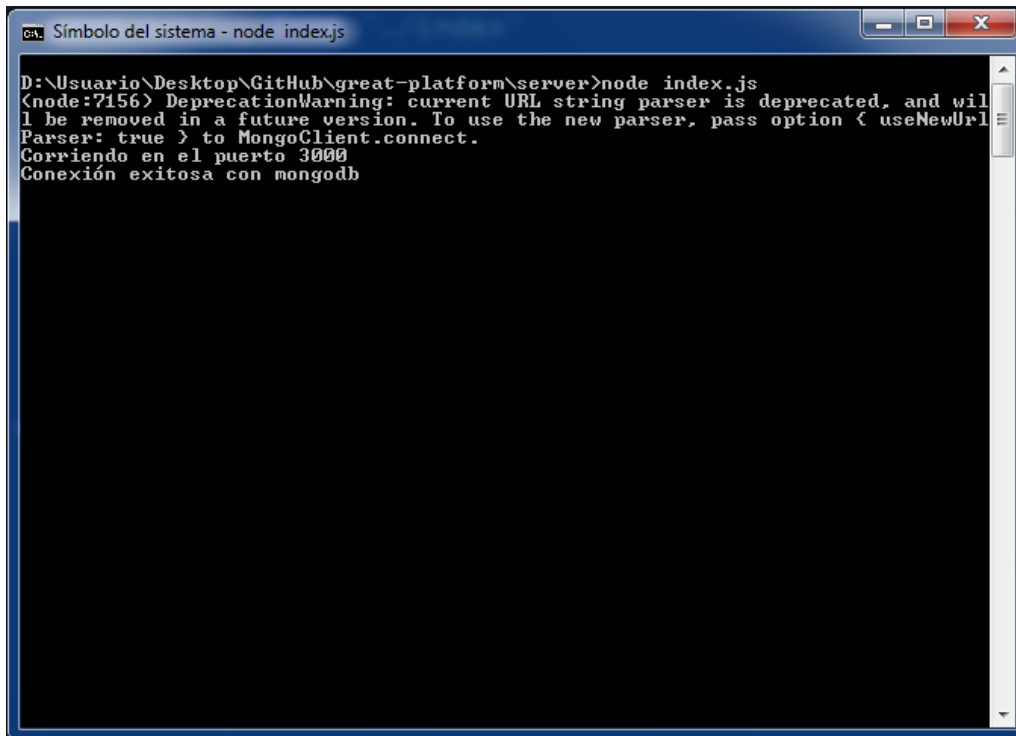
con esto se nos descargarían e instalarías todo lo que necesitamos para hacer que nuestra aplicación funciones correctamente.

Una vez que tenemos instalado mongo pasamos a posicionarnos en la dirección en la que fue instalado y entramos a la carpeta bin, y corremos el comando `mongod`, con esto haremos correr nuestra base de datos para hacer uso del mismo.



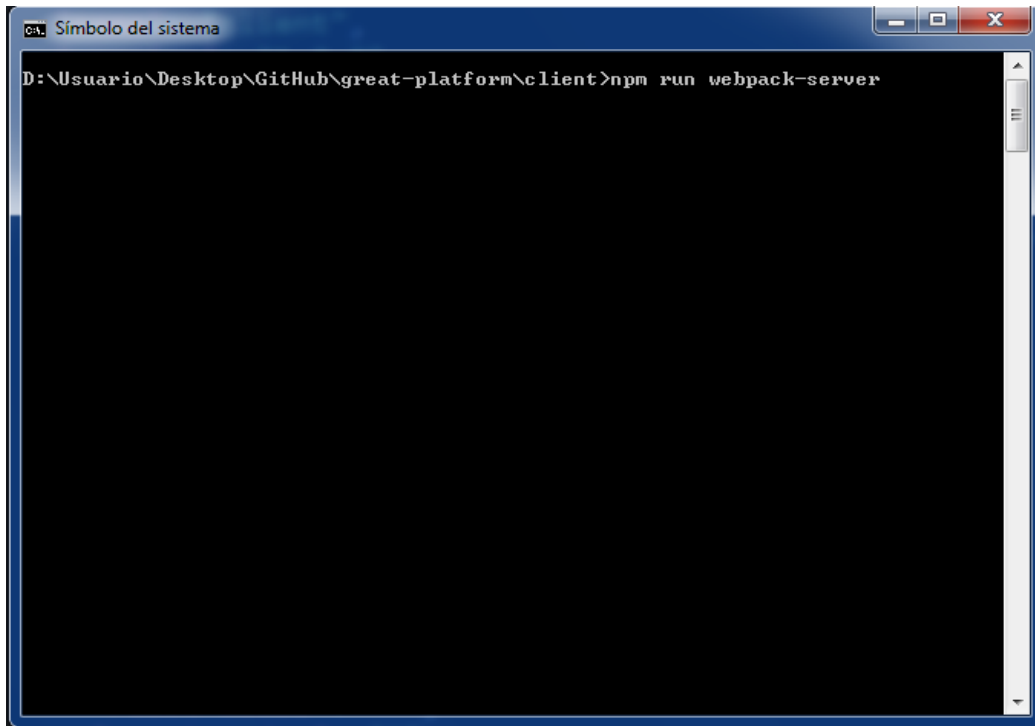
```
C:\> Símbolo del sistema - mongod
C:\> mongod
2018-07-09T14:22:39.557-0400 I CONTROL [main]
2018-07-09T14:22:39.559-0400 W CONTROL [main] 32-bit servers don't have journal
ing enabled by default. Please use --journal if you want durability.
2018-07-09T14:22:39.559-0400 I CONTROL [main]
2018-07-09T14:22:39.568-0400 I CONTROL [main] Hotfix KB2731284 or later update
is not installed. will zero-out data files
2018-07-09T14:22:39.573-0400 I CONTROL [initandlisten] MongoDB starting : pid=7
832 port=27017 dbpath=C:\data\db\ 32-bit host=Usuario-PC
2018-07-09T14:22:39.573-0400 I CONTROL [initandlisten] targetMinOS: Windows Vis
ta/Windows Server 2008
2018-07-09T14:22:39.573-0400 I CONTROL [initandlisten] db version v3.2.10
2018-07-09T14:22:39.573-0400 I CONTROL [initandlisten] git version: 79d9b3ab5ce
20f51c272b4411202710a082d0317
2018-07-09T14:22:39.573-0400 I CONTROL [initandlisten] allocator: tcmalloc
2018-07-09T14:22:39.573-0400 I CONTROL [initandlisten] modules: none
2018-07-09T14:22:39.573-0400 I CONTROL [initandlisten] build environment:
2018-07-09T14:22:39.574-0400 I CONTROL [initandlisten] distarch: i386
2018-07-09T14:22:39.574-0400 I CONTROL [initandlisten] target_arch: i386
2018-07-09T14:22:39.574-0400 I CONTROL [initandlisten] options: {}
2018-07-09T14:22:39.576-0400 I - [initandlisten] Detected data files in C
:\data\db\ created by the 'mmapv1' storage engine, so setting the active storage
engine to 'mmapv1'.
2018-07-09T14:22:39.615-0400 I CONTROL [initandlisten]
2018-07-09T14:22:39.615-0400 I CONTROL [initandlisten] ** WARNING: This 32-bit
MongoDB binary is deprecated
2018-07-09T14:22:39.615-0400 I CONTROL [initandlisten]
2018-07-09T14:22:39.616-0400 I CONTROL [initandlisten]
2018-07-09T14:22:39.616-0400 I CONTROL [initandlisten] ** NOTE: This is a 32 bi
t MongoDB binary.
2018-07-09T14:22:39.616-0400 I CONTROL [initandlisten] ** 32 bit builds a
re limited to less than 2GB of data (or less with --journal).
2018-07-09T14:22:39.616-0400 I CONTROL [initandlisten] ** Note that journ
aling defaults to off for 32 bit and is currently off.
2018-07-09T14:22:39.616-0400 I CONTROL [initandlisten] ** See http://doch
ub.mongodb.org/core/32bit
```


Luego entramos a la siguiente dirección \great-platform\server, para correr nuestra aplicación en el puerto 3000. Por medio de node index.js



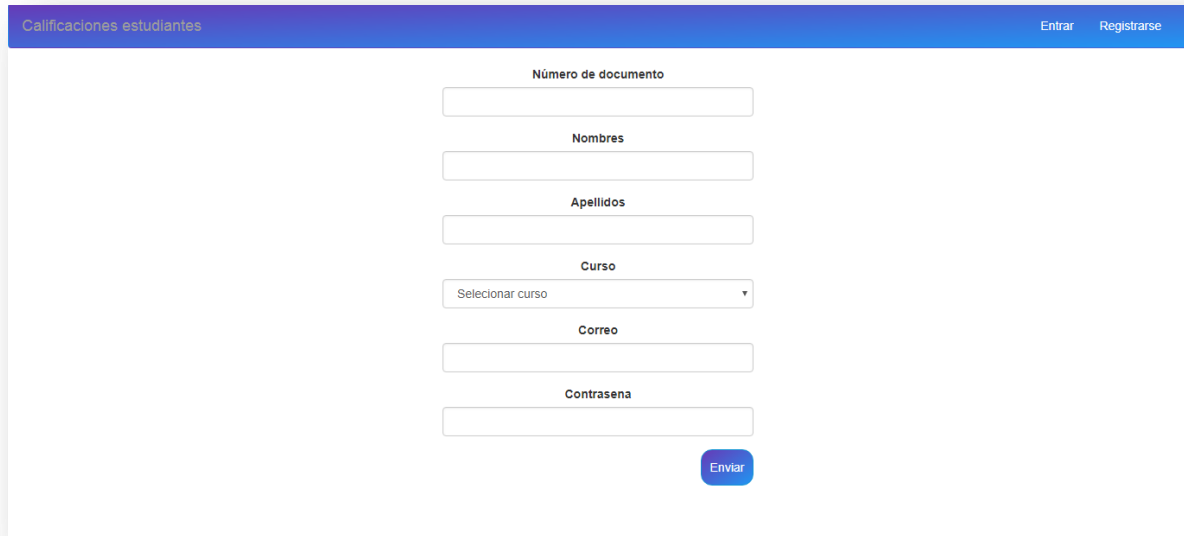
```
Símbolo del sistema - node index.js
D:\Usuario\Desktop\GitHub\great-platform\server>node index.js
(node:7156) DeprecationWarning: current URL string parser is deprecated, and will
be removed in a future version. To use the new parser, pass option { useNewUrl
Parser: true } to MongoClient.connect.
Corriendo en el puerto 3000
Conexión exitosa con mongodh
```

Una vez que tenemos nuestro servidor corriendo, ahora tendremos que dirigirnos a la siguiente dirección `\great-platform\client`, y corremos el comando **npm run webpack-server**, esto lo que hará será correr nuestra parte del cliente en donde se encuentran las vistas que se mostrará al usuario, en **localhost:8080**.



```
Símbolo del sistema
D:\Usuario\Desktop\GitHub\great-platform\client>npm run webpack-server
```

Luego para probar la aplicación nos vamos a localhost:8080, al irnos a **localhost:8080/registrarse** se mostrará el lugar en donde podrán registrarse los usuarios.



The image shows a web application interface for 'Calificaciones estudiantes'. At the top, there is a blue header bar with the text 'Calificaciones estudiantes' on the left and two links, 'Entrar' and 'Registrarse', on the right. Below the header, the registration form is centered. It consists of several input fields: 'Número de documento', 'Nombres', 'Apellidos', 'Curso' (a dropdown menu with 'Seleccionar curso' as the placeholder), 'Correo', and 'Contraseña'. Each field is a simple white rectangle with a thin border. At the bottom right of the form, there is a blue button with the text 'Enviar' in white.

```
Símbolo del sistema - mongo
C:\mongo\bin>mongo
2018-07-10T21:49:31.222-0400 I CONTROL [main] Hotfix KB2731284 or later update
is not installed, will zero-out data files
MongoDB shell version: 3.2.10
connecting to: test
Server has startup warnings:
2018-07-10T21:37:27.966-0400 I CONTROL [initandlisten]
2018-07-10T21:37:27.967-0400 I CONTROL [initandlisten] ** WARNING: This 32-bit
MongoDB binary is deprecated
2018-07-10T21:37:27.967-0400 I CONTROL [initandlisten]
2018-07-10T21:37:27.968-0400 I CONTROL [initandlisten]
2018-07-10T21:37:27.968-0400 I CONTROL [initandlisten] ** NOTE: This is a 32 bi
t MongoDB binary.
2018-07-10T21:37:27.969-0400 I CONTROL [initandlisten] **      32 bit builds a
re limited to less than 2GB of data (or less with --journal).
2018-07-10T21:37:27.970-0400 I CONTROL [initandlisten] **      Note that journ
aling defaults to off for 32 bit and is currently off.
2018-07-10T21:37:27.971-0400 I CONTROL [initandlisten] **      See http://doch
ub.mongodb.org/core/32bit
2018-07-10T21:37:27.971-0400 I CONTROL [initandlisten]
> show dbs
agencia      0.078GB
contabilidad 0.078GB
local        0.078GB
notas        0.078GB
notastudent 0.078GB
> use notastudent
switched to db notastudent
> db.usuarios.find()
{ "_id" : ObjectId("5b3439bdae93b91e0c6e7dd5"), "rol" : "alumno", "nroDocumento"
: 888888, "nombres" : "pablo", "apellidos" : "cabrera", "curso" : "3", "correo"
: "pablo@gmail.com", "contrasena" : "holamundo", "calificaciones" : [ { "nota"
: 5, "_id" : ObjectId("5b42668550bb330eccc81702"), "materia" : ObjectId("5b0db99
bfb03611e6850d55f"), "oportunidad" : "3º", "observaciones" : "", "docente" : Obj
ectId("5b3439a4ae93b91e0c6e7dd4"), "fechaCreacion" : ISODate("2018-07-08T19:31:1
7.674Z"), "fechaExamen" : null, "fechaActualizacion" : ISODate("2018-07-09T19:26
:30.214Z") } ], "_v" : 0 }
{ "_id" : ObjectId("5b3439a4ae93b91e0c6e7dd4"), "rol" : "admin", "nroDocumento"
: 555555, "nombres" : "juan carlos", "apellidos" : "miranda", "correo" : "gama@g
mail.com", "contrasena" : "holamundo", "calificaciones" : [ ], "_v" : 0 }
{ "_id" : ObjectId("5b42956fc57b4605f8c57afe"), "rol" : "docente", "nroDocumento"
: 333333, "nombres" : "Nidia", "apellidos" : "peralta", "correo" : "nidia@gmai
l.com", "contrasena" : "holamundo", "calificaciones" : [ ], "_v" : 0 }
{ "_id" : ObjectId("5b4283e1a3bef31c98fe6111"), "rol" : "alumno", "nroDocumento"
: 222222, "nombres" : "ruth andrea", "apellidos" : "motomori", "curso" : "4", "
correo" : "ruth@gmail.com", "contrasena" : "holamundo", "calificaciones" : [ {
"nota" : 5, "_id" : ObjectId("5b42953ec57b4605f8c57afd"), "materia" : ObjectId("5
b0e39d41e151a1af433212b"), "oportunidad" : "1º", "observaciones" : "felicitado",
"docente" : ObjectId("5b3439a4ae93b91e0c6e7dd4"), "fechaCreacion" : ISODate("20
18-07-08T22:50:38.535Z") }, { "nota" : 3, "_id" : ObjectId("5b429658c57b4605f8c5
7aff"), "materia" : ObjectId("5b3462efdfd0351df04da592"), "oportunidad" : "2º",
"docente" : ObjectId("5b42956fc57b4605f8c57afe"), "fechaCreacion" : ISODate("201
8-07-08T22:55:20.607Z") } ], "_v" : 0 }
>
```

Entramos a la consola de mongo para comprobar que se guardaron correctamente los datos en la base de datos.

show dbs muestra todas las base de datos que tenemos creados.

Importante: Una vez que corremos nuestra aplicación en node, automáticamente se nos creará nuestra base de datos si no existe aún, en este caso nos creó la base de datos llamada **notastudent**

use notastudent Este comando nos permite hacer uso de esa base de datos para poder consultas al mismo.

db.usuarios.find() Nos muestra una colección de objetos en donde está almacenada todos los usuarios registrados dentro de nuestro sistema.

Luego si queremos saber cuáles son las colecciones de datos que están dentro de nuestra base de datos simplemente hacemos uso del siguiente comando:

show collections

Importante: las colecciones se mostrarán únicamente si se guardaron datos dentro de alguna colección.

En **localhost:8080/entrar** se mostrará el formulario con el usuario podrá iniciar sesión e ingresar al sistema.

The image shows a login form for a system titled 'Calificaciones estudiantes'. The form is centered on a white background with a blue header bar. The header bar contains the text 'Calificaciones estudiantes' on the left and 'Entrar' and 'Registrarse' on the right. The form consists of two input fields: 'Correo' (Email) and 'Contraseña' (Password). Below the password field is a blue button labeled 'Enviar' (Send).

Una vez se inició sesión se le mostrará al usuarios las opciones de materias, usuarios, y el perfil de usuario que inició sesión. En este caso se muestran todas las materias que podrán crear los administradores o docentes, pero no los alumnos.

The image shows the 'Materias' (Subjects) management page. The page has a blue header bar with the text 'Calificaciones estudiantes' on the left and 'Materias', 'Usuarios', 'Perfil', and 'Salir' on the right. The main content area is titled 'Materias'. It features a form for adding a new subject with a 'Nombre' (Name) input field and 'Guardar' (Save) and 'Cancelar' (Cancel) buttons. Below the form is a blue button labeled 'Agregar' (Add). A table lists the existing subjects with their names and options to edit or delete them.

Nombre	Opciones
matemática I	<button>Editar</button> <button>Eliminar</button>
historia	<button>Editar</button> <button>Eliminar</button>
tecnología I	<button>Editar</button> <button>Eliminar</button>
investigación	<button>Editar</button> <button>Eliminar</button>
proyecto	<button>Editar</button> <button>Eliminar</button>

Cada usuario tendrá la posibilidad de editar sus datos personales en este caso el usuario Juan Carlos es el administrador, y solamente a él se le mostrará la opción de editar el rol de cualquier usuario, para así poder agregar más administradores o docentes.

Calificaciones estudiantes

Materias

Usuarios

Perfil

Salir

Número de Documento: 555555

Nombres: juan carlos

Apellidos: miranda

Correo: gama@gmail.com

Curso:

Número de documento

555555

Nombres

juan carlos

Apellidos

miranda

Curso

Seleccionar curso

Rol

admin

Correo

gama@gmail.com

Contraseña

holamundo

Guardar

Cancelar

Los docentes y administradores podrán entrar al perfil del alumno/a y editar sus calificaciones, en este caso el que inició sesión es un administrador y es por eso que se muestra la opción de editar los datos personales del alumno, de los contrarios si fuese un docente es opción no estaría disponible.

Calificaciones estudiantes

Materias

Usuarios

Perfil

Salir

Número de Documento: 222222

Editar

Nombres: ruth andrea

Apellidos: motomori

Correo: ruth@gmail.com

Curso: 4

Calificaciones

Materia

Seleccionar

Oportunidad

Seleccionar

Nota

Nota

Observaciones

Observaciones

Guardar

Cancelar

Materia	Nota	Oportunidad	Observaciones	Docente	Creado	Actualizado	Opciones
historia	5	1*	felicitado	juan carlos miranda	8-7-2018	9-7-2018	<div>Editar</div> <div>Eliminar</div>
tecnologia I	3	2*		Nidia peralta	8-7-2018	9-7-2018	<div>Editar</div> <div>Eliminar</div>

En este caso la usuaria Ruth Andrea tiene el rol de alumno, y al ir a su perfil puede ver sus calificaciones obtenidas. Y también otra cosa muy importante es que las opciones de agregar calificaciones y editar o eliminarlas quedan deshabilitados ya que es una alumna y no una administradora o docente.

Calificaciones estudiantes

Perfil

Salir

Número de Documento: 222222

Editar

Nombres: ruth andrea

Apellidos: motomori

Correo: ruth@gmail.com

Curso: 4

Calificaciones

Materia	Nota	Oportunidad	Observaciones	Docente	Creado	Actualizado	Opciones
historia	5	1°	felicitado	juan carlos miranda	8-7-2018	9-7-2018	
tecnología I	3	2°		Nidia peralta	8-7-2018	9-7-2018	