

Escuela de Ingeniería en Computación

Bases de Datos II - IC4302 - Grupo 2

**Proyecto 1: Documentación Final Primer Proyecto**

**Estudiantes:**

Stephannie Alanis

José Antonio Salas

Julio Viquez

**Profesor:** José Navas Sú

Cartago, Costa Rica

I Semestre , 2018

Las colecciones de la base de datos son las siguientes:

- Instituciones
- Escuelas
- Grupos
- Usuarios
- Asistencia
- Evaluaciones
- Mensajes
- Topics

El modelo de instituciones quedó conformado de la siguiente forma:

```
export class Institucion {
  _id?: string;
  nombre: string;
  sedes: Sede[];
  periodo : {
    año : string;
    semestre : string;}}
```

Las sedes corresponden a este modelo:

```
export class Sede
{
  nombre: string;
  id_escuelas: string[];
}
```

En `id_escuelas` se almacena el id de las escuelas que corresponden a la sede, por efectos prácticos y de independencia de programas las escuelas son únicas por cada sede, esto quiere decir que la escuela de Ing. Computación de la sede de Cartago es distinta a la escuela de Ing. Computación de la sede de San José por poner un ejemplo. Las sedes no tienen su propia colección en la base de datos, la multiplicidad de las sedes se trabaja desde el arreglo sedes de la colección Instituciones. `Id_escuelas` se refiere al ObjectId que crea mongo.

El modelo de escuelas está conformado de la siguiente manera:

```
export class Escuela {
  _id?: string;
  nombre: string;
  programas: Programa[]}
```

El modelo de programas es el siguiente:

```
export class Programa {
  codigo_programa: string;
  nombre: string;
```

```
    malla_curricular: Curso[];}
```

codigo\_programa es un código definido por el profesor para identificar cada programa de la escuela, no es generado automáticamente ya que programa no tiene su propia colección de mongo.

El modelo de curso es el siguiente:

```
export class Curso {
  codigo_curso: string;
  nombre: string;
  temas: Tema[]
}
```

codigo\_curso es un código que especifica el profesor para identificar el curso, cada codigo\_curso y nombre son únicos, no son generados automáticamente debido a que curso no tiene una colección en mongo, curso está incrustada en la colección grupos de mongo, temas es un arreglo que contiene todos los temas de ese curso.

El modelo de tema es el siguiente:

```
export class Tema {
  nombre: string;
  subtemas: string[];
}
```

Cuenta con un nombre y un arreglo que almacena todos los subtemas pertenecientes a ese Tema.

El modelo de grupo es el siguiente:

```
export class Grupo {
  _id?: string;
  numero: number;
  id_institucion: string;
  id_escuela: string;
  sede: string;
  id_profesor: string;
  profesor : string;
  cupos: number;
  periodo:{
    ano: string;
    semestre : string;
  };
  horario: [
    {
      dia: string;
      hora_inicio: string;
      hora_final: string;
    }
  ]
}
```

```

curso: {
  codigo_programa : string,
  codigo_curso: string;
  nombre: string;
}
rubros: Rubro[]
lista_estudiantes: ListaEstudiantes[];
lista_asistencia: ListaAsistencia[];
}

```

Este modelo si cuenta con una colección en la base de datos, que tiene los mismos atributos que el modelo, el atributo `_id?` es el que genera mongo automáticamente para identificar, numero se refiere al número del grupo que es asignado por el profesor, `id_institucion`, `id_escuela`, `id_profesor` son los identificadores de cada uno para poder asociarlos correctamente, sede es la sede a la que ese grupo pertenece, así como profesor se refiere al nombre del profesor que impartirá dicho grupo, cupos es la cantidad total de cupos con la que cuenta el grupo, periodo es un atributo incrustado que tiene el año y el semestre en el que se impartirá ese grupo, horario es otro atributo incrustado que es el día y la hora en que se da el curso, curso es otro atributo incrustado donde se encuentra el código del programa al que pertenece, `codigo_curso` y el nombre, estos atributos son incrustados porque cada grupo necesita de esa información y es más fácil de localizar y buscar esa información a tener varias colecciones con información repetitiva, rubros es un arreglo con los rubros de evaluación asignados por el profesor, `lista_estudiantes` es un arreglo con todos los estudiantes que matricularon en ese grupo y `lista_asistencia` es un arreglo que controla la asistencia de cada estudiante al curso.

El modelo de rubro es el siguiente:

```

export class Rubro {
  nombre: string;
  porcentaje: number;
  evaluaciones: [
    {
      id_evaluacion: number;
      nombre: string;
      fecha_entrega: Date;
      porcentaje: number;
    }
  ]
}

```

El modelo de rubro cuenta con el nombre del rubro, por ejemplo Exámenes, Proyectos, Tareas, el valor de porcentaje de cada rubro, contiene incrustado las evaluaciones, ya que cada rubro tiene evaluaciones y estas tienen un identificador único, un nombre, una fecha de entrega y el valor del porcentaje. Todos estos valores son creados por el profesor.

El modelo de lista-estudiantes es el siguiente:

```
export class ListaEstudiantes {  
  
    id_estudiante: string;  
    nombre: string;  
    evaluaciones: EvaluacionEstudiante[];  
}
```

Este modelo cuenta con el `id_estudiante` que es el identificador único de cada estudiante, que sería el carnet, el nombre del estudiante y un arreglo de evaluaciones, que son del tipo `EvaluacionEstudiante`, donde en ese arreglo se almacenan todas las evaluaciones de un estudiante específico, esto para que se facilitara la consulta de las evaluaciones de cada estudiante, debido a que cada estudiante tiene las evaluaciones del curso pero todos obtienen notas diferentes.

El modelo de lista-asistencia es el siguiente:

```
export class ListaAsistencia {  
    _id?: string;  
    id_grupo : string;  
    fecha: Date;  
    asistencia: Asistencia[];  
}
```

Este modelo cuenta con una fecha, que es el día que se dio la clase y un arreglo asistencia, que está conformado por todos los estudiantes que están en el curso, cada clase cuenta con una fecha y una lista de asistencia únicas, así como también posee el id del grupo al que pertenecen los estudiantes y el id que identifica cada lista.

El modelo de asistencia es el siguiente:

```
export class Asistencia {  
  
    id_estudiante: string;  
    nombre: string;  
    estado: string;  
}
```

Este modelo es sencillo y también cuenta con una colección en la base de datos solo cuenta con el `id_estudiante` que es el carnet con el que cuenta el estudiante, nombre del estudiante y un estado que indica si estuvo presente o ausente, esto se ocupa en el modelo

de listaAsistencia, ya que cada objeto lleva fecha y el id\_estudiante y estado, dando la facilidad de consultar la asistencia por fecha del curso.

El modelo de matrícula es el siguiente:

```
export class MatriculaComponent implements OnInit {
  session: Usuario
  cursos: any[];
  selectedCurso: string;
  grupos: Grupo[];
  nombresProfesor: string;
}
```

Este modelo tiene session que es el usuario que se encuentra conectado en el sistema, un arreglo con los cursos que puede matricular, selectedCurso que es el curso que matriculo, el grupo y el nombre del profesor que dará ese curso.

El modelo de usuario es el siguiente:

```
export class Usuario {
  _id?: string;
  nombre: string;
  tipo: string;
  carnet: string;
  password: string;
  email: string;
  institucion: {
    id_institucion : string;
    sede: string;
  }
  escuela: string;
  programa: {
    codigo_programa: string;
    nombre: string;
  }
  historial_cursos: HistorialCurso[]
}
```

Este modelo tiene una colección en la base de datos que se llama igual usuarios y cuenta con los mismos atributos, \_id? es el identificador que asigna la base de datos MongoDB, nombre es el nombre del usuario, tipo indica si el usuario es un estudiante o un profesor, carnet es el identificador del usuario ya sea estudiante o profesor, email y password son los valores que usa el usuario para iniciar sesión en la aplicación ambos deben ser únicos para cada usuario, institución se encuentra incrustado y cuenta con el identificador de la institución y la sede, esto porque cuando el usuario se registra indica la institución y la sede a la que pertenece, por lo que hacerlo de esta manera es bastante eficiente al igual que programa que también se encuentra incrustado con los valores del código del programa y el nombre, igual el usuario indica estos valores cuando se registra, escuela que es el nombre de la escuela a la que el usuario pertenece, y el historial\_cursos que es un arreglo de tipo

HistorialCurso que lleva todos los cursos llevados por el estudiante y si es un profesor son todos los cursos que ha impartido, teniendo en cuenta que un usuario profesor también puede ser un usuario estudiante pero con diferente programa y cursos.

El modelo de historial-curso es el siguiente:

```
export class HistorialCurso {
  periodo:{
    ano: string;
    semestre : string;};
  codigo_curso: string;
  nombre: string;
  estado: string;
  nota_final: number;
  id_grupo: string;
}
```

Este modelo crea cursos con su estado y nota, esta estructura sirve para poder crear objetos de este tipo y almacenarlos en otra clase, período es incrustado y tiene el año y el semestre donde se lleva ese curso, se incrusta porque son dos valores que pertenecen a periodo y que cada curso se lleva en un periodo diferente, codigo\_curso es el código del curso que se encuentra creado, así como el nombre del curso, estado es si ese curso es aprobado, pendiente, reprobado, nota\_final es la nota obtenida por el estudiante en ese curso y el id\_grupo que lo asocia con grupo para indicar el número de grupo en el que el estudiante llevó el curso.

El modelo de evaluación-estudiante es el siguiente:

```
export class EvaluacionEstudiante{
  id_evaluacion : string;
  porcentaje : number;
  nota : number;
}
```

Este modelo almacena las evaluaciones del estudiante, donde cuenta con el id\_evaluacion que se usa para identificar la evaluación que tiene asociada, así como el porcentaje y la nota obtenida por el estudiante.

El modelo de comentario es el siguiente:

```
export class Comentario {
  autor: {
    id_usuario: String;
    nombre: String;
    tipo: String;
  }
  fecha: Date;
  texto: String;
}
```

Este modelo almacena los comentarios que se dan en los foros de discusión, cuenta con una estructura incrustada del autor que lleva el id\_usuario, el nombre y el tipo de usuario que es para indicar el usuario que realizó el comentario en el foro, así como la fecha del día que lo escribió y el texto que conlleva.

El siguiente es el modelo de topic:

```
export class Topic {
  _id?: String;
  id_grupo: String;
  autor: {
    id_usuario: String;
    nombre: String;
    tipo: String;
  }
  fecha: Date;
  titulo: String;
  texto: String;
  estado: String;
  comentarios: Comentario[];
}
```

Este modelo cuenta con una colección en la base de datos mongo, y se encarga de almacenar los temas de discusión del foro, contiene un \_id que lo identifica en la base de datos, el id\_grupo que asocia al grupo al que ese foro pertenece, así como el autor que es una estructura incrustada, que cuenta con el id del usuario, el nombre y el tipo de usuario que sirve para identificar el usuario que realizó ese topic, así como la fecha, el título, el texto, el estado del topic y un arreglo de comentarios, que son todos los comentarios que se han hecho en ese tema en particular.

El modelo de mensaje es el siguiente:

```
export class Mensaje {
  _id?: String;
  id_usuario: String;
  id_destinatario: String;
  mensaje: String;
  fecha: Date;
}
```

Este modelo cuenta con una colección en la base de datos de mongo, almacena todos los mensajes que se envían los profesores y estudiantes, tiene una estructura donde \_id es el identificador que se usa en la base de datos, id\_usuario es el identificador de la persona que envía el mensaje y el id\_destinatario es el identificador de la persona que recibe el mensaje, mensaje almacena todo el cuerpo del mensaje que se envía y fecha es la fecha de cuando se envió el mensaje.



El modelo de evaluaciones es el siguiente:

```
export class evaluaciones{
  _id?: string;
  id_grupo: string;
  nombre: string;
  tipo: string;
  fecha_entrega: Date;
  porcentaje: number;
}
```

Este modelo cuenta con una colección en la base de datos de mongo, almacena todas las evaluaciones del curso, cuenta con una estructura donde el `_id` es el identificador que se usa en la base de datos, `id_grupo` es el identificador del grupo al que pertenecen esas evaluaciones, `nombre` de la evaluación, el `tipo` de la evaluación, ya sea un examen, proyecto, tarea, `fecha_entrega`, es la fecha límite que tiene el estudiante para entregar la evaluación y el valor del porcentaje que tiene la evaluación.

## Interfaz Gráfica:

La interfaz gráfica usa la herramienta de bootstrap para el diseño con estilos de los formularios, esto ayuda a hacer la navegación por la página más intuitiva y sólida. La página de inicio del sistema es la siguiente:



The screenshot shows a web application interface. At the top, there is a dark navigation bar with three links: 'Inicio Sesion' (with a key icon), 'Registro Profesor' (with a person icon), and 'Registro Estudiante' (with a person icon). Below the navigation bar, the main content area has the title 'Inicio de sesion'. Under the title, there are two input fields: 'Email' and 'Password'. Below the 'Password' field is a blue 'Login' button.

La interfaz valida que los cambios de email y password sean rellenados antes de ejecutar el botón de Login, arriba se tienen dos pestañas extras, el de Registro de Profesor y Registro de Estudiante:

## Formulario para el registro de un profesor

**Nombre Completo:**

**Carnet:**

**Email:**

**Contraseña:**

Institucion

Sede:

Escuela:

Guardar

Los campos de Institución, Sede y Escuela son de la clase dropdownlist que son rellenos con los elementos de la base de datos automáticamente, por indicaciones puntuales de la especificación, el primer campo de cada dropdownlist es una opción default en caso de no encontrar en la lista lo que buscaba, así se habilita otro campo para la creación de una nueva instancia ya sea de Institución, Sede o Escuela.

## Formulario para el registro de un profesor

**Nombre Completo:**

**Carnet:**

**Email:**

**Contraseña:**

Institucion	No se encuentra mi Institucion ▼
	Nueva Institucion
	<input type="text"/>
Sede:	No se encuentra mi Sede ▼
	Nueva Sede
	<input type="text"/>
Escuela:	No se encuentra mi Escuela ▼
	Nueva Escuela
	<input type="text"/>

Guardar

El estudiante, a diferencia del profesor, está asociado únicamente a un programa de curso, este debe ser seleccionado por medio de un dropdownlist, luego de completar los dropdownlist anteriores:

## Formulario para registro de un estudiante

Nombre:

Carnet:

Correo Electrónico:

Password:

Institucion:

Sede:

Escuela:

Programas:

Registrar Estudiante

Una vez ingresado el usuario en el módulo de Login, hay una barra de navegación para acceder a los módulos de mantenimiento, el formulario para mantenimiento de escuela es el siguiente:

## Mantenimiento Escuelas

Institucion:

Sede:

Nombre de Escuela:

Crear

Editar

Borrar

Nombre

Artes Dramáticas

Tal como los formularios anteriores, primero se elige Institución, luego Sede y se habilita el campo para escribir el nombre de la escuela a crear o editar. Una vez ingresado el nombre de la escuela, se puede crear la instancia que va a ser insertada en la base de datos, debajo de estos botones el usuario tiene la opción de escoger una instancia por medio de una tabla para editar los datos. Nos pareció una buena manera unir el formulario de creación con el formulario de editar y eliminar componentes, este patrón se repite en todo el sistema, ya que de esta forma nos ahorramos componentes innecesarios y le ahorramos molestias al usuario de buscar cada formulario que corresponda. A continuación los formularios que siguen esta misma dinámica:

#### Formulario de Programas:

Sistema de Registro

Mantenimiento ▾

LogOut

### Mantenimiento Programas

Institucion:

Sede:

Escuela:

Codigo de Programa:

Nombre de Programa:

Crear

Editar

Borrar

Codigo	Nombre
--------	--------

## Formulario de Cursos:

Sistema de Registro

Mantenimiento ▾

LogOut

### Mantenimiento Cursos

Institucion:

Sede:

Escuela:

Programa:

Codigo de Curso:

Nombre de Curso:

Crear Editar Borrar

Codigo	Nombre
--------	--------

En cuanto al formulario de temas, lo único distinto que posee es la posibilidad de añadir subtemas por cada tema que el usuario ingrese:

Sistema de Registro

Mantenimiento ▾

LogOut

### Mantenimiento Temas

Institucion:

Sede:

Escuela:

Programa:

Curso:

Nombre de Tema:

Crear Editar Borrar

Tema
subtemas

En cuanto al formulario de Grupos está conformado por una tabla que muestra los horarios que ingresa el profesor, esto permite darle mantenimiento a los horarios del grupo, ya que se pueden definir varios horarios a un grupo. También se puede visualizar todos los grupos para editarlos dentro de este mismo formulario.

## Formulario para el mantenimiento de un grupo

Programa:

Cursos:

Cupos:

Numero de grupo:

Horarios:

Dia:

Hora de inicio

Hora de fin:

Agregar Horario

Borrar horario

Seleccione el horario que desee eliminar

Dia

Hora Inicio

Hora Fin

Crear Grupo

Editar Grupo

Eliminar Grupo

Seleccione el grupo que desea darle mantenimiento

# grupo

Curso

Codigo

15

prueba3

IC-02

La ventana de mantenimiento del perfil, cuenta con los únicos tres campos que consideramos que el usuario puede modificar: el nombre, el email y la contraseña. Se rellenan los campos con los datos del usuario logueado. Se habilita la opción de guardar los campos. No consideramos que el usuario pueda cambiar su carnet, tipo de usuario, institución o sede.

## Formulario para mantenimiento de un usuario

Nombre:

Son Goku

Correo Electrónico:

goku@gmail.com

Password:

\*\*\*

Guardar Cambios

La matrícula queda de la siguiente manera:

Sistema de Registro

Mi perfil

Matricular

LogOut

## Matricula TEC

### Julio

Grupo	Curso	Profesor	Horario	Cupos	
1	IC-02 prueba3	Son Goku	Lunes : 12:45 - 13:45 Lunes : 14:45 - 12:00	40	
1	IC-01 prueba1	Son Goku	Lunes : 12:45 - 13:45	57	

Grupo	Curso	Profesor	Horario	Cupos
-------	-------	----------	---------	-------

Confirmar Matricula

La primera tabla despliega los cursos en los cuales el usuario tiene permitido matricularse. La segunda tabla muestra los cursos en los cuales el usuario ya se ha matriculado en el periodo actual.

En el mantenimiento de grupos un profesor puede editar los horarios de los grupos, así como ver los grupos y los horarios a los que pertenece.

Agregar Horario

Borrar horario

Seleccione el horario que desee eliminar

Día	Hora Inicio	Hora Fin
Lunes	12:45	13:45
Lunes	14:45	12:00

Crear Grupo

Editar Grupo

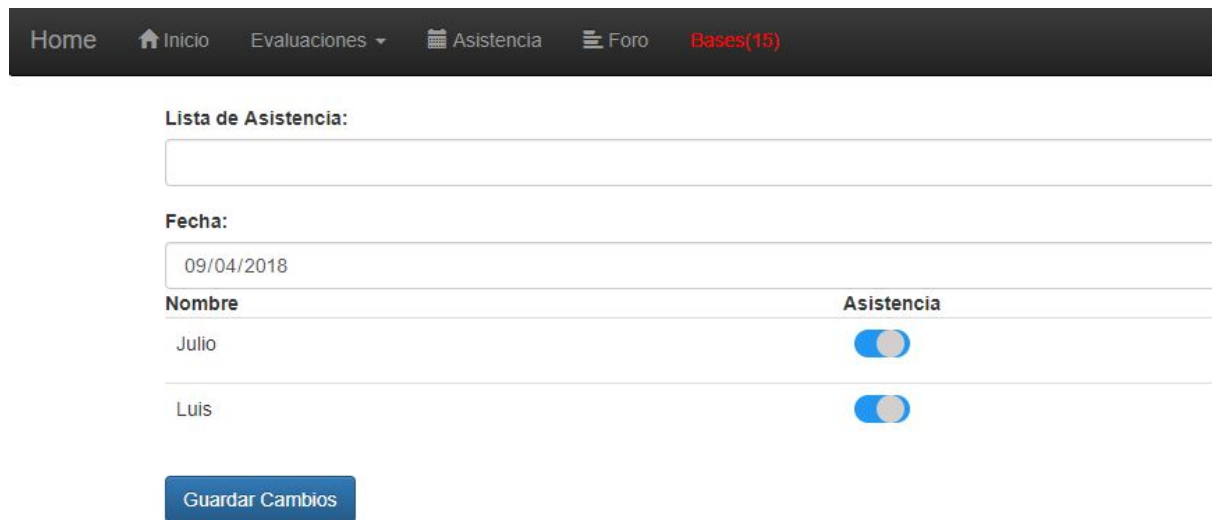
Eliminar Grupo

Seleccione el grupo que desea darle mantenimiento

# grupo	Curso	Codigo
1	prueba3	IC-02
1	prueba1	IC-01
2	prueba1	IC-01
15	Bases	05

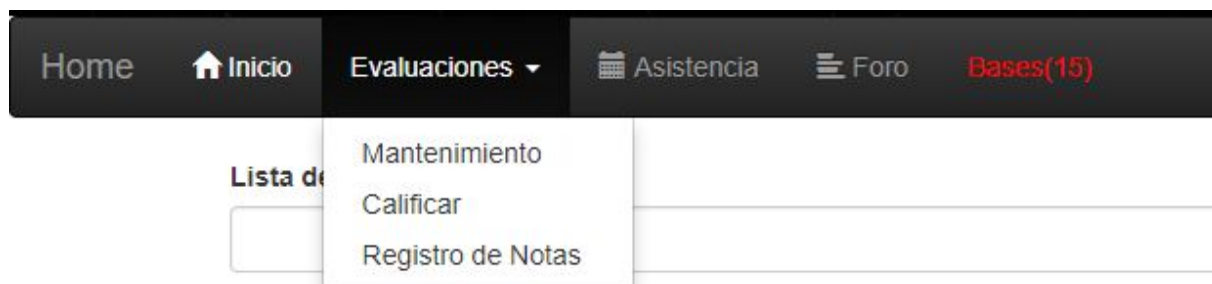


El profesor al ingresar a un grupo puede llevar el control de asistencia, selecciona la fecha si solo quiere ver la asistencia de una fecha anterior o puede crear una nueva sesión y tomar la asistencia fácilmente.

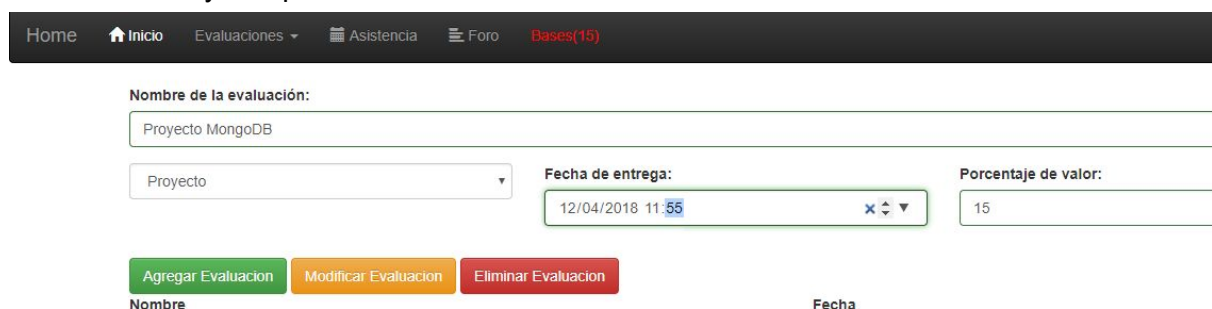


Nombre	Asistencia
Julio	<input checked="" type="checkbox"/>
Luis	<input checked="" type="checkbox"/>

En la pestaña de evaluaciones el profesor selecciona lo que desea realizar, ya sea el mantenimiento de las evaluaciones, calificar las evaluaciones asignadas o ver el registro de notas.



Al seleccionar mantenimiento, el profesor crea una nueva evaluación, al llenar todos los datos que son solicitados, como el nombre, tipo de evaluación, fecha, hora de entrega y el valor entonces ya se puede crear la evaluación.



Al seleccionar Calificar se muestran todas las evaluaciones creadas.

[Home](#) [Inicio](#) [Evaluaciones](#) [Asistencia](#) [Foro](#) [Bases\(15\)](#)

Seleccione la evaluación

Nombre	Fecha
Examen1	2018-04-01T15:48
Tarea2	2018-04-01T15:48
Proyecto 1	2018-04-01T15:45
Proyecto MongoDB	2018-04-12T11:55
Proyecto MongoDB	2018-04-12T11:55

Calificación:

Nombre	Nota (0 a 100)
<a href="#">Guardar Cambios</a>	

Al seleccionar una evaluación se despliegan los estudiantes y se les puede asignar una nota.

Seleccione la evaluación

Nombre	Fecha
Examen1	2018-04-01T15:48
Tarea2	2018-04-01T15:48
Proyecto 1	2018-04-01T15:45
Proyecto MongoDB	2018-04-12T11:55
Proyecto MongoDB	2018-04-12T11:55

Calificación:

Nombre	Nota (0 a 100)
Julio	<input type="text" value="0"/>
Luis	<input type="text" value="0"/>
<a href="#">Guardar Cambios</a>	

Al seleccionar registro de notas se puede ver el historial de cada estudiante y su porcentaje obtenido en cada evaluación, logrando generar las notas finales.

[Home](#) [Inicio](#) [Evaluaciones](#) [Asistencia](#) [Foro](#) [Bases\(15\)](#)

Nombre de estudiante	Examen1 (25)	Tarea2 (10)	Proyecto 1 (65)	Proyecto MongoDB (15)	Proyecto MongoDB (15)	Nota
Julio	11.25	2.3000000000000003	65	14.25	0	92.8
Luis	18.75	0	32.5	10.5	0	61.75

[Generar notas finales](#)

Al seleccionar foro se muestran los foros que se encuentran activos, así como también se puede crear un nuevo topic para discusión.

[Home](#) [Inicio](#) [Evaluaciones](#) [Asistencia](#) [Foro](#) [Bases\(15\)](#)

Estado	Topic	Autor	Publicado	Comentarios
Abierto	Dudas de examen	Kakaroto	2018-04-08 18:33	0
Abierto	Dudas del proyecto	Kakaroto	2018-04-09 12:54	0

**Título:**

**Texto:**

Comenzar Topic

Al seleccionar un tema se puede crear un comentario, así como ver los comentarios que se encuentran en ese foro, se puede editar o eliminar un comentario y se puede cerrar el foro.

Dudas del proyecto

Kakaroto

2018-04-09 12:54

Abierto

AbrirCerrar

En este tema se discutirán las dudas del proyecto de mongoDB

Comentarios

Kakaroto

2018-04-09 12:55

Se debe utilizar Stack MEAN

**Comentario:**

Comentar

Al seleccionar mensajes, se muestran todos los contactos del usuario a los que puede enviar un mensaje, así como también muestra las conversaciones de mensajes que ya ha tenido.

Contactos

Rodrigo Nuñez Nuñez vsurak@gmail.com

Juan Juan@gmail.com

Danny aaaaa@dsdsd.com

Maria maria@gmail.com

Luis Felipe Soto feli@gmail.com

Estudiante Prueba est

**Julio jviquez2@gmail.com**


Luis luis2

RTER

Mike mike@gmail.com

Wilson wil@gmail.com

Stephannie Alanis stef.alanis03@gmail.com

Julio 

hola

hola julio

hola profe

Mensaje

Mensaje

Enviar

En mantenimiento de temas se puede crear un nuevo tema, indicando todos los datos solicitados.

## Mantenimiento Temas

Institucion:

TEC

Sede:

Central

Escuela:

Ing. Computacion

Programa:

Programa Prueba Programa Prueba

Curso:

05 Bases

Nombre de Tema:

HyperGrahDB|

Crear

Editar


Borrar

Tema

Mongo

Oracle

subtemas



Al crear el tema y seleccionarlo se asocian los subtemas de ese tema en particular.

## Mantenimiento Temas

**Institucion:**  
TEC

**Sede:**  
Central

**Escuela:**  
Ing. Computacion

**Programa:**  
Programa Prueba Programa Prueba

**Curso:**  
05 Bases

**Nombre de Tema:**  
HyperGrahDB

[Crear](#) [Editar](#) [Borrar](#)

Tema	subtemas
Mongo	Historia
Oracle	
HyperGrahDB	

Mostrando la jerarquía de cada subtema perteneciente a ese tema en particular, y cada tema cuenta con una jerarquía de subtemas distinto.

**Curso:**  
05 Bases

**Nombre de Tema:**  
HyperGrahDB

[Crear](#) [Editar](#) [Borrar](#)

Tema	subtemas
Mongo	Aplicaciones
Oracle	Historia
HyperGrahDB	Diseñadores
	Ejemplos
	Aplicaciones

Como estudiante al matricular los cursos, selecciona la pestaña de mis grupos y se muestran todos los grupos que ha matriculado, con la nota obtenida a ese curso.

Sistema de Registro [Mi perfil](#) [Mis mensajes](#) [Mis grupos](#) [Matricular](#)

### Seleccione el grupo

Código	Nombre	Periodo	No un
05	Bases	1 semestre 2018	0
IC-02	prueba3	1 semestre 2018	0

Al seleccionar un grupo se muestran los temas y subtemas de ese grupo, así como las evaluaciones asignadas por el profesor.

HomeInicioForoBases(15)

FechaEstado

Temas y subtemas del curso:

- Mongo
  - Sharding
  - Queries
  - Security
- Oracle
  - Security
- HyperGrahDB
  - Historia
  - Diseñadores
  - Ejemplos
  - Aplicaciones

Seleccione la evaluacion que desea observar

- Examen1
- Tarea2
- Proyecto 1
- Proyecto MongoDB
- Proyecto MongoDB

Al seleccionar una evaluación se muestra el detalle de esa evaluación.

Seleccione la evaluacion que desea observar

- Examen1
- Tarea2
- Proyecto 1
- Proyecto MongoDB
- Proyecto MongoDB

Nombre:

Tipo de Evaluacion:

Fecha de Entrega:

Valor: %

Nota Obtenida (sobre 100):

Porcentaje obtenido:

Al seleccionar foro se muestran los foros activos y el estudiante puede escoger el foro del que desea participar, así como también puede crear un nuevo topic de discusión.

[Home](#) [Inicio](#) [Foro](#) [Bases\(15\)](#)

Estado	Topic	Autor	Publicado	Comentarios
Abierto	Dudas de examen	Kakaroto	2018-04-08 18:33	0
Abierto	Dudas del proyecto	Kakaroto	2018-04-09 12:54	2

**Titulo:**

**Texto:**

Comenzar Topic

Al seleccionar el foro, puede dejar un comentario y ver los comentarios de los otros usuarios que han participado de ese foro.

[Home](#) [Inicio](#) [Foro](#) [Bases\(15\)](#)

**Dudas del proyecto**  
Kakaroto  
2018-04-09 12:54  
Abierto

En este tema se discutirán las dudas del proyecto de mongoDB

**Comentarios**

Kakaroto  
2018-04-09 12:55

Se debe utilizar Stack MEAN

Stephannie Alanis  
2018-04-09 13:13

No entiendo como desea que se muestren las notas finales del usuario

**Comentario:**  

No entiendo como desea que se muestren las notas finales del usuario

Al seleccionar los mensajes, el estudiante también puede ver todos los contactos, ver los mensajes recibidos y poder enviar un mensaje.

**Sistema de Registro** [Mi perfil](#) [Mis mensajes](#) [Mis grupos](#) [Matricular](#) [LogOut](#)

**Contactos**

Rodrigo Nuñez Nuñez vsurak@gmail.com

Juan Juan@gmail.com

Danny aaaaa@dsdsd.com

Maria maria@gmail.com

Luis Felipe Soto feli@gmail.com

**Kakaroto goku**

Estudiante Prueba est

Julio jviquez2@gmail.com

Luis luis2

RTER

Mike mike@hotmail.com

**Kakaroto**

Su tarea a sido recibida

Gracias profesor

**Mensaje**

Gracias profesor

Enviar

Cuando un profesor genera las notas finales, al estudiante le aparece la nota final del curso.

**Sistema de Registro** [Mi perfil](#) [Mis mensajes](#) [Mis grupos](#) [Matricular](#) [Cerrar](#)

**Seleccione el grupo**

Código	Nombre	Periodo	No un
05	Bases	1 semestre 2018	5
IC-02	prueba3	1 semestre 2018	0



Finalmente cuando el profesor revisa una evaluación, el estudiante al seleccionar la evaluación, puede ver su nota obtenida.

[Home](#) [Inicio](#) [Foro](#) [Bases\(15\)](#)

Fecha

Estado

### Temas y subtemas del curso:

- Mongo
  - Sharding
  - Queries
  - Security
- Oracle
  - Security
- HyperGrahDB
  - Historia
  - Diseñadores
  - Ejemplos
  - Aplicaciones

### Seleccione la evaluacion que desea observar

- Examen1
- Tarea2
- Proyecto 1
- Proyecto MongoDB
- Proyecto MongoDB

Tarea corta 3:Oracle

Nombre: Tarea corta 3:Oracle

Tipo de Evaluacion: tarea

Fecha de Entrega: 2018-04-26T11:55

Valor: %

Nota Obtenida (sobre 100): 100

Porcentaje obtenido: 5