

ESCUELA POLITÉCNICA NACIONAL

Facultad de Sistemas

Tecnologías Web con JavaScript

Fecha: 13 – 06 – 2017 :GR2

Examen – 1er Bimestre

1) Terminar el curso con su correo institucional de Docker

CodeSchool es una empresa de cursos pagados y gratuitos con una interfaz dinámica y ejercicios interactivos.

Usted debe de crear una cuenta en el siguiente enlace:

https://www.codeschool.com/users/sign_up

Y luego debe de terminar el siguiente curso:

<https://www.codeschool.com/courses/try-docker>

2) Terminar el curso con su correo institucional de DevTools

En este paso debe de terminar el siguiente curso:

<https://www.codeschool.com/courses/discover-devtools>

3) Levantar un Servidor Rest en sails.js

Revisar el ejercicio que tiene que hacer dependiendo de la siguiente tabla:

	Apellido(s)	Número de Ejercicio
1	CABASCANGO VIRACocha CRISTIAN RICARDO	1
2	CARDENAS TORRES JORGE ANDRES	2
3	CASTILLO ABAD ANA CRISTINA	3
4	DAVILA SILVA DAVID ALEJANDRO	4
5	ENRIQUEZ PEREZ FREDDY FERNANDO	5
6	FLORES SANCHEZ BRYAN HAROLD	6
7	FREIRE GONZALEZ DANIEL ENRIQUE	7
8	GALLARDO ANDRADE TANIA SOLEDAD	8
9	GARZON FREIRE DAVID ALEJANDRO	9
10	GONZALEZ GAVILANEZ JOSE DAVID	10
11	GUALOTO ANDRANGO ADEMIR ALEXIS	1
12	JIMENEZ GUERRERO BRAYAN ANDRES	6
13	MENDOZA PUERTAS PATRICIO KEVIN	3
14	MORA BARCENES JULIO CESAR	4
15	MOROCHO SIMBAÑA DARWIN VINICIO	5
16	MOTOCHÉ MACAS CRISTHIAN ALBERTO	6
17	NARANJO MORENO DARIO ALEXANDER	7
18	NIETO JARA HENRY DANILO	8
19	NUÑEZ LARGO BYRON FERNANDO	9
20	OCAMPO MOGOLLON ANGEL DANILO	10
21	PACHACAMA QUINGA JONATHAN PAUL	1

Profesor: Msc. Sebastián Guerrero

22	PAREDES RUANO ANDRES DAVID	6
23	PROAÑO CHAVIZNAN JAIRO GEOVANNY	3
24	QUIMBITA QUIMBITA LILIAN RAQUEL	4
25	RAMOS ACARO DANIELA PAULINA	5
26	RECALDE CAICEDO JULIA PATRICIA	2
27	REGALADO POZO EVELYN GUISELA	7
28	SAMPEDRO MORA CARLOS EDUARDO	8
29	TINAJERO GUAJAN JOSE ANDRES	9
30	VEGA MEJIA FERNANDA MISHHELL	10
31	VERA MORALES JAIRO RICARDO	1
32	ZURITA CABASCANGO JOSE RICARDO	3

a. Ejercicio 1

Crear 1 tabla en Sailsjs de **Materia**.

La **Materia** tendrá los campos:

- NombreMateria (string)
- TopicoMateria (string)
- FechaCreacion (date)

b. Ejercicio 2

Crear 1 tabla en Sailsjs de **Pokemon**.

Relacionar de uno a muchos Entrenador con Pokemon.

El **Entrenador** tendrá los campos:

- Nombre (string)
- Region (string)
- FechaInicioMaestroPokemon (date)

c. Ejercicio 3

Crear 1 tabla en Sailsjs de **Pastelería**.

La **Pastelería** tendrá los campos:

- Nombre (string)
- Ciudad (string)
- Correo (email)

d. Ejercicio 4

Crear 1 tablas en Sailsjs de **Héroe**.

El **Héroe** tendrá los campos:

- Nombre (string)
- Castillo (string)
- Nivel (integer)

e. Ejercicio 5

Crear 1 tabla en Sailsjs de **Equipo**.

El **Equipo** tendrá los campos:

- Nombre (string)
- FechaCreacion (date)
- PaisResidencia (string)

f. Ejercicio 6

Crear 1 tabla en Sailsjs de **Artista**.

El **Artista** tendrá los campos:

- Nombre (string)
- Estilo (string)
- PaisResidencia (string)

g. Ejercicio 7

Crear 2 tablas en Sailsjs de **Usuario**.

La **Borrachera** tendrá los campos

- Motivo (string)
- LatitudDondeEmpezo (integer)
- LongitudDondeEmpezo (integer)

h. Ejercicio 8

Crear 2 tablas en Sailsjs de **Bodega**.

La **Bodega** tendrá los campos:

- Nombre (string)
- Direccion (string)
- CapacidadEnToneladas (integer)

i. Ejercicio 9

Crear 1 tabla en Sailsjs de **Celular**.

El **Celular** tendrá los campos:

- Nombre (string)
- SistemaOperativo (string)
- Version (integer)

j. Ejercicio 10

Crear 1 tabla en Sailsjs de **Agarre**.

El **Agarre** tendrá los campos

- Nombre (string)
- Veces (integer)
- DineroGastado (float)

Después de encontrar su ejercicio haga lo siguiente:

1. Cree los modelos con el comando "sails generate api modelo" (1pt)
 - a. Documentación: <http://sailsjs.com/documentation/reference/command-line-interface/sails-generate>
2. Cree los atributos dependiendo de su ejercicio (3 pt)
 - a. Documentación: <http://sailsjs.com/documentation/concepts/models-and-orm/attributes>

4) FrontEnd con Angular de su Ejercicio

1. Cree una aplicación con angular con el nombre en formato: "twj-examen-apellido-nombre" (1pt)
 - a. Ejemplo: ng new twj-examen-eguez-adrian
 - b. Documentación: <https://github.com/angular/angular-cli>
2. Importar el Bootstrap (1 pt)
3. Crear una clase (en typescript) para su ejercicio. (2 pt)
 - a. Por ejemplo, si su ejercicio es de bodegas la clase seria BodegasClass y los atributos que tendrían serían: id, nombre, dirección, capacidad, createdAt, updatedAt
4. Crear un componente para mostrar su ejercicio (por ejemplo, si es bodega para mostrar las bodegas). Dentro del componente deben de tener los siguientes ítems (4 pt):
 - a. El nombre del componente debe de ser singular de su Ejercicio (por ejemplo si es bodega el componente debería de llamarse bodegas.component.ts y los demás archivos con el mismo formato)
 - b. Dentro del componente deben de mostrarse todos los elementos del modelo en sails (por ejemplo, si es bodega debería de tener nombre, dirección y capacidad en toneladas)
5. Dentro del componente recibir un "@Input" para llenar los datos del componente (1 pt)
6. Importar el servicio Http en su componente principal (1 pt)
7. Crear el método para llamar a los datos dentro del componente principal con el evento ngOnInit
 - a. Dentro de ngOnInit utilizar el servicio Http para llamar a todos los registros desde sails.
8. Crear el método para crear nuevos datos en su ejercicio
9. Crear el html para crear datos en su ejercicio (inputs y botón) (4 pt)
10. Utilizar *ngFor para presentar los datos cargados con ngOnInit en su html (sea responsive) (2 pt)

5) Crear otra aplicación con un api

Puede utilizar cualquier api de cualquier tema (como el de star wars) <https://swapi.co> para mostrar en un componente los datos de ese tema.

1. Cree una aplicación con angular con el nombre en formato: "twj-examen-api-apellido-nombre" (1pt)
 - a. Ejemplo: ng new twj-examen-api-eguez-adrian
 - b. Documentación: <https://github.com/angular/angular-cli>
2. Importar el Bootstrap (1 pt)
3. Crear una clase (en typescript) para su api. (2 pt)
 - a. Por ejemplo, si su ejercicio es de star wars la clase seria PlanetasStarWarsClass y los atributos que tendrían serían los que de el api.
4. Crear un componente para mostrar su ejercicio (por ejemplo, si es planetas de starwars para mostrar los planetas). Dentro del componente deben de tener los siguientes ítems (6 pt):

- a. El nombre del componente debe de ser singular de su Ejercicio (por ejemplo si es planetas el componente debería de llamarse planeta.component.ts y los demás archivos con el mismo formato)
 - b. Dentro del componente deben de mostrarse 4 de los elementos del modelo que tenga su api
 - c. La presentación será calificada (hágalo de la mejor manera)
- 5. Dentro del componente recibir un “@Input” para llenar los datos del componente (1 pt)
- 6. Importar el servicio Http en su componente principal (1 pt)
- 7. Crear el método para llamar a los datos dentro del componente principal con el evento ngOnInit
 - a. Dentro de ngOnInit utilizar el servicio Http para llamar a todos los registros desde sails.