--­­­



Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Manejo e implementación de archivos

Segundo Semestre 202

Catedrático: Ing. Luis Espino

Aux: Luis Daniel Castellanos

**PROYECTO 1**

|  |  |
| --- | --- |
| **Nombre** | **Carnet** |
| Maxwellt Joel Ramírez Ramazzini | 201709328 |
|  |  |

­

**ESQUEMA CONCEPTUAL**

Diagrama

Descripción generada automáticamente­­­

**ESQUEMA LOGICO**

Diagrama

Descripción generada automáticamente

**ESQUEMA FISICO**

Diagrama, Esquemático

Descripción generada automáticamente

­

**DESCRIPCIÓN DE TABLAS**

**Estudiantes**

La tabla estudiantes , posee información única y personal de cada uno de los estudiantes registrados que se inscribe en la universidad de tal forma que se pueda obtener su información al realizar una consulta

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Nombre | Tipo | Primary | Foreign | Unique | Not Null |
| 1 | Carnet | Bigint | x |  |  | x |
| 2 | Nombres | Varchar (150) |  |  |  | x |
| 3 | Apellidos | Varchar (150) |  |  |  | x |
| 4 | FechaNac | Date |  |  |  | x |
| 5 | Correo | Varchar(150) |  |  |  | x |
| 6 | Telefono | Int |  |  |  | x |
| 7 | Dirección | Varchar (100) |  |  |  | x |
| 8 | Dpi | Bigint |  |  |  | x |
| 9 | Carrera | Int |  |  |  |  |

**Docente**

La tabla docentes , posee información única y personal de cada uno de los docentes registrados que ha sido contratado por la facultad de ingeniería de tal forma que al consultar información de un curso el cual dicho docente imparta este se muestre

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Nombre | Tipo | Primary | Foreign | Unique | Not Null |
| 1 | Carnet | Bigint | x |  |  | x |
| 2 | Nombres | Varchar (150) |  |  |  | x |
| 3 | Apellidos | Varchar (150) |  |  |  | x |
| 4 | FechaNac | Date |  |  |  | x |
| 5 | Correo | Varchar(150) |  |  |  | x |
| 6 | Telefono | Int |  |  |  | x |
| 7 | Dirección | Varchar (100) |  |  |  | x |
| 8 | Dpi | Bigint |  |  |  | x |
| 9 | REGSIIF | Int |  |  |  | x |

**Carrera**

La tabla carrera , posee datos importantes de las carreras creadas en la facultad de ingeniera, de tal forma que puedan ser acreditadas a los estudiantes

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Nombre | Tipo | Primary | Foreign | Unique | Not Null |
| 1 | Id | Integer | x |  |  | x |
| 2 | Nombre | Varchar (150) |  |  |  | x |

**HorCurso**

La tabla HorCurso, posee la información importante sobre los horarios que se presentan para cada uno de los cursos creados e impartidos por los docentes en el periodo actual.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Nombre | Tipo | Primary | Foreign | Unique | Not Null |
| 1 | IdHabCurso | Integer | x |  |  | x |
| 2 | Dia | int |  |  |  | x |
| 3 | Horario | Varchar(12) |  |  |  | x |

**HabCurso**

La tabla HabCurso posee información que se encuentra vinculada a cada uno de los cursos que han sido creados y se encuentran actualmente como habilitados en el periodo actual.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Nombre | Tipo | Primary | Foreign | Unique | Not Null |
| 1 | Id | Integer | x |  |  | x |
| 2 | CodigoCurs | Int |  |  |  | x |
| 3 | Ciclo | Varchar(2) |  |  |  | x |
| 4 | Docente | Int |  |  |  | x |
| 5 | CupoMax | Int |  |  |  | x |
| 6 | Sección | Char(1) |  |  |  | x |

**AsigMateria**

La tabla Asig Materi posee la información de cada uno de los estudiantes y la materia a la que se han asignado en el ciclo actual.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Nombre | Tipo | Primary | Foreign | Unique | Not Null |
| 1 | CodigoCurs | Integer | x |  |  | X |
| 2 | Ciclo | Varchar (2) |  |  |  | X |
| 3 | Sección | Char(1) |  |  |  | X |
| 4 | Carnet | Bigint |  | x |  | x |

**Historial**

La tabla historial posee los registros de las acciones realizadas a cada una de las diferentes tablas del sistema.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Nombre | Tipo | Primary | Foreign | Unique | Not Null |
| 1 | Id | Integer | x |  |  | x |
| 2 | Fecha | Date |  |  |  | x |
| 3 | Descripcion | Varchar(150) |  |  |  | x |
| 4 | Acción | Varchar(50) |  |  |  | x |

**DESCRIPCION API**

La API se realizo en base a las funciones requeridas por el sistema de ventas, en donde se conecta a la base de datos de dicho sistema y en este realiza la búsqueda a través de los diferentes endpoints que se pueden enviar desde POSTMAN, de tal forma que se obtenga la información requerida por el usuario dependiendo del interés de este.

**DESCRIPCION ENDPOINTS**

* **getConsultaUno:**

A través de este endpoint se busca mostrar el cliente que más ha comprado en el sistema-

* **getConsultaDos:**

A través de este endpoint se busca mostrar la lista del producto más y menos comprado en el sistema

* **getConsultaTres:**

A través de este endpoint se busca mostrar el vendedor que más ha vendido en el sistema

* **getConsultaCuatro:**

A través de este endpoint se busca mostrar el país en donde más y menos se ha vendido en el sistema.

* **getConsultaCinco:**

A través de este endpoint se busca mostrar el Top 5 de países en donde más se ha comprado en forma ascendente.

* **getConsultaSeis:**

A través de este endpoint se busca mostrar la categoría en donde más y menos se ha comprado en el sistema.

* **getConsultaSiete:**

A través de este endpoint se busca mostrar la categoría que más fue comprada por cada país en el sistema.

* **getConsultaOcho:**

A través de este endpoint se busca mostrar las ventas por mes de Inglaterra en el sistema.

* **getConsultaNueve:**

A través de este endpoint se busca mostrar le mes con más y menos ventas en el sistema

* **getConsultaDiez:**

A través de este endpoint se busca mostrar las ventas de cada producto de la categoría deportes

**MANUAL DE USUARIO**

**Levantar el servicio**

Al cargar los archivos a nuestro editor preferido, debemos acceder a la carpeta correspondiente de estos archivos:

Texto

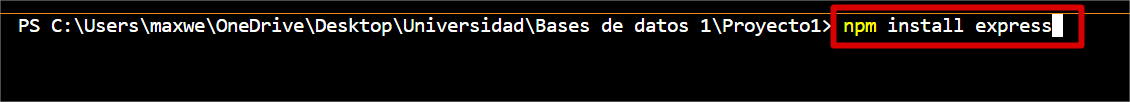
Descripción generada automáticamente

Como siguiente paso, debemos ejecutar le comando **npm install**, este instalara todas las dependencias que el proyecto necesite :

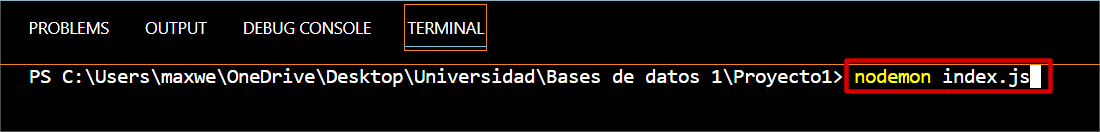
Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Como siguiente paso, debemos ejecutar le comando **npm install express**, algunas veces no suele instalarse express al ejecutar el comando anterior:



Luego de instalar todas las dependencias, ejecutaremos el comando **nodemon index.js** para correr el proyecto:



El proyecto se ejecutara y estará en modo escucha, para mostrar los resultados:

Captura de pantalla de un celular

Descripción generada automáticamente

**USO DE ENDPOINTS**

**ENDPOINTS**

A través de postman, podremos realizar el uso de los endpoints y obtener los resultados de cada uno de ellos. Accedemos a postman con el puerto 5000 con la dirección especificada en la API:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Escribimos el endpoint que querramos utilizar luego de la dirección principal, en esta caso usaremos el **getConsultaUno** y seleccionamos el botón send:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente con confianza media

Los resultados se mostrarán en la parte inferior de la consola que tiene Postman

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Podemos realizar el uso de cada endpoint de la misma forma, repitiendo los pasos anteriores.

**Codigo para encender los motores**