

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD INGENIERIA
ESCUELA DE CIENCIAS Y SISTEMAS
ANALISIS Y DISEÑO DE SISTEMAS 1
SECCION N
ING. JOSE MANUEL RUIZ JUAREZ
AUX. CESAR DIONICIO SAZO MAYEN

Manual técnico

All for one - Grupo No. 7	
Nombre	Carné
Erick Fernando Sánchez Mejía	201503878
Melyza Alejandra Rodríguez Contreras	201314821
Helmut Efraín Najarro Álvarez	201712350
Christopher Alexander Acajabon Gudiel	201404278

Guatemala, 7 de junio de 2,021

Tabla de contenido

Contenido

Objetivo general.....	4
Objetivos específicos	4
Descripción General	4
Arquitectura y flujo de la aplicación	5
Tecnologías utilizadas.....	6
Docker.....	6
Docker compose	6
Node.js	7
React JS.....	7
MongoDB	8
Git	8
Git flow.....	8
Justificación de tecnologías utilizadas.....	9
Docker.....	9
Docker compose	9
Node.js	9
React JS.....	10
Mongo	10
Git	10
Git flow.....	10
Requerimientos.....	11
Requerimientos funcionales	11
Requerimientos no funcionales	14
Requisitos	16
Hardware.....	16
Docker:	16
Node.js:	16
Visual studio code:.....	17
Software	17
Herramientas de desarrollo (instalación de tecnologías)	18
Descargar proyecto	18
Forma 1: Clonar repositorio	18

Forma 2: Descargar ZIP.....	18
Abrir proyecto con Visual Studio Code	19
Backend	19
Frontend.....	19
Backend corriendo.....	20
Frontend corriendo	20
Explicación código	21
Comandos Git.....	40
Erick Fernando Sánchez Mejia - 201503878	40
Primer release	40
Segundo release.....	43
Melyza Alejandra Rodríguez Contreras - 201314821.....	45
Primer Release	45
Segundo Release	46
Christopher Alexander Acajabon Gudiel - 201404278	51
Primer Release	51
Segundo Release	53
Helmut Efrain Najarro Alvarez - 201712350	56
Primer release	56
Segundo release.....	58
Diagrama de git Flow	60
Anexos.....	61
Ramas de corrección (hotfix).....	61
Ejemplo	62

Objetivo general

Comprender la importancia de utilizar un sistema de control de versiones en los proyectos, y entender cada uno de los elementos que conforman un sistema de este tipo para saber de qué forma se almacena y estructura la información.

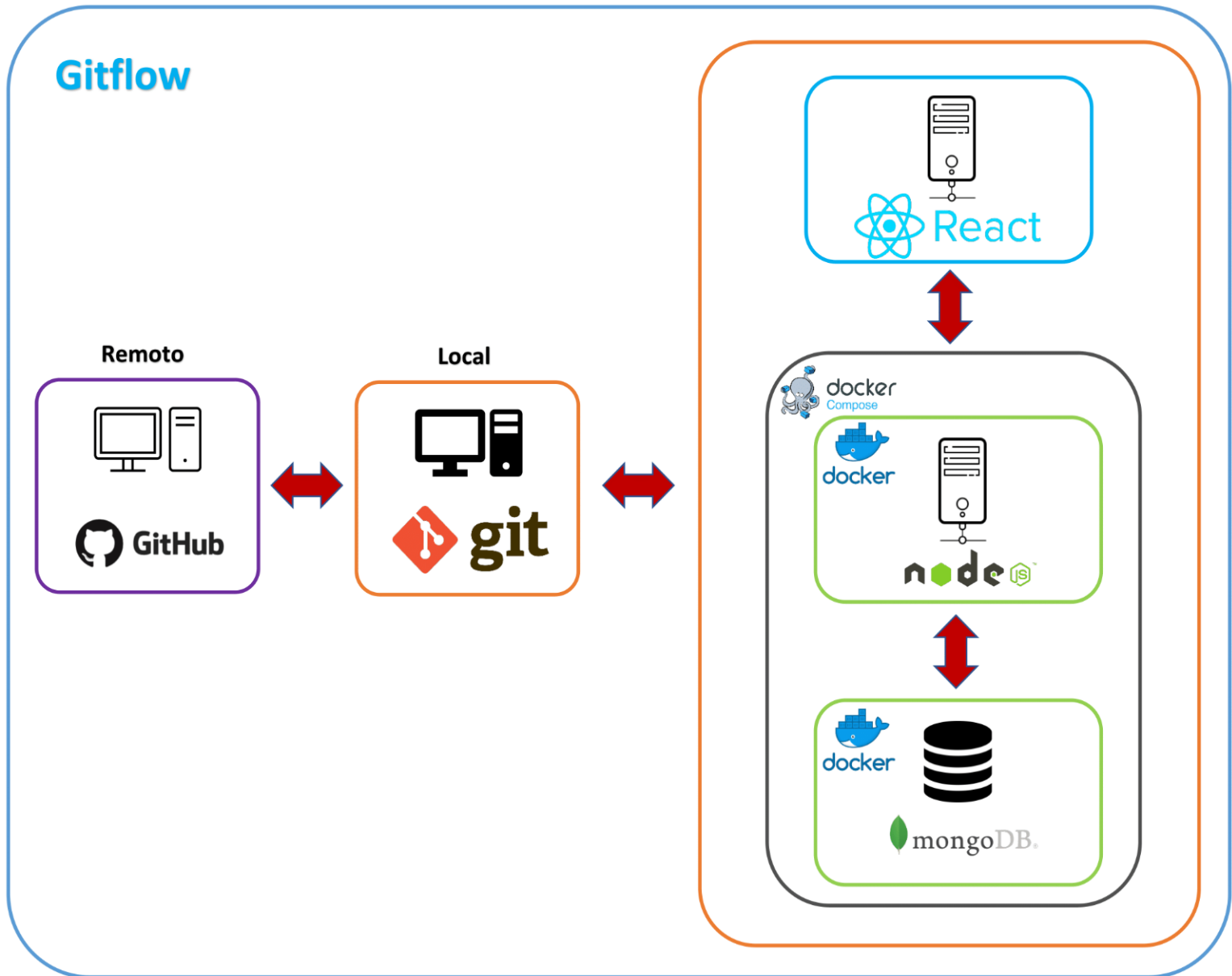
Objetivos específicos

- Implementar un repositorio para alojar el código de la presente práctica.
- Aplicar la técnica de “branching” llamada Git Flow.
- Trabajar en los grupos creados en el laboratorio.
- Aplicar de forma correcta la funcionalidad de cada una de las ramas de un flujo en Git Flow.
- Volver a una versión o versiones anteriores de nuestro código en caso de existir algún error en el código.

Descripción General

La aplicación consiste en crear una solución con una arquitectura cliente-servidor, la cual está comúnmente compuesta por dos componentes, el proveedor y el consumidor, se le conoce como proveedor al servidor que nos provee una serie de servicios o recursos los cuales son consumidos por el cliente o consumidor.

Arquitectura y flujo de la aplicación

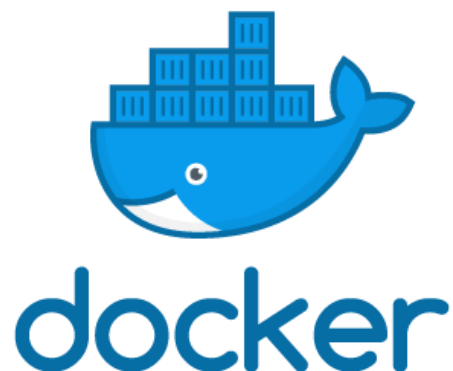


Tecnologías utilizadas

Docker

Es una plataforma abierta de desarrollo, envío, y ejecución de aplicaciones. Docker te permite separar tus aplicaciones de tu infraestructura de trabajo, para así entregar el software de forma rápida. Con Docker se puede manejar la infraestructura de la misma forma que se manejan las aplicaciones. Tomando ventaja de la metodología de envío, prueba, y despliegue de código de forma rápida de Docker, se puede significativamente reducir el retraso entre escritura de código y ejecución en producción.

Docker provee la capacidad de empaquetar y ejecutar aplicaciones en un entorno vagamente aislado llamado contenedor. La aislación y seguridad permite ejecutar muchos contenedores simultáneamente en un host dado. Los contenedores son ligeros y contienen lo necesario para ejecutar la aplicación.



Docker compose

Compose es una herramienta para definir y ejecutar un multi contenedor de aplicaciones de Docker. Con compose se utiliza un archivo YAML para configurar los servicios de las aplicaciones. Luego con un comando, se crea e inician todos los servicios de la configuración.



Node.js

Diseñado para crear aplicaciones de red escalables, utilizando un entorno de ejecución de JavaScript orientado a eventos asíncronicos.

Node.js lleva el modelo de eventos un poco más allá. Incluye un bucle de eventos como runtime de ejecución en lugar de una biblioteca. En otros sistemas siempre existe una llamada de bloqueo para iniciar el bucle de eventos. Por lo general, el comportamiento se define mediante devoluciones callbacks de llamada al iniciarse un script y al final se inicia un servidor a través de una llamada de bloqueo. Node.js simplemente entra en el bucle de eventos después de ejecutar el script de entrada y sale cuando no hay más devoluciones callbacks de llamada para realizar.



React JS

React es una librería de JavaScript declarativa, eficiente y flexible para construir interfaces de usuario. Permite componer IUs complejas de pequeñas y aisladas piezas de código llamadas "componentes".



MongoDB

MongoDB es una base de datos distribuida, basada en documentos y de uso general que ha sido diseñada para desarrolladores de aplicaciones modernas y para la era de la nube.



Git

Git es un controlador de versiones es un sistema que graba cambios a un archivo o conjunto de archivos a través del tiempo, permitiendo hacer llamadas a versiones específicas. Esto funciona gracias a que el sistema de control de versiones tiene una simple base de datos que mantiene todos los cambios en archivos bajo una versión de control.



Git flow

Es un flujo de trabajo de Git. El flujo de GitFlow define un modelo estricto de ramificación diseñado alrededor de la publicación del proyecto. Proporciona un marco sólido para gestionar proyectos más grandes.

En realidad, es una especie de idea abstracta de un flujo de trabajo de Git. Quiere decir que ordena qué tipo de ramas se deben configurar y cómo fusionarlas.

Justificación de tecnologías utilizadas

Docker

- Se puede programar localmente y compartir el trabajo con los compañeros, usando contenedores.
- Impulsar aplicaciones a un entorno de prueba y ejecutarlas de forma manual y automáticas
- Al encontrar errores, pueden ser corregidos en el entorno de desarrollo para volver a implementarlos.

Docker compose

- Múltiples entornos aislados en un solo host.
- Mantiene el volumen de datos cuando los contenedores son creados, al momento de ejecutar docker-compose up, si encuentra algún contenedor de alguna ejecución previa, copia el volumen del contenedor antiguo al nuevo contenedor. Este proceso asegura que cualquier dato que se haya creado en volumen no esté perdido.
- Soporta variables en el archivo compose. Se puede utilizar estas variables para personalizar la composición para diferentes entornos, o diferentes usuarios.
- Documentar y configurar todos los servicios de dependencias de la aplicación (bases de datos, colas, caches, APIs, etc).

Node.js

- Utiliza HTTP, diseñado teniendo en cuenta la transmisión de operaciones con streaming y baja latencia. Lo cual hace que Node.js sea muy adecuado para la base de una librería o un framework web.
- Orientado a eventos asíncronos.
- Incluye un bucle de eventos como runtime de ejecución en lugar de una biblioteca.

React JS

- Flexible.
- Dinámico.
- Ligero.
- Reutilización de código.

Mongo

- Utiliza documentos JSON, la forma más productiva de trabajar con datos.
- Admite matrices y objetos anidados como valores.
- Trabaja con esquemas dinámicos y flexibles.
- Lenguaje de consulta rico y expresivo que permite filtrar y ordenar por cualquier campo, independientemente de cómo esté incrustado en un documento.
- Las propias consultas son también JSON, por lo que se programan fácilmente.

Git

- Permite tener múltiples ramas locales que pueden ser completamente independientes entre sí. La creación, fusión y eliminación de esas líneas de desarrollo toma segundos.
- Las operaciones se realizan localmente, lo que le da una gran ventaja de velocidad en sistemas centralizados que constantemente tienen que comunicarse con un servidor en algún lugar.
- Garantiza la integridad criptográfica de cada parte de su proyecto. Cada archivo y confirmación se suma de verificación y se recupera mediante su suma de verificación cuando se vuelve a verificar.

Git flow

- Este flujo de trabajo es ideal para los flujos de trabajo de software basados en publicaciones
- Ofrece un canal específico para las correcciones de producción.
- Es ideal para los proyectos que tienen un ciclo de publicación programado.
- Asigna funciones muy específicas a las diferentes ramas y define cómo y cuándo deben interactuar.

Requerimientos

Requerimientos funcionales

Requerimiento	Especificación
Apartados de visualización	<p>Se debe contar con dos pantallas de visualización, en la primera de ellas se le dará la bienvenida al usuario y se le mostrará los datos personales de cada uno de los miembros del equipo de desarrollo. La pantalla subsiguiente será utilizada para las diferentes funcionalidades con las que debe contar la aplicación.</p>
Datos del equipo	<p>Se requiere mostrar los datos con los que se identifica cada uno de los miembros del equipo en una pantalla inicial que se mostrará cuando el usuario ingresa a la aplicación.</p> <p>Los datos para mostrar son:</p> <ul style="list-style-type: none">• Nombre completo• Carnet• Universidad• Carrera• Teléfono• Dirección

<p>Administración de libros</p>	<p>La aplicación permitirá la administración de libros. Se requiere llevar control de los siguientes datos con relación a cada uno de los libros:</p> <ul style="list-style-type: none"> • Título • Autor • Editorial • Año de publicación • Edición • Número de páginas • Género literario • ISBN
<p>Funcionalidad CREATE Libro</p>	<p>Se requiere una funcionalidad que permita ingresar nuevos libros al sistema, para ello se deberá solicitar al usuario, por medio de un formulario, el ingreso de los datos correspondientes al nuevo registro. Posteriormente el libro ingresado deberá ser almacenado.</p>
<p>Funcionalidad READ Libro</p>	<p>Se requiere mostrar cada uno de los registros de libros existentes, mostrando sus datos al usuario.</p>
<p>Funcionalidad UPDATE Libro</p>	<p>Se debe poder actualizar los datos correspondientes a los libros almacenados, para ello, se debe mostrar los datos actualmente registrados y se podrá modificar los mismos a conveniencia del usuario. Posteriormente se actualizarán los datos en los que el usuario haya realizado cambios.</p>

Funcionalidad DELETE Libro	Por medio de esta funcionalidad se podrán eliminar registros de libros almacenados, el usuario deberá indicar por medio de la interfaz el libro que desea eliminar y tras una confirmación, el libro se eliminará de forma permanente.
Almacenamiento de Libros	Cada uno de los registros ingresados por el usuario deberán ser almacenados en una base de datos para su posterior manipulación y consulta.
Aplicación web	La aplicación debe ser de tipo web, es decir, se podrá acceder desde un navegador.

Requerimientos no funcionales

Requerimiento	Especificación
Arquitectura Cliente-Servidor	<p>Se requiere que la aplicación posea una arquitectura de tipo Cliente-Servidor en la que el cliente podrá manipular los datos desde una aplicación web, realizando peticiones a un servidor, el cual consultará en el almacenamiento y generará respuestas en base a la operación realizada.</p> <p>La comunicación entre estos se llevará a cabo mediante el uso de una API REST, la cual contará con diferentes endpoints en los cuales se llevarán a cabo las diferentes consultas a la base de datos.</p>
Base de datos	<p>Se solicita el almacenamiento persistente de los registros ingresados por el usuario, para ellos se requiere la utilización de una base de datos, esta podrá ser consultada por el servidor, lo que permitirá realizar las operaciones de las diferentes funcionalidades desarrolladas.</p>
Interfaz amigable e intuitiva	<p>El sistema debe ser presentado de forma amigable al usuario, es decir, de manera cómoda a la vista de este. Así mismo, la disposición de los elementos debe dar pauta a que el usuario realice acciones dentro de la aplicación de manera intuitiva, haciendo uso de íconos descriptivos de las funciones a las cuales representan, o bien, conceptos que pueden ser claramente interpretados.</p>
Disponibilidad	<p>La aplicación debe estar disponible en cualquier momento en que el usuario desea utilizarla, siempre y cuando se cumpla con los requerimientos técnicos necesarios, los</p>

	cuales de detallan anteriormente en el apartado de REQUISITOS.
Integridad de la información	Los datos manipulados por la aplicación deben ser almacenados en la base de datos de tal manera que puedan ser reflejados exactamente de la misma manera en que el usuario los ingresó, sin alteraciones que puedan crear conflicto en el entendimiento de estos por parte del cliente.
Rendimiento	La aplicación debe ser capaz de funcionar ante el constante uso de esta sin mermar sus funciones o limitarlas.
Aplicación Responsiva	La aplicación por desarrollar debe ser adaptable a las características del dispositivo desde el cual sea accedida, esta debe sufrir las alteraciones necesarias para ser presentada de manera legible y cómoda para el usuario.
Funcionalidad	La aplicación debe cumplir a cabalidad con cada uno de los aspectos requeridos y garantizar el correcto funcionamiento de estos.
Documentación	Se deberá documentar cada uno de los aspectos útiles para el desarrollo de la aplicación, siendo este aspecto de gran interés para la comprensión del sistema y su posterior mantenimiento. Se requiere documentar las funcionalidades, tecnologías utilizadas, incluyendo detalles técnicos acerca de estas, y los procedimientos necesarios para el correcto funcionamiento del sistema.

Requisitos

Hardware

Docker:

Backend de WSL 2:

- Windows 10 de 64 bits: Home, Pro, Enterprise o Education, versión 1903 (compilación 18362 o superior).
- Habilite la función WSL 2 en Windows, para obtener instrucciones detalladas consulte <https://docs.microsoft.com/en-us/windows/wsl/install-win10>
- Se requieren los siguientes requisitos previos de hardware para ejecutar correctamente WSL 2 en Windows 10:
 - Procesador de 64 bits.
 - RAM del sistema de 4GB
 - El soporte de virtualización de hardware a nivel de BIOS debe estar habilitado en la configuración del BIOS.

Backend de Hyper-V y contenedores de Windows:

- Windows 10 de 64 bits: Pro, Enterprise o Education (compilación 17134 o superior).
- Las características de Hyper-V y contenedores de Windows deben estar habilitadas.
- Se requieren los siguientes requisitos previos de hardware para ejecutar correctamente Client Hyper-V en Windows 10:
 - Procesador de 64 bits.
 - RAM del sistema de 4GB
 - El soporte de virtualización de hardware a nivel de BIOS debe estar habilitado en la configuración del BIOS

Node.js:

Node.js puede funcionar en la mayoría de las computadoras de la actualidad, ya que se ha visto funcionando en dispositivos con 256 o 128 MB de RAM y con tan solo un núcleo en el procesador.

Visual studio code:

Es una pequeña descarga (<100 MB) y ocupa un espacio en disco de 200 MB, VS Code es liviano y debería ejecutarse fácilmente en el hardware actual.

Se recomienda:

- Procesador de 1,6 GHz o más rápido
- 1 GB de RAM

Software

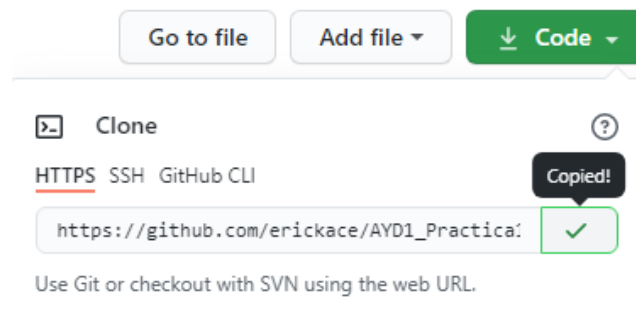
Sistema operativo	Windows 10
Docker	versión 20.10.6
Docker-compose	versión 1.29.1
Node.js	versión 14.16.1
Visual Studio Code	versión 1.56.2
Git	versión 2.31.1
Git flow	versión 1.12.3
Kernel de Linux para Windows (necesario para Docker desktop)	Puede descargar en https://docs.microsoft.com/en-us/windows/wsl/install-win10 (paso 4)
Google Chrome	Versión 72 o una más reciente (puede utilizar su navegador de preferencia)

Herramientas de desarrollo (instalación de tecnologías)

Descargar proyecto

Forma 1: Clonar repositorio

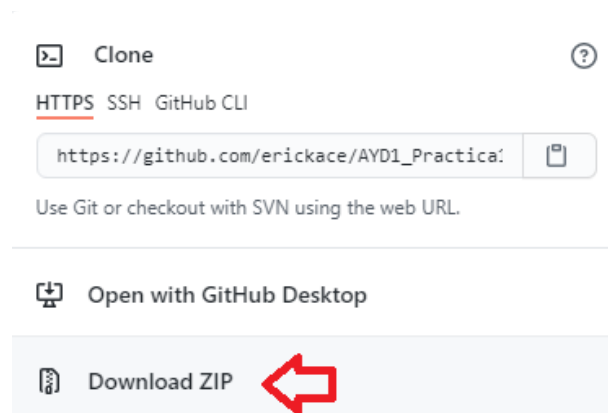
1. Ve a Github y obtén la ruta del repositorio:



2. Crea el directorio donde quieras meter el proyecto.
3. Acceder al directorio desde tu consola y escribir:
`git clone https://github.com/erickace/AYD1_Practica1 7.git`
4. Listo, ya tienes el proyecto clonado en tu ordenador.

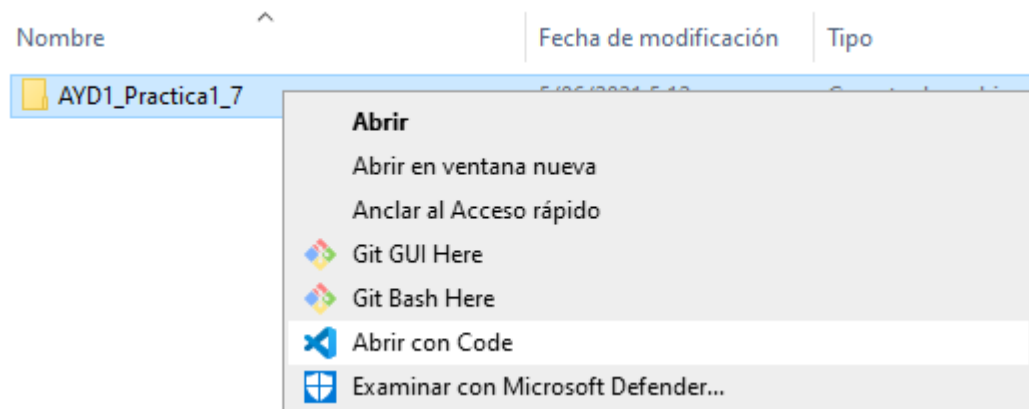
Forma 2: Descargar ZIP

1. Descargar el fichero.



2. Mover el zip al directorio que desees.
3. Descomprimir el fichero.
4. Listo, ya tienes el proyecto clonado en tu ordenador.

Abrir proyecto con Visual Studio Code



Backend

Abrir consola integrada de VS Code y acceder a carpeta 'node-docker-mongo':

- `cd node-docker-mongo`

Instalamos dependencias:

- `npm install`

Compilamos:

- `docker-compose build`
- `docker-compose up`

Nota: automáticamente nos descargara la imagen de Mongo.

Frontend

Abrir nueva consola integrada de VS Code y acceder a carpeta 'appweb':

- `cd appweb`

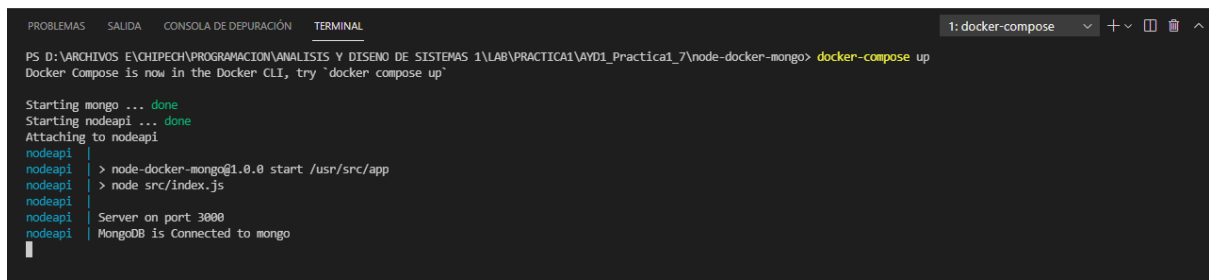
Instalamos dependencias:

- `npm install`

Compilamos:

- `npm start`

Backend corriendo

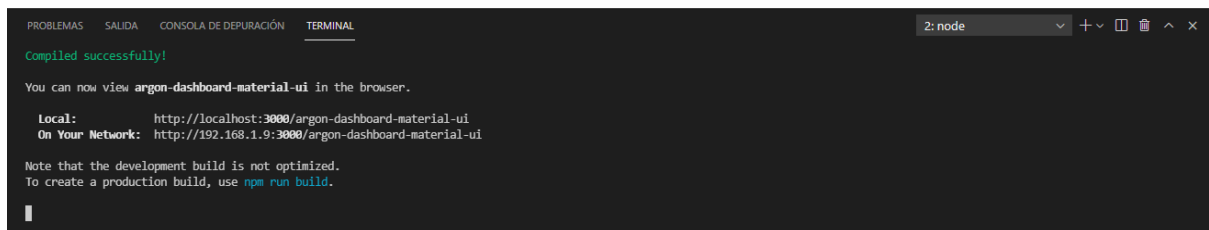


```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  1: docker-compose  +  -  [ ]  ^

PS D:\ARCHIVOS EVCHPECH\PROGRAMACION\ANALISIS Y DISEÑO DE SISTEMAS 1\LAB\PRACTICA1\AYD1_Practica1_7\node-docker-mongo> docker-compose up
Docker Compose is now in the Docker CLI, try 'docker compose up'

Starting mongo ... done
Starting nodeapi ... done
Attaching to nodeapi
nodeapi |
nodeapi | > node-docker-mongo@1.0.0 start /usr/src/app
nodeapi | > node src/index.js
nodeapi |
nodeapi | Server on port 3000
nodeapi | MongoDB is Connected to mongo
nodeapi |
```

Frontend corriendo



```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  2: node  +  -  [ ]  ^  x

Compiled successfully!

You can now view argon-dashboard-material-ui in the browser.

  Local:            http://localhost:3000/argon-dashboard-material-ui
  On Your Network:  http://192.168.1.9:3000/argon-dashboard-material-ui

Note that the development build is not optimized.
To create a production build, use npm run build.


```

Explicación código

Backend

Modelo de base de datos:

El CRUD se basa en un solo objeto, el cual cuenta con 8 atributos sin contar el `_id` que se genera automáticamente en la colección de mongo, a continuación se muestra la estructura del objeto localizada en `/src/model/item.js` del backend:

```
const ItemSchema = new Schema({
  titulo: {
    type: String,
    required: true
  },
  anio_publicacion: {
    type: String,
    required: false
  },
  edicion: {
    type: String,
    required: true
  },
  numero_paginas: {
    type: String,
    required: false
  },
  autor: {
    type: String,
    required: true
  },
  editorial: {
    type: String,
    required: false
  },
  generoLiterario: {
    type: String,
    required: false
  },
  isbn: {
    type: String,
    required: true
  }
})
```

```
});
```

Cadena de conexión hacia MongoDB

Esta se encuentra ubicada en `/src/database.js` del backend:

Utilizamos **Mongoose** que es una biblioteca de JavaScript que nos permite definir esquemas con datos fuertemente tipados. Una vez que se define un esquema, **Mongoose** nos permite crear un Modelo basado en un esquema específico. Un modelo de mangosta se asigna a un documento MongoDB a través de la definición del esquema del modelo.

```
const mongoose = require('mongoose');

mongoose.connect('mongodb://mongo/mydatabase', {
  useNewUrlParser: true,
  useUnifiedTopology: true
})
.then(db => console.log('MongoDB is Connected to', db.connection.host))
.catch(err => console.error(err));
```

Centro del control del Backend

Código necesario para levantar el servidor, el cual se encarga de unificar y conectar las dependencias necesarias según requiera el proyecto, en este caso se utilizó nodejs el cual hace uso de express, también utilizamos CORS que es un mecanismo que utiliza cabeceras HTTP adicionales para permitir que un user agent obtenga permiso para acceder a recursos seleccionados desde un servidor, en un origen distinto (dominio) al que pertenece, el código se encuentra ubicado en `/src/index.js` del backend:

```
const express = require('express');
const cors = require('cors');

const app = express();
app.use(cors());
app.use(express.json({ extended: true }));
require('./database');

app.use(require('./routes/index.routes'));

app.listen(3000);
console.log('Server on port', 3000);
```

Rutas (APIS) - Peticiones HTTP

Create:

Ejemplo de objeto aceptado:

```
{
  "titulo": "análisis 1",
  "anio_publicacion": "1990",
  "edicion": "1ra",
  "numero_paginas": 1000,
  "autor": "N/A",
  "editorial": "N/A",
  "generoLiterario": "Historia",
  "isbn": 555556
}
```

El endpoint el cual se puede acceder desde <http://localhost:3200/> por medio de una petición POST, es el encargado de guardar un libro, el cual recibe un json que tiene la forma del modelo de base de datos mostrado anteriormente, este debe de ser enviado por body, el código se encuentra ubicado en `/src/routes/index.routes.js` del backend:

Al hacer la petición obtenemos el listado de los datos ya guardados en la bd luego lo recorremos y comparamos el isbn de la bd con el isbn que se envió en la petición, si el isbn enviado ya existe no dejará guardar ya que debe de ser único para un libro y retorna el mensaje para mostrar, de lo contrario, procederá a realizar el respectivo save en la colección de la base de datos.

```
router.post('/', async (req, res) => {
  var ver = await Item.find().sort({ _id: -1 });
  var verificacion = 0;
  for(var i = 0; i < ver.length; i++){
    if(ver[i].isbn == req.body.isbn){
      verificacion = 1;
      break;
    }
  }
  if(verificacion == 1){
    verificacion = 0;
    res.json("Ya existe");
    //res.json("Ya existe el libro con isbn: " + req.body.isbn);
  }else{
    const newItem = new Item({
      titulo: req.body.titulo,
      anio_publicacion: req.body.anio_publicacion,
```

```

        edicion: req.body.edicion,
        numero_paginas: req.body.numero_paginas,
        autor: req.body.autor,
        editorial: req.body.editorial,
        generoLiterario: req.body.generoLiterario,
        isbn: req.body.isbn
    });
    console.log(newItem);
    await newItem.save().then(item => res.json(item));
}
});

```

Resultado obtenido al hacer la petición (Prueba realizada con postman):

The screenshot shows the Postman interface for a POST request to `http://localhost:3200/`. The request body is a JSON object with the following fields: `titulo`, `anio_publicacion`, `edicion`, `numero_paginas`, `autor`, `editorial`, `generoLiterario`, and `isbn`. The response is a 200 OK status with a response time of 305 ms and a body size of 452 B. The response body is a JSON object containing the same fields as the request, plus an additional `_id` field.

Request:

```

1  {
2    "titulo": "analisis 1",
3    "anio_publicacion": "1990",
4    "edicion": "1ra",
5    "numero_paginas": 1000,
6    "autor": "N/A",
7    "editorial": "N/A",
8    "generoLiterario": "Historia",
9    "isbn": "555556"
10 }

```

Response:

```

1  {
2    "_id": "60bea277a3e4ff0015b92561",
3    "titulo": "analisis 1",
4    "anio_publicacion": "1990",
5    "edicion": "1ra",
6    "numero_paginas": "1000",
7    "autor": "N/A",
8    "editorial": "N/A",
9    "generoLiterario": "Historia",
10   "isbn": "555556",

```


Read:

Endpoint encargado de obtener la información, se puede acceder con <http://localhost:3200/items> por medio de una petición GET, el código se encuentra ubicado en `/src/routes/index.routes.js` del backend:

Al hacer la petición obtenemos el listado de los datos ya guardados en la bd luego lo retornamos en formato json como respuesta.

```
router.get('/items', async (req, res) => {
  await Item.find().then(items => res.json(items));
});
```

Resultado obtenido al hacer la petición (Prueba realizada con postman):

The screenshot shows a Postman interface with a GET request to `http://localhost:3200/items`. The response is a JSON array of two book objects. The status is 200 OK, with a response time of 14 ms and a body size of 660 B.

KEY	VALUE	DESCRIPTION	...	Bulk Edit
KEY	VALUE	DESCRIPTION	...	Bulk Edit

Body: Cookies Headers (7) Test Results

200 OK 14 ms 660 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "_id": "60bea277a3e4ff0015b92561",
4     "titulo": "analisis 1",
5     "anio_publicacion": "1990",
6     "edicion": "1ra",
7     "numero_paginas": "1000",
8     "autor": "N/A",
9     "editorial": "N/A",
10    "generoLiterario": "Historia",
11    "isbn": "555556",
12    "__v": 0
13  },
14  {
15    "_id": "60bea2f6a3e4ff0015b92562",
16    "titulo": "analisis",
17    "anio_publicacion": "1989",
18    "edicion": "1ra",
19    "numero_paginas": "1000",
```

Update:

Endpoint encargado de actualizar un ítem, se puede acceder con <http://localhost:3200/edit/id> siendo el id el ISBN del libro, acá se realiza una petición POST, también es necesario enviar por body el objeto con los datos que se reemplazara con el original, el código se encuentra ubicado en `/src/routes/index.routes.js` del backend:

Al hacer la petición obtenemos el listado de los datos ya guardados luego lo recorremos y comparamos el isbn de la bd con el isbn que se envió en la petición, para luego obtener el `_id`

que genera mongo automáticamente, seguidamente se procede a hacer el respectivo update del ítem en la bd.

```
router.post('/edit/:id', async (req, res) => {
  var ver = await Item.find().sort({ _id: -1 });
  var verificacion = 0;
  const { id } = req.params;
  for(var i = 0; i < ver.length; i++){
    if(ver[i].isbn == req.body.isbn){
      verificacion = ver[i]._id;
      break;
    }
  }

  console.log(verificacion);

  await Item.updateOne({ _id: verificacion }, req.body).then(items =>
res.json(items));
  //res.json("Editado!");
});
```

Resultado obtenido al hacer la petición (Prueba realizada con postman):

The screenshot shows the Postman interface for a POST request to `http://localhost:3200/edit/555556`. The request body is a JSON object with the following fields:

```
{
  "titulo": "análisis y diseño, aprobado",
  "anio_publicacion": "1989",
  "edicion": "1ra",
  "numero_paginas": 1000,
  "autor": "N/A",
  "editorial": "N/A",
  "generoLiterario": "Historia",
  "isbn": 55589
}
```

The response is a JSON object with the following fields:

```
{
  "n": 1,
  "nModified": 1,
  "ok": 1
}
```

Delete:

Endpoint encargado de eliminar un ítem, se puede acceder con <http://localhost:3200/delete/id> siendo el id el ISBN del libro, acá se realiza una petición POST, el código se encuentra ubicado en **/src/routes/index.routes.js** del backend:

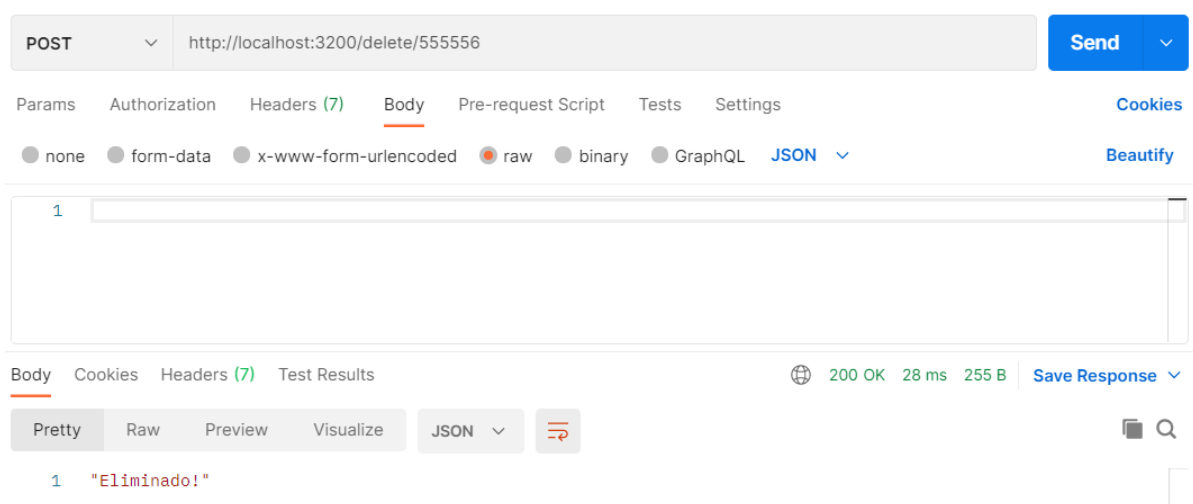
Al hacer la petición obtenemos el listado de los datos ya guardados luego lo recorremos y comparamos el isbn de la bd con el isbn que se envió en la petición, para luego obtener el `_id` que genera mongo automáticamente, seguidamente se procede a hacer el respectivo remove de la bd.

```
router.post('/delete/:id', async (req, res) => {
  var ver = await Item.find().sort({ _id: -1 });
  var verificacion = 0;
  const { id } = req.params;
  for(var i = 0; i < ver.length; i++){
    if(ver[i].isbn == id){
      verificacion = ver[i]._id;
      break;
    }
  }

  console.log(verificacion);

  await Item.remove({ _id: verificacion });
  res.json("Eliminado!");
});
```

Resultado obtenido al hacer la petición (Prueba realizada con postman):



Dockerfile

Ya que se utiliza docker para el backend, utilizamos un dockerfile para obtener la imagen de nodejs y correrla en nuestro backend dentro del contenedor que contiene nodejs:

Explicación del Dockerfile:

```
# La siguiente instrucción FROM para configurar la imagen base de la
aplicación:
FROM node:10

# Asignar el workdir de la api, colocamos la bandera -p para que cree el
directorio:
RUN mkdir -p /usr/src/app

# Nos posicionamos en el directorio creado dentro del container:
WORKDIR /usr/src/app

# Copiar el package-json y el lock.json:
COPY package*.json ./

# Correr 'npm install' para instalar todas las dependencias que
necesitemos:
RUN npm install

# Acá pueden existir muchos mas comandos "run"
# Copiar todos los archivos al workdir:
COPY . .

# Exponer el puerto 3000 al mundo:
EXPOSE 3000

# Correr NodeJS:
CMD [ "npm", "start" ]
```

docker-compose

Nuestra aplicación no es tan simple como para ejecutarse en un solo proceso: necesitamos servidores de bases de datos, de aplicaciones, en este caso nodejs y mongodb, entonces para esto utilizaremos docker-compose que nos permite tener multi contenedores en red.

Nuestra arquitectura ideal para esta aplicación pasa por crear dos contenedores: uno para la base de datos mongoDB y otro para el servidor web nodejs.

La mejor forma de gestionar aplicaciones que requieren de varios contenedores es utilizando la herramienta Docker Composer, un CLI para automatizar despliegues que usa un archivo de configuración en formato YAML.

Explicación del código:

El apartado 'services' contiene los contenedores que se crearán para desplegar nuestra aplicación. En este caso se crearán dos: uno que vamos a llamar "nodeapi" y otro que llamaremos "mongo". Estos nombres pueden ser los que queramos y servirán para identificar a nuestros servicios.

Para el servicio 'nodeapi' utilizamos el archivo Dockerfile mencionado arriba, este se encuentra también en la carpeta raíz del proyecto (build: .).

Cómo necesitaremos comunicarnos con el servicio, abrimos el puerto 3200 del contenedor.

Para el servicio 'mongo' no tenemos un proyecto entero con su Dockerfile, sino que vamos a coger la imagen mongo de hub.docker.com que ya viene preparada para ejecutar la base de datos no relacional.

En este caso, abrimos el puerto 27017. Luego especificamos que el servidor web esta conectado con el servicio 'mongo' a través de links, por lo que eso nos servirá para la conexión mostrada arriba en la parte de **Cadena de conexión hacia MongoDB**.

```
version: "3"

services:
  web:
    container_name: nodeapi
    image: nodeapi
    restart: always
    build: .
    ports:
      - "3200:3000"
    links:
      - mongo
    volumes:
      - ../usr/src/app
  mongo:
```

```
container_name: mongo
image: mongo
ports:
  - "27017:27017"
logging:
  driver: none
```

Fronted

Componente Dashboard:

Utilizamos componentes de material-ui react para realizar nuestro componente de bienvenida, el cual muestra una tabla con los datos de los estudiantes que participaron en la realización de la práctica. El código se encuentra en el fronted en la ruta </src/views/practica1/Dashboard.js>.

Los imports son los siguientes:

```
// @material-ui/core components
import { makeStyles } from "@material-ui/core/styles"; //Import para
utilizar la solución de estilo de materia ui
import { useTheme } from "@material-ui/core/styles"; //Import para
utilizar template de material ui
import Avatar from "@material-ui/core/Avatar"; //Import para poder
utilizar el ICONO avatar
import Box from "@material-ui/core/Box"; //Import para poder utilizar el
ICONO box
import Card from "@material-ui/core/Card"; //Import para poder utilizar
el ICONO card
//import CardActions from "@material-ui/core/CardActions";
import CardHeader from "@material-ui/core/CardHeader"; // Import para
poder utilizar el Componente que coloca header dentro de un card o tarjeta
import Container from "@material-ui/core/Container"; // Import para poder
utilizar el componente contenedor de material ui
import Table from "@material-ui/core/Table"; // Import para poder
utilizar el componente tabla de material ui
import TableBody from "@material-ui/core/TableBody"; // Import para poder
utilizar el componente body de una tabla de material ui
import TableCell from "@material-ui/core/TableCell"; // Import para poder
utilizar el componente cell de una tabla de material ui
import TableContainer from "@material-ui/core/TableContainer"; // Import
para poder utilizar el componente container de una tabla de material ui
```

```
import TableHead from "@material-ui/core/TableHead"; // Import para poder
utilizar el componente head de una tabla de material ui
import TableRow from "@material-ui/core/TableRow"; // Import para poder
utilizar el componente row de una tabla de material ui
```

Componente:

Hacemos uso de una función flecha el cual tiene el nombre del componente Dashboard. Entonces la forma más sencilla de definir un componente es escribir una función de JavaScript:

```
const Dashboard = () => {
  const classes = useStyles();
  const theme = useTheme();
  return (
```

Al crear el componente Dashboard que renderiza el componente Header que dentro de él existe otro componente llamado container, y en componente existe otro componente llamado box y así sucesivamente formando la tabla con los importes antes mencionados, el componente `TableContainer` es la tabla:

```
const Dashboard = () => {
  const classes = useStyles();
  const theme = useTheme();
  return (
    <>
      <Header />
      { /* Page content */ }
      <Container
        maxWidth={false}
        component={Box}
        marginTop="-11.5rem"
        classes={{ root: classes.containerRoot }}
      >
        <Box
          component={Card}
          marginTop="3rem"
          classes={{ root: classes.cardRoot + " " +
classes.cardRootDark }}
        >
          <CardHeader
            className={classes.cardHeader}
            title="BIENVENIDO, SOMOS EL GRUPO: All for one"
            titleTypographyProps={{
```

```

        component: Box,
        marginBottom: "0!important",
        variant: "h3",
        textAlign: "center",
      })
    </CardHeader>
    <TableContainer>

```

Por último en el componente de “bienvenida” exportamos el componente Dashboard para poder ser utilizado en otros componentes:

```
export default Dashboard;
```

Servicio:

El url a utilizar que nos proporciona el servicio para realizar las peticiones POST y GET se encuentra en el fronted en la ruta </src/services/apiREST.js>

Ruta mencionada en el apartado de APIS del backend:

```
export const Apiurl = "http://localhost:3200";
```

Componente Libros (CRUD):

Utilizamos componentes de material-ui react, assets de la plantilla predefinidos de serie, axios que es una librería JavaScript que puede ejecutarse en el navegador y que nos permite hacer sencillas las operaciones como cliente HTTP, por lo que podremos configurar y realizar solicitudes a un servidor y recibiremos respuestas fáciles de procesar

El componente de Libros (CRUD), muestra las operaciones de create, read, update y delete. El código se encuentra en el fronted en la ruta </src/views/practica1/Libros.js>.

Los imports son los siguientes:

Utilizamos el hook **useState** para añadir un estado a nuestro componente, para que puedan tener comportamiento y sean dinámicos.

Utilizamos el hook **useEffect**, un hook que recibe como parámetro una función que se ejecutará cada vez que nuestro componente se renderice, ya sea por un cambio de estado, por recibir props nuevas o, y esto es importante, porque es la primera vez que se monta.


```

import React, { useState, useEffect } from "react";

// @material-ui/core components
import { makeStyles } from "@material-ui/core/styles";
import { useTheme } from "@material-ui/core/styles";
import Box from "@material-ui/core/Box";
import Container from "@material-ui/core/Container";

// core components
import Header from "components/Headers/Header.js";

import componentStyles from "assets/theme/views/admin/tables.js";

// imports generales
import axios from "axios";
import { Apiurl } from '../../services/apirest';
import './style.css';

// imports para hacer el añadir un libro
import PostAddIcon from '@material-ui/icons/PostAdd';
import { Button } from '@material-ui/core';
import Dialog from "@material-ui/core/Dialog";
import Slide from "@material-ui/core/Slide";
import Typography from "@material-ui/core/Typography";
import IconButton from "@material-ui/core/IconButton";
import Clear from "@material-ui/icons/Clear";
import DialogActions from "@material-ui/core/DialogActions";
import DialogContent from "@material-ui/core/DialogContent";
import componentStyles_ from "assets/theme/components/dialog.js";
import Grid from "@material-ui/core/Grid";
import FormGroup from "@material-ui/core/FormGroup";
import FormLabel from "@material-ui/core/FormLabel";
import FormControl from "@material-ui/core/FormControl";
import FilledInput from "@material-ui/core/FilledInput";
import Swal from 'sweetalert2'

// imports para mostrar el listado de los libros
import MaterialTable from "material-table";

// imports para hacer el editar un libro
import EditAttributesIcon from '@material-ui/icons/EditAttributes';

// imports para hacer el eliminar un libro

```

```
import DeleteForeverIcon from '@material-ui/icons/DeleteForever';
import {Modal} from '@material-ui/core';
```

Como el componente de Libros utilizamos una función fecha que renderice nuestro componente, en esta ocasión se llama Libros, el código se encuentra en el fronted en la ruta </src/views/practica1/Libros.js>

```
const Libros = () => {
  const styles = useMoreStyles();
```

Exportando el componente:

El código se encuentra en el fronted en la ruta </src/views/practica1/Libros.js>

```
export default Libros;
```

Estado para modales create, edit y delete

Hacemos uso del hook useState para conocer el estado del modal de creación de un nuevo libro, este tipo de operación se usa para los modales de editar y confirmación de eliminar un libro:

El código se encuentra en el fronted en la ruta </src/views/practica1/Libros.js>

```
const [open, setOpen] = React.useState(false);
const [openEdit, setOpenEdit] = React.useState(false);
const [modalEliminar, setModalEliminar] = useState(false);
```

Petición POST para la operación create

Para la petición hicimos uso de axios y de servicios Apiurl, lo ejecutamos como asíncrono para que se ejecute en 2do plano y pasamos como parámetros la url del endpoint y el objeto a guardar, hacemos uso de los métodos `alertSession_` para indicar que el libro ya existe y el método `alertSession` para indicar que el libro se insertó correctamente.

El método `handleClose()` cierra el modal si el libro se guardó correctamente.

Hacemos uso del hook `useState` utilizando el método `setData` ya que data que se encuentra en `useState` ya contiene los datos, es decir actualizamos esa data con un registro más.

El código se encuentra en el fronted en la ruta </src/views/practica1/Libros.js>

```
let baseUrl_ = Apiurl + "/";
const petitionPost = async () => {
  await axios.post(baseUrl_, libroSeleccionado)
    .then(response => {
      if (response.data == "Ya existe") {
```

```

        alertSession_("Ya existe el libro con isbn: " +
libroSeleccionado.isbn);
    } else {
        handleClose();
        setData(data.concat(response.data));
        alertSession("Libro insertado con exito!");
    }
}).catch(error => {
    console.log(error);
})
}

```

Utilizamos la `const bodyInsertar` la cual cuenta con el componente que va dentro del modal, contiene todos los campos.

El código se encuentra en el fronted en la ruta </src/views/practica1/Libros.js>

```

const bodyInsertar = (
    <div>
        <DialogContent>
            <Grid container>

```

Petición GET para la operación read

Para la petición hicimos uso de axios y de servicios Apiurl, lo ejecutamos como asíncrono para que se ejecute en 2do plano y pasamos como parámetros la url del endpoint, la respuesta la agregamos a la variable data por medio de setData que utiliza el hook useState y useEffect para que nuestro componente se renderice al detectar cambios.

El código se encuentra en el fronted en la ruta </src/views/practica1/Libros.js>

```

const [data, setData] = useState([]);

```

```

let baseUrl = Apiurl + "/items";
const peticionGet = async () => {
    await axios.get(baseUrl)
        .then(response => {
            setData(response.data);
        }).catch(error => {
            console.log(error);
        })
}

useEffect(() => {

```

```
    peticiónGet();
  }, [])
```

En esta parte se crea la tabla que muestra los datos y las columnas utilizando el componente importado de material ui llamado `MaterialTable`

El código se encuentra en el fronted en la ruta </src/views/practica1/Libros.js>

```
const columns = [
  { title: 'Título', field: 'titulo' },
  { title: 'Fecha de publicación', field: 'anio_publicacion' },
  { title: 'Edición', field: 'edicion' },
  { title: 'Número de páginas', field: 'numero_paginas' },
  { title: 'Autor', field: 'autor' },
  { title: 'Editorial', field: 'editorial' },
  { title: 'Genero literario', field: 'generoLiterario' },
  { title: 'ISBN', field: 'isbn' }
];
```

```
<MaterialTable
  columns={columns}
  data={data}
  title="Libros agregados"
```

Petición POST para la operación update

Para la petición hicimos uso de axios y de servicios Apiurl, lo ejecutamos como asíncrono para que se ejecute en 2do plano y pasamos como parámetros la url del endpoint, pasamos como parámetros el isbn del libro y el objeto editado, en el response hace un backup de la data en dataNueva para posteriormente buscar el isbn y hacer el edit de table si todo ocurrió de manera correcta.

El código se encuentra en el fronted en la ruta </src/views/practica1/Libros.js>

```
let baseUrlEdit_ = Apiurl + "/edit";
const peticiónPut = async () => {
  await axios.post(baseUrlEdit_ + "/" + libroSeleccionado.isbn,
libroSeleccionado)
    .then(response => {
      var dataNueva = data;
      dataNueva.map(libro => {
        if (libro.isbn === libroSeleccionado.isbn) {
          libro.anio_publicacion =
libroSeleccionado.anio_publicacion;
          libro.autor = libroSeleccionado.autor;
          libro.edicion = libroSeleccionado.edicion;
```

```

        libro.editorial = libroSeleccionado.editorial;
        libro.generoLiterario =
libroSeleccionado.generoLiterario;
        libro.numero_paginas =
libroSeleccionado.numero_paginas;
        libro.titulo = libroSeleccionado.titulo;
    }
    });
    setData(dataNueva);
    alertSession("Libro editado correctamente!");
    handleClose_();
}).catch(error => {
    console.log(error);
})
}

```

Utilizamos la `const bodyEditor` la cual cuenta con el componente que va dentro del modal, contiene todos los campos

El código se encuentra en el fronted en la ruta </src/views/practica1/Libros.js>

```

const bodyEditor = (
    <div>
        <DialogContent>
            <Grid container>
                <Grid item xs={12} lg={6}>
                    <FormGroup>
                        <FormLabel>Titulo *</FormLabel>

```

Petición POST para la operación delete

Para la petición hicimos uso de axios y de servicios Apiurl, lo ejecutamos como asíncrono para que se ejecute en 2do plano y pasamos como parámetros la url del endpoint, pasamos como parámetros el isbn del libro, en el response si todo fue correcto quita el libro de la tabla. El código se encuentra en el fronted en la ruta </src/views/practica1/Libros.js>

```

let baseUrlDelete_ = Apiurl + "/delete";
const peticionDelete = async () => {
    await axios.post(baseUrlDelete_ + "/" + libroSeleccionado.isbn)
        .then(response => {

setData(data.filter(libro=>libro.isbn!==libroSeleccionado.isbn));
            alertSession("Libro eliminado con éxito!");
            abrirCerrarModalEliminar();
        }).catch(error => {

```

```

        console.log(error);
    })
}

```

Utilizamos la `const bodyEliminar` la cual cuenta con el componente que va dentro del modal, contiene el mensaje de confirmación de eliminar libro

El código se encuentra en el fronted en la ruta </src/views/practica1/Libros.js>

```

const bodyEliminar = (
  <div className={styles.modal}>
    <p>Estás seguro que deseas eliminar este libro
    <b>{libroSeleccionado && libroSeleccionado.titulo}</b>? </p>
    <div align="right">
      <Button color="primary" onClick={() =>
peticionDelete()}>Sí</Button>
      <Button onClick={() =>
abrirCerrarModalEliminar()}>No</Button>
    </div>
  </div>
)

```

Al agregar las funcionalidades de edit y delete el componente de tabla queda de la siguiente forma, donde se utiliza el evento `onClick` para llamar al respectivo método y hacer la petición:

El código se encuentra en el fronted en la ruta </src/views/practica1/Libros.js>

```

<MaterialTable
  columns={columns}
  data={data}
  title="Libros agregados"
  actions={[
    {
      icon: EditAttributesIcon,
      tooltip: 'Editar Libro',
      onClick: (event, rowData) =>
seleccionarLibro(rowData, "Editar")
    },
    {
      icon: DeleteForeverIcon,
      tooltip: 'Eliminar Libro',
      onClick: (event, rowData) =>
seleccionarLibro(rowData, "Eliminar")
    }
  ]}

```

```

        options={{
          actionsColumnIndex: -1,
          sorting: false,
          search: false,
          paging: false,
          draggable: false
        }}
        localization={{
          header: {
            actions: "Acciones"
          }
        }}
      />

```

Componente Dialog que muestra el modal, se utilizó el mismo componente con diferente open para las diferentes funciones:

El código se encuentra en el fronted en la ruta </src/views/practica1/Libros.js>

```

<Dialog
  open={open}
  TransitionComponent={Transition}
  keepMounted
  onClose={handleClose}
  aria-labelledby="alert-dialog-slide-title"
  aria-describedby="alert-dialog-slide-description"
>
  <div className={classes_.dialogHeader}>
    <Typography
      variant="h5"
      component="h5"
      className={classes_.dialogTitle}
    >
      Agregar nuevo libro
    </Typography>
    <IconButton onClick={handleClose}>
      <Clear />
    </IconButton>
  </div>
  {bodyInsertar}
</Dialog>

```

Comandos Git

Erick Fernando Sánchez Mejía - 201503878

Primer release

Iniciando proyecto para utilizar comandos de git flow

```
ferna@DESKTOP-DK3SE91 MINGW64 ~/OneDrive/Documentos/git/AYDI_Practical_7 (master)
$ git flow init

which branch should be used for bringing forth production releases?
- master
Branch name for production releases: [master]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/Users/ferna/OneDrive/Documentos/git/AYDI_Practical_7/.git/hooks]

ferna@DESKTOP-DK3SE91 MINGW64 ~/OneDrive/Documentos/git/AYDI_Practical_7 (develop)
$
```

Creamos la **feature tablaEstudiantes**

```
ferna@DESKTOP-DK3SE91 MINGW64 ~/OneDrive/Documentos/git/AYDI_Practical_7 (develop)
$ git flow feature start tablaEstudiantes
Switched to a new branch 'feature/tablaEstudiantes'

Summary of actions:
- A new branch 'feature/tablaEstudiantes' was created, based on 'develop'
- You are now on branch 'feature/tablaEstudiantes'

Now, start committing on your feature. When done, use:

    git flow feature finish tablaEstudiantes

ferna@DESKTOP-DK3SE91 MINGW64 ~/OneDrive/Documentos/git/AYDI_Practical_7 (feature/tablaEstudiantes)
$
```

Verificamos el estado y subimos los cambios


```
fern@DESKTOP-0K3SE91 R1N6A64 ~/OneDrive/Documentos/git/AV01_Practical_7 (feature/tablaEstudiantes)
$ git status
On branch feature/tablaEstudiantes
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   appweb/src/views/practical/Dashboard.js

no changes added to commit (use "git add" and/or "git commit -a")

fern@DESKTOP-0K3SE91 R1N6A64 ~/OneDrive/Documentos/git/AV01_Practical_7 (feature/tablaEstudiantes)
$ git add .

fern@DESKTOP-0K3SE91 R1N6A64 ~/OneDrive/Documentos/git/AV01_Practical_7 (feature/tablaEstudiantes)
$ git commit -m "[ADD] tabla y datos personales de 201503878"
[feature/tablaEstudiantes 332bf66] [ADD] tabla y datos personales de 201503878
1 file changed, 171 insertions(+)

fern@DESKTOP-0K3SE91 R1N6A64 ~/OneDrive/Documentos/git/AV01_Practical_7 (feature/tablaEstudiantes)
$
```

Publicamos la feature **tablaEstudiantes**

```
fern@DESKTOP-0K3SE91 R1N6A64 ~/OneDrive/Documentos/git/AV01_Practical_7 (feature/tablaEstudiantes)
$ git flow feature publish tablaEstudiantes
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 1.93 KiB | 657.00 KiB/s, done.
Total 7 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
remote:
remote: Create a pull request for 'feature/tablaEstudiantes' on GitHub by visiting:
remote:   https://github.com/erickace/AV01_Practical_7/pull/new/feature/tablaEstudiantes
remote:
To https://github.com/erickace/AV01_Practical_7.git
 * [new branch]   feature/tablaEstudiantes -> feature/tablaEstudiantes
Branch 'feature/tablaEstudiantes' set up to track remote branch 'feature/tablaEstudiantes' from 'origin'.
Already on 'feature/tablaEstudiantes'
Your branch is up to date with 'origin/feature/tablaEstudiantes'.

Summary of actions:
- The remote branch 'feature/tablaEstudiantes' was created or updated
- The local branch 'feature/tablaEstudiantes' was configured to track the remote branch
- You are now on branch 'feature/tablaEstudiantes'

fern@DESKTOP-0K3SE91 R1N6A64 ~/OneDrive/Documentos/git/AV01_Practical_7 (feature/tablaEstudiantes)
$
```

Al finalizar todos los commits además del respectivo merge con la rama develop, esto ya después de que todos hicimos commit y el último hizo merge, creamos el primer release con la version v1.0.0

```
fern@DESKTOP-0K3SE91 R1N6A64 ~/OneDrive/Documentos/git/AV01_Practical_7 (develop)
$ git flow release start v1.0.0
Switched to a new branch 'release/v1.0.0'

Summary of actions:
- A new branch 'release/v1.0.0' was created, based on 'develop'
- You are now on branch 'release/v1.0.0'

Follow-up actions:
- Bump the version number now!
- Start committing last-minute fixes in preparing your release
  When done, run:

    git flow release finish 'v1.0.0'

fern@DESKTOP-0K3SE91 R1N6A64 ~/OneDrive/Documentos/git/AV01_Practical_7 (release/v1.0.0)
$
```

Terminamos el relese No. 1

- Fusionamos la rama release de nuevo en master.
- Etiquetamos el comunicado con su nombre.
- Realizamos la función inversa del release en develop

```
fern@DESKTOP-0C3SE91 R106a64 ~/OneDrive/Documentos/git/AY01_Practical_7 (release/v1.0.0)
$ git flow release finish -m "Primera versión que muestra los datos de los estudiantes en una tabla como mensaje de bienvenida" v1.0.0
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
Merge made by the 'recursive' strategy.
 appweb/src/views/practical1/dashboard.js | 394 +++++
 1 file changed, 394 insertions(+)
Already on 'master'
Your branch is ahead of 'origin/master' by 7 commits.
  (use "git push" to publish your local commits)
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.
Already up to date!
Merge made by the 'recursive' strategy.
Deleted branch release/v1.0.0 (was e842b65).

Summary of actions:
- Release branch 'release/v1.0.0' has been merged into 'master'
- The release was tagged 'v1.0.0'
- Release tag 'v1.0.0' has been back-merged into 'develop'
- Release branch 'release/v1.0.0' has been locally deleted
- You are now on branch 'develop'

fern@DESKTOP-0C3SE91 R106a64 ~/OneDrive/Documentos/git/AY01_Practical_7 (develop)
$
```

```
fern@DESKTOP-0C3SE91 R106a64 ~/OneDrive/Documentos/git/AY01_Practical_7 (develop)
$ git push origin develop
Enumerating objects: 2, done.
Counting objects: 100% (2/2), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 417 bytes | 208.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/erickace/AY01_Practical_7.git
   e842b65..9386c99  develop -> develop

fern@DESKTOP-0C3SE91 R106a64 ~/OneDrive/Documentos/git/AY01_Practical_7 (develop)
$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 7 commits.
  (use "git push" to publish your local commits)

fern@DESKTOP-0C3SE91 R106a64 ~/OneDrive/Documentos/git/AY01_Practical_7 (master)
$ git push origin master
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/erickace/AY01_Practical_7.git
   c2bcc45..da8783b  master -> master

fern@DESKTOP-0C3SE91 R106a64 ~/OneDrive/Documentos/git/AY01_Practical_7 (master)
$
```

Segundo release

Creamos la **feature backend**

```
fern@DESKTOP-OK3SE91 PTMG464 ~/OneDrive/Documentos/git/AY01_Practical_7 (master)
$ git checkout develop
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.

fern@DESKTOP-OK3SE91 PTMG464 ~/OneDrive/Documentos/git/AY01_Practical_7 (develop)
$ git flow feature start backend
Switched to a new branch 'feature/backend'

Summary of actions:
- A new branch 'feature/backend' was created, based on 'develop'
- You are now on branch 'feature/backend'

Now, start committing on your feature. When done, use:

    git flow feature finish backend

fern@DESKTOP-OK3SE91 PTMG464 ~/OneDrive/Documentos/git/AY01_Practical_7 (feature/backend)
$
```

Público la feature en el remoto para que los demás usuarios puedan usarlas

```
fern@DESKTOP-OK3SE91 PTMG464 ~/OneDrive/Documentos/git/AY01_Practical_7 (feature/backend)
$ git flow feature publish backend
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'feature/backend' on Github by visiting:
remote:   https://github.com/erickace/AY01_Practical_7/pull/new/feature/backend
remote:
To https://github.com/erickace/AY01_Practical_7.git
 * [new branch]      feature/backend -> feature/backend
Branch 'feature/backend' set up to track remote branch 'feature/backend' from 'origin'.
Already on 'feature/backend'
Your branch is up to date with 'origin/feature/backend'.

Summary of actions:
- The remote branch 'feature/backend' was created or updated
- The local branch 'feature/backend' was configured to track the remote branch
- You are now on branch 'feature/backend'

fern@DESKTOP-OK3SE91 PTMG464 ~/OneDrive/Documentos/git/AY01_Practical_7 (feature/backend)
$
```

Creamos la **feature fronted**

```
fern@DESKTOP-OK3SE91 PTMG464 ~/OneDrive/Documentos/git/AY01_Practical_7 (develop)
$ git flow feature start fronted
Switched to a new branch 'feature/fronted'

Summary of actions:
- A new branch 'feature/fronted' was created, based on 'develop'
- You are now on branch 'feature/fronted'

Now, start committing on your feature. When done, use:

    git flow feature finish fronted
```

Público la feature en el remoto para que los demás usuarios puedan usarlas

```
fern@DESKTOP-0K3SE91 PING664 ~/OneDrive/Documentos/git/AYD1_Practical_7 (feature/fronted)
$ git flow feature publish fronted
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'feature/fronted' on GitHub by visiting:
remote:   https://github.com/erickace/AYD1_Practical_7/pull/new/feature/fronted
remote:
To https://github.com/erickace/AYD1_Practical_7.git
 * [new branch]      feature/fronted -> feature/fronted
Branch 'feature/fronted' set up to track remote branch 'feature/fronted' from 'origin'.
Already on 'feature/fronted'
Your branch is up to date with 'origin/feature/fronted'.

Summary of actions:
- The remote branch 'feature/fronted' was created or updated
- The local branch 'feature/fronted' was configured to track the remote branch
- You are now on branch 'feature/fronted'

fern@DESKTOP-0K3SE91 PING664 ~/OneDrive/Documentos/git/AYD1_Practical_7 (feature/fronted)
$
```

Añadir las funcionalidades de CRUD (create) y luego efectuó el respectivo commit

```
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   appweb/package-lock.json
        modified:   appweb/package.json
        modified:   appweb/src/views/practical/libros.js

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        appweb/src/views/practical/style.css

no changes added to commit (use "git add" and/or "git commit -a")

fern@DESKTOP-0K3SE91 PING664 ~/OneDrive/Documentos/git/AYD1_Practical_7 (feature/fronted)
$ git add .
warning: LF will be replaced by CRLF in appweb/package-lock.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in appweb/package.json.
The file will have its original line endings in your working directory

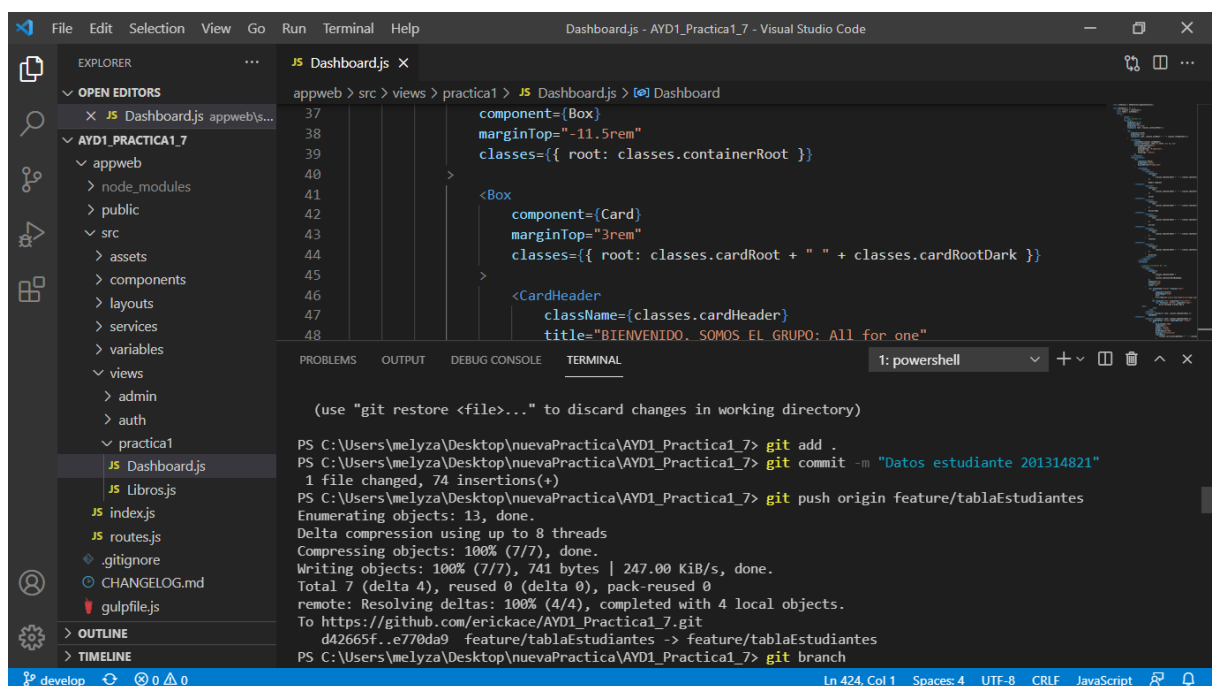
fern@DESKTOP-0K3SE91 PING664 ~/OneDrive/Documentos/git/AYD1_Practical_7 (feature/fronted)
$ git status
On branch feature/fronted
Your branch is up to date with 'origin/feature/fronted'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   appweb/package-lock.json
        modified:   appweb/package.json
        modified:   appweb/src/views/practical/libros.js
        new file:   appweb/src/views/practical/style.css
```

Melyza Alejandra Rodriguez Contreras - 201314821

Primer Release

- git checkout --track origin/feature/tablaEstudiantes
- git status
- git pull
- git add .
- git commit -m "Datos estudiante 201314821"



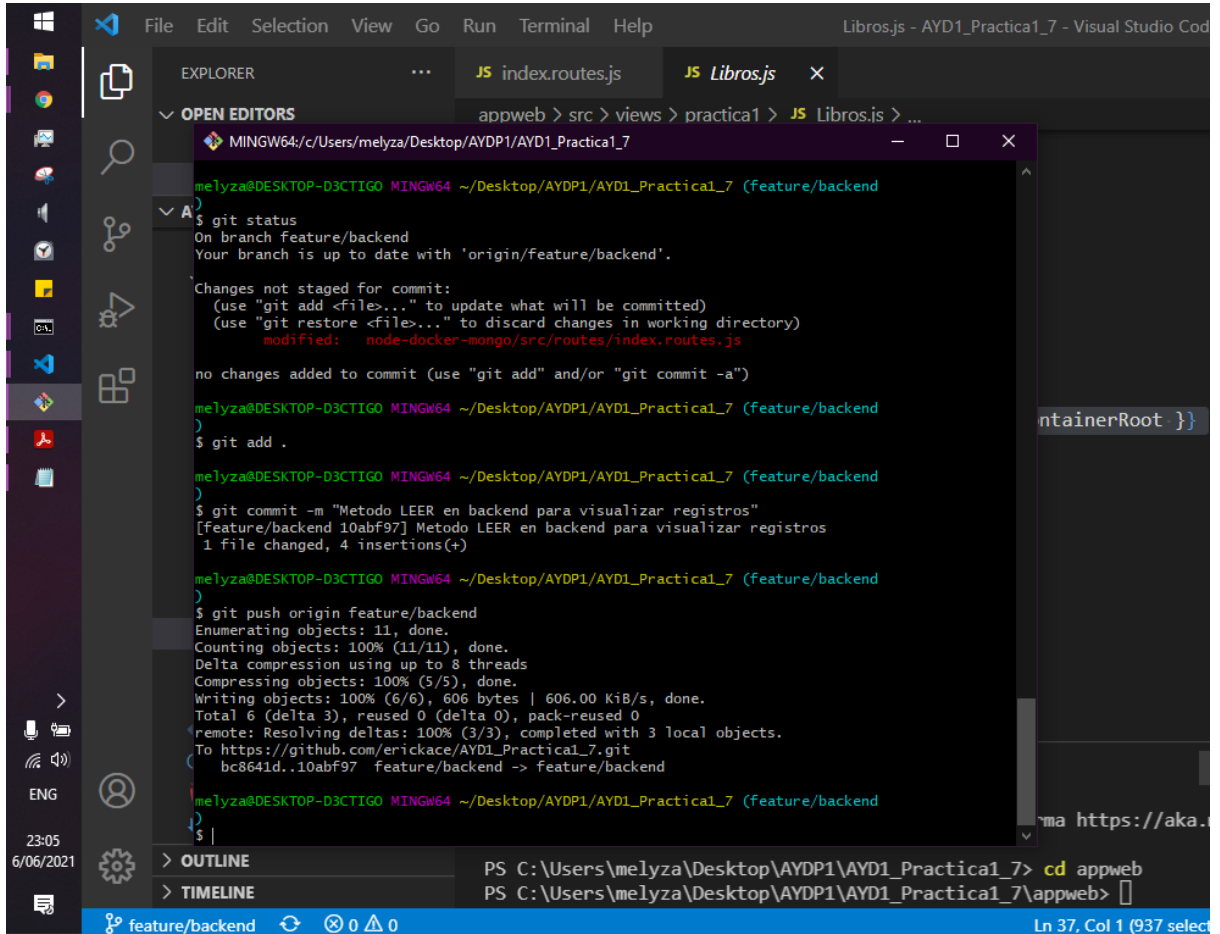
The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows the project structure with folders like 'appweb', 'src', 'views', and 'practica1'. The 'practica1' folder is expanded, showing 'Dashboard.js'. The main editor displays the content of 'Dashboard.js', which includes JSX code for a 'Box' component and a 'CardHeader' component. The Terminal panel at the bottom shows the execution of several git commands in a PowerShell session:

```
(use "git restore <file>..." to discard changes in working directory)
PS C:\Users\melyza\Desktop\nuevaPractica\AYD1_Practica1_7> git add .
PS C:\Users\melyza\Desktop\nuevaPractica\AYD1_Practica1_7> git commit -m "Datos estudiante 201314821"
1 file changed, 74 insertions(+)
PS C:\Users\melyza\Desktop\nuevaPractica\AYD1_Practica1_7> git push origin feature/tablaEstudiantes
Enumerating objects: 13, done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 741 bytes | 247.00 KiB/s, done.
Total 7 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/erickace/AYD1_Practica1_7.git
d42665f..e770da9 feature/tablaEstudiantes -> feature/tablaEstudiantes
PS C:\Users\melyza\Desktop\nuevaPractica\AYD1_Practica1_7> git branch
```

The status bar at the bottom indicates the current branch is 'develop', with 0 changes and 0 conflicts. The file encoding is UTF-8, and the line endings are CRLF.

Segundo Release

- git checkout --track origin/feature/backend
- git status
- git add .
- git commit -m "Metodo LEER en backend para visualizar registros"
- git push origin feature/backend



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left shows the file structure of the project. The main editor area displays the 'index.routes.js' file. A terminal window is open at the bottom, showing the execution of the following git commands:

```
melyza@DESKTOP-D3CTIGO MINGW64 ~/Desktop/AYDP1/AYD1_Practica1_7 (feature/backend)
$ git status
On branch feature/backend
Your branch is up to date with 'origin/feature/backend'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   node-docker-mongo/src/routes/index.routes.js

no changes added to commit (use "git add" and/or "git commit -a")

melyza@DESKTOP-D3CTIGO MINGW64 ~/Desktop/AYDP1/AYD1_Practica1_7 (feature/backend)
$ git add .

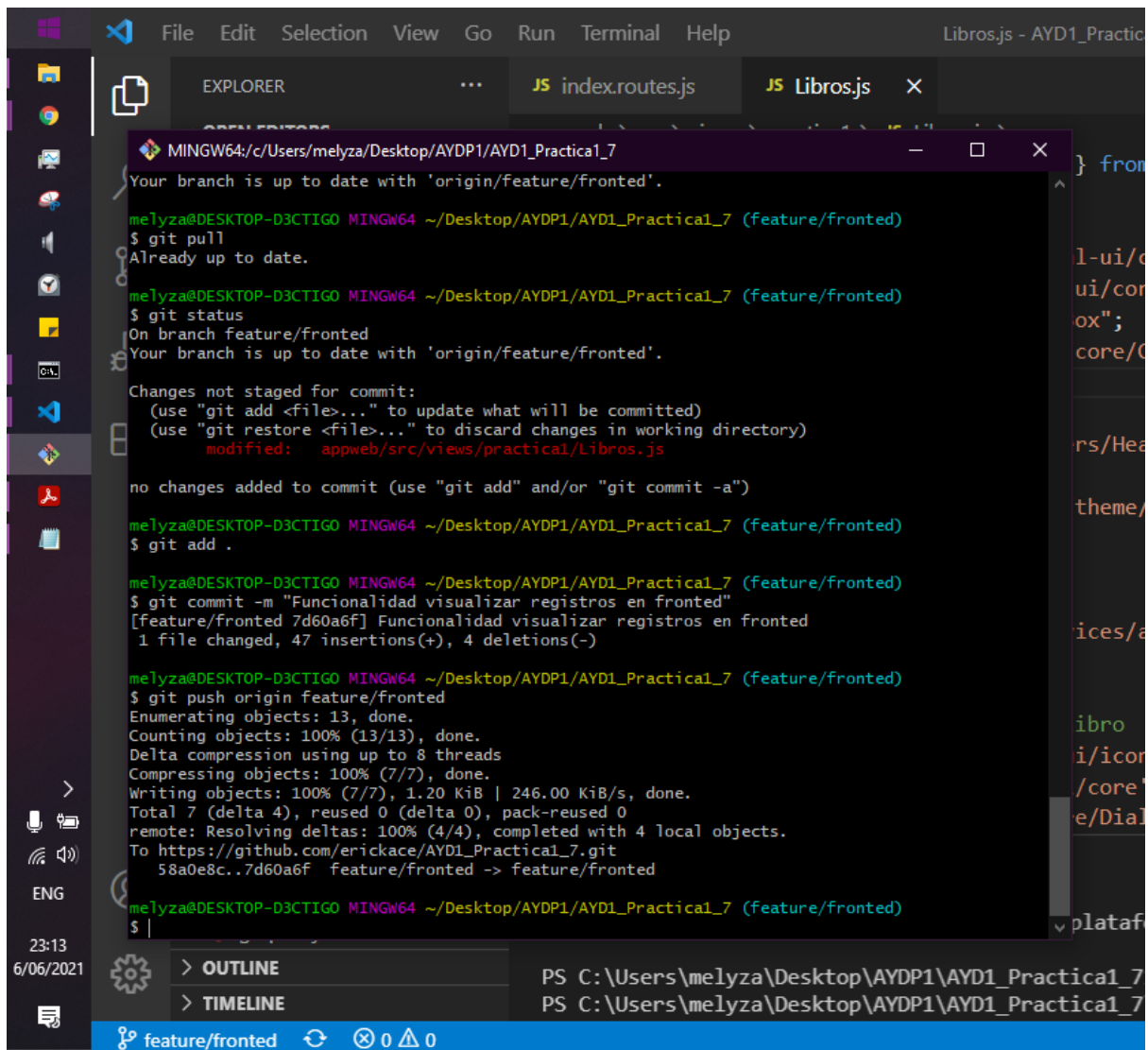
melyza@DESKTOP-D3CTIGO MINGW64 ~/Desktop/AYDP1/AYD1_Practica1_7 (feature/backend)
$ git commit -m "Metodo LEER en backend para visualizar registros"
[feature/backend 10abf97] Metodo LEER en backend para visualizar registros
 1 file changed, 4 insertions(+)

melyza@DESKTOP-D3CTIGO MINGW64 ~/Desktop/AYDP1/AYD1_Practica1_7 (feature/backend)
$ git push origin feature/backend
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 606 bytes | 606.00 KiB/s, done.
Total 6 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/erickace/AYD1_Practica1_7.git
   bc8641d..10abf97  feature/backend -> feature/backend

melyza@DESKTOP-D3CTIGO MINGW64 ~/Desktop/AYDP1/AYD1_Practica1_7 (feature/backend)
$ |
```

The terminal output shows the successful execution of the git commands, including the push to the remote repository. The status bar at the bottom indicates the current branch is 'feature/backend' and the file 'index.routes.js' is modified.

- git checkout --track origin/feature/frontend
- git status
- git add .
- git commit -m "Funcionalidad visualizar registros en frontend"
- git push origin feature/frontend



The screenshot shows a Visual Studio Code editor with a terminal window open. The terminal is running a series of Git commands in a MINGW64 environment. The user is currently on the 'feature/fronted' branch. The commands and their outputs are as follows:

```
mingw64~/Desktop/AYDP1/AYD1_Practica1_7 (feature/fronted)
$ git pull
Already up to date.

mingw64~/Desktop/AYDP1/AYD1_Practica1_7 (feature/fronted)
$ git status
On branch feature/fronted
Your branch is up to date with 'origin/feature/fronted'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   appweb/src/views/practical1/Libros.js

no changes added to commit (use "git add" and/or "git commit -a")

mingw64~/Desktop/AYDP1/AYD1_Practica1_7 (feature/fronted)
$ git add .

mingw64~/Desktop/AYDP1/AYD1_Practica1_7 (feature/fronted)
$ git commit -m "Funcionalidad visualizar registros en fronted"
[feature/fronted 7d60a6f] Funcionalidad visualizar registros en fronted
1 file changed, 47 insertions(+), 4 deletions(-)

mingw64~/Desktop/AYDP1/AYD1_Practica1_7 (feature/fronted)
$ git push origin feature/fronted
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 1.20 KiB | 246.00 KiB/s, done.
Total 7 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/erickace/AYD1_Practica1_7.git
  58a0e8c..7d60a6f  feature/fronted -> feature/fronted

mingw64~/Desktop/AYDP1/AYD1_Practica1_7 (feature/fronted)
$ |
```

The bottom status bar of VS Code shows the current branch as 'feature/fronted' and the file explorer as 'OUTLINE' and 'TIMELINE'.

Creación de segundo Release

- git checkout develop
- git branch release/v1.2.0
- git branch
- git checkout release/v1.2.0
- git push origin release/v1.2.0

```
MINGW64:/c/Users/melyza/Desktop/AYDP1/AYD1_Practical_7
Already up to date.

melyza@DESKTOP-D3CTIGO MINGW64 ~/Desktop/AYDP1/AYD1_Practical_7 (develop)
$ git branch release/v1.2.0

melyza@DESKTOP-D3CTIGO MINGW64 ~/Desktop/AYDP1/AYD1_Practical_7 (develop)
$ git branch
* develop
  feature/backend
  feature/Backend
  feature/tablaEstudiantes
  master
  release/v1.2.0

melyza@DESKTOP-D3CTIGO MINGW64 ~/Desktop/AYDP1/AYD1_Practical_7 (develop)
$ git checkout release/v1.2.0
Switched to branch 'release/v1.2.0'

melyza@DESKTOP-D3CTIGO MINGW64 ~/Desktop/AYDP1/AYD1_Practical_7 (release/v1.2.0)
$ git push origin release/v1.2.0
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'release/v1.2.0' on GitHub by visiting:
remote:   https://github.com/erickace/AYD1_Practical_7/pull/new/release/v1.2.0
remote:
remote: To https://github.com/erickace/AYD1_Practical_7.git
* [new branch]      release/v1.2.0 -> release/v1.2.0

melyza@DESKTOP-D3CTIGO MINGW64 ~/Desktop/AYDP1/AYD1_Practical_7 (release/v1.2.0)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

melyza@DESKTOP-D3CTIGO MINGW64 ~/Desktop/AYDP1/AYD1_Practical_7 (master)
$ git pull
Already up to date.

melyza@DESKTOP-D3CTIGO MINGW64 ~/Desktop/AYDP1/AYD1_Practical_7 (master)
$ git merge release/v1.2.0
Updating da8703b..510b522
Fast-forward
 appweb/package-lock.json      | 276 ++++++++
 appweb/package.json           | 8 +-
 appweb/src/views/practical1/Libros.js | 745 ++++++++
 appweb/src/views/practical1/style.css | 17 +
 node-docker-mongo/src/model/item.js | 8 +
 node-docker-mongo/src/routes/index.routes.js | 67 +++
 6 files changed, 1109 insertions(+), 12 deletions(-)
 create mode 100644 appweb/src/views/practical1/style.css

melyza@DESKTOP-D3CTIGO MINGW64 ~/Desktop/AYDP1/AYD1_Practical_7 (master)
$
```

- git checkout master
- git pull
- git merge release/v1.2.0
- git push origin master
- git checkout develop
- git merge release/v1.2.0
- git push origin develop


```
MINGW64:/c/Users/melyza/Desktop/AYDP1/AYD1_Practica1_7
$ git push origin release/v1.2.0
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'release/v1.2.0' on GitHub by visiting:
remote:   https://github.com/erickace/AYD1_Practica1_7/pull/new/release/v1.2.0
remote:
To https://github.com/erickace/AYD1_Practica1_7.git
 * [new branch]      release/v1.2.0 -> release/v1.2.0

melyza@DESKTOP-D3CTIGO MINGW64 ~/Desktop/AYDP1/AYD1_Practica1_7 (release/v1.2.0)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

melyza@DESKTOP-D3CTIGO MINGW64 ~/Desktop/AYDP1/AYD1_Practica1_7 (master)
$ git pull
Already up to date.

melyza@DESKTOP-D3CTIGO MINGW64 ~/Desktop/AYDP1/AYD1_Practica1_7 (master)
$ git merge release/v1.2.0
Updating da8703b..510b522
Fast-forward
 appweb/package-lock.json      | 276 ++++++++
 appweb/package.json           | 8 +-
 appweb/src/views/practical/Libros.js | 745 +++++
 appweb/src/views/practical/style.css | 17 +
 node-docker-mongo/src/model/item.js | 8 +
 node-docker-mongo/src/routes/index.routes.js | 67 +++
 6 files changed, 1109 insertions(+), 12 deletions(-)
 create mode 100644 appweb/src/views/practical/style.css

melyza@DESKTOP-D3CTIGO MINGW64 ~/Desktop/AYDP1/AYD1_Practica1_7 (master)
$ git push origin master
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/erickace/AYD1_Practica1_7.git
 da8703b..510b522 master -> master

melyza@DESKTOP-D3CTIGO MINGW64 ~/Desktop/AYDP1/AYD1_Practica1_7 (master)
$ git checkout develop
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.

melyza@DESKTOP-D3CTIGO MINGW64 ~/Desktop/AYDP1/AYD1_Practica1_7 (develop)
$ git merge release/v1.2.0
Already up to date.

melyza@DESKTOP-D3CTIGO MINGW64 ~/Desktop/AYDP1/AYD1_Practica1_7 (develop)
$ git push origin develop
Everything up-to-date

melyza@DESKTOP-D3CTIGO MINGW64 ~/Desktop/AYDP1/AYD1_Practica1_7 (develop)
$ |
```

Creación de release desde Github

Releases

Tags

Latest release

v1.2.0

510b522

Compare

Funcionalidades CRUD Libros

MelyzaRC released this now

Segunda versión

Se agregaron funcionalidades CRUD para libros.



Assets 2

[Source code \(zip\)](#)

[Source code \(tar.gz\)](#)

Christopher Alexander Acajalon Gudiel - 201404278

Primer Release

- Git fetch
- Git checkout feature/tablaEstudiante
- Git status
- Git pull

```
MINGW64: C:\ARCHIVOS ECHPECH\PROGRAMACION\ANALISIS Y DISEÑO DE SISTEMAS\LAB\PRACTICA1\W01_Practica1_7
C:\ARCHIVOS ECHPECH\PROGRAMACION\ANALISIS Y DISEÑO DE SISTEMAS\LAB\PRACTICA1\W01_Practica1_7
PS C:\ARCHIVOS ECHPECH\PROGRAMACION\ANALISIS Y DISEÑO DE SISTEMAS\LAB\PRACTICA1\W01_Practica1_7> git branch
* master

C:\ARCHIVOS ECHPECH\PROGRAMACION\ANALISIS Y DISEÑO DE SISTEMAS\LAB\PRACTICA1\W01_Practica1_7
PS C:\ARCHIVOS ECHPECH\PROGRAMACION\ANALISIS Y DISEÑO DE SISTEMAS\LAB\PRACTICA1\W01_Practica1_7> git fetch
remote: Counting objects: 11, done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 7 (delta 0), reused 7 (delta 0), pack-reused 0
Unpacking objects: 100% (7/7), 1.91 KiB | 3.00 KiB/s, done.
From https://github.com/echpech/ANALISIS_Y_DISEÑO_DE_SISTEMAS
* [new branch]      develop    -> origin/develop
+ [new branch]      Feature/tablaEstudiantes -> origin/Feature/tablaEstudiantes

C:\ARCHIVOS ECHPECH\PROGRAMACION\ANALISIS Y DISEÑO DE SISTEMAS\LAB\PRACTICA1\W01_Practica1_7
PS C:\ARCHIVOS ECHPECH\PROGRAMACION\ANALISIS Y DISEÑO DE SISTEMAS\LAB\PRACTICA1\W01_Practica1_7> git branch
git: 'branch' is not a git command. See 'git --help'.

The most similar command is
  branch

C:\ARCHIVOS ECHPECH\PROGRAMACION\ANALISIS Y DISEÑO DE SISTEMAS\LAB\PRACTICA1\W01_Practica1_7
PS C:\ARCHIVOS ECHPECH\PROGRAMACION\ANALISIS Y DISEÑO DE SISTEMAS\LAB\PRACTICA1\W01_Practica1_7> git branch
* master

C:\ARCHIVOS ECHPECH\PROGRAMACION\ANALISIS Y DISEÑO DE SISTEMAS\LAB\PRACTICA1\W01_Practica1_7
PS C:\ARCHIVOS ECHPECH\PROGRAMACION\ANALISIS Y DISEÑO DE SISTEMAS\LAB\PRACTICA1\W01_Practica1_7> git checkout develop
Switched to a new branch 'develop'
branch 'develop' set up to track remote branch 'develop' from 'origin'.

C:\ARCHIVOS ECHPECH\PROGRAMACION\ANALISIS Y DISEÑO DE SISTEMAS\LAB\PRACTICA1\W01_Practica1_7
PS C:\ARCHIVOS ECHPECH\PROGRAMACION\ANALISIS Y DISEÑO DE SISTEMAS\LAB\PRACTICA1\W01_Practica1_7> git checkout Feature/tablaEstudiantes
Switched to a new branch 'Feature/tablaEstudiantes'
branch 'Feature/tablaEstudiantes' set up to track remote branch 'Feature/tablaEstudiantes' from 'origin'.

C:\ARCHIVOS ECHPECH\PROGRAMACION\ANALISIS Y DISEÑO DE SISTEMAS\LAB\PRACTICA1\W01_Practica1_7
PS C:\ARCHIVOS ECHPECH\PROGRAMACION\ANALISIS Y DISEÑO DE SISTEMAS\LAB\PRACTICA1\W01_Practica1_7> git status
On branch Feature/tablaEstudiantes
Your branch is up to date with 'origin/Feature/tablaEstudiantes'.

nothing to commit, working tree clean

C:\ARCHIVOS ECHPECH\PROGRAMACION\ANALISIS Y DISEÑO DE SISTEMAS\LAB\PRACTICA1\W01_Practica1_7
PS C:\ARCHIVOS ECHPECH\PROGRAMACION\ANALISIS Y DISEÑO DE SISTEMAS\LAB\PRACTICA1\W01_Practica1_7> git pull
Already up to date.

C:\ARCHIVOS ECHPECH\PROGRAMACION\ANALISIS Y DISEÑO DE SISTEMAS\LAB\PRACTICA1\W01_Practica1_7
PS C:\ARCHIVOS ECHPECH\PROGRAMACION\ANALISIS Y DISEÑO DE SISTEMAS\LAB\PRACTICA1\W01_Practica1_7>
```

- Git add .
- Git commit
- Git push origin feature/tablaEstudiantes

```
MINIARRA\ARCHIVOS E-CRIPCEN\PROGRAMACION\ANALISIS Y DISEÑO DE SISTEMAS\LAB\PRACTICA\WD1_Prac2\7
C:\Users\alvaro\source\repos\WD1_Prac2\7>git status
# SE ENCONTRO EN LA RAMA DE ORIGEN (feature/Tabladestudiantes)
# git status
On branch feature/Tabladestudiantes
Your branch is up to date with 'origin/feature/Tabladestudiantes'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   src\view\practical\dashboard.js

no changes added to commit (use "git add" and/or "git commit -a")

C:\Users\alvaro\source\repos\WD1_Prac2\7>git add src\view\practical\dashboard.js
# SE ENCONTRO EN LA RAMA DE ORIGEN (feature/Tabladestudiantes)
# git add
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 426 bytes | 426.00 KiB/s, done.
Total 7 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/sericidaco/WD1_Prac2\7.git
   31250c..d428d7  feature/Tabladestudiantes -> feature/Tabladestudiantes

C:\Users\alvaro\source\repos\WD1_Prac2\7>git push
remote: Enumerating objects: 11, done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 7 (delta 4), reused 0 (delta 4), pack-reused 0
uploading objects: 100% (7/7), 722 bytes | 2.90 KiB/s, done.
From https://github.com/sericidaco/WD1_Prac2\7
   d428d7..e770d6  feature/Tabladestudiantes -> origin/feature/Tabladestudiantes
Updating d428d7..e770d6
Fast-forward
   src\view\practical\dashboard.js | 74 +++++++++++++++++++++
   1 file changed, 74 insertions(+)
```

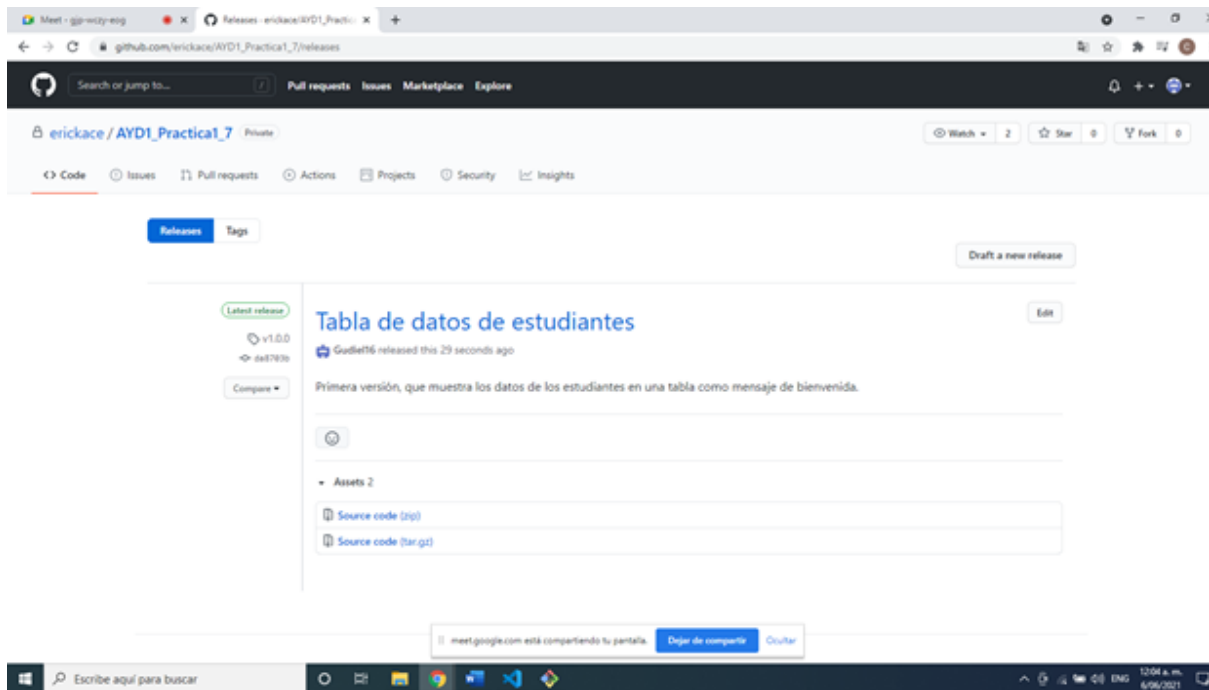
- Git checkout
- Git pull

```
MINIARRA\ARCHIVOS E-CRIPCEN\PROGRAMACION\ANALISIS Y DISEÑO DE SISTEMAS\LAB\PRACTICA\WD1_Prac2\7
C:\Users\alvaro\source\repos\WD1_Prac2\7>git pull
remote: Enumerating objects: 11, done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 7 (delta 4), reused 0 (delta 4), pack-reused 0
unpacking objects: 100% (7/7), 485 bytes | 1.04 KiB/s, done.
From https://github.com/sericidaco/WD1_Prac2\7
   625d32..d428d7  feature/Tabladestudiantes -> origin/feature/Tabladestudiantes
Updating 625d32..d428d7
Fast-forward
   src\view\practical\dashboard.js | 2 +-
   1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\alvaro\source\repos\WD1_Prac2\7>git checkout develop
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.

C:\Users\alvaro\source\repos\WD1_Prac2\7>git pull
remote: Enumerating objects: 1, done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
unpacking objects: 100% (1/1), 212 bytes | 1.30 KiB/s, done.
From https://github.com/sericidaco/WD1_Prac2\7
   c20c43..48d3d3  develop -> origin/develop
Updating c20c43..48d3d3
Fast-forward
   src\view\practical\dashboard.js | 104 +++++++++++++++++++++++++++++++++++++
   1 file changed, 104 insertions(+)
```

- Realice 1 creado desde GitHub



Segundo Release

- Git fetch
- Git checkout feature/backend
- Git pull



- Git status

- ```

MINGW64 ~/ARCHIVOS/E/CHIPCIN/PROGRAMACION/ANALISIS/VIDEO DE SISTEMAS 1/LAB/PRACTICAL/AVDL_Practical_7
$ git pull
remote: Enumerating objects: 47, done.
remote: Counting objects: 100% (47/47), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 27 (delta 15), reused 27 (delta 15), pack-reused 0
Unpacking objects: 100% (27/27), 4.55 KiB | 1.50 MiB/s, done.
From https://github.com/erickcruiz/AVDL_Practical_7
 10d9f07..abd01ad feature/backed ~> origin/feature/backed
 10d9f07..abd01ad feature/fronked ~> origin/feature/fronked
Updating 10d9f07..abd01ad
Fast-forward
 index.html ~> index.html
 1 file changed, 18 insertions(+)

$ git push --set-upstream origin feature/backed
$ git status
On branch feature/backed
Your branch is up to date with 'origin/feature/backed'.

Changes not staged for commit:
 (use "git add <file>..." to update what will be committed)
 (use "git restore <file>..." to discard changes in working directory)

no changes added to commit (use "git add" and/or "git commit -a")

$ git push --set-upstream origin feature/backed
$ git status
On branch feature/backed
Your branch is ahead of 'origin/feature/backed' by 1 commit.
 (use "git push" to publish your local commits)

nothing to commit, working tree clean

$ git push --set-upstream origin feature/backed
$ git push origin feature/backed
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 424 bytes | 313.96 KiB/s, done.
Total 3 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/erickcruiz/AVDL_Practical_7.git
 abd01ad..7f6ab6a feature/backed ~> feature/backed

```

- ```

C:\WINDOWS\system32\cmd.exe /s /c cd C:\Users\jorge\Documents\ANALISIS Y DISEÑO DE SISTEMAS I\LAB\PRACTICA1\WVL_PRACTICA1_7
C:\Users\jorge\Documents\ANALISIS Y DISEÑO DE SISTEMAS I\LAB\PRACTICA1\WVL_PRACTICA1_7>
C:\Users\jorge\Documents\ANALISIS Y DISEÑO DE SISTEMAS I\LAB\PRACTICA1\WVL_PRACTICA1_7>git status
On branch Feature/Frontend
Your branch is up to date with 'origin/Feature/Frontend'.

nothing to commit, working tree clean
C:\Users\jorge\Documents\ANALISIS Y DISEÑO DE SISTEMAS I\LAB\PRACTICA1\WVL_PRACTICA1_7>git checkout Feature/Backend
Switched to a new branch 'Feature/Backend'
C:\Users\jorge\Documents\ANALISIS Y DISEÑO DE SISTEMAS I\LAB\PRACTICA1\WVL_PRACTICA1_7>git pull
Already up to date.
C:\Users\jorge\Documents\ANALISIS Y DISEÑO DE SISTEMAS I\LAB\PRACTICA1\WVL_PRACTICA1_7>git branch
* Feature/Backend
  Feature/Frontend
  master
C:\Users\jorge\Documents\ANALISIS Y DISEÑO DE SISTEMAS I\LAB\PRACTICA1\WVL_PRACTICA1_7>git pull
Already up to date.
C:\Users\jorge\Documents\ANALISIS Y DISEÑO DE SISTEMAS I\LAB\PRACTICA1\WVL_PRACTICA1_7>git status
On branch Feature/Frontend
Your branch is up to date with 'origin/Feature/Frontend'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   appweb/src/views/practica1/3/index.js

no changes added to commit (use "git add" and/or "git commit -a")
C:\Users\jorge\Documents\ANALISIS Y DISEÑO DE SISTEMAS I\LAB\PRACTICA1\WVL_PRACTICA1_7>git add .
C:\Users\jorge\Documents\ANALISIS Y DISEÑO DE SISTEMAS I\LAB\PRACTICA1\WVL_PRACTICA1_7>git commit -m "Agregando funciones para eliminar registros en el frontend"
[master 7eb3683] Agregando funciones para eliminar registros en el frontend
 3 files changed, 47 insertions(+), 47 deletions(-)
C:\Users\jorge\Documents\ANALISIS Y DISEÑO DE SISTEMAS I\LAB\PRACTICA1\WVL_PRACTICA1_7>git status
On branch Feature/Frontend
Your branch is ahead of 'origin/Feature/Frontend' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

```

- Git push origin feature/fronted
- Git checkout
- Git merge -> feature/backend
- Git merge -> feature/fronted

```

MINGW64/ARCHIVOS E/CHPECH/PROGRAMACION/ANALISIS Y DISEÑO DE SISTEMAS 1/LAB/PRACTICA1/AVDL_Practica1_7
C:\Users\KINGM4>cd /d C:\Users\KINGM4\Documents\E/CHPECH/PROGRAMACION/ANALISIS Y DISEÑO DE SISTEMAS 1/LAB/PRACTICA1/AVDL_Practica1_7 (Feature/fronted)
C:\Users\KINGM4>git push origin Feature/fronted
error: src refspec Feature/fronted does not match any
error: failed to push some refs to 'https://github.com/serchada/AVDL_Practica1_7.git'

C:\Users\KINGM4>cd /d C:\Users\KINGM4\Documents\E/CHPECH/PROGRAMACION/ANALISIS Y DISEÑO DE SISTEMAS 1/LAB/PRACTICA1/AVDL_Practica1_7 (Feature/fronted)
C:\Users\KINGM4>git push origin Feature/fronted
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 2.18 KiB | 435.00 KiB/s, done.
Total 7 (delta 4), reused 5 (delta 0), push-reused 5
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/serchada/AVDL_Practica1_7.git
  6c16174..f6c3f43 Feature/fronted -> Feature/fronted

C:\Users\KINGM4>cd /d C:\Users\KINGM4\Documents\E/CHPECH/PROGRAMACION/ANALISIS Y DISEÑO DE SISTEMAS 1/LAB/PRACTICA1/AVDL_Practica1_7 (Feature/fronted)
C:\Users\KINGM4>git checkout develop
Switched to branch 'develop'
Your branch is behind 'origin/develop' by 2 commits, and can be fast-forwarded.
  (use "git pull" to update your local branch)

C:\Users\KINGM4>cd /d C:\Users\KINGM4\Documents\E/CHPECH/PROGRAMACION/ANALISIS Y DISEÑO DE SISTEMAS 1/LAB/PRACTICA1/AVDL_Practica1_7 (develop)
C:\Users\KINGM4>git branch
* develop
  Feature/Backend
  Feature/fronted
  Feature/tablaEstudiantes
  master

C:\Users\KINGM4>cd /d C:\Users\KINGM4\Documents\E/CHPECH/PROGRAMACION/ANALISIS Y DISEÑO DE SISTEMAS 1/LAB/PRACTICA1/AVDL_Practica1_7 (develop)
C:\Users\KINGM4>git merge --no-ff Feature/backend
Merge made by the 'recursive' strategy.
 node-docker-mongo/src/models/vistas.js | 8 +++
 node-docker-mongo/src/routes/index.routes.js | 67 ++++++
 2 files changed, 75 insertions(+)

C:\Users\KINGM4>cd /d C:\Users\KINGM4\Documents\E/CHPECH/PROGRAMACION/ANALISIS Y DISEÑO DE SISTEMAS 1/LAB/PRACTICA1/AVDL_Practica1_7 (develop)
C:\Users\KINGM4>git merge --no-ff Feature/fronted
Merge: Feature/fronted - not something we can merge

C:\Users\KINGM4>cd /d C:\Users\KINGM4\Documents\E/CHPECH/PROGRAMACION/ANALISIS Y DISEÑO DE SISTEMAS 1/LAB/PRACTICA1/AVDL_Practica1_7 (develop)
C:\Users\KINGM4>git merge --no-ff Feature/fronted
Merge made by the 'recursive' strategy.
 appweb/package-lock.json | 276 ++++++
 appweb/package.json | 8 +
 appweb/src/views/practica1/vistas.js | 741 ++++++
 appweb/src/views/practica1/style.css | 27 +
 4 files changed, 1034 insertions(+), 12 deletions(-)

```

- Git push origin develop
- Git fetch
- Git checkout
- Git pull

```

MINGW64/ARCHIVOS E/CHPECH/PROGRAMACION/ANALISIS Y DISEÑO DE SISTEMAS 1/LAB/PRACTICA1/AVDL_Practica1_7
C:\Users\KINGM4>cd /d C:\Users\KINGM4\Documents\E/CHPECH/PROGRAMACION/ANALISIS Y DISEÑO DE SISTEMAS 1/LAB/PRACTICA1/AVDL_Practica1_7
C:\Users\KINGM4>git push origin develop
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (3/3), 348 bytes | 348.00 KiB/s, done.
Total 3 (delta 0), reused 5 (delta 0), push-reused 5
To https://github.com/serchada/AVDL_Practica1_7.git
  8106131..5120522 develop -> develop

C:\Users\KINGM4>cd /d C:\Users\KINGM4\Documents\E/CHPECH/PROGRAMACION/ANALISIS Y DISEÑO DE SISTEMAS 1/LAB/PRACTICA1/AVDL_Practica1_7 (develop)
C:\Users\KINGM4>git branch
* develop
  Feature/Backend
  Feature/fronted
  Feature/tablaEstudiantes
  master

C:\Users\KINGM4>cd /d C:\Users\KINGM4\Documents\E/CHPECH/PROGRAMACION/ANALISIS Y DISEÑO DE SISTEMAS 1/LAB/PRACTICA1/AVDL_Practica1_7 (develop)
C:\Users\KINGM4>git fetch
From https://github.com/serchada/AVDL_Practica1_7
  develop: 5120522 => origin/develop
  (new branch)
  release/v1.0.0 -> origin/release/v1.0.0

C:\Users\KINGM4>cd /d C:\Users\KINGM4\Documents\E/CHPECH/PROGRAMACION/ANALISIS Y DISEÑO DE SISTEMAS 1/LAB/PRACTICA1/AVDL_Practica1_7 (develop)
C:\Users\KINGM4>git branch
* develop
  Feature/Backend
  Feature/fronted
  Feature/tablaEstudiantes
  master

C:\Users\KINGM4>cd /d C:\Users\KINGM4\Documents\E/CHPECH/PROGRAMACION/ANALISIS Y DISEÑO DE SISTEMAS 1/LAB/PRACTICA1/AVDL_Practica1_7 (develop)
C:\Users\KINGM4>git checkout master
Switched to branch 'master'
Your branch is behind 'origin/master' by 12 commits, and can be fast-forwarded.
  (use "git pull" to update your local branch)

C:\Users\KINGM4>cd /d C:\Users\KINGM4\Documents\E/CHPECH/PROGRAMACION/ANALISIS Y DISEÑO DE SISTEMAS 1/LAB/PRACTICA1/AVDL_Practica1_7 (master)
C:\Users\KINGM4>git pull
Updating 8106131..5120522
Fast-forward
 appweb/package-lock.json | 276 ++++++
 appweb/package.json | 8 +
 appweb/src/views/practica1/vistas.js | 741 ++++++
 appweb/src/views/practica1/style.css | 27 +
 node-docker-mongo/src/models/vistas.js | 8 +
 node-docker-mongo/src/routes/index.routes.js | 67 +
 6 files changed, 1034 insertions(+), 12 deletions(-)
 create mode 100644 appweb/src/views/practica1/style.css

```

Helmut Efrain Najarro Alvarez - 201712350

Primer release

1. Clonación de repositorio remoto
2. Clonación de remota develop

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19042.985]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\efrai>cd C:\Users\efrai\Desktop\Repos

C:\Users\efrai\Desktop\Repos>git clone https://github.com/erickace/AYD1_Practica1_7
Cloning into 'AYD1_Practica1_7'
remote: Enumerating objects: 179, done.
remote: Counting objects: 100% (179/179), done.
remote: Compressing objects: 100% (138/138), done.
remote: Total 179 (delta 29), reused 178 (delta 28), pack-reused 0 eceiving objects: 93% (167/179), 2.35 MiB | 590.00 KiB/s
Receiving objects: 100% (179/179), 2.52 MiB | 591.00 KiB/s, done.
Resolving deltas: 100% (29/29), done.

C:\Users\efrai\Desktop\Repos>git cheackout

C:\Users\efrai\Desktop\Repos>git checkout
fatal: not a git repository (or any of the parent directories): .git

C:\Users\efrai\Desktop\Repos>git branch
fatal: not a git repository (or any of the parent directories): .git

C:\Users\efrai\Desktop\Repos>cd C:\Users\efrai\Desktop\Repos\AYD1_Practica1_7

C:\Users\efrai\Desktop\Repos\AYD1_Practica1_7>git branch
* master

C:\Users\efrai\Desktop\Repos\AYD1_Practica1_7>git checkout --track origin/develop
Switched to a new branch 'develop'
Branch 'develop' set up to track remote branch 'develop' from 'origin'.
```

3. Clonación de rama remota feature/tablaEstudiantes
4. Agregación de datos en la tabla
5. Commit
6. Push


```
Símbolo del sistema

C:\Users\efrai\Desktop\Repos\AYD1_Practica1_7>git checkout --track origin/feature/tablaEstudiantes
Switched to a new branch 'feature/tablaEstudiantes'
Branch 'feature/tablaEstudiantes' set up to track remote branch 'feature/tablaEstudiantes' from 'origin'.

C:\Users\efrai\Desktop\Repos\AYD1_Practica1_7>git pull
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 7 (delta 4), reused 7 (delta 4), pack-reused 0
Unpacking objects: 100% (7/7), 721 bytes | 1024 bytes/s, done.
From https://github.com/erickace/AYD1_Practica1_7
   d42665f..e770da9  feature/tablaEstudiantes -> origin/feature/tablaEstudiantes
Updating d42665f..e770da9
Fast-forward
 appweb/src/views/practica1/Dashboard.js | 74 ++++++
 1 file changed, 74 insertions(+)

C:\Users\efrai\Desktop\Repos\AYD1_Practica1_7>git add .

C:\Users\efrai\Desktop\Repos\AYD1_Practica1_7>git commit -m "Agregando datos personales - 201712350"
[feature/tablaEstudiantes 62b1c92] Agregando datos personales - 201712350
 1 file changed, 74 insertions(+), 2 deletions(-)

C:\Users\efrai\Desktop\Repos\AYD1_Practica1_7>git push
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 729 bytes | 729.00 KiB/s, done.
Total 7 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/erickace/AYD1_Practica1_7
   e770da9..62b1c92  feature/tablaEstudiantes -> feature/tablaEstudiantes
```

7. Agregación de datos corregidos en la tabla
8. Commit
9. Push
10. Merge la rama develop con la rama feature/tablaEstudiantes
11. Push

```

C:\Users\efrai\Desktop\Repos\AYD1_Practica1_7>git add .
C:\Users\efrai\Desktop\Repos\AYD1_Practica1_7>git commit -m "Correccion Datos - 201712350"
[feature/tablaEstudiantes d386ed9] Correccion Datos - 201712350
1 file changed, 1 insertion(+), 1 deletion(-)
C:\Users\efrai\Desktop\Repos\AYD1_Practica1_7>git push
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 625 bytes | 625.00 KiB/s, done.
Total 7 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/erickace/AYD1_Practica1_7
62b1c92..d386ed9 feature/tablaEstudiantes -> feature/tablaEstudiantes
C:\Users\efrai\Desktop\Repos\AYD1_Practica1_7>git checkout develop
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.
C:\Users\efrai\Desktop\Repos\AYD1_Practica1_7>git merge --no-ff feature/tablaEstudiantes
Merge made by the 'recursive' strategy.
 appweb/src/views/practica1/Dashboard.js | 394 +++++
1 file changed, 394 insertions(+)
C:\Users\efrai\Desktop\Repos\AYD1_Practica1_7>git push
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 243 bytes | 243.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/erickace/AYD1_Practica1_7
c2bcc45..e842b65 develop -> develop
C:\Users\efrai\Desktop\Repos\AYD1_Practica1_7>

```

Segundo release

1. Después de clonar las ramas remotas feature/backend y feature/fronted
2. Agregacion de datos al frontend
3. Commit
4. Push
5. Cambio de rama a backend
6. Agregacion de datos al backend
7. Commit
8. Push

Simbolo del sistema

```
C:\Users\efrai\Desktop\Nueva\AYD1_Practica1_7>git add .
```

```
C:\Users\efrai\Desktop\Nueva\AYD1_Practica1_7>git commit -m "Agregando funciones para editar, en el frontend 2"
[feature/fronted 6c161fa] Agregando funciones para editar, en el frontend 2
```

```
C:\Users\efrai\Desktop\Nueva\AYD1_Practica1_7>git push
```

```
Enumerating objects: 26, done.
Counting objects: 100% (26/26), done.
Delta compression using up to 8 threads
Compressing objects: 100% (14/14), done.
Writing objects: 100% (14/14), 2.87 KiB | 1.44 MiB/s, done.
Total 14 (delta 8), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (8/8), completed with 4 local objects.
To https://github.com/erickace/AYD1_Practica1_7
   7d60a6f..6c161fa  feature/fronted -> feature/fronted
```

```
C:\Users\efrai\Desktop\Nueva\AYD1_Practica1_7>git checkout feature/backend
```

```
Switched to branch 'feature/backend'
Your branch is up to date with 'origin/feature/backend'.
```

```
C:\Users\efrai\Desktop\Nueva\AYD1_Practica1_7>git add .
```

```
C:\Users\efrai\Desktop\Nueva\AYD1_Practica1_7>git commit -m "Agregando funcion Edit, para el backend"
[feature/backend abd91e0] Agregando funcion Edit, para el backend
 1 file changed, 18 insertions(+)
```

```
C:\Users\efrai\Desktop\Nueva\AYD1_Practica1_7>git push
```

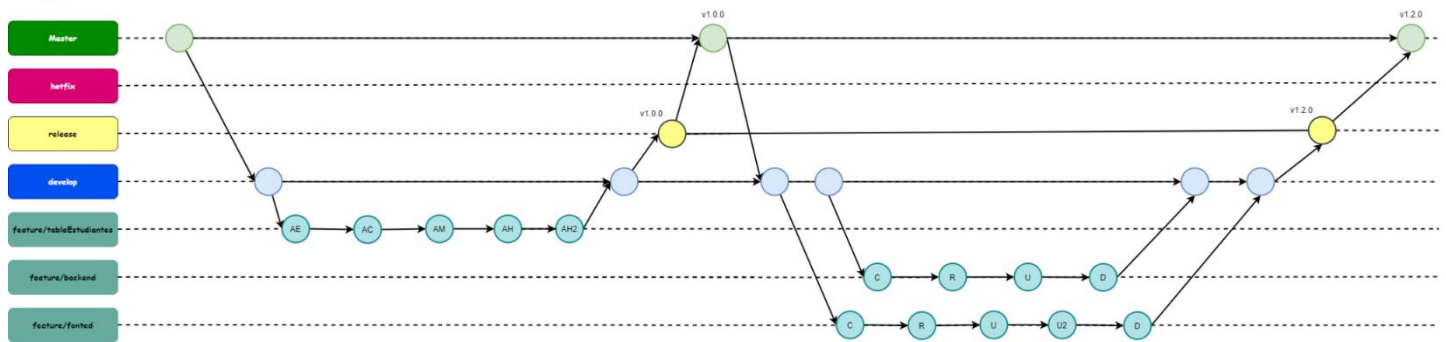
```
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 680 bytes | 680.00 KiB/s, done.
Total 6 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/erickace/AYD1_Practica1_7
   10abf97..abd91e0  feature/backend -> feature/backend
```

```
C:\Users\efrai\Desktop\Nueva\AYD1_Practica1_7>
```

Diagrama de git Flow

AE = Agregar Datos - 201503878
 AC = Agregar Datos - 201404278
 AM = Agregar Datos - 201314821
 AH = Agregar Datos - 201712350

AH2 = Corrección Datos - 201712350



C = Funcionalidad CREATE - 201503878
 R = Funcionalidad READ - 201314821
 U = Funcionalidad UPDATE - 201712350
 U2= Actualización de Funcionalidad UPDATE - 201712350
 D= Funcionalidad DELETE - 201404278

Anexos

Ramas de corrección (hotfix)

Las ramas de mantenimiento o "corrección" (hotfix) se utilizan para reparar rápidamente las publicaciones de producción, las ramas de corrección son muy similares a las ramas de publicación y a las de función, salvo porque se basan en la maestra en vez de la de desarrollo.

Es la única rama que debería bifurcarse directamente a partir de la maestra.

Una vez que la solución esté completa, debería fusionarse en la maestra y la de desarrollo (o la rama de publicación actual), y la maestra debería etiquetarse con un número de versión actualizado.

Tener una línea de desarrollo específica para la solución de errores permite que tu equipo aborde las incidencias sin interrumpir el resto del flujo de trabajo ni esperar al siguiente ciclo de publicación, puedes considerar las ramas de mantenimiento como ramas de publicación ad hoc que trabajan directamente con la maestra.

Una rama de corrección puede crearse utilizando los siguientes métodos:

Usando Git:

- `git checkout master`
- `git checkout -b hotfix_branch`

Usando Git Flow:

- `git flow hotfix start hotfix_branch`

Al igual que al finalizar una rama de publicación, una rama de corrección se fusiona tanto en la maestra como en la de desarrollo:

- `git checkout master`
- `git merge hotfix_branch`
- `git checkout develop`
- `git merge hotfix_branch`
- `git branch -D hotfix_branch`
- `git flow hotfix finish hotfix_branch`

Ejemplo

A continuación, se incluye un ejemplo completo que demuestra un flujo de ramas de función. Vamos a asumir que tenemos una configuración del repositorio con una rama maestra:

- `git checkout master`
- `git checkout -b develop`
- `git checkout -b feature_branch`
- `git checkout develop`
- `git merge feature_branch`
- `git checkout master`
- `git merge develop`
- `git branch -d feature_branch`

Además del flujo de función y publicación, un ejemplo de corrección sería de la siguiente manera:

- `git checkout master`
- `git checkout -b hotfix_branch`
- `git checkout develop`
- `git merge hotfix_branch`
- `git checkout master`
- `git merge hotfix_branch`