/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*////////\*

**INICIAMOS EL PROYECTO**

**Angular 12 ultimas versiones de node, con scss y material**

\*Crearemos nuevo repositorio en GIT HUB con todo.

\*carpeta del proyectoName

\*git clone url de git

\*Parado en carpeta del proyectoName con cd espacio, o abro aca mismo Power shell, o entrar en la consola de Git Bash y  ls

Creo rama, etc…

\*en VS arrastro carpeta proyectoName y abro terminal.

\*ng new projectAngular

Ver versions ng version, npm –v y node –v.

npm install -g @angular/cli

(no se opcion: Dentro de la carpeta en vs > npm init)

\*dentro de esta carpeta projectAngular

\*npm version patch. Agrego cambios en package.json.

\*

**Backup del proyecto y reléase con versión**

GitHub sistema de control de versiones, funciona como un hosting donde se guarda tu proyecto,

Crear new repository >

Como subir tu proyecto a GitHub

1\*

Crearemos nuevo repositorio en GIT HUB:

En GitHub> +> tus repo > new> publico > nombre> publico >add readme> crear.

Después del Readme (ej: *Backend de tienda onLine, desarrollado con GraphQL y Node Typescript.*

*Este proyecto cuenta con licencia conforme a los términos de la licencia MIT.),* podría agregar **topics** referentes para facilitar búsquedas. En **repo** > about >**topics** > agrego > Save **changed**

También clone repo a la carpeta del Proyecto que luego guardo en carpeta que quiero, pero usando el mismo VS.

Me da una dirección http donde almacenar el código. Sino voy agregando los archivos que tenga con su commit correspondiente.

Copiamos URL.

Entonces en repo > code> clone > copiar. Ahora abro VS, close folder si tengo algo abierto > open folder > busco la carpeta que quiero > seleccione > abro terminal > git clone lo copiado [*https://github.com/aAndres441/onLine\_Shop\_backend.git*](https://github.com/aAndres441/onLine_Shop_backend.git)*.*

Ahora desde terminal entrar a la carpeta del proyecto que queremos subir.

> .ls, a través del comando cd ruta, > cd nombreCarpeta o

Parado en carpeta del proyecto con cd espacio, abro aca mismo Power shell, o

O entrar en la consola de Git Bash y  ls

2\*

git init (repo\_local) (se crea una carpeta git como repo local)

Entro a la carpeta y ng new nombreProyecto

git status (si ya teníamos algo, sigo hacia abajo, si no creamos paquetes, etc y luego seguimos)

git add . (espacio entre add y el punto)

git status

git commit -m ‘entre comillas el comentario’

git remote add origin URL (URL copiada anteriormente)

git push –u origin master

actualizar github

o mejor luego del paso status

$ git remote add origin (URL copiada ej:https://github.com/aAndres441/onLine\_Shop.git)

git pull origin master (“quiero una actualización desde el repo remoto de rama master”)

ya en los sucesivos commit solo git add . > git commit –m “xx” > git push origin master

Crearemos definimos package.json, versiones dependencias y mas propiedades del proyecto:

3\*

Dentro de la carpeta en vs > **npm init**

Aparece unas palabras … save it as a dependency in the **package.json** file.Press ^C at any time to quit.

package name: (online\_shop\_backend) y voy completando y enter, si me equivoco podría cambiar luego de creado.

package name: (online\_shop\_backend) Nuevo nombre backend-tienda

version: (1.0.0) nuevo0.0.1

description: Proyecto backend API GraphQL de tiendaOnLine MEAN+G

Versiones minimas para ejecutar este proyecto entonces.

entry point: (index.js) build/server.js //aca es mi punto de entrada, luego lo creo

test command: no pongo nada solo enter.

git repository: (<https://github.com/aAndres441/onLine_Shop_backend.git>) enter.

keywords: *graphql,meang,api-graphql,stripe,cloudinary,netlify,zeitnow*

author: Andres Arias [andrarias@gmail.com](mailto:andrarias@gmail.com) // asi escribo mi mail <[and..@gmail.com](mailto:and..@gmail.com)> y asi si tengo pagina web (nombremipag –web.com)

license: (ISC) MIT

enter

Veremos que se creó el archivo package.json, también podría hacer cambios al abrirlo.

Verificamos versiones con npm –v y node –v.

NPM es un administrador de paquetes para paquetes Node.js, o módulos si lo desea.

El programa NPM se instala en su computadora cuando instala Node.js

Node.js® es un entorno de ejecución  en tiempo real incluye todo lo que se necesita para ejecutar un programa escrito en JavaScript [V8 de Chrome](https://v8.dev/).

Debemos tener última versión de Node lts, desde la 14 o sino debería instalar node,js

npm install -g npm o npm update –g o npm install npm@latest -g

Instalar cli.

Version de AngularCli

ng version || ng -v // instalar global

npm install -g @angular/cli@latest

y luego debo hacerlo **local**

npm install **--save**-dev @angular/cli@latest

Entonces debemos agregar en package.json al final , puede ser después de “homepage:”:

"engines": {

    "node": ">=14.0.0",

    "npm": ">=6.0.0"  }

Crearemos configuramos typescript con tsConfig.json asi :

Como estamos desarrollando con node y ts , deberíamos tener instalado dentro de las dependencias del proyecto o sea local, como recomendado pues usa versiones mas actuales, pero puede ser general.

Entonces npm install typescript (sin –g) (-D es para desarrollo)

npm i --save -D@angular/cli

Debe aparecer en package.json esta propiedad

   "dependencies":{

    "typescript":"^4.2.4"

  }

Vemos en terminal a comprobar con cual versión estoy (por ser node va npx),

version npx tsc - -version , debe ser la misma que instale o sea 4.2.4.

Las dependecias de producción que se usaran tanto en producción cuando subimos nuestro proyecto a la nube como en desarrollo. Por otro lado, existen dep solo de desarrollo en nuestra maquina.

4\* Ahora en la carpeta y después de verificar node y Angular cli ya puedo inicializar proyecto desde consola

ng new projectName

Git **VERSIONADO** semántico del proyecto con Git

Guardo todo:

git add –all git .d >> git commit –m”comentario”

git branch //veo ramas

git checkout –b rama-nueva //crea y -b para cambiar a esa rama nueva

git push origin rama- nueva //push todo a esa rama nueva

git push origin –delete ramaCualquiera//eliminar ramaCualquiera

git checkout master o main //ir a la master

git merge rama- nueva // trae todo desde la rama-nueva a la master

git push origin master // guarda todo en master

Lo observo en package .json del VS

"name": "on-line-shop-meang",

  "version": "0.0.0",

En Terminal VS

Tags 0.0.0 llamado

**npm version major** - 1.0.0

npm version **minor** - 1.1.0

npm version **patch** - 1.1.1

En **code** > **releases** > **Tag** > agrego la versión con info

*“Proyecto Base de Angular 12 con ultimas versiones de node.*

*Inicializamos con scss.*

*Agregamos angular material.*”> créate a new reléase

O veo en consola bash con comando **git tag** >enter

Actualizo con npm version minor > enter. En terminal VS

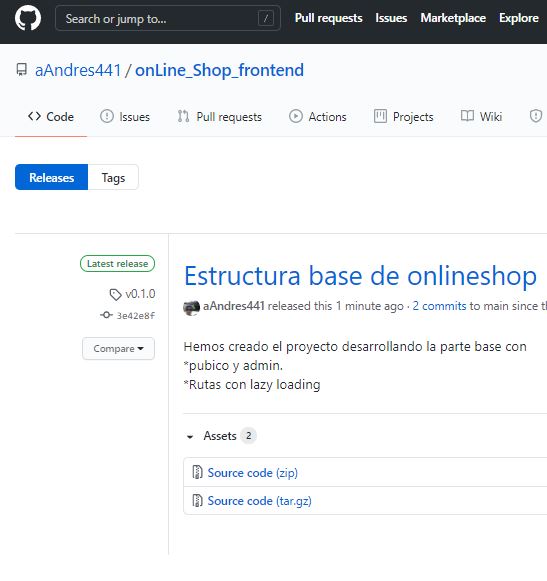
Aparece v0.1.0 que será el tag creado

De nuevo git tag >enter , observo el cambio.

**git push – - tags && git push origin master**

Después voy al GitHub y veo el tag > click el tag y edit tag > añado titulo, info > **Publicar**

Obtendremos fichero zip por si lo necesitamos en algún momento, descargando e instalando dependencias para usar.



**CREAR RAMAS** en Git: para desarrollar software en equipo.

Git branch muestra las versiones

Git branch nuevoNombre crea versión nueva.

Git checkout nuevoNombre

git checkout -b , lleva dos parametros: -b y el nombre de la rama sin espacios. Esto crea una nueva rama y se cambia a esa nueva rama.

Eliminar rama: git push origin --delete <NombreRama> o $ git branch -D [branch-name]

Cambiar de puerto: ng serve -o --port 4300

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*# Angular12

Proyecto base creado con la ultima version de Angular 12.

Proyecto base que sera creciendo como mi hija en el 4 de Julio.

Inicializado con scss.

Agregamos Angular material.

Este proyecto cuenta con licencia conforme a los términos de la licencia MIT. Angular 12 ultimas versiones de node, con scss, Angular\_material

Proyecto base creado con la ultima version de Angular 12

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Angular Material

<https://material.angular.io/>

Ver material en: <https://material.angular.io/guide/getting-started>

Buen Guia Material y Bootstrap: https://lab.morioh.com/morioh/preview/light/dashboard-01.html

ng add @angular/material

//-- flat no crea carpeta

//--skipTests

//--routing crea rutas para lazyLoading

ng g m public/pages --routing --flat

ng g c public/pages --module pages.module --skip-tests

ng g m public/pages/clientes –routing

Mostrar un componente, ej MatSliderModule

Debe importar el que desea mostrar agregando las siguientes líneas a su archivo app.module.ts.

**import { MatSliderModule } from '@angular/material/slider';**

**…**

**@NgModule ({....**

**imports: [...,**

**MatSliderModule,**

**…]**

**})**

Ej para usar input type =file **npm install --save @angular-material-components/file-input**

import { NgxMatFileInputModule } from '@angular-material-components/file-input';

@NgModule({

  declarations: [ ],

  imports: [

    CommonModule,

    NgxMatFileInputModule

  ]})

<mat-form-field>

  <mat-label>Choose imagen</mat-label>

  <ngx-mat-file-input   accept=".png, .jpg, .jpeg" (click)='capturarImagen($event)'name='capturarImagen'>

  </ngx-mat-file-input>

</mat-form-field>

**Sombra, Shadow**: Defina tamaño, desenfoque, radio de propagación, color:

#example1 { box-shadow: 10px 10px 8px 10px #888888;}

Cambia la sombra de exterior (inicio) a interior

#example1 {box-shadow: 20px 20px 20px 10px blue inset;}

Defina múltiples sombras:

#example1 {box-shadow: 5px 5px blue, 10px 10px red, 15px 15px green;}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

K2BTools tutorial Para genexus 17 upgrade >3 , url: <http://wiki.k2btools.com/k2btools/servlet/wiki?20616,Toc%3AK2BToolsTutorial>

escribinos a [info@k2btools.com](mailto:info@k2btools.com)

Pueden ver las Release notes de la versión aca: [wiki.k2btools.com/k2btools/servlet/wiki?22013,K2BTools+13.1+-+Release+notes](http://wiki.k2btools.com/k2btools/servlet/wiki?22013,K2BTools+13.1+-+Release+notes),

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//////\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

API de prueba

{JSON} Placeholder es una API REST en línea gratuita que puede usar siempre que necesite datos falsos. Puede estar en un README en GitHub, para una demostración en CodeSandbox, en ejemplos de código en Stack Overflow, ... o simplemente para probar cosas localmente.: <https://jsonplaceholder.typicode.com/>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Hash **HA-256 y SHA-1**

tabla hash de tamaño m=11m=11

La función hash tomará cualquier ítem de la colección y devolverá un número entero en el rango de nombres de las ranuras, entre 0 y m-1

Nuestra primera función hash, a veces denominada “método del residuo”, simplemente toma un ítem y lo divide por el tamaño de la tabla, devolviendo el residuo (resto) como su valor hash (h(item)=item%11h(item)=item%11).

[**https://runestone.academy/runestone/static/pythoned/SortSearch/TransformacionDeClaves.html**](https://runestone.academy/runestone/static/pythoned/SortSearch/TransformacionDeClaves.html)

**Generador de hash online** [**https://hash.online-convert.com/es**](https://hash.online-convert.com/es)

**Con HA-256**

Paulina:

hex: 006e3a93b8f63344095bc538966c67539282df6d4f566ccff5d983acb0c18be8

HEX: 006E3A93B8F63344095BC538966C67539282DF6D4F566CCFF5D983ACB0C18BE8

h:e:x: 00:6e:3a:93:b8:f6:33:44:09:5b:c5:38:96:6c:67:53:92:82:df:6d:4f:56:6c:cf:f5:d9:83:ac:b0:c1:8b:e8

base64: AG46k7j2M0QJW8U4lmxnU5KC321PVmzP9dmDrLDBi+g=

Andres:

hex: 84d4dce0d11bc319bc61143cd0849a1464cd7dcef4c73393d155fb27879df26b

HEX: 84D4DCE0D11BC319BC61143CD0849A1464CD7DCEF4C73393D155FB27879DF26B

h:e:x: 84:d4:dc:e0:d1:1b:c3:19:bc:61:14:3c:d0:84:9a:14:64:cd:7d:ce:f4:c7:33:93:d1:55:fb:27:87:9d:f2:6b

base64: hNTc4NEbwxm8YRQ80ISaFGTNfc70xzOT0VX7J4ed8ms=

Imagen reflejo en el mar; en D:\Imagenes\Pauli\Dibujos

hex: f960a6a1b388a686cad7e0de014d9c9b4744c6204813874671b98d61ec7a0023

HEX: F960A6A1B388A686CAD7E0DE014D9C9B4744C6204813874671B98D61EC7A0023

h:e:x: f9:60:a6:a1:b3:88:a6:86:ca:d7:e0:de:01:4d:9c:9b:47:44:c6:20:48:13:87:46:71:b9:8d:61:ec:7a:00:23

base64: +WCmobOIpobK1+DeAU2cm0dExiBIE4dGcbmNYex6ACM=

**Con SHA-1**

**Andres**

hex: 9a0a874d87c1273bcf4d18c9ae2c245f6072efbd

HEX: 9A0A874D87C1273BCF4D18C9AE2C245F6072EFBD

h:e:x: 9a:0a:87:4d:87:c1:27:3b:cf:4d:18:c9:ae:2c:24:5f:60:72:ef:bd

base64: mgqHTYfBJzvPTRjJriwkX2By770=

**Paulina**

hex: a9aedaa3ae066cebeb5475fb4a100bc534e38402

HEX: A9AEDAA3AE066CEBEB5475FB4A100BC534E38402

h:e:x: a9:ae:da:a3:ae:06:6c:eb:eb:54:75:fb:4a:10:0b:c5:34:e3:84:02

base64: qa7ao64GbOvrVHX7ShALxTTjhAI=

Imagen reflejo en el mar; en D:\Imagenes\Pauli\Dibujos

hex: dbb6d17b67f968a4a26fb0f8b27df02692e9b510

HEX: DBB6D17B67F968A4A26FB0F8B27DF02692E9B510

h:e:x: db:b6:d1:7b:67:f9:68:a4:a2:6f:b0:f8:b2:7d:f0:26:92:e9:b5:10

base64: 27bRe2f5aKSib7D4sn3wJpLptRA=

background: #557E29;