



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE SOFTWARE
ASIGNATURA: CLOUD COMPUTING - NIVEL: 06

Nombres completos:	Luis Eduardo Benavides Puetate
Fecha:	2 de Diciembre de 2021
Tema:	Examen Bimestre I
Objetivo de esta actividad:	Crear una aplicación web que contenga una página donde se pueda ingresar una dirección URL de una imagen y luego consumir API.

INDICACIONES:

Crear una aplicación web que contenga una página donde se pueda ingresar una dirección URL de una imagen (ej. un paisaje), luego consumir API de computer vision de Microsoft Azure para describir el contenido de la imagen. El resultado debe ser un párrafo de texto que describa el contenido de la imagen.

La aplicación web debe publicarse en Azure y se debe enviar por este medio la URL de la aplicación web, adicional le documento PDF del informe, no olvidar incluir en el PDF el código fuente de toda la solución

DESARROLLO:

Al iniciar la aplicación:

Iniciamos el Proyecto con Git Bash

```
npm install
npm i
npm run dev
```

```
MINGW64 C:\xampp\htdocs\node\Benavides-api-computer-vision
Found 0 vulnerabilities

$ npm i
npm WARN EBADENGINE Unsupported engine {
  npm WARN EBADENGINE   package: 'next@12.0.4',
  npm WARN EBADENGINE   required: { node: '>=12.22.0' },
  npm WARN EBADENGINE   current: { node: 'v12.18.4', npm: '7.20.3' }
}
npm WARN EBADENGINE
up to date, audited 420 packages in 2s
68 packages are looking for funding
run npm fund for details
Found 0 vulnerabilities

$ npm run dev
npm WARN EBADENGINE Unsupported engine {
  npm WARN EBADENGINE   package: 'next@12.0.4',
  npm WARN EBADENGINE   required: { node: '>=12.22.0' },
  npm WARN EBADENGINE   current: { node: 'v12.18.4', npm: '7.20.3' }
}
npm WARN EBADENGINE
ready - started server on 0.0.0.0:3000, url: http://localhost:3000
info - Loaded env from C:\xampp\htdocs\node\Benavides-api-computer-vision\.env
local

warn - You have enabled the JIT engine which is currently in preview.
warn - Preview features are not covered by server, may introduce breaking change
      in, and can change at any time.
event - compiled successfully in 9.3s (172 modules)
attention - Next.js now collects completely anonymous telemetry regarding usage.
            This information is used to shape Next.js' roadmap and prioritize features.
            You can learn more, including how to opt-out if you'd not like to participate in
            this anonymous program, by visiting the following link:
            https://nextjs.org/telemetry

wait - compiling /...
event - compiled successfully in 1474 ms (243 modules)
key = 232f1df3b56643ca41dc8287785f80
endpoint = https://modelodeanalizadoimagen.cognitiveservices.azure.com/
key = 232f1df3b56643ca41dc8287785f80
endpoint = https://modelodeanalizadoimagen.cognitiveservices.azure.com/
ComputerVision.jsConfigured = true
warning: Invalid DOM property `class`. Did you mean `className`?
  at input
  at div
  at div
  at div
  at Analyze
  at App (webpack-internal:///./src/pages/index.js:15:102)
  at MyApp (webpack-internal:///./src/pages/_app.js:11:18)
  at InnerApp
  at StyleRegistry (C:\xampp\htdocs\node\Benavides-api-computer-vision\node_modules\styled-jsx\dist\stylesheet-registry.js:231:34)
  at AppContainer (C:\xampp\htdocs\node\Benavides-api-computer-vision\node_modules\next\dist\server\render.js:139:29)
  at AppContainerWithMorphCfIberStructure (C:\xampp\htdocs\node\Benavides-api-computer-vision\node_modules\next\dist\server\render.js:369:57)
```

Api

computerVision.js

images.js

```
import { computerVision } from "../../helpers/computerVision";

export default async function tasks(res, req) {
  const { method, body } = req;
  switch (method) {
    case "GET":
      try {
        const query = "SELECT * FROM tasks";
        const response = await conn.query(query);
        return res.status(200).json(response.rows);
      } catch (error) {
        return res.status(400).json({ error: error.message });
      }
    case "POST":
      try {
        const { url } = body;
        const response = await computerVision(url);
        return res.status(200).json(response);
      } catch (error) {
        return res.status(400).json({ error: error.message });
      }
    default:
      return res.status(400).json("invalid method");
  }
}
```



Imagen a Analizar

URL Ingrese la URL de una Imagen

Analizar



Al momento de Insertar una URL:

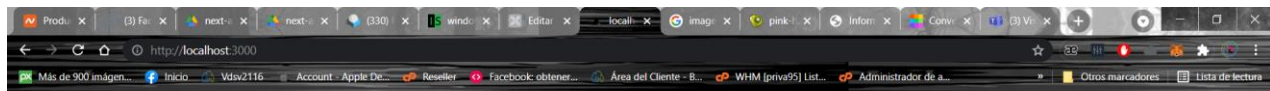


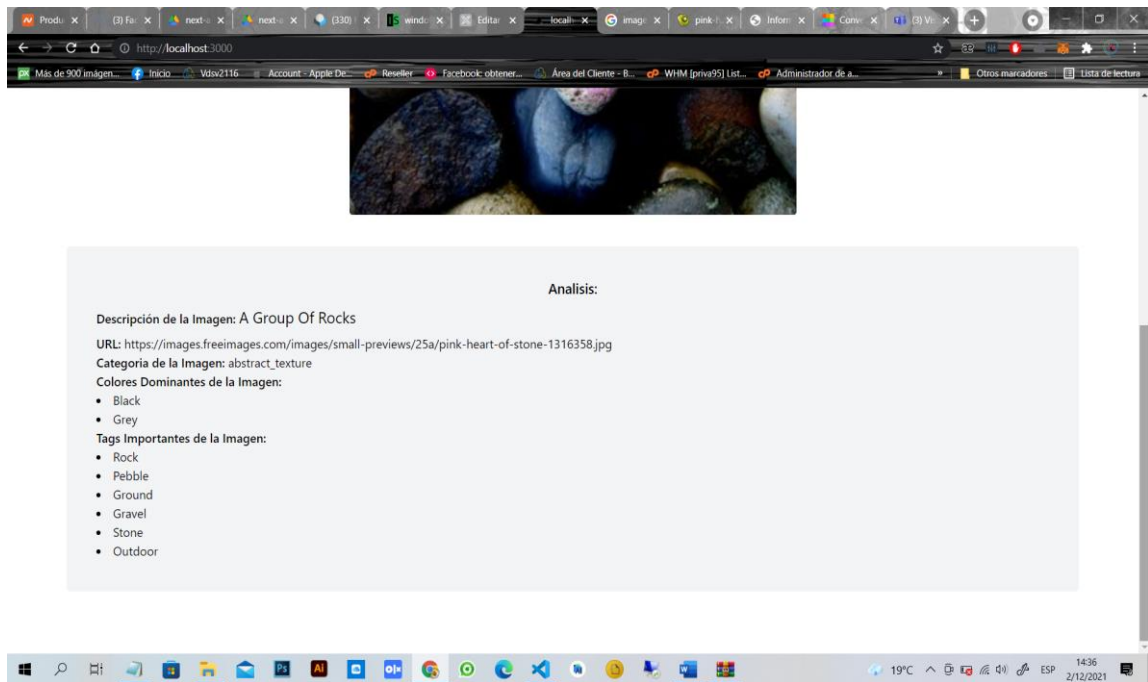
Imagen a Analizar

URL <https://images.freeimages.com/images/small-previews/25a/pink-heart-of-stone-1316358.jpg>

Analizar

Imagen analizada





Con esto se obtienen todos los resultados de una imagen.