

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

INTRODUCCIÓN A LA PROGRAMACIÓN Y COMPUTACIÓN 1

CATEDRÁTICO: ING. WILLIAM ESTUARDO ESCOBAR ARGUETA

TUTOR ACADÉMICO: AYESER CRISTIÁN OXLAJ JUÁREZ



MANUAL TÉCNICO **PROYECTO 2**

Álvaro Gabriel Ceballos Gil

CARNÉ: 202300673

SECCIÓN: F

GUATEMALA, 27 DE ABRIL DEL 2,024

ÍNDICE

ÍNDICE	1
INTRODUCCIÓN	1
OBJETIVOS	3
1. GENERAL	3
2. ESPECÍFICOS	3
ALCANCES DEL SISTEMA	4
ESPECIFICACIÓN TÉCNICA	5
• REQUISITOS DE HARDWARE	5
• REQUISITOS DE SOFTWARE	5
TABLA DE ENDPOINTS	6

INTRODUCCIÓN

Este manual se encontrará de forma más desarrollada la explicación del código del proyecto, con el fin de que un programador pueda entender el código del programa, y así poder implementarlo si así lo desea.

Este manual le permite al programador ver la lógica del programa de una forma mucho más comprensible, lo que permitirá un mejor desarrollo de la aplicación

en el caso de que el programador desee implementar alguna función de este programa en el suyo.

OBJETIVOS

1. GENERAL

- 1.1. Ayudar al programador a poder entender el programa de una mejor manera

2. ESPECÍFICOS

- 2.1. Explicar la lógica de las funcionalidades del programa de una forma más sencilla
- 2.2. Brindar información necesaria para la comprensión del proyecto de una forma más técnica

ALCANCES DEL SISTEMA

Este manual tiene el objetivo de explicar de una forma mucho más explícita las funcionalidades del código del proyecto 2, con el fin de que el programador sea capaz de entender correctamente las líneas de código utilizadas para el desarrollo del programa.

También se tiene como objetivo que el programador logre entender mejor la lógica del programa, y que de esta forma el programador sea capaz de implementar varias funcionalidades de este programa en su código.

ESPECIFICACIÓN TÉCNICA

- **REQUISITOS DE HARDWARE**

Para poder correr este programa, es necesario que el usuario tenga instalado Visual Studio Code, junto con node js, y todas sus dependencias como express.

También debe de instalar el framework de react con todas sus dependencias para poder correr el programa

Se debe de contar con un sistema operativo Windows de 64 bits, con una memoria RAM recomendada de 8GB, ya que con esto se garantiza de que el programa pueda correr sin ningún tipo de dificultad.

- **REQUISITOS DE SOFTWARE**

- Debe contar con un sistema operativo medianamente moderno, como Windows 10, esto con el fin de que a la hora de desarrollar el proyecto no tenga algún tipo de error.

TABLA DE ENDPOINTS

Método	Dirección y tipo de método	Body	Respuesta
Registrar Usuarios por JSON: Para poder registrar los datos, se manda un archivo JSON el cual contiene varios usuarios, con sus respectivos	/Publicacion /CargaMasiva a /POST	<pre>{ "codigo": "3", "nombres": "Michael", "apellidos": "Jackson", "genero": "asas", "facultad": "comunicacion", "carrera": "derecho", "correo": "@nminoso", "contrasenia": "1", "campoextra": "1234" },</pre>	<pre>{ "codigo": "3", "nombres": "Michael", "apellidos": "Jackson", "genero": "asas", "facultad": "comunicacion", "carrera": "derecho", "correo": "@nminoso", "contrasenia": "1", "campoextra": "1234" },</pre>

poder ser mostra dos en la tabla			
Inicio de Sesión: Para el inicio de sesión se manda el JSON completo con todos los campos del usuario , y se valida que si el carnet	/accesos/Log in	<pre> { "apellidos": "perez", "carrera": "veterinario", "codigo": "2", "correo": "@nmeidcinoso", "facultad": "medicina", "genero": "asasas", "nombres": "Lusi miguel" } </pre>	<pre> { "codigo": "2", "nombres": "Lusi miguel", "genero": "asasas", "facultad": "medicina", "carrera": "veterinario", "correo": "@nmeidcinoso", "apellidos": "perez" } </pre>

<p>ingresa do y la contras eña ingresa da coincid en con la de algun usuario , este permiti rá el acceso</p>			
<p>Creación de Publicaciones: Para crear publicaciones se creó un arreglo donde</p>	<p>/Publicacion /createPost /POST</p>	<pre>▼ {mensaje: 'Todod bien con mensaje: "Todod bien con</pre>	<pre>▼ {mensaje: 'Todod bien con La pub mensaje: "Todod bien con la pu ▼ [[Prototype]]: Object ► constructor: f Object() ► hasOwnProperty: f hasOwnProp ► isPrototypeOf: f isPrototype ► propertyIsEnumerable: f prop ► toLocaleString: f toLocaleSt ► toString: f toString() ► valueOf: f valueOf() ► __defineGetter__: f __define ► __defineSetter__: f __define ► __lookupGetter__: f __lookup ► __lookupSetter__: f __lookup __proto__: (...) ► get __proto__: f __proto__() ► set __proto__: f __proto__()</pre>

se almace naban los campos , y tambié n cookies para mostrar los datos. En la publica cion se puede almace nar texto e imágen es (base 64)			
Elimin ación de usuari os	/delete /DELETE		<pre> ▼ [{...}] ⓘ ► 0: {codigo: '3', nombres: 'Mich length: 1 ► [[Prototype]]: Array(0) </pre>

Para la elimina cion de usuario s se busca al usuario por medio de su codigo único y se elimina .			
--	--	--	--

