

2º Curso DESARROLLO DE APLICACIONES WEB
DESARROLLO WEB EN ENTORNO SERVIDOR
SOLUCIÓN Prueba 3 y 4

Nombre:
Apellidos:

Tiempo estimado: 160 minutos

5 Diciembre 2023

Normas:

- Pon tu nombre y apellidos en la parte superior de este documento.
- Para cada ejercicio realiza las capturas de pantalla necesarias donde se muestre que lo has resuelto de forma satisfactoria.
- Copia las capturas dentro de este documento.
- Al finalizar:
 - **Subir a la plataforma Moodle este documento relleno en formato PDF con la resolución de esta prueba.**
 - **Sube a Github el código e indica la URL a continuación**

URL de repositorio de Github → _____

Genera en tu PC o portátil la siguiente estructura de carpetas:

examen-dwes-3-4

```
|— ej1  
|— ej2  
|— ej3  
|— ej4  
|__ .git
```

En cada carpeta trabajarás el código correspondiente a cada ejercicio.

AYUDA para GIT

No haremos control de versiones para cada ejercicio, sino que lo hacemos de una vez para todo.

Para inicializar y guardar cambios:

```
jose@slimbook ~/Proyectos/examen-dwes-3-4> ls  
ej1 ej2 ej3 ej4  
jose@slimbook ~/Proyectos/examen-dwes-3-4> git init  
Initialized empty Git repository in /home/jose/Proyectos/examen-dwes-3-4/.git/  
jose@slimbook ~/Proyectos/examen-dwes-3-4> git add .  
jose@slimbook ~/Proyectos/examen-dwes-3-4> git commit -m "Solución a Prueba DWES - Tems 3, 4"
```

El resto de pasos puedes obtenerlos desde Github al crear un repositorio nuevo.

1 TEMA 3. Ejercicio 1 (NODE+EXPRESS)

Trajabaremos con la API REST de Rick and Morty.

- Documentación: <https://rickandmortyapi.com/documentation/#rest>

Crea una aplicación que muestre un pequeño formulario con un input de tipo number o de tipo range y un botón de submit.

El rango de números admitido está entre 1 y 826, que es total de personajes de la serie.

← → ↻ localhost:3000

Personajes de Rick and Morty

Personaje

Consultar

El formulario enviará mediante POST dicho número al servidor, el cual obtendrá de la API REST la información del personaje solicitado y la mostrará por pantalla, tal como se ve en la captura.

← → ↻ localhost:3000

Personajes de Rick and Morty

Personaje 414

Consultar

Datos del personaje 414

Nombre: **Trunkphobic guy**

Especie: **Human**

Imágen:



2 TEMA 3. Ejercicio 2 (NODE+EXPRESS)

Crea una API REST que gestione información acerca de alumnos, con las propiedades que se muestran en el siguiente ejemplo:

```
{
  id: 1,
  nombre: "Eva",
  edad: 22,
  domicilio: {
    calle: "Antigua",
    num: 3,
    localidad: 'Lucena'
  }
}
```

IMPORTANTE: Las propiedades id, edad y num se guardarán en formato Number, aunque se envíen a la API en formato string. En este caso deberás realizar la conversión necesaria.

Al iniciar la aplicación, estarán dados de alta 4 alumnos.

Los endpoints deben permitir:

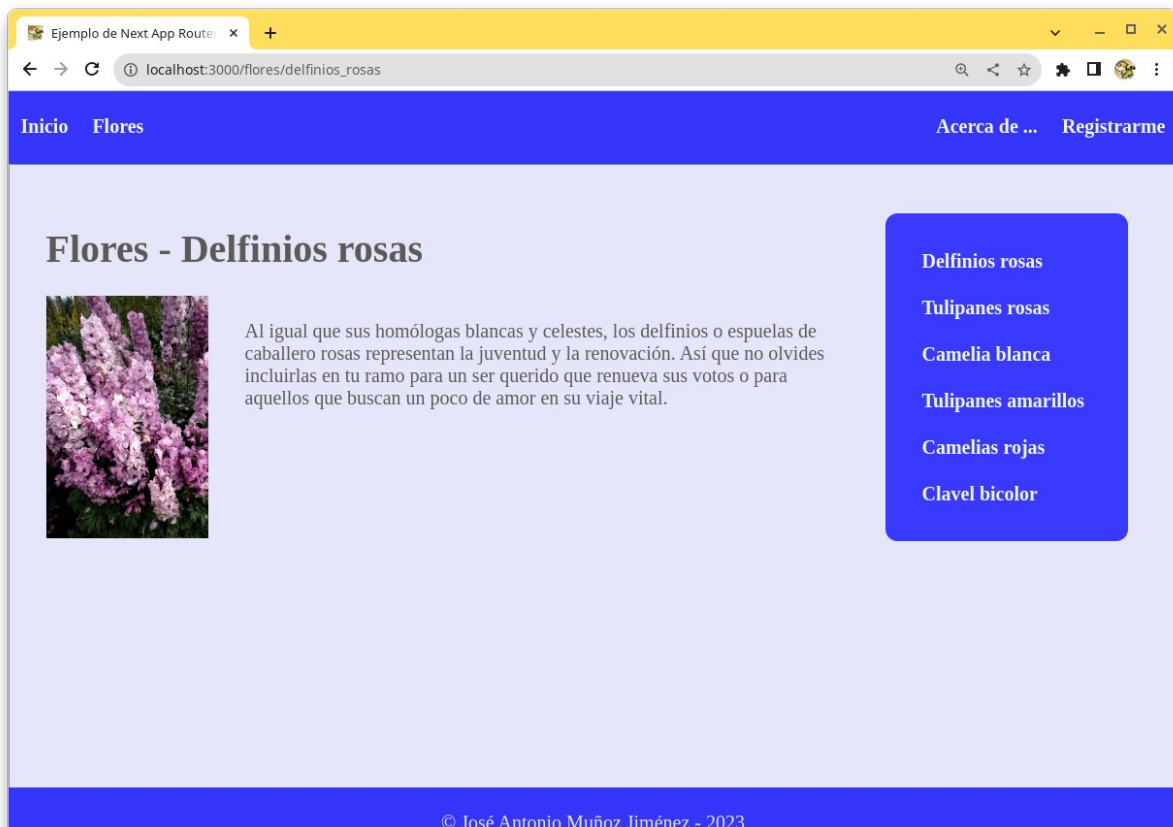
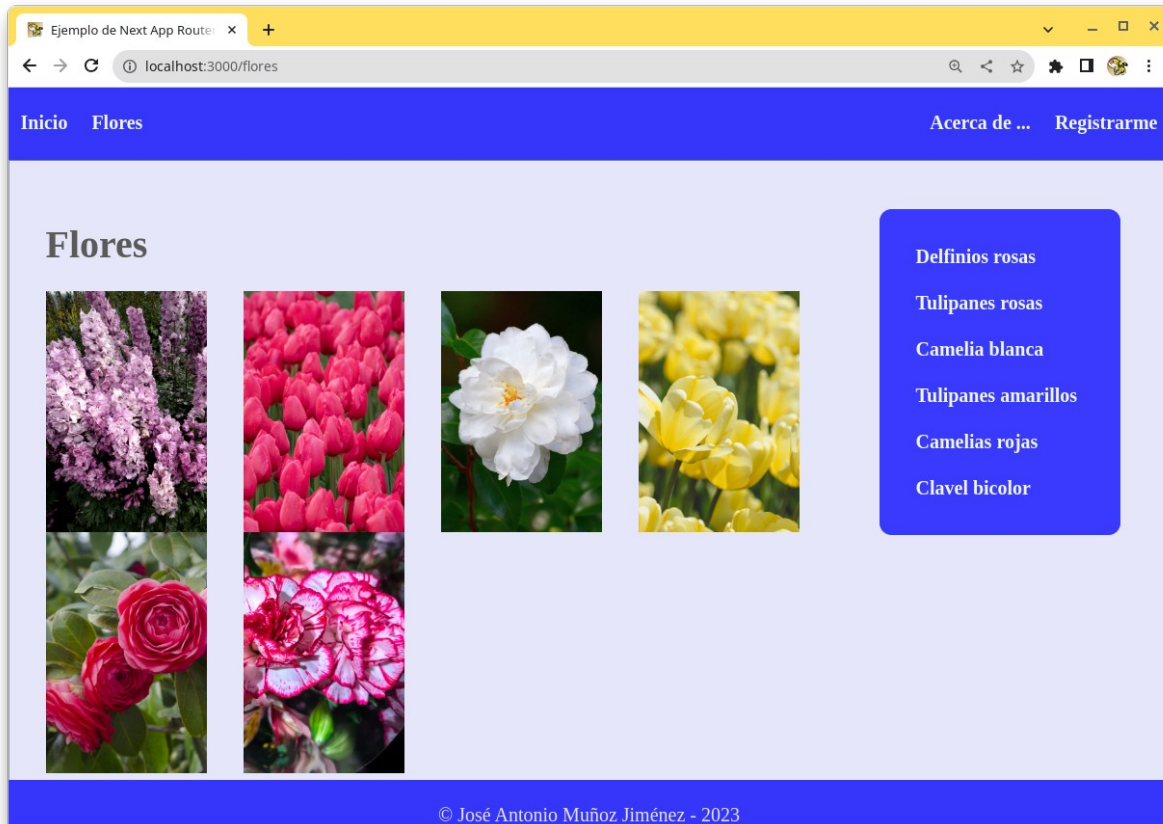
1. Listar todos los alumnos
2. Listar un alumno por id
3. Insertar un alumno (el id se generará de forma automática)
4. Modificar un alumno por id
5. Eliminar un alumno por id

Instala el plugin Rest Client de VSCode, si no lo tienes.

Crea un archivo llamado **api.rest** con las 5 peticiones anteriores a la API. Haz una captura de cada una de ellas.

3 TEMA 4. Ejercicio 3 (NEXTJS, APP ROUTER)

Usando App Router, elabora la siguiente aplicación.



El texto e imágenes puede obtenerse desde

<https://www.micasarevista.com/plantas-flores/g42345957/significado-flores-plantas/>

Por ejemplo, la primera imagen tiene la URL siguiente:

https://hips.hearstapps.com/hmg-prod/images/flower-meanings-delphiniums-pink-1671556178.jpg?crop=0.502xw:1.00xh;0.151xw,0&resize=980:*

Requisitos:

- En todas las páginas aparecerá la barra de navegación y el footer.
- En las páginas de Flores y subpáginas aparecerá el menú a la derecha.
- El menú anterior no aparecerá en las páginas Inicio, Acerca de... y Registrarme.
- Se valorará que el resultado final se aproxime lo máximo posible a las capturas mostradas.

IMPORTANTE: Deberás modificar el siguiente archivo para que quede así:

Puesto que las imágenes se descargaran desde [hearstapps.com](https://hips.hearstapps.com/) debemos configurar NextJS para que lo tenga en cuenta.

Archivo next.config.js

```
/** @type {import('next').NextConfig} */
const nextConfig = {
  images: {
    remotePatterns: [
      {
        protocol: 'https',
        hostname: '**.hearstapps.com',
      },
    ],
  },
}
module.exports = nextConfig
```

4 TEMA 4. Ejercicio 4 (NEXTJS, API, ROUTE HANDLER)

En este ejercicio se pide lo mismo que en el Ejercicio 2, pero esta vez lo haremos con NextJS.

IMPORTANTE: Se comprobará que los identificadores de constantes, variables, rutas y funciones se refieren a alumnos y no a users, products u otros.

Crea una API REST que gestione información acerca de alumnos, con las propiedades que se muestran en el siguiente ejemplo:

```
{
  id: 1,
  nombre: "Eva",
  edad: 22,
  domicilio: {
    calle: "Antigua",
    num: 3,
    localidad: 'Lucena'
  }
}
```

IMPORTANTE: Las propiedades id, edad y num se guardarán en formato Number, aunque se envíen a la API en formato string. En este caso deberás realizar la conversión necesaria.

Al iniciar la aplicación, estarán dados de alta 4 alumnos.

Los endpoints deben permitir:

1. Listar todos los alumnos
2. Listar un alumno por id
3. Insertar un alumno (el id se generará de forma automática)
4. Modificar un alumno por id
5. Eliminar un alumno por id

Instala el plugin Rest Client de VSCode, si no lo tienes.

Crea un archivo llamado **api.rest** con las 5 peticiones anteriores a la API. Haz una captura de cada una de ellas.