

UNIVERSIDAD AUTÓNOMA DE MADRID

DEPARTAMENTO DE INFORMÁTICA

Proyecto de Sistemas Informáticos

Práctica - 2

Roberto MARABINI
Alejandro BELLOGÍN

Registro de Cambios

Versión ¹	Fecha	Autor	Descripción
1.0	10.10.2022	RM	Primera versión.
1.1	17.11.2022	RM	Pequeñas modificaciones en la descripción de la práctica.
1.2	5.12.2022	RM	Renumerar práctica 3 ->2
1.3	12.12.2022	AB	Traducción al inglés
1.4	26.1.2023	RM	Revisión antes de subir el documento a <i>Moodle</i>

¹La asignación de versiones se realizan mediante 2 números *X.Y*. Cambios en *Y* indican aclaraciones, descripciones más detalladas de algún punto o traducciones. Cambios en *X* indican modificaciones más profundas que o bien varían el material suministrado o el contenido de la práctica.

Índice

1. Objetivo	3
2. Introducción	3
3. Organización de la Práctica	4
4. Asignación de tiempo para las diferentes tareas	4
5. Material a entregar al finalizar la practica	5
6. Criterios de evaluación	6

1. Objetivo

En esta práctica aprenderás a crear una aplicación desde cero usando *Vue.js*. Veremos primero los conceptos fundamentales de *Vue.js* así como la anatomía de las aplicaciones creadas con este framework. Seguidamente aprenderás a crear una aplicación CRUD paso a paso capaz de conectarse a un API REST. Cuando finalice la practica, habrás aprendido a crear **componentes**, **eventos**, **métodos**, **propiedades computadas** y **formularios**, además de saber gestionar el **estado de los componentes**, acceder a **APIs** externos a la aplicación y gestionar el **enrutado**.

2. Introducción

Vue.js es un entorno de código abierto desarrollado en JavaScript que se encarga de proporcionar las herramientas necesarias para la creación de las vistas en una arquitectura MVC.

Para esta práctica necesitarás tener instalado tanto Node como npm. Nota: npm se instala automáticamente al instalar Node.

Listado 1: Cómo instalar npm desde la línea de comandos. La flecha roja indica un salto de línea introducido para mejorar la visibilidad del listado.

```
# install node (and npm)
# node is already installed in the lab computers (linux partition)
curl -fsSL https://deb.nodesource.com/setup_14.x | sudo -E bash -
→ sudo apt-get install -y nodejs
```

En esta práctica se creará una aplicación de gestión con *Vue.js*. La aplicación permitirá gestionar los datos de diferentes personas. Veremos **cómo instalar *Vue.js*** y cómo configurarlo. Seguidamente verás cuál es la **estructura de un archivo *Vue.js***. Luego aprenderás a crear componentes, propiedades, métodos, eventos y diferentes tipos de sentencias de *Vue.js*.

Para crear la interfaz de la aplicación, usaremos tablas y formularios, agregando campos a estos últimos junto con las validaciones pertinentes. Por último, también aprenderás a desplegar una aplicación creada con *Vue.js* en *Render.com* y como

conectarla a un API.

En <https://tutorialvue.onrender.com/> puedes ver el resultado de la aplicación que vas a crear.

3. Organización de la Práctica

La práctica se ha dividido en tres secciones que se describen por separado en tres ficheros pdf adicionales. En la primera se crea la aplicación con toda la funcionalidad requerida excepto la habilidad de persistir los datos. En la segunda parte la aplicación accederá a un API REST creada en *Django* que permitirá la persistencia de los datos. Finalmente en la tercera parte se discute sobre como se puede navegar entre diversos URLs sin necesidad de contactar con el servidor.

4. Asignación de tiempo para las diferentes tareas

Se han asignado tres semanas (unas 9 horas de clase) para la realización de esta práctica. Aproximadamente se espera que la mitad del esfuerzo se dedique a la primera parte y un cuarto a cada una de las dos partes restantes.

5. Material a entregar al finalizar la practica

1. Se debe indicar claramente, y en el momento de la entrega en *Moodle*, las URLs donde están desplegados el proyecto de *Vue.js* y el de *Django* en *Render.com*. Esta información debe aparecer correctamente configurada en el fichero `settings.py` entregado con el proyecto de *Django* (variables `CORS_ORIGIN_WHITELIST` (*Vue.js*) y `ALLOWED_HOSTS` (*Django*)). En *Render.com*, la base utilizada debe estar albergada en <https://neon.tech> y la interfaz de administración de *Django* debe ser accesible mediante el usuario y contraseña (`alumnodb`).
2. En esta practica se crean tres proyectos, dos en *Vue.js* y otro en *Django* llamados `tutorial-vue`, `twopages` y `persona` respectivamente. Para cada uno de estos proyectos subid a *Moodle* el fichero obtenido al ejecutar el comando `zip -r ../xxxxx.zip .git` desde el directorio raíz del proyecto. Recordad que hay que añadir (`git add`) y enviar (`git commit`) los ficheros a *git* antes de ejecutar el comando. Comprobad que el contenido del fichero zip es correcto ejecutando la orden: `cd /tmp; rm -rf tmpDir; unzip xxxxx.zip; git clone . tmpDir; ls tmpDir`. El directorio `tmpDir` contendrá el proyecto.

Dado que se crean tres proyectos, dos en *Vue.js* y otro en *Django*, deberás subir a *Moodle* tres ficheros ZIP llamados `tutorial-vue.zip`, `twopages.zip` y `persona.zip` respectivamente.
3. No subáis, ni añadáis al repository el entorno virtual de python (*Django*) ni el directorio `node_modules` (*Vue.js*) puesto que sólo son válidos en el ordenador en el que se han creado.

6. Criterios de evaluación

La calificación de esta práctica será *Apto* o *No Apto*. Para aprobar, y obtener así una calificación de *Apto*, es necesario satisfacer TODOS los siguientes criterios:

- El código creado para esta practica debe ejecutarse correctamente usando las versiones de los diferentes programas que se describen en la memoria. En particular Python 3.9, *Django* 3.2 y *Vue.js* 3.2
- Todas las funcionalidades solicitadas en la guía de la práctica deben estar satisfechas.
- Los datos de la aplicación deben persistirse en una base de datos PostgreSQL almacenada en <https://neon.tech>.
- La aplicación dedicada a la gestión de personas (proyectos *tutorial-vue* y *persona*) debe estar desplegada en *Render.com*, y debe ser accesible a través de los URL suministrados. En *Render.com* la interfaz de administración (*Django*) debe ser completa, pudiendo visualizar y cambiar los datos a conveniencia mediante el superusuario creado (nombre de usuario: `alumnodb`, clave `alumnodb`).
- No es necesario desplegar la aplicación *twopages* en *Render.com*.
- El código creado en esta práctica debe estar almacenado en un repositorio privado creado mediante `git`. Debe haber constancia, por tanto, del trabajo realizado en `git` a través de la carpeta `.git`, que debe entregarse obligatoriamente. **Al menos debe haber un envío a git por cada día de clase por cada miembro del grupo.**
- El código necesario para ejecutar la aplicación deben entregarse en *Moodle*, y debe ser posible ejecutarlo localmente.

NOTA: El código usado para corregir esta práctica será el subido a *Moodle*. En ningún caso se usará el código existente en el repositorio *Git* o en *Render.com*.