

Bienvenido al curso de automatización de tareas con **Justo.js**. Nuestro objetivo es describir detalladamente cómo automatizar tareas bajo esta plataforma. Asume del estudiante conocimientos previos de **JavaScript** y **Node.js**.

La lección comienza presentando el concepto de automatización, sus usos más frecuentes y sus ventajas. Y a continuación, se muestra el programa del curso.

Al finalizar la lección, el estudiante sabrá:

- En qué consiste la automatización.
- Por qué es importante automatizar.
- Cuál es el programa del curso.

Introducción

La **automatización** (*automation*) es el proceso mediante el cual se convierte tareas manuales en automáticas. Al automatizar las tareas, podemos ejecutarlas más fácil y rápidamente que si tuviéramos que hacerlo una y otra vez manualmente.

La automatización se lleva a cabo mediante **herramientas de automatización** (*automation tools*), aplicaciones de software que permiten convertir e implementar tareas manuales repetitivas en automáticas. Entre otras, tenemos **Ansible**, **Grun**, **Gulp**, **Justo** o **Puppet**, cada una de ellas orientadas a un determinado mercado.

Usos de la automatización

Entre los principales usos de la automatización encontramos:

- Construcción de software.
- Despliegue o instalación de software.
- Realización de pruebas.
- Administración y monitorización de configuración.
- Realización de empaquetados.
- Extracción, transformación y carga de datos.
- Generación de archivos y/o directorios a partir de plantillas.

Cuando construimos software, es muy común automatizar determinadas tareas como, por ejemplo, la compilación, la validación de buenas prácticas y de gramática, etc. En estos casos, se suele utilizar algún tipo de herramienta de automatización para, con un simple comando, solicitar la invocación de una determinada tarea o proceso automatizado. Por ejemplo, **Justo** viene con *plugins* para la automatización de tareas de compilación, de validación de buenas prácticas, de copia de archivos, etc., ayudando así a ser más productivos y eficientes.

Independientemente del tamaño de la organización, es común realizar instalaciones o despliegues de aplicaciones webs, de instancias de bases de datos y otros productos. Por buenas prácticas, se recomienda seguir guías de instalación específicas de la organización para que todas las instalaciones sigan los convenios internos. Estas guías se pueden automatizar parcial o totalmente mediante algún tipo de herramienta de automatización. Con ello, se instalará más fácilmente y se reducirá los errores. Con **Justo** se puede utilizar *plugins* o, mejor todavía, desarrollar *plugins* específicos que sigan los estándares de la organización. Así, cuando vayamos a realizar una nueva instalación de, digamos, un motor de bases de datos como, por ejemplo, **ArangoDB**, **CouchDB**, **PostgreSQL**, **Redis** o **RethinkDB**, usaremos su versión automatizada, probada y estandarizada. Actualmente, **Justo** dispone de *plugins*

oficiales para instalar, desinstalar o actualizar paquetes con **APT**, así como para crear usuarios y grupos de usuarios, lo que ayudará a desplegar nuevo software de manera muy sencilla.

La realización de pruebas es un aspecto muy importante tanto en el desarrollo de software como en las instalaciones. En el primer caso, los desarrolladores deben probar su software. Deben comprobar que cada componente hace lo que se espera de él sin efectos inesperados. Por otra parte, los administradores deben probar que sus instalaciones se han realizado correctamente ejecutando determinadas consultas a su entorno, validando la configuración, etc. En ambos casos, la ejecución de pruebas se debe automatizar para hacerlas más rápidas y eficientes. **Justo** viene con un componente inherente que ayuda a escribir y ejecutar las pruebas de manera muy sencilla.

La realización de paquetes es también una tarea muy común en la comunidad de sistemas y de desarrollo. Por ejemplo, cuando un desarrollador finaliza un componente de software, debe empaquetarlo para poder utilizarlo y/o reutilizarlo en otros proyectos. Este empaquetado se suele automatizar. **Justo** cuenta con tareas que ayudan, por ejemplo, a copiar, borrar y comprimir archivos permitiendo así crear el paquete. También proporciona tareas para convertir archivos de formato **.odt** a **.pdf** y para publicar paquetes de **Node.js** mediante **NPM**.

La extracción, transformación y carga de datos, también conocida como **ETL** (*extract, transform, load*), es un proceso ampliamente utilizado en bases de datos para transferir datos de uno o más orígenes de datos a un almacén de datos. El proceso comienza extrayendo los datos de sus orígenes. A continuación, los transforma o adapta al formato y/o estructura que tiene el destino. Finalmente, los carga en el almacén destino. Este tipo de proceso es muy fácil automatizarlo mediante **Justo**.

Otro aspecto muy automatizable es la generación de archivos y/o directorios. Generalmente, cuando se comienza un proyecto de software o empaquetado de algún tipo, se necesita crear una estructura de directorios específica. Este tipo de tareas es muy común automatizarla de alguna manera para reducir errores y aumentar la productividad. Esto se puede realizar en **Justo** mediante los generadores. Por ejemplo, se puede utilizar un generador para crear la estructura de un proyecto con el que desarrollar una aplicación **Express**, **Horizon**, **Node.js** o **React**.

Ventajas de la automatización

Las principales ventajas de la automatización son las siguientes:

- Aumento de la productividad.
- Aumento de la calidad.
- Aumento de la precisión o exactitud.
- Aumento de la robustez.
- Reducción de costes.
- Reducción de errores.
- Reducción de riesgos.
- Reducción de tareas manuales.
- Reducción de tiempos de ejecución, de operación, de despliegue...
- Reducción de mantenimiento.
- Reducción de la duplicidad de trabajos.
- Reducción de la complejidad.

Son muchas las organizaciones, independientemente de su tamaño y sector, que utilizan algún tipo de herramienta de automatización como, por ejemplo, **Adobe**, **Amazon**, **ARM**, **AT&T**, **BBC**, **BT**, **Costa Coffee**, **Dell**, **eBay**, **Facebook**, **Financial Times**, **GitHub**, **Google**, **IBM**, **Intel**, **Lidl**, **Microsoft**, **MongoDB**, **Mozilla**, **NASA**, **Netflix**, **Red Hat**, **Sony**, **Spotify**, **Telefónica**, **The Guardian**, **Twitter**, **Virgin**, **Walmart**, **Yahoo!** y **Zalando**.

Información del curso

Este curso tiene básicamente dos objetivos. El primero es enseñar en qué consiste la automatización. Y

el segundo, cómo automatizar con **Justo.js**. Dejándose para otros específicos los generadores y las pruebas de software.

Al finalizarlo, el estudiante sabrá:

- En qué consiste la automatización.
- Cómo instalar **Justo.js**.
- Cómo automatizar mediante **Justo.js**.
- Cómo utilizar y desarrollar *plugins* y módulos para extender la funcionalidad de **Justo.js**.

Conocimientos previos

El estudiante debe tener conocimiento de **JavaScript** y **Node.js**, pues los *scripts*, los *plugins* y los módulos, con los que extender **Justo**, se redactan con este lenguaje y esta plataforma.

Plan de estudio

El curso tiene una duración aproximada de **4 horas**. Se divide en **7 lecciones**, cada una de ellas con una parte de teoría y generalmente una de práctica. Al finalizar el curso, puede realizar **un examen** de tipo test, con el que evaluar los conocimientos adquiridos.

El enfoque a seguir es muy sencillo: ir lección a lección; primero hay que leer la teoría y, después, realizar la práctica. Se recomienda encarecidamente que el estudiante realice cada lección, tanto teoría como práctica, en el mismo día, con el menor número de interrupciones a lo largo de su estudio. Al finalizar el curso, se recomienda realizar el examen.

A continuación, se enumera las distintas lecciones y el tiempo estimado para su estudio:

Lección	Teoría	Práctica	Descripción
1 Introducción	10min	-	Esta lección.
2 Introducción a Justo.js	15min	10min	Introducción a Justo.js , cuáles son sus principales características y cómo instalarlo.
3 Tareas simples	30min	15min	Qué son las tareas simples y cómo definir las y trabajar con ellas.
4 Macros	10min	10min	Qué son las macros y cómo definir las y trabajar con ellas.
5 Flujos de trabajo	10min	10min	Qué son los flujos de trabajo, cómo definirlos y trabajar con ellos.
6 Plugins	15min	20min	Qué son los <i>plugins</i> y cómo definir <i>plugins</i> de usuario.
7 Módulos	15min	20min	Qué son los módulos y cómo desarrollarlos y trabajar con ellos.
Examen	30min		Evaluación de los conocimientos adquiridos.

Información de publicación

Título **Automatización de tareas con Justo.js**

Autor **Raúl G. González - raulggonzalez@nodemy.com**

Revisión **Nacho del Rey y Enrique Gil**

Primera edición **Agosto de 2016**

Última actualización **Enero de 2017**

Versión actual **1.1.0**

Versión de **Justo** **0.19**

Contacto **hola@nodemy.com**