

¿Preparado para el reto de la **Agenda**?



En este gran reto crearemos una herramienta muy útil: una agenda.

La agenda está relacionada con el ser humano desde que se comenzó a medir el tiempo. De hecho, se crearon específicamente para eso. En un mundo lleno de demandas laborales, personales y familiares, tener la posibilidad de almacenar los eventos en un entorno virtual, según la hora y la fecha, es de gran utilidad. Es por este motivo que te invitamos a crear una propia y a que, una vez adquieras la habilidad, la incluyas en tus proyectos de desarrollo.

## Objetivo

Desarrolla una agenda que permita crear, editar y eliminar eventos con su información asociada, almacenándolos en una base de datos.

## Requerimientos Generales

- Para realizar correctamente este proyecto debes desarrollar dos versiones de la agenda: Una en PHP y una base de datos en MySQL; la segunda versión en Node.js y una base de datos en MongoDB; de esta manera podrás aplicar tus conocimientos en ambas tecnologías y motores de bases de datos.
- Para la versión en PHP de la agenda debes descargar el archivo [NextUcodBaseAgenda\\_versionPHP.zip](#) en donde encontrarás todo el desarrollo Front-End de la agenda. De la misma manera, para la versión en Node.js debes descargar el archivo [NextUcodBaseAgenda\\_versionNode.zip](#) que contiene el Front-End de la agenda para esta versión.

Los requerimientos específicos de este proyecto son:

### REQUERIMIENTOS VERSIÓN PHP-MYSQL:

1

Creará una base de datos mediante la herramienta **phpMyAdmin** en la que se pueda almacenar usuarios con sus contraseñas y eventos de la agenda asociados a cada usuario. Cada evento comprende un id, título, fecha de inicio, hora de inicio, fecha de finalización, hora de finalización (los últimos 3 pueden ser nulos), y un campo booleano que indique si el evento es de día completo o no. Para cada usuario la base de datos debe almacenar el correo electrónico que funcionará como el nombre de usuario único ante el sistema, el nombre completo, su contraseña, y su fecha de nacimiento.

2

Creará 3 usuarios en la base de datos desde el servidor ejecutando un script de PHP, las contraseñas deben almacenarse encriptadas.

3

Valida el ingreso al sistema al iniciar sesión, comparando los datos ingresados en el formulario con los almacenados en la base de datos.

4

Al iniciar una sesión exitosamente, el sistema debe cargar todos los eventos asociados al usuario en la sesión desde la base de datos y mostrarlos en el calendario. La librería usada para el calendario es **FullCalendar**. Puedes encontrar su documentación, si lo consideras necesario, en el siguiente enlace: [fullcalendar.io](http://fullcalendar.io)

U\_CODE CSS ▾

## Inicia sesión

Usuario

jp@mail.com

Contraseña

Enviar

5  
Al diligenciar el formulario de la derecha, se debe crear un nuevo evento en el calendario y debe ser almacenado en la base de datos.

## Agenda

&lt; &gt; today

November 2016

month week day

Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	31	1	2	3 13:00 Cita Veterinario	4	5
6 18:00 Partido de Futb	7	8 Recoger Ropa	9	10 14:30 Recoger Mesa	11	12 7:30 Entrevista de trat
13	14	15	16	17	18 10:00 Cita con el mec	19 18:00 Peluquería
20	21	22	23	24 7:00 Recoger flores	25	26
27	28	29 Cumpleaños de Santi	30	1	2	3

U\_CODE

CSS

5

6

7

8

9

10

6

Al arrastrar un evento al ícono de la basura en la parte derecha de la página, éste debe ser eliminado del calendario y su registro debe eliminarse de la base de datos.

# Agenda

<

>

today

November 2016

month

week

day

Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	31	1	2	3	4	5
				13:00 Cita Veterinario		

## Estatus de la entrega

Número de intento	Éste es el intento 1.
Estatus de la entrega	Sin intento
Estatus de calificación	No calificado
Criterio para calificar	<div>Completitud</div> <div>Funcional</div>

U\_CODE

CSS



	<p>Experto. El proyecto cumple con: VERSIÓN PHP-MYSQL: (1) Crea una base de datos para almacenar usuarios, correo electrónico, contraseña, fecha de nacimiento y eventos de la agenda. Cada evento comprende un id, título, fecha de inicio y fin, hora de inicio y fin, campo booleano que indique si el evento es de día completo o no. (2) Guarda 3 usuarios en la base de datos desde el servidor ejecutando un script de PHP, las contraseñas deben almacenarse encriptadas. (3) Crea un nuevo evento en el calendario y se almacena en la base de datos. (4) Elimina eventos. (5) Actualiza eventos. VERSIÓN NODE.JS - MONGODB: (1) Crea un script que al ejecutarse por consola, cree un nuevo usuario para el sistema en una base de datos en MongoDB. (2) Crea un nuevo evento en el calendario y se almacena en la base de datos. (3) Elimina eventos. (4) Actualiza eventos.</p> <p><b>50 puntos</b></p>	<p>Avanzado. El proyecto cumple de 6 a 8 de los siguientes items: VERSIÓN PHP-MYSQL: (1) Crea una base de datos para almacenar usuarios, correo electrónico, contraseña, fecha de nacimiento y eventos de la agenda. Cada evento comprende un id, título, fecha de inicio y fin, hora de inicio y fin, campo booleano que indique si el evento es de día completo o no. (2) Guarda 3 usuarios en la base de datos desde el servidor ejecutando un script de PHP, las contraseñas deben almacenarse encriptadas. (3) Crea un nuevo evento en el calendario y se almacena en la base de datos. (4) Elimina eventos. (5) Actualiza eventos. VERSIÓN NODE.JS - MONGODB: (1) Crea un script que al ejecutarse por consola, cree un nuevo usuario para el sistema en una base de datos en MongoDB. (2) Crea un nuevo evento en el calendario y se almacena en la base de datos. (3) Elimina eventos. (4) Actualiza eventos.</p> <p><b>37 puntos</b></p>	<p>Aprendiz. El proyecto cumple de 3 a 5 de los siguientes items: VERSIÓN PHP-MYSQL: (1) Crea una base de datos para almacenar usuarios, correo electrónico, contraseña, fecha de nacimiento y eventos de la agenda. Cada evento comprende un id, título, fecha de inicio y fin, hora de inicio y fin, campo booleano que indique si el evento es de día completo o no. (2) Guarda 3 usuarios en la base de datos desde el servidor ejecutando un script de PHP, las contraseñas deben almacenarse encriptadas. (3) Crea un nuevo evento en el calendario y se almacena en la base de datos. (4) Elimina eventos. (5) Actualiza eventos. VERSIÓN NODE.JS - MONGODB: (1) Crea un script que al ejecutarse por consola, cree un nuevo usuario para el sistema en una base de datos en MongoDB. (2) Crea un nuevo evento en el calendario y se almacena en la base de datos. (3) Elimina eventos. (4) Actualiza eventos.</p> <p><b>19 puntos</b></p>	<p>Novato. El proyecto cumple de 1 a 2 de los siguientes items: VERSIÓN PHP-MYSQL: (1) Crea una base de datos para almacenar usuarios, correo electrónico, contraseña, fecha de nacimiento y eventos de la agenda. Cada evento comprende un id, título, fecha de inicio y fin, hora de inicio y fin, campo booleano que indique si el evento es de día completo o no. (2) Guarda 3 usuarios en la base de datos desde el servidor ejecutando un script de PHP, las contraseñas deben almacenarse encriptadas. (3) Crea un nuevo evento en el calendario y se almacena en la base de datos. (4) Elimina eventos. (5) Actualiza eventos. VERSIÓN NODE.JS - MONGODB: (1) Crea un script que al ejecutarse por consola, cree un nuevo usuario para el sistema en una base de datos en MongoDB. (2) Crea un nuevo evento en el calendario y se almacena en la base de datos. (3) Elimina eventos. (4) Actualiza eventos.</p> <p><b>10 puntos</b></p>	<p>Incumplido. El proyecto enviado no corresponde a los requerimientos solicitados.</p> <p><b>0 puntos</b></p>

U\_CODE

CSS



<b>Correctitud Funcional</b>	<p>Experto. El proyecto cumple con todos los siguientes items:</p> <p>VERSIÓN PHP-MYSQL: (1) Valida el ingreso al sistema al iniciar sesión, comparando los datos ingresados en el formulario con los almacenados en la base de datos. (2) Al iniciar una sesión, el sistema debe cargar todos los eventos asociados al usuario en la sesión desde la base de datos y mostrarlos en el calendario.</p> <p>VERSIÓN NODE.JS - MONGODB: (1) Valida el ingreso al sistema al iniciar sesión, comparando los datos ingresados en el formulario con los almacenados en la base de datos. (2) Al iniciar una sesión exitosamente, el sistema debe cargar todos los eventos asociados al usuario en la sesión desde la base de datos y mostrarlos en el calendario.</p> <p><b>30 puntos</b></p>	<p>Avanzado. El proyecto del estudiante cumple con 3 de los siguientes items:</p> <p>VERSIÓN PHP-MYSQL: (1) Valida el ingreso al sistema al iniciar sesión, comparando los datos ingresados en el formulario con los almacenados en la base de datos. (2) Al iniciar una sesión, el sistema debe cargar todos los eventos asociados al usuario en la sesión desde la base de datos y mostrarlos en el calendario.</p> <p>VERSIÓN NODE.JS - MONGODB: (1) Valida el ingreso al sistema al iniciar sesión, comparando los datos ingresados en el formulario con los almacenados en la base de datos. (2) Al iniciar una sesión exitosamente, el sistema debe cargar todos los eventos asociados al usuario en la sesión desde la base de datos y mostrarlos en el calendario.</p> <p><b>26 puntos</b></p>	<p>Aprendiz. El proyecto del estudiante cumple con 2 de los siguientes items:</p> <p>VERSIÓN PHP-MYSQL: (1) Valida el ingreso al sistema al iniciar sesión, comparando los datos ingresados en el formulario con los almacenados en la base de datos. (2) Al iniciar una sesión, el sistema debe cargar todos los eventos asociados al usuario en la sesión desde la base de datos y mostrarlos en el calendario.</p> <p>VERSIÓN NODE.JS - MONGODB: (1) Valida el ingreso al sistema al iniciar sesión, comparando los datos ingresados en el formulario con los almacenados en la base de datos. (2) Al iniciar una sesión exitosamente, el sistema debe cargar todos los eventos asociados al usuario en la sesión desde la base de datos y mostrarlos en el calendario.</p> <p><b>10 puntos</b></p>	<p>Novato. El proyecto del estudiante cumple con 1 de los siguientes items:</p> <p>VERSIÓN PHP-MYSQL: (1) Valida el ingreso al sistema al iniciar sesión, comparando los datos ingresados en el formulario con los almacenados en la base de datos. (2) Al iniciar una sesión, el sistema debe cargar todos los eventos asociados al usuario en la sesión desde la base de datos y mostrarlos en el calendario.</p> <p>VERSIÓN NODE.JS - MONGODB: (1) Valida el ingreso al sistema al iniciar sesión, comparando los datos ingresados en el formulario con los almacenados en la base de datos. (2) Al iniciar una sesión exitosamente, el sistema debe cargar todos los eventos asociados al usuario en la sesión desde la base de datos y mostrarlos en el calendario.</p> <p><b>5 puntos</b></p>	<p>El proyecto enviado no corresponde a los requerimientos solicitados.</p> <p><b>0 puntos</b></p>
<b>Capacidad de Ser Modificado</b>	<p>Experto. El proyecto fue publicado en Github y se evidencia el uso de versiones.</p> <p><b>10 puntos</b></p>	<p>Avanzado. El proyecto fue publicado en Github.</p> <p><b>9 puntos</b></p>	<p>Aprendiz. Existe el enlace del proyecto en Github, pero no se encuentra actualizado.</p> <p><b>5 puntos</b></p>	<p>Novato. El proyecto no fue publicado en Github.</p> <p><b>2 puntos</b></p>	<p>Incumplido. El proyecto enviado no corresponde a los requerimientos solicitados.</p> <p><b>0 puntos</b></p>

U\_CODE 

CSS ▾

Legibilidad y Calidad de Código					
	Experto. El proyecto cumple con: (1) Código ordenado. (2) Nombra los elementos, clases y recursos, de manera descriptiva y concisa. <i>10 puntos</i>	Avanzado. El 80% proyecto cumple con: (1) Código ordenado. (2) Nombra los elementos, clases y recursos, de manera descriptiva y concisa. <i>8 puntos</i>	Aprendiz. El proyecto cumple con solo 1 de las siguientes: (1) Código ordenado. (2) Nombra los elementos, clases y recursos, de manera descriptiva y concisa. <i>5 puntos</i>	Novato. El proyecto no cumple con: (1) Código ordenado. (2) Nombra los elementos, clases y recursos, de manera descriptiva y concisa. <i>2 puntos</i>	Incumplido. El proyecto enviado no corresponde a los requerimientos solicitados. <i>0 puntos</i>

Última modificación

-

Añadir envío

Hacer cambios a su envío

Copyright © 2016 Next U LLC.

¿Quiénes somos?   Contáctenos   Condiciones de servicio   Política de privacidad

