

Desarrollo web independiente

Alberto Benavides

San Nicolás, N. L.

20 de noviembre de 2019

Temario

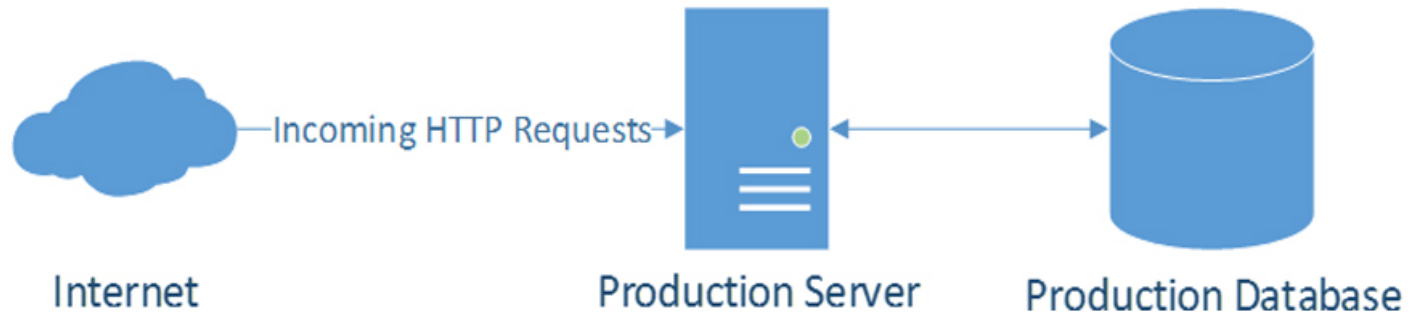
- Conceptos fundamentales
- Consejos basados en mi experiencia
- Recomendaciones técnicas
- Mini tutorial

Conceptos fundamentales

Página web VS Aplicación web

Página web	Aplicación web
Publicitaria	Administrativa
Sin acceso de usuarios	Con acceso de usuarios
Contenido estático	Contenido dinámico
Visual	Interactiva
Bajo coste	Elevado coste
Poco tiempo	Mucho tiempo
Mínima responsabilidad	Gran responsabilidad

Servidor



Peticiones

- Get
 - Petición por barra de búsqueda
`http://ruta.com?llave=valor`
 - Compartir información
 - Identificar recursos
 - Tamaño limitado (no archivos multimedia)
- Post
 - Petición en el cuerpo del mensaje
 - Oculta
 - Información confidencial
 - Tamaño "ilimitado" (servidor)

Consejos basados en mi experiencia

Habilidades ideales

- Programador: Conoce distintos lenguajes (mín. 3)
- Artista visual: Crea contenidos estéticos
- Mercadólogo: Crea contenidos intuitivos
- Matemático: Resuelve problemas con algoritmos
- Vendedor: Negocia precios con los clientes

Ventajas

- Libertad de horarios y "oficinas"
- Distintos proyectos = Mucho aprendizaje
- Aprendes a negociar = Valorar tu trabajo
- Dinero negociado = Dinero ganado

Desventajas

- Hay que convencer al cliente
- ¡Hay que aprender a cobrar!
<https://www.cuantocuestamiweb.com/>
- Proyectos grandes: Necesidad de equipo multidisciplinario

Recomendaciones

- Ser sincero: Capacidades, inconvenientes, limitaciones
- No aceptar trabajos mal pagados
- Establecer un documento con entregables

Recomendaciones técnicas

Software

- Visual Studio Code: Editor de textos pensado para programadores
- XAMPP: Plataforma independiente para servidores locales
- Sourcetree: Manejador visual de control de versiones
- Consola o terminal: `cd`

Tecnologías indispensables

- HTML
 - *Hiper Text Markup Language*
 - Lenguaje de formateo de texto
 - No es un lenguaje de programación
- CSS
 - *Cascading Style Sheet*
 - Hojas de estilo de cascadeo
 - Estilo de páginas web
- JS: Lenguaje de programación integrado en navegadores

Paquete básico

- PHP
 - *Programming Hypertext Preprocessor*
 - Lenguaje de programación para HTML incrustado
- jQuery
 - Librería de JS
 - Facilita la manipulación de los elementos HTML
 - Añade atajos de funciones de JS
- MySQL:
 - Base de datos relacional
 - Funciona con tablas
- Bootstrap: Plantilla de diseño utilizada por Twitter

Mini tutorial de HTML a Laravel

Estructura mínima HTML

```
<html>
  <head>
    <title>Ejemplo HTML</title>
    <style>
      #rojo{
        color: red;
      }
    </style>
  </head>
  <body>
    <h1>Características</h1>
    <ul>
      <li>Usa etiquetas</li>
      <li id="rojo">Estilos por selectores</li>
    </ul>
    <script>
      alert("Mira esta alerta.");
    </script>
  </body>
</html>
```

Recomendación **Laravel**

- *Framework* PHP = Marco de trabajo en PHP
- Conjunto de programas, métodos, instrucciones... que realizan una tarea específica
- Tarea específica: Crear una aplicación web



Requisitos

- [Git](#): Software de control de versiones
- [Composer](#): Herramienta para manipulación de dependencias de PHP



Instalación

```
composer create-project --prefer-dist laravel/laravel nombreApp
```

Servidor local

```
php artisan serve
```

```
rem Entrar a http://localhost:8000
```

Archivo `.env`

- (Revisar) Renombrar archivo `.env.example` a `.env`
- Contiene la información del entorno de la aplicación
- Obtener variables de entorno

```
// El segundo parámetro de env() es el valor por defecto  
'debug' => env('APP_DEBUG', false), // en config/app.php  
  
$environment = App::environment();
```

Llave de la aplicación

Permite asegurar sesiones y datos encriptados

1. Generar llave de la aplicación

```
php artisan key:generate
```

2. Agregar la llave al archivo `.env`

```
APP_KEY=base64:...
```

Carpetas principales

- `app` : Modelos de tablas (Usuario)
- `config` : Todas las configuraciones de la aplicación
- `database` :
 - `migrations` : Especificación de tablas (usuarios)
 - `seeds` : Registros predeterminados de tablas
- `public` :
 - `css` , `js` : Archivos de estilo y scripts
 - `index.php` : Llegada de peticiones de usuarios

- `resources` :
 - `lang` : Contiene los idiomas instalados
 - `views` : Contiene las vistas (páginas) de la aplicación
- `routes/web.php` : Contiene las redirecciones de la aplicación
- `storage` : Archivos generados por cada sesión
 - `storage/app/public` : Usada generalmente para almacenar archivos generados por el usuario

Creación de base de datos con MySQL

1. Abrir XAMPP
2. Ejecutar MySQL
3. Entrar al shell de XAMPP
4. Entrar a la base de datos:

```
mysql -u root -p
```

5. Crear base de datos

```
create database ejemplo;
```

6. Actualizar `.env`

Configuración de autenticación

```
// app/Providers/AppServiceProvider.php

use Illuminate\Support\Facades\Schema;

public function boot()
{
    Schema::defaultStringLength(191);
}
```

```
rem cd carpeta/del/proyecto

rem Prepara vistas, modelos y migrations para auth
composer require laravel/ui --dev
npm install && npm run dev
php artisan ui vue --auth

rem Corre las migraciones
php artisan migrate
```

Subir repositorio a GitHub

1. Crear cuenta en [GitHub](#)
2. Clic en + (arriba a la derecha) -> New repository
3. Agregar un nombre
4. Create repository
5. `git init`
6. `git remote add origin url`

Cada cambio significativo:

7. `git add *`
8. `git commit -m "Mensaje"`
9. `git push origin master`

Referencias

- <https://www.progress.com/documentation/sitefinity-cms/reference-architecture-diagrams>
- <https://developer.mozilla.org/es/docs/Web/HTML>
- <https://developer.mozilla.org/es/docs/Web/CSS>
- https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/Qué_es_JavaScript
- <https://jquery.com/>
- <https://laravel.com/>
- <https://vuejs.org/>