

# GUÍA PARA COMPLETAR EL PDF DE ENTREGA

---

## Estructura Sugerida para el PDF

### 1. PORTADA

- Título: "Asignación 1: Fundamentos de Laravel - Rutas, Controladores y Vistas"
  - Tu nombre completo
  - Número de estudiante
  - Nombre del curso
  - Nombre del profesor
  - Fecha: 15 de Noviembre, 2025
- 

### 2. INTRODUCCIÓN (1 página)

#### Contenido sugerido:

- Breve descripción del objetivo de la asignación
- Resumen de lo que implementaste (2 rutas: estática y dinámica)
- Tecnologías utilizadas: Laravel, PHP, Blade

#### Ejemplo:

Esta asignación tiene como objetivo implementar los fundamentos de Laravel para manejar solicitudes web mediante el uso de rutas, controladores y vistas Blade. Se desarrollaron dos funcionalidades principales: una ruta estática de bienvenida y una ruta dinámica de saludo que acepta parámetros.

### 3. DESARROLLO DEL PROYECTO (3-4 páginas)

#### 3.1 Estructura del Proyecto

- Diagrama o descripción de la estructura de archivos creados
- Explicación de dónde se encuentra cada componente (controlador, rutas, vistas)

#### 3.2 Implementación de la Ruta Estática

##### Código del controlador:

```
public function bienvenida()
{
    return view('bienvenida');
}
```

**Ruta definida:**

```
Route::get('/bienvenida', [PaginaController::class, 'bienvenida']);
```

**Explicación:**

- Describe cómo funciona esta ruta
- Explica el flujo: URL → Ruta → Controlador → Vista

**3.3 Implementación de la Ruta Dinámica****Código del controlador:**

```
public function saludo($nombre)
{
    return view('saludo', ['nombre' => $nombre]);
}
```

**Ruta definida:**

```
Route::get('/saludo/{nombre}', [PaginaController::class, 'saludo']);
```

**Explicación:**

- Describe cómo se captura el parámetro de la URL
- Explica cómo se pasa el dato a la vista
- Menciona el uso de sintaxis Blade: {{ \$nombre }}

**3.4 Vistas Blade**

- Descripción de las vistas creadas
- Explicación del uso de Blade
- Mención de los estilos CSS aplicados

---

**4. CAPTURAS DE PANTALLA (2-3 páginas)****4.1 Ruta de Bienvenida****Instrucciones para tomar la captura:**

1. Inicia el servidor: `php artisan serve`
2. Abre el navegador en: `http://localhost:8000/bienvenida`
3. Toma captura de pantalla que muestre:
  - La URL completa en la barra de direcciones

- El contenido de la página: "Bienvenido a mi primera aplicación de Laravel"

**Incluir en el PDF:**

- Captura de pantalla clara
- Pie de foto: "Figura 1: Ruta estática /bienvenida mostrando mensaje de bienvenida"

**4.2 Ruta de Saludo - Ejemplo 1****Instrucciones:**

1. Visita: <http://localhost:8000/saludo/Carlos>
2. Toma captura mostrando "Hola, Carlos"

**Incluir en el PDF:**

- Captura de pantalla
- Pie de foto: "Figura 2: Ruta dinámica /saludo/Carlos con parámetro 'Carlos'"

**4.3 Ruta de Saludo - Ejemplo 2****Instrucciones:**

1. Visita: <http://localhost:8000/saludo/Maria>
2. Toma captura mostrando "Hola, Maria"

**Incluir en el PDF:**

- Captura de pantalla
- Pie de foto: "Figura 3: Ruta dinámica /saludo/Maria con parámetro 'Maria'"

**4.4 Capturas Adicionales (Opcional pero recomendado)**

- Captura del código del controlador en el editor
- Captura del archivo de rutas (web.php)
- Captura de las vistas Blade

---

**5. PREGUNTAS DE REFLEXIÓN (2-3 páginas)****Pregunta 1: ¿Qué es el enrutamiento en Laravel y por qué es importante?****Tu respuesta debe incluir:**

- Definición de enrutamiento
- Función principal (capturar peticiones HTTP)
- Importancia en la arquitectura de la aplicación
- Ejemplo de cómo se usa en tu proyecto

**Ejemplo de respuesta:**

El enrutamiento en Laravel es el sistema que captura las peticiones HTTP entrantes y las dirige hacia la lógica de aplicación apropiada. Es importante porque define los puntos de entrada de la aplicación web y permite organizar de manera clara cómo responde la aplicación a diferentes URLs. En mi proyecto, las rutas definidas en web.php conectan las URLs /bienvenida y /saludo/{nombre} con los métodos correspondientes del PaginaController.

**Pregunta 2: ¿Cuál es la diferencia entre una ruta estática y una ruta dinámica?****Tu respuesta debe incluir:**

- Definición de ruta estática
- Definición de ruta dinámica
- Ejemplos de tu proyecto
- Casos de uso para cada tipo

**Pregunta 3: ¿Qué papel juega el controlador en el patrón MVC de Laravel?****Tu respuesta debe incluir:**

- Explicación del patrón MVC
- Rol específico del controlador
- Cómo el controlador interactúa con rutas y vistas
- Ejemplo de tu PaginaController

**Pregunta 4: ¿Cómo se pasan datos desde un controlador a una vista en Laravel?****Tu respuesta debe incluir:**

- Métodos para pasar datos (array, with(), compact())
- Ejemplo específico de tu método saludo()
- Cómo se reciben los datos en la vista Blade

**Pregunta 5: ¿Qué es Blade y qué ventajas ofrece para el desarrollo de vistas?****Tu respuesta debe incluir:**

- Definición de Blade
- Ventajas principales (sintaxis, seguridad, herencia)
- Ejemplos de sintaxis Blade usada en tu proyecto
- Comparación con PHP puro

---

## 6. CONCLUSIONES (1 página)

**Contenido sugerido:**

- Resumen de lo aprendido
- Desafíos encontrados y cómo los superaste

- Aplicaciones prácticas de estos conceptos
- Próximos pasos en tu aprendizaje de Laravel

### Ejemplo:

A través de esta asignación, he comprendido los fundamentos del flujo de trabajo en Laravel: rutas, controladores y vistas. Aprendí cómo Laravel organiza el código siguiendo el patrón MVC y cómo el sistema de enrutamiento permite crear tanto rutas estáticas como dinámicas. El uso de Blade como motor de plantillas simplifica significativamente el desarrollo de vistas...

---

## 7. REFERENCIAS

- Documentación oficial de Laravel: <https://laravel.com/docs>
  - Guía de rutas: <https://laravel.com/docs/routing>
  - Guía de controladores: <https://laravel.com/docs/controllers>
  - Guía de vistas Blade: <https://laravel.com/docs/blade>
- 

## CHECKLIST ANTES DE ENTREGAR EL PDF

- Portada con todos los datos completos
  - Introducción que explica el propósito del proyecto
  - Desarrollo detallado de la implementación
  - Código fuente incluido y explicado
  - Al menos 3 capturas de pantalla claras
  - Las 5 preguntas de reflexión respondidas completamente
  - Conclusiones personales
  - Referencias bibliográficas
  - Formato consistente y profesional
  - Revisión ortográfica y gramatical
  - Numeración de páginas
  - Índice (opcional pero recomendado)
- 

## TIPS PARA UN PDF DE CALIDAD

1. **Formato:** Usa un procesador de textos profesional (Word, Google Docs, LaTeX)
  2. **Fuente:** Arial o Times New Roman, tamaño 11-12 pt
  3. **Márgenes:** 2.5 cm en todos los lados
  4. **Interlineado:** 1.5 líneas
  5. **Capturas:** Alta resolución, con bordes si es necesario
  6. **Código:** Usa formato de código (Courier New o similar)
  7. **Secciones:** Numeradas claramente (1, 1.1, 1.2, etc.)
  8. **Longitud:** 10-15 páginas aproximadamente
-

# INSTRUCCIONES PARA INICIAR EL SERVIDOR

Antes de tomar las capturas de pantalla, debes instalar Laravel y ejecutar el servidor:

Opción 1: Si tienes Composer instalado

```
cd c:\Users\Nefta\practica1  
composer install  
php artisan serve
```

Opción 2: Si NO tienes Composer

1. Descarga e instala Composer desde: <https://getcomposer.org/>
2. Descarga e instala PHP 8.0 o superior
3. Luego ejecuta los comandos de la Opción 1

Opción 3: Usar Laravel Herd (Recomendado para Windows)

1. Descarga Laravel Herd: <https://herd.laravel.com/>
2. Instala Herd
3. Añade la carpeta practica1 a Herd
4. Accede a: <http://practica1.test>

---

## REPOSITORIO DE GITHUB

Pasos para subir tu proyecto a GitHub:

**1. Crear repositorio en GitHub**

- Ve a [github.com](https://github.com) y crea un nuevo repositorio
- Nómbralo: "laravel-practica1-rutas-controladores"
- No inicialices con README (ya tienes uno)

**2. Inicializar Git localmente**

```
cd c:\Users\Nefta\practica1  
git init  
git add .  
git commit -m "Asignación 1: Fundamentos de Laravel - Rutas, Controladores y  
Vistas"
```

**3. Conectar con GitHub y subir**

```
git branch -M main  
git remote add origin https://github.com/TU_USUARIO/laravel-practica1-rutas-
```

```
controladores.git  
git push -u origin main
```

#### 4. Verificar

- Asegúrate de que el directorio /vendor NO esté en GitHub (gracias al .gitignore)
  - Verifica que todos los archivos importantes estén subidos
- 

¡Buena suerte con tu entrega! 