

Informe de Despliegue de Aplicación Web

Alejandro Roncancio Roldan
María Fernanda Rios
Ivan Camilo Aguilar

Universidad de san Buenaventura
Facultad de ingeniería

Bogotá, Colombia
Mayo del 2025

Contenido

Informe de Despliegue de Aplicación Web	3
1. Introducción	3
2. Descripción de la Aplicación	3
3. Requisitos de Despliegue.....	4
4. Presupuesto	4
5. Herramientas Utilizadas	5
6. Proceso de Despliegue	5
7. Consideraciones de Seguridad.....	6
8. Conclusiones	7
9. Referencias.....	8

Informe de Despliegue de Aplicación Web

1. Introducción

El presente informe tiene como propósito describir de forma detallada el proceso de despliegue de la aplicación web desarrollada en el marco del proyecto académico de la asignatura Construcción de Software, correspondiente al programa de Tecnología en Desarrollo de Software de la Universidad de San Buenaventura. Esta aplicación, titulada Checklist, tiene como finalidad automatizar el proceso de control de asistencia de los estudiantes en los salones de clase.

Este informe expone no solo los pasos técnicos realizados para desplegar la aplicación, sino también los requisitos del entorno, las herramientas empleadas, consideraciones de seguridad, presupuesto estimado y conclusiones derivadas de la experiencia. Se utilizó Render, una plataforma de alojamiento gratuita y moderna, para facilitar el despliegue del backend de la aplicación.

El despliegue de una aplicación web es una fase crucial, ya que implica trasladar el desarrollo local hacia un entorno de producción accesible para los usuarios finales. Por tanto, se documenta cada etapa con el fin de que sirva como guía para futuras implementaciones.

2. Descripción de la Aplicación

La aplicación Checklist fue diseñada con el objetivo de brindar una solución tecnológica al proceso manual de control de asistencia de los estudiantes universitarios. Tradicionalmente, este proceso depende del llamado de lista por parte del docente o del uso de planillas físicas. Checklist elimina este procedimiento obsoleto mediante la integración de tecnología web y lectura de carnés estudiantiles.

Funcionalidades principales:

- Registro automático de asistencia al escanear el código del carnet del estudiante.
- Panel de administración para consultar las asistencias por nombre, clase o fecha.
- Registro y autenticación de profesores mediante sistema seguro basado en JWT.
- Interfaz visual atractiva y amigable, adaptada tanto para docentes como para personal administrativo.

Tecnologías utilizadas:

- **Frontend:** HTML5, CSS3, Bootstrap 5, Handlebars (HBS), Google Fonts.
- **Backend:** Node.js con Express.js.
- **Base de datos:** MySQL.
- **Seguridad:** JSON Web Tokens (JWT), Bcrypt para el cifrado de contraseñas.
- **Otros:** Dotenv para gestión de variables de entorno, Cookie-parser para manejo de tokens.

3. Requisitos de Despliegue

El correcto funcionamiento de la aplicación en un entorno productivo requiere de ciertos elementos de hardware, software y configuraciones específicas. A continuación, se detallan los requisitos necesarios para su implementación.

Requisitos del servidor:

- Sistema operativo recomendado: **Linux (Ubuntu 20.04 o superior)**.
- Entorno de ejecución: **Node.js v16 o superior**.
- Memoria mínima: **512MB (ideal 1GB)**.
- Espacio en disco: **1GB mínimo para archivos de aplicación y logs**.

Software y configuraciones:

- Motor de base de datos: MySQL 5.7 o superior (puede ser remoto).
- Dependencias principales: Express, Bcryptjs, JWT, HBS, Dotenv, Cookie-parser.
- Configuración de variables de entorno: Se deben definir las siguientes variables:
- hostdb, usuariodb, passworddb, nombredb: datos de conexión a la base de datos.
- CLAVE: clave secreta usada para firmar los tokens JWT.

Compatibilidad:

- Render como plataforma de hosting es totalmente compatible con aplicaciones Node.js.
- Requiere que se definan correctamente los comandos de build (`npm install`) y de arranque (`node app.js`).

4. Presupuesto

Una de las ventajas de este proyecto es que fue desplegado utilizando herramientas y plataformas que ofrecen planes gratuitos para desarrolladores, lo que permite minimizar los costos sin comprometer la funcionalidad.

Detalles del presupuesto:

- **Plataforma de hosting:** Render (plan gratuito) – \$0.
- **Repositorio Git:** GitHub – \$0.

- **Base de datos:** MySQL alojado localmente o con opción gratuita en Railway/PlanetScale – \$0.
- **Dominio personalizado:** No se adquirió. Render proporciona una URL gratuita
- **Herramientas auxiliares:** Visual Studio Code, Node.js, Git, todos gratuitos.

5. Herramientas Utilizadas

Durante el desarrollo y despliegue de la aplicación se utilizaron diversas herramientas y entornos de trabajo. A continuación se listan las principales:

- Visual Studio Code (IDE): Editor de código fuente ampliamente utilizado para proyectos JavaScript.
- Git y GitHub: Sistema de control de versiones distribuido y plataforma para alojar repositorios y conectar con Render.
- Node.js: Entorno de ejecución del lado del servidor para JavaScript.
- Express.js: Framework para crear servidores web de forma rápida y modular.
- Handlebars (HBS): Motor de plantillas utilizado para renderizar páginas desde el servidor.
- MySQL: Sistema gestor de base de datos relacional.
- Render: Plataforma de despliegue y hosting para aplicaciones Node.js.
- Dotenv: Para gestionar credenciales sensibles desde un archivo .env.
- Cookie-parser y JSON Web Token: Para manejo seguro de autenticación y autorización.

6. Proceso de Despliegue

El despliegue se realizó en la plataforma **Render**, que permite publicar servicios web con conexión directa a un repositorio GitHub. A continuación se describe el proceso paso a paso:

1. Paso 1: Preparar el código

- Subir todo el proyecto a un repositorio privado o público en GitHub.
- Asegurarse de que `app.js` esté correctamente configurado como punto de entrada.
- Incluir un archivo `.env.example` que muestre las variables necesarias.

2. Paso 2: Crear cuenta en Render

- Ingresar a <https://render.com>
- Crear cuenta gratuita y vincular la cuenta de GitHub.

3. Paso 3: Crear un nuevo servicio web

- Seleccionar la opción “Web Service” y escoger el repositorio del proyecto.
- Elegir Node como entorno y configurar los siguientes parámetros:
 - **Build Command:** `npm install`
 - **Start Command:** `node app.js`
 - **Root Directory:** / (si todo está en la raíz)

4. Paso 4: Configurar variables de entorno

- Añadir las variables del archivo `.env` en el panel de Render para que el servidor pueda conectarse a la base de datos y firmar tokens.

5. Paso 5: Despliegue automático

- Render clona el repositorio, ejecuta la instalación y lanza la aplicación.
- Una vez desplegado, Render genera una URL pública del tipo:
`https://nombre-del-proyecto.onrender.com`

7. Consideraciones de Seguridad

La aplicación cuenta con múltiples medidas de seguridad para proteger tanto los datos de los usuarios como el acceso al sistema:

- Cifrado de contraseñas: Se utiliza `bcryptjs` para almacenar las contraseñas de forma segura en la base de datos.
- Autenticación basada en JWT: Cada sesión válida genera un token que se guarda en cookies HTTP Only, evitando accesos por JavaScript malicioso.
- Protección de rutas: Solo usuarios autenticados pueden acceder a páginas privadas como `/admin`.
- Validaciones del lado servidor: Se valida que los campos requeridos estén completos y se evita la duplicación de registros (por ejemplo, una cédula ya registrada).
- Uso de `.env`: Las credenciales sensibles como contraseñas de base de datos y claves secretas no se almacenan en el código fuente.
- Evita ataques comunes: Implementaciones simples de protección frente a inyecciones SQL (uso de placeholders en consultas) y XSS.

8. Conclusiones

El despliegue de la aplicación web Checklist representó un proceso enriquecedor y clave dentro del ciclo de desarrollo, ya que permitió trasladar el proyecto desde un entorno local de pruebas hacia una plataforma de producción accesible públicamente. Esta etapa no solo confirmó la viabilidad técnica del sistema desarrollado, sino que también consolidó la comprensión de herramientas modernas de despliegue como Render, la gestión de versiones mediante GitHub, y la importancia del uso de variables de entorno para proteger información sensible.

Uno de los logros más relevantes fue comprobar que, con los recursos disponibles en planes gratuitos, es posible montar una aplicación web funcional, segura y profesional. Se garantizaron aspectos esenciales como la autenticación de usuarios mediante JWT, el cifrado de contraseñas con bcrypt, y el uso de cookies seguras para el manejo de sesiones. Asimismo, se logró mantener una arquitectura modular y escalable que puede ser fácilmente adaptada o ampliada en el futuro, gracias a la correcta estructuración del backend con Express.js y la implementación de un patrón limpio en la lógica del servidor.

Durante el proceso, se identificaron oportunidades de mejora que pueden ser abordadas en versiones futuras del proyecto, como la migración de la base de datos local hacia un servicio gestionado en la nube, la incorporación de un dominio propio con certificado SSL para navegación segura (HTTPS), y la implementación de mejores prácticas de monitoreo y logging para facilitar el mantenimiento en producción.

En definitiva, el despliegue no solo validó los aspectos técnicos de la solución, sino que también reforzó el aprendizaje práctico de los estudiantes en cuanto a herramientas de desarrollo modernas, seguridad en aplicaciones web, automatización del flujo de integración continua y buenas prácticas de publicación de software. Este proceso evidencia que el trabajo colaborativo, bien planificado y con enfoque en la calidad, puede materializarse en aplicaciones web reales que resuelven necesidades concretas en contextos educativos.

9. Referencias

- Render Documentation – <https://render.com/docs>
- Node.js Official Docs – <https://nodejs.org>
- Express.js – <https://expressjs.com>
- JWT Official Site – <https://jwt.io>
- MySQL – <https://dev.mysql.com>
- Handlebars – <https://handlebarsjs.com>
- Bootstrap – <https://getbootstrap.com>