

DESTINIX

Sistema logístico de tours, consultas, y publicación de destinos turísticos

Destinix (D.X)

Manual Técnico 2025

**Tabla de contenido**

[**Introducción** 4](#_Toc200396211)

[1. Objetivos 5](#_Toc200396212)

[**1.1 Objetivos Específicos** 5](#_Toc200396213)

[2. Alcance 6](#_Toc200396214)

[2.1 Módulos Personas 6](#_Toc200396215)

[3. Requerimientos mínimos de Hardware 6](#_Toc200396216)

[**3.1 Requerimientos mínimos de Software** 7](#_Toc200396217)

[4. Recursos empleados durante el proceso de desarrollo 7](#_Toc200396218)

[**4.1 Frontend** 7](#_Toc200396219)

[**4.1.1 JavaScript** 8](#_Toc200396220)

[**4.1.2 React.JS** 8](#_Toc200396221)

[**4.1.3 Bootstrap** 9](#_Toc200396222)

[4.2 Backend 9](#_Toc200396223)

[**4.2.1 Node.js** 9](#_Toc200396224)

[**4.2.2 PHP** *(lenguaje principal de las APIs)* 10](#_Toc200396225)

[5. Base de Datos 10](#_Toc200396226)

[**5.1 XAMPP** 10](#_Toc200396227)

[6. Instalación del gestor de bases de datos (XAMPP) 11](#_Toc200396228)

[7. Instalación de Node.js 11](#_Toc200396229)

[8. Instalación de folders del aplicativo 11](#_Toc200396230)

[9. Casos de uso 12](#_Toc200396231)

[ *Ilustración 1 -UML Casos de uso* 12](#_Toc200396232)

[10. Contingencias y Soluciones en Destinix 13](#_Toc200396233)

[11. Modelo relacional 13](#_Toc200396234)

[ *Ilustración 2 -MER Modelo Relacional* 14](#_Toc200396235)

[12. Diccionario de datos 14](#_Toc200396236)

[Conclusión 15](#_Toc200396237)

## **Introducción**

Destinix es una plataforma web de gestión turística premium que centraliza y optimiza los servicios turísticos en Bogotá, ofreciendo a anunciantes (hoteles, restaurantes, atracciones) la posibilidad de promocionar sus establecimientos mediante un modelo de suscripción o pago por publicación. Funciona como un sistema logístico integral que permite a los usuarios finales acceder a información verificada y actualizada sobre destinos turísticos.

Desarrollada bajo metodologías ágiles, la plataforma utiliza tecnologías modernas como React.js para un frontend dinámico, Node.js con API REST para el backend, y una base de datos relacional (MySQL) normalizada, garantizando escalabilidad, rendimiento y seguridad (incluyendo encriptación de contraseñas).

Este manual técnico documenta la arquitectura, componentes clave y procesos de Destinix, proporcionando pautas para su implementación, personalización y mantenimiento. Está dirigido a desarrolladores, administradores de sistemas y equipos de soporte técnico.

Para más detalles, consultar los repositorios en GitHub y la documentación técnica asociada en los enlaces proporcionados.

# **1. Objetivos**

Proporcionar la documentación técnica completa para la instalación, configuración y comprensión de la arquitectura del aplicativo Destinix, garantizando una implementación exitosa en diferentes entornos.

## **1.1 Objetivos Específicos**

* **Definir requisitos técnicos:**
  + Especificar los mínimos requerimientos de hardware (ej: RAM, almacenamiento) y software (SO, versiones de Node.js, MySQL) para el despliegue.
* **Documentar el proceso de instalación:**
  + Guía paso a paso para configurar el backend (API REST), frontend (React) y base de datos (MySQL), incluyendo dependencias y variables de entorno.
* **Describir herramientas de desarrollo:**
  + Tecnologías clave: React.js, Node.js, Bootstrap, librerías de encriptación y herramientas auxiliares (Git, Postman).
* **Explicar la estructura técnica:**
  + Diagramas de arquitectura (BPMN, MER), flujos de datos, y documentación de endpoints API para integraciones.

# **2. Alcance**

El manual técnico de Destinix cubre los siguientes módulos y componentes:

* **Frontend:** Interfaz web responsive (React.js) para usuarios turistas y anunciantes.
* **Backend:** Servicios API REST (Node.js) para gestión de reservas, pagos y autenticación.
* **Base de datos:** Modelo relacional (MySQL) con scripts SQL y normalización (1FN-3FN).

**Seguridad:** Protocolos de encriptación y gestión de roles (administrador, anunciante, turista (usuario).

## **2.1 Módulos Personas**

* Administrador
* Anunciante
* Usuario

# **3. Requerimientos mínimos de Hardware**

* Procesador: Ryzen 5 2600 - 3,4 Ghz - 6 núcleos
* Memoria Ram(mínimo) :8.0GB
* Conexión a internet
* Espacio mínimo de disco 50GB

## **3.1 Requerimientos mínimos de Software**

* Sistema operativo: Sistema operativo: Windows 10/11 – Linux (Distribuciones) Ubuntu/Debian
* Navegador Web: Chrome/Opera/Firefox/Edge
* MySQL con servidor
* NodeJS

# **4. Recursos empleados durante el proceso de desarrollo**

**4.1 Frontend**

El Frontend es la parte del software con la que los usuarios interactúan directamente, incluyendo la interfaz gráfica, el diseño y la experiencia de usuario.

## **4.1.1 JavaScript**

JavaScript es el lenguaje de programación estándar para crear interactividad en páginas web. No requiere un entorno de ejecución adicional, ya que se ejecuta directamente en el navegador del cliente.

* **Ventajas:**
  + Permite crear páginas web dinámicas y eficientes.
  + Es un lenguaje ampliamente adoptado y fácil de aprender para los desarrolladores.
  + Compatible con todos los navegadores modernos.

### **4.1.2 React.JS**

ReactJS es una biblioteca de JavaScript utilizada para construir interfaces de usuario interactivas y reutilizables mediante componentes.

* **Ventajas:**
  + Facilita el desarrollo de interfaces complejas mediante componentes modulares.
  + Ofrece un rendimiento optimizado gracias al uso del Virtual DOM.
  + Gran comunidad de soporte y amplia documentación disponible.

### **4.1.3 Bootstrap**

Bootstrap es un framework de diseño frontend que proporciona componentes predefinidos y herramientas para crear interfaces de manera rápida y eficiente.

* **Ventajas:**
  + Acelera el desarrollo con plantillas y estilos predefinidos.
  + Garantiza compatibilidad con dispositivos móviles y desktop.
  + Integración sencilla con ReactJS para un diseño consistente.

# **4.2 Backend**

El backend de destinix es el núcleo de sistema, encargado de procesar la lógica, gestionar los datos y comunicarse con el frontend mediante las APIs seguras. Opera el servidor y maneja solicitudes de usuarios, autenticación.

## **4.2.1 Node.js**

Entorno de ejecución JavaScript para Destinix, utilizado para la creación o realización de APIs.

* **Ventajas:**
  + Amplio ecosistema de paquetes (npm), para integraciones como (ej: envío de emails, etc)

## **4.2.2 PHP** *(lenguaje principal de las APIs)*

Lenguaje en cual se utilizó para el desarrollo de las APIs de destinix

* **Ventajas:**
  + Amplia contabilidad con servidores web (ej: Apache)
  + Sintaxis clara y gran documentación disponible
  + Integración nativa con MySQL

# **5. Base de Datos**

## **5.1 XAMPP**

Es una distribución popular de software de código abierto que facilita el desarrollo web. Especialmente en entornos de prueba. Incluye un servidor web (Apache), un sistema de gestión de base de datos (MySQL), y los lenguajes de programación de PHP y Perl.

Sus principales ventajas son su facilidad de uso e instalación ofreciendo un paquete de todo en uno, sin necesidad de un servidor externo, es multiplataforma es decir funciona con Windows, Linux y macOS, por lo que se puede desarrollar en cualquier sistema operativo sin problemas también es ideal para desarrollo local y contiene una configuración flexible.

# **6. Instalación del gestor de bases de datos (XAMPP)**

El proceso de instalación se encuentra en un documento aparte, con el fin de no sobrecargar de contenido del presente manual técnico; enlace del documento: ‘[1.1 Manual de instalación - XAMPP.docx - Documentos de Google](https://docs.google.com/document/d/1A9hTyOT7E0G5mQvbNzx-plaIMFSsBBYB/edit)’.

# **7. Instalación de Node.js**

Node.js es un entorno de ejecución multiplataforma enfocado al lado del servidor (Backend). Permitiendo crear aplicaciones escalables y gestionar múltiples conexiones al mismo tiempo. [Node.js se utiliza para ejecutar código JavaScript en los servidores sin necesidad de un navegador web.](https://www.bing.com/ck/a?!&&p=8760398d0978a83b8415a3c62ae73a24406e302f127e80ca0a5bbd49aabe5254JmltdHM9MTc0ODk5NTIwMA&ptn=3&ver=2&hsh=4&fclid=38092487-ff62-6823-3016-3172fecd6999&psq=que+es+node%2cjs&u=a1aHR0cHM6Ly93d3cuaXRkby5jb20vYmxvZy9xdWUtZXMtbm9kZS1qcy15LXBhcmEtcXVlLXNpcnZlLw&ntb=1)

El proceso de instalación se encuentra en un documento aparte, con el fin de no sobrecargar de contenido el manual técnico; enlace del documento: ‘[1.2 Manual de instalación - NodeJS.docx - Documentos de Google](https://docs.google.com/document/d/1cDcEdBTHycOWBSjf8HCz48kGynROtazv/edit).

# **8. Instalación de folders del aplicativo**

En este paso se proporcionará los archivos del aplicativo Destinix comprimidos en formato .zip. A continuación, se detalla cómo desplegarlos correctamente, asegurándose de tener previamente instalados Node.js y MySQL, como se indicó en las secciones anteriores.

🔹 *Importante:*

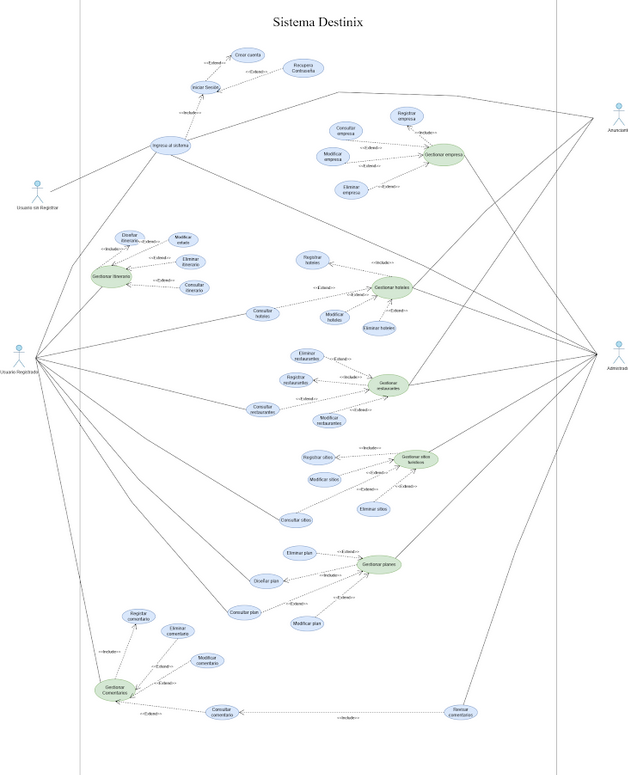
El proceso de instalación completa se documenta en un manual aparte ("Manual de Instalación de Destinix") para mantener este documento técnico claro y conciso. Puede acceder a la guía detallada en el siguiente enlace:

# **9. Casos de uso**

Los casos de uso extendidos se documentaron en un archivo independiente para facilitar su consulta y mantener este manual técnico conciso.

Acceder al siguiente link: <https://drive.google.com/drive/u/1/folders/1DIByH9jecWJnYtj1XXEhuy0CRx_XaOgK>

## *Ilustración 1 -UML Casos de uso*



# **10. Contingencias y Soluciones en Destinix**

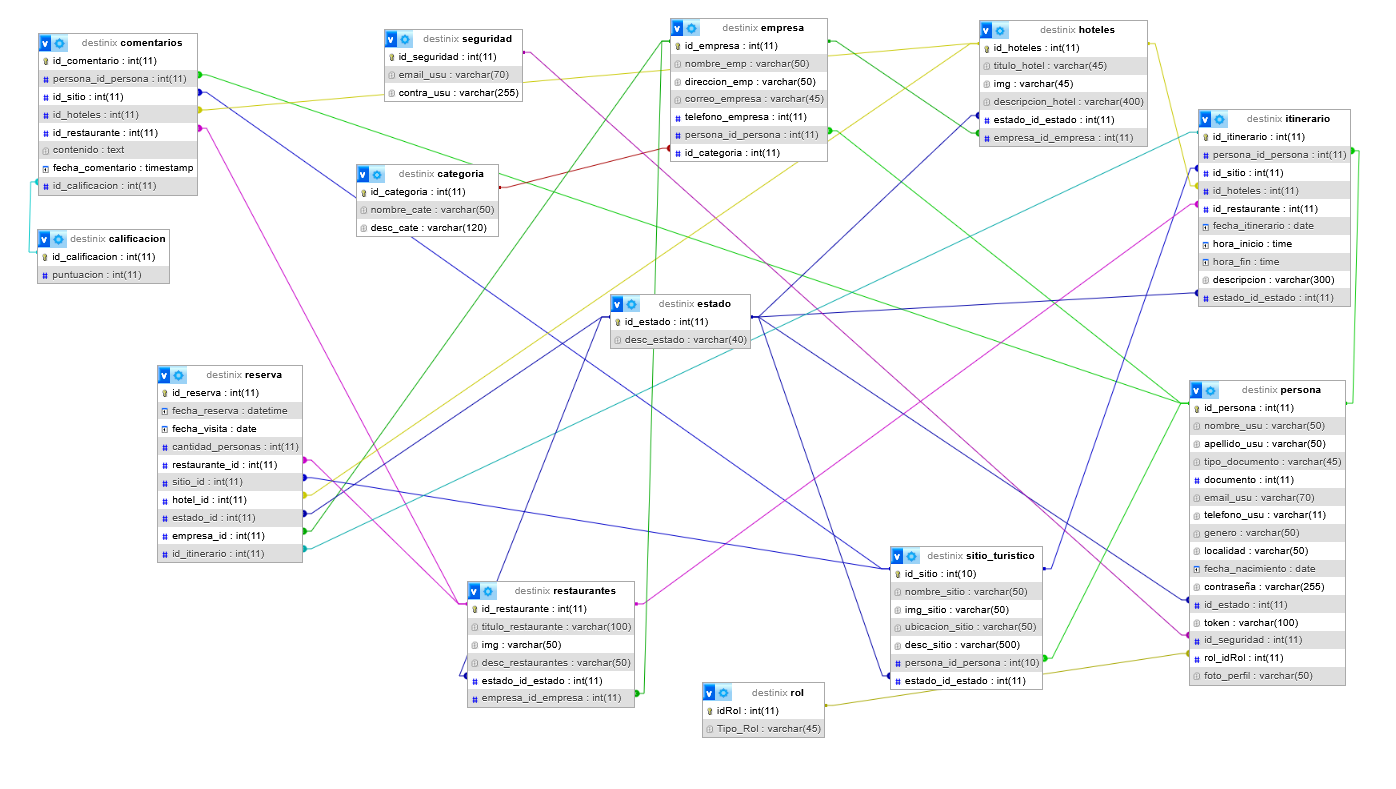
Durante la implementación local de Destinix, podrían presentarse los siguientes errores al configurar los entornos (Frontend y Backend):

* XAMPP no instalado o configurado incorrectamente: Verifique que los servicios de Apache y MySQL estén activos en el panel de control de XAMPP, ya que este paquete es esencial para el servidor local y la gestión de la base de datos. Consulte el Manual de Instalación de XAMPP para guías detalladas.
* Node.js no instalado: Confirme su instalación ejecutando node -v en la terminal. Este runtime es crítico para ejecutar el backend (API) y gestionar dependencias. Si falta, siga el Tutorial de Instalación de Node,js.
* **Dependencias no instaladas**: Asegúrese de que la carpeta node\_modules exista en el directorio del proyecto. Si no está presente, ejecute npm install para restaurar los paquetes requeridos.
* Si el problema persiste tras validar estos puntos, contacte al **equipo de soporte técnico** de Destinix mediante.

# **11. Modelo relacional**

El modelo relacional de Destinix representa la estructura lógica de la base de datos mediante tablas interconectadas que garantizan la integridad y eficiencia en el manejo de datos.

## *Ilustración 2 -MER Modelo Relacional*



# **12. Diccionario de datos**

El Diccionario de Datos de Destinix documenta de manera estructurada todas las entidades (tablas), atributos (campos), relaciones y restricciones que componen la base de datos del sistema. Este recurso es indispensable para desarrolladores, administradores de bases de datos y auditores, ya que proporciona una referencia técnica detallada para el mantenimiento, escalabilidad y validación de la integridad de los datos.

Acceder al siguiente link: [Diccionario de datos - Google Drive](https://drive.google.com/drive/u/1/folders/1daHQ0a1Nh9smZe-v6V2swL37vuPnEbH8)

**Conclusión**

Destinix se consolida como una plataforma turística innovadora diseñada para conectar a anunciantes (hoteles, restaurantes y sitios) con viajeros, facilitando la promoción de destinos y la consulta de información turística verificada. A diferencia de sistemas tradicionales, Destinix se enfoca en la **publicación y descubrimiento de experiencias turísticas** mediante un modelo flexible de suscripción o pago por publicación, sin incluir módulos de reservas, lo que simplifica su operación y reduce complejidades técnicas.

Desarrollado con tecnologías modernas como **React.js** para una interfaz dinámica, **Node.js** y **PHP** para APIs robustas, y una base de datos relacional en **MySQL**, el sistema garantiza escalabilidad, rendimiento y seguridad. Su arquitectura modular, respaldada por herramientas como **XAMPP** para entornos locales y **Bootstrap** para diseño responsive, asegura una experiencia intuitiva tanto para turistas como para anunciantes.

La documentación técnica, que incluye modelos de datos, casos de uso y guías de instalación detalladas, refleja el compromiso con la transparencia y la facilidad de implementación. **Destinix** no solo satisface las necesidades actuales del mercado turístico en Bogotá, sino que también ofrece una solución adaptable y centrada en la **promoción eficiente de destinos**, potenciando la visibilidad de los negocios locales y enriqueciendo la planificación de viajes para los usuarios finales.

Con su enfoque claro en la **publicación y consulta de información turística**, Destinix se posiciona como una herramienta esencial para dinamizar el sector, sin las limitaciones operativas de sistemas de reserva, marcando así una diferencia clave en su categoría.