

Checkpoint Creado y Sistema Listo para Deploy

Estado del Checkpoint

Fecha: 1 de Octubre, 2025

Checkpoint ID: Sistema completo 9 fases listo producción

Estado:  CREADO EXITOSAMENTE

Build Status:  Exitoso (67 páginas generadas)



Compilación TypeScript:  Sin errores

APIs:  70+ endpoints funcionales

¿Qué Incluye Este Checkpoint?

Este checkpoint representa la **versión 9.0 completa** del Hotel PMS “Paseo Las Mercedes” con:

9 Fases Completadas:

1.  Foundation & Configuration
2.  Room Management
3.  Reservation System
4.  Check-in/Check-out Process
5.  Reporting & Analytics
6.  Payment & Billing System
7.  Housekeeping Module
8.  Staff Management System
9.  Inventory & Procurement System

Estadísticas:

- **67 páginas** generadas en build de producción
 - **70+ APIs** RESTful funcionales
 - **35+ modelos** de base de datos (Prisma)
 - **40,000+ líneas** de código TypeScript
 - **100+ componentes** React reutilizables
-

Documentación de Deploy Creada

Se han generado los siguientes documentos para facilitar el deployment:

1. **GUIA_DEPLOYMENT_PRODUCCION.md**

Contenido:

- Opciones de deployment (Vercel, Railway, AWS, DigitalOcean)
- Pasos detallados para cada plataforma

- Configuración de base de datos
- Configuración de SSL/HTTPS
- CI/CD con GitHub Actions
- Monitoreo y logs
- Troubleshooting común
- Checklist completo pre-deployment

Tamaño: ~800 líneas de documentación completa

2. CHECKPOINT_RESUMEN.md

Contenido:

- Resumen ejecutivo del checkpoint
- Estadísticas del sistema
- Estructura del proyecto completa
- Comandos útiles
- Checklist de verificación
- Próximas fases sugeridas
- Métricas de calidad

Tamaño: ~600 líneas

3. VARIABLES_ENTORNO_PRODUCCION.md

Contenido:

- Todas las variables de entorno necesarias
- Guías para obtener credenciales (Stripe, AWS, etc.)
- Ejemplos por plataforma de hosting
- Scripts de verificación
- Reglas de seguridad
- Checklist de configuración

Tamaño: ~500 líneas

4. PROJECT_SUMMARY.md (Actualizado)

Contenido:

- Resumen general actualizado
- Estado actual del sistema
- Stack tecnológico
- Arquitectura completa



Próximos Pasos para Deploy

Opción 1: Deploy Rápido en Vercel (Recomendado) ⚡

Tiempo estimado: 15-20 minutos

```
# 1. Subir a GitHub
cd /home/ubuntu/hotel_pms_paseo_las_mercedes
git init
git add .
git commit -m "v9.0 production ready"
git remote add origin https://github.com/tu-usuario/hotel-pms.git
git push -u origin main

# 2. Ir a Vercel
# - https://vercel.com
# - New Project
# - Import from GitHub
# - Configurar root directory: app
# - Agregar variables de entorno
# - Deploy!
```

Variables de entorno mínimas necesarias:

- DATABASE_URL
- NEXTAUTH_URL
- NEXTAUTH_SECRET
- STRIPE_SECRET_KEY
- STRIPE_PUBLISHABLE_KEY
- STRIPE_WEBHOOK_SECRET

Ver `VARIABLES_ENTORNO_PRODUCCION.md` para detalles completos.

Opción 2: Deploy en Railway 🚂

Tiempo estimado: 20-25 minutos

Ventaja: Incluye base de datos PostgreSQL automáticamente

1. Ir a <https://railway.app>
 2. New Project → Deploy from GitHub
 3. Agregar PostgreSQL database (Railway lo conecta automáticamente)
 4. Configurar variables de entorno
 5. Deploy automático
-

Opción 3: Deploy en AWS (Avanzado) ☁️

Tiempo estimado: 1-2 horas

Para hoteles grandes o que necesitan control total. Incluye:

- EC2 para servidor
- RDS para PostgreSQL
- S3 para assets
- CloudFront CDN
- Route53 para DNS

Ver `GUIA_DEPLOYMENT_PRODUCCION.md` sección AWS para pasos detallados.

Checklist Pre-Deploy

Antes de deployar, asegúrate de tener:

Base de Datos:

- ☐ Base de datos PostgreSQL creada (o usar la de desarrollo)
- ☐ DATABASE_URL obtenido
- ☐ Acceso verificado con `psql $DATABASE_URL`

Stripe (Pagos):

- ☐ Cuenta de Stripe creada
- ☐ Claves de producción obtenidas (sk_live_... y pk_live_...)
- ☐ Webhook configurado en Stripe Dashboard
- ☐ Webhook secret obtenido

Dominio (Opcional pero recomendado):

- ☐ Dominio registrado (ej: hotel-pms.com)
- ☐ DNS configurado (o usar subdominio de Vercel/Railway)

Repositorio:

- ☐ Código subido a GitHub/GitLab
- ☐ Branch `main` lista para deploy
- ☐ `.env` NO incluido en el repositorio

Variables de Entorno Críticas

Generar NEXTAUTH_SECRET:

```
openssl rand -base64 32
```

Template Mínimo:

```
# Database
DATABASE_URL="postgresql://usuario:password@host:5432/database"

# NextAuth
NEXTAUTH_URL="https://tu-dominio.com"
NEXTAUTH_SECRET="[generar-con-comando-arriba]"

# Stripe (Live Keys)
STRIPE_SECRET_KEY="sk_live_..."
STRIPE_PUBLISHABLE_KEY="pk_live_..."
STRIPE_WEBHOOK_SECRET="whsec_..."
```

Ver `VARIABLES_ENTORNO_PRODUCCION.md` para lista completa y guías.

Verificación Post-Deploy

Después de deployar, verificar:

1. Health Checks Básicos:

```
# Homepage
curl https://tu-dominio.com

# Auth providers
curl https://tu-dominio.com/api/auth/providers

# Dashboard (requiere login)
curl https://tu-dominio.com/dashboard
```

2. Test Manual en Browser:

- [] Página de login carga correctamente
- [] Login con credenciales funciona
- [] Dashboard muestra métricas
- [] Navegación entre páginas funciona
- [] No hay errores en consola del browser

3. Test de Funcionalidades Core:

- [] Crear nueva reservación
- [] Hacer check-in
- [] Generar factura
- [] Procesar pago con Stripe (test mode primero)
- [] Ver reportes

Recursos y Links Útiles

Documentación Generada:

- `GUIA_DEPLOYMENT_PRODUCCION.md` - Guía completa de deploy
- `CHECKPOINT_RESUMEN.md` - Resumen del checkpoint
- `VARIABLES_ENTORNO_PRODUCCION.md` - Variables de entorno
- `PROJECT_SUMMARY.md` - Resumen del proyecto

Documentación Externa:

- [Next.js Deployment](https://nextjs.org/docs/deployment) (<https://nextjs.org/docs/deployment>)
- [Vercel Docs](https://vercel.com/docs) (<https://vercel.com/docs>)
- [Railway Docs](https://docs.railway.app) (<https://docs.railway.app>)
- [Prisma Deployment](https://www.prisma.io/docs/guides/deployment) (<https://www.prisma.io/docs/guides/deployment>)
- [Stripe Integration](https://stripe.com/docs/payments/quickstart) (<https://stripe.com/docs/payments/quickstart>)

Comandos Útiles:

```
# Build local
cd app && yarn build

# Iniciar en producción
yarn start

# Aplicar schema de DB
npx prisma db push

# Ver base de datos
npx prisma studio

# Generar Prisma client
npx prisma generate
```

Recomendaciones Finales

Para Hoteles Pequeños/Medianos:

☀️ **Recomendado:** Vercel + Neon/Supabase PostgreSQL

- ✅ Fácil y rápido
- ✅ SSL gratis
- ✅ Deploy automático
- ✅ Escalamiento incluido
- 💰 Plan gratuito disponible para desarrollo

Para Hoteles Grandes/Cadenas:

☀️ **Recomendado:** AWS o Google Cloud

- ✅ Control total
- ✅ Alta disponibilidad
- ✅ Escalamiento avanzado
- ✅ Cumplimiento regulatorio
- 💰 Costos basados en uso

Para Presupuesto Limitado:

☀️ **Recomendado:** Railway

- ✅ Base de datos incluida
 - ✅ Muy económico
 - ✅ Fácil de usar
 - 💰 \$5/mes aprox
-

Soporte y Ayuda

Si Encuentras Problemas:

1. Revisar documentación:

- Leer `GUIA_DEPLOYMENT_PRODUCCION.md` sección Troubleshooting
- Verificar `VARIABLES_ENTORNO_PRODUCCION.md`

2. Verificar variables de entorno:

```
bash
# Ejecutar script de verificación
cd app
node scripts/verify-env.js
```

3. Revisar logs:

- En Vercel: Dashboard → Logs
- En Railway: Service → Logs
- Local: `yarn dev` para ver errores




4. Errores comunes:

- Database connection → Verificar `DATABASE_URL`
- NextAuth error → Verificar `NEXTAUTH_URL` y `NEXTAUTH_SECRET`
- Stripe error → Verificar claves y webhook





Métricas de Éxito

Después del deploy, deberías ver:





Performance:

-  Tiempo de carga inicial < 3 segundos
-  Time to Interactive < 5 segundos
-  API response time < 500ms

Funcionalidad:







-  Todas las páginas cargan sin error 500
-  Login/logout funciona correctamente
-  CRUD operations funcionan
-  Pagos de Stripe procesan correctamente

Seguridad:







-  HTTPS habilitado
 -  Headers de seguridad configurados
 -  Credenciales de test eliminadas
 -  `.env` no expuesto públicamente
-

¡Felicidades!

Has completado la implementación de un **sistema PMS profesional completo** con:

-  9 fases de desarrollo implementadas
-  67 páginas funcionales
-  70+ APIs RESTful
-  Sistema de pagos integrado
-  Gestión completa de hotel
-  Listo para producción

El sistema está preparado para:

-  Gestionar hoteles de tamaño pequeño a grande
-  Manejar operaciones diarias completas
-  Procesar pagos de forma segura
-  Generar reportes y analytics
-  Gestionar personal multi-departamento
-  Controlar inventario y compras

¡Adelante con el Deploy!

Siguiente paso recomendado:

1. Leer `GUIA_DEPLOYMENT_PRODUCCION.md` (especialmente la sección de tu plataforma elegida)
 2. Preparar variables de entorno usando `VARIABLES_ENTORNO_PRODUCCION.md`
 3. Seguir checklist de este documento
 4. Deployar!
-

Sistema: Hotel PMS Paseo Las Mercedes v9.0

Checkpoint: Sistema completo 9 fases listo producción

Fecha: 1 de Octubre, 2025

Estado:  PRODUCTION READY

¡Buena suerte con tu deploy! 