

# Cómo Usar el Plan de Pruebas - Guía Práctica






---

## ¿Cuándo Usar Cada Documento?

---






### **PLAN\_DE\_PRUEBAS.md (Completo - 45-60 min)**

#### Usar cuando:

-  Antes del despliegue a producción
-  Después de cambios importantes en el código
-  Pruebas de regresión completas
-  Validación de nuevas funcionalidades
-  Después de actualizar dependencias





### **CHECKLIST\_PRUEBAS\_RAPIDAS.md (Rápido - 15 min)**

#### Usar cuando:

-  Cambios menores en código
-  Fixes de bugs pequeños
-  Verificación rápida antes de commit
-  Testing de funcionalidades específicas
-  Validación después de actualizaciones menores

### **SCRIPTS\_DATOS\_PRUEBA.md (Preparación)**

#### Usar cuando:



-  Necesites datos consistentes para testing
-  Base de datos esté vacía
-  Quieras resetear datos de prueba
-  Simular diferentes escenarios

---



## Flujo Recomendado de Pruebas

---


### ESCENARIO 1: Desarrollo Diario

1. Hacer cambios en código
2. CHECKLIST\_PRUEBAS\_RAPIDAS.md (15 min)
3. Si pasa  Commit
4. Si falla  Corregir y repetir

### ESCENARIO 2: Release/Deploy

1. Ejecutar SCRIPTS\_DATOS\_PRUEBA.md (setup)
2. Ejecutar PLAN\_DE\_PRUEBAS.md completo (60 min)
3. Documentar resultados
4. Si **todo** pasa  Deploy
5. Si algo falla  Fix y repetir desde paso 2

## ESCENARIO 3: Hotfix en Producción

1. CHECKLIST\_PRUEBAS\_RAPIDAS.md
2. Probar específicamente la funcionalidad afectada
3. Si pasa  Deploy inmediato
4. Ejecutar plan completo en siguiente ventana



## Preparación del Entorno de Pruebas

### Paso 1: Base de Datos

```
# Opción A: Base de datos limpia
npx prisma db push --force-reset
npx prisma db seed

# Opción B: Agregar datos de prueba a BD existente
# Ejecutar scripts del SCRIPTS_DATOS_PRUEBA.md
```

### Paso 2: Variables de Entorno

Verificar que estén configuradas:

```
DATABASE_URL=...
NEXTAUTH_SECRET=...
SMTP_USER=...
SMTP_PASS=...
```


### Paso 3: Aplicación

```
yarn dev
# Verificar que carga en http://localhost:3000
```



## Cómo Documentar Resultados

### Durante las Pruebas:

1. **Marcar cada test:** ☐ → ☒ o ☐ → 
2. **Screenshot de errores:** Guardar en carpeta `pruebas/screenshots/`
3. **Tomar tiempos:** Especialmente para pruebas de rendimiento
4. **Anotar observaciones:** Comportamientos extraños o mejoras

## Formato de Reporte:

### REPORTE DE PRUEBAS

Fecha: 2024-08-17

Ejecutado por: [Tu nombre]

Tipo: [Completo/Rápido]

### RESULTADOS:

- ✓ Autenticación: PASS
- ✗ Gestión Empleados: FAIL - Error en creación
- ✓ Solicitudes: PASS
- ...

### ERRORES CRÍTICOS:

- No se pueden crear empleados nuevos
- Error 500 en formulario

### OBSERVACIONES:

- Dashboard carga lento (5 segundos)
- Emails se demoran 30 segundos

ESTADO FINAL: ✗ REQUIERE CORRECCIONES

---

## Criterios de Aprobación

### Para Deploy a Producción:

- ✓ **100% de casos críticos** deben pasar
- ✓ **95% de casos totales** deben pasar
- ✓ **Sin errores 500** en flujos principales
- ✓ **Emails funcionando** correctamente
- ✓ **Login/logout** sin problemas

### Casos Críticos (NO pueden fallar):

1. Login admin y empleado
2. Crear solicitudes
3. Aprobar/rechazar solicitudes
4. Envío de emails
5. Creación de empleados
6. Dashboard principal carga

---

## Automatización (Futuro)

### Próximos Pasos:

1. **Cypress/Playwright:** Para automatizar pruebas de UI
2. **Jest:** Para pruebas unitarias
3. **API Testing:** Con Postman/Newman
4. **CI/CD:** Ejecutar pruebas automáticas en GitHub Actions

## Scripts de Automatización:

```
# package.json
{
  "scripts": {
    "test:quick": "run checklist rápido",
    "test:full": "run plan completo",
    "test:setup": "run scripts de datos",
    "test:api": "run tests de API"
  }
}
```

## Qué Hacer Cuando Algo Falla

### 1. Error en Autenticación

- Verificar NEXTAUTH\_SECRET
- Verificar DATABASE\_URL
- Revisar logs del servidor

### 2. Error en Emails

- Verificar SMTP\_USER y SMTP\_PASS
- Verificar conexión a Gmail
- Revisar App Passwords de Google

### 3. Error en Base de Datos

- Verificar conexión PostgreSQL
- Ejecutar `npx prisma db push`
- Verificar datos con `npx prisma studio`

### 4. Error de UI/Frontend

- Verificar console del navegador
- Revisar Network tab para APIs fallidas
- Verificar responsive design



## Métricas de Calidad

### Seguimiento Mensual:

- **Tiempo promedio de pruebas:** \_\_\_\_ min
- **% de casos que pasan primero:** \_\_\_\_ %
- **Errores encontrados por deploy:** \_\_\_\_
- **Tiempo de corrección promedio:** \_\_\_\_ horas

### Objetivos:

- Pruebas rápidas < 15 minutos
- Pruebas completas < 60 minutos

- 95% casos exitosos antes de deploy
  - 0 errores críticos en producción
- 

## Checklist de Implementación

---

Para implementar este sistema en tu equipo:

- ☐ **Equipo capacitado** en el uso de los documentos
  - ☐ **Entorno de testing** configurado
  - ☐ **Datos de prueba** disponibles
  - ☐ **Proceso documentado** en wiki del equipo
  - ☐ **Responsables asignados** para ejecución
  - ☐ **Calendario de testing** establecido
  - ☐ **Tools de reporte** definidas
- 

## Tips y Mejores Prácticas

---

### Durante las Pruebas:

1. **No saltar pasos** aunque parezcan obvios
2. **Documentar TODO**, incluso los pasos
3. **Probar con datos reales** ocasionalmente
4. **Variar navegadores** entre pruebas
5. **Simular usuarios reales** (no solo casos perfectos)

### Gestión de Errores:

1. **Crear ticket/issue** para cada error
2. **Priorizar por severidad** (crítico/alto/medio/bajo)
3. **Asignar responsables** para cada fix
4. **Retesting obligatorio** después de fixes

### Mejora Continua:

1. **Actualizar plan** después de cada release
  2. **Agregar nuevos casos** cuando encuentres bugs
  3. **Optimizar tiempos** de ejecución
  4. **Solicitar feedback** del equipo
- 

 **Recuerda: Las pruebas no son un obstáculo, son una herramienta de calidad y confianza en tu aplicación.**

---

## Resumen de Archivos Creados

Archivo	Propósito	Tiempo	Uso
PLAN_DE_PRUEBAS.md	Plan completo	60 min	Pre-deploy
CHECK-LIST_PRUEBAS_RAPIDAS.md	Verificación rápida	15 min	Desarrollo
SCRIPTS_DATOS_PRUEBA.md	Preparación de datos	5 min	Setup
GUIA_COMO_USAR_PLAN_PRUEBAS.md	Esta guía	-	Referencia

¡Tu sistema de pruebas está listo! 🚀