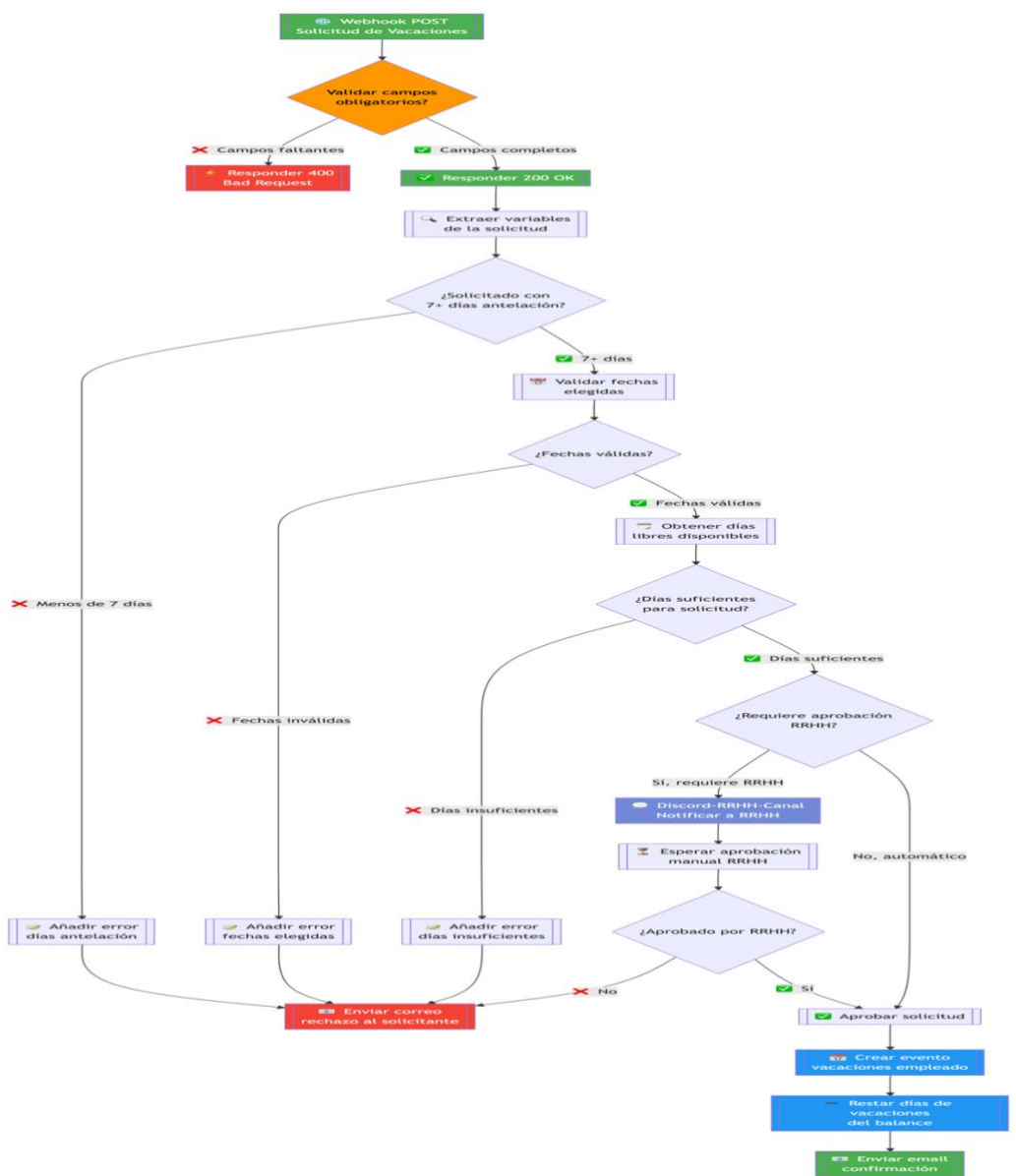


Documentación del Sistema de Automatización de Solicitudes de Vacaciones

Workflow Completo de Validación y Aprobación de Días Libres

🔍 Diagrama de Flujo del Sistema de Vacaciones



Módulos del Sistema de Vacaciones

1. Trigger: Webhook POST - Solicitud Vacaciones

- **Endpoint:** POST /api/vacaciones/solicitud
- **Payload esperado:**

json

```
{  
  "empleado_id": "EMP-12345",  
  "empleado_email": "ana.garcia@empresa.com",  
  "fecha_inicio": "2025-07-15",  
  "fecha_fin": "2025-07-22",  
  "dias_solicitados": 6,  
  "tipo_vacaciones": "ordinarias",  
  "motivo": "Descanso anual",  
  "jefe_inmediato": "jefe.rodriguez@empresa.com"  
}
```

- **Autenticación:** API Key o JWT Token

2. Validación de Datos del Webhook

Verificación de campos obligatorios:

javascript

```
const camposRequeridos = [  
  'empleado_id',  
  'empleado_email',  
  'fecha_inicio',  
  'fecha_fin',  
  'dias_solicitados'  
];
```

// Validación

```

for (let campo of camposRequeridos) {
  if (!payload[campo]) {
    return responderError(400, `Campo requerido faltante: ${campo}`);
  }
}

```

3. Extracción de Variables

```

javascript
const solicitud = {
  empleado: payload.empleado_id,
  email: payload.empleado_email,
  inicio: new Date(payload.fecha_inicio),
  fin: new Date(payload.fecha_fin),
  dias: parseInt(payload.dias_solicitados),
  // Cálculo días hábiles
  dias_habiles: calcularDiasHabiles(inicio, fin)
};

```

4. Validaciones de Política Empresarial

Regla 1: Antelación Mínima (7 días)

```

javascript
const hoy = new Date();
const antelacionMinima = 7; // días
const diasAntelacion = diferenciaDias(hoy, solicitud.inicio);

if (diasAntelacion < antelacionMinima) {
  registrarError('ANTELACION_INSUFICIENTE', {
    dias_necesarios: antelacionMinima,
    dias_proporcionados: diasAntelacion
  });
}

```

Regla 2: Validación de Fechas

javascript

// Fechas válidas

```
if (solicitud.inicio >= solicitud.fin) {  
  registrarError('FECHAS_INVERTIDAS');  
}
```

// No fines de semana

```
if (esFinDeSemana(solicitud.inicio) || esFinDeSemana(solicitud.fin)) {  
  registrarError('FECHAS_FIN_SEMANA');  
}
```

// No días festivos

```
const festivos = obtenerFiestasNacionales();  
if (festivos.some(f => esMismaFecha(f, solicitud.inicio))) {  
  registrarError('FECHA_FESTIVO');  
}
```

Regla 3: Días Disponibles

javascript

```
async function verificarDiasDisponibles(empleadold) {  
  const query = `  
    SELECT  
      dias_totales,  
      dias_tomados,  
      dias_pendientes,  
      dias_disponibles  
    FROM balances_vacaciones  
    WHERE empleado_id = $1  
  `;  
  
  const resultado = await db.query(query, [empleadold]);  
  
  if (!resultado.rows[0]) {  
    registrarError('EMPLEADO_NO_ENCONTRADO');  
    return false;  
  }  
}
```

```

const balance = resultado.rows[0];

if (solicitud.dias > balance.dias_disponibles) {
  registrarError('DIAS_INSUFICIENTES', {
    solicitados: solicitud.dias,
    disponibles: balance.dias_disponibles
  });
  return false;
}

return true;
}

```

5. Flujos de Aprobación

Aprobación Automática (Condiciones):

javascript

```

const aprobacionAutomatica = (
  solicitud.dias <= 3 &&           // Máximo 3 días
  diasAntelacion >= 14 &&         // Con 14+ días antelación
  balance.dias_disponibles >= 10 && // Suficientes días disponibles
  !empleadoEsNuevo(empleadoId)    // No es empleado nuevo (< 6 meses)
);

```

Aprobación Manual por RRHH:

javascript

```

if (!aprobacionAutomatica) {
  // Notificar a RRHH vía Discord
  await enviarNotificacionDiscord({
    canal: 'rrrh-vacaciones',
    mensaje: `📄 Nueva solicitud requiere revisión:

```

- **Empleado:** `${solicitud.empleado}`
- **Período:** `${solicitud.inicio}` al `${solicitud.fin}`
- **Días:** `${solicitud.dias}`
- **Enlace:** `${enlaceAprobacion}`

```
});

// Esperar respuesta RRHH (timeout: 72h)
const aprobado = await esperarAprobacionRRHH(72);
}
```

6. Ejecución de Acciones

Si Aprobado:

1. Crear Evento Calendario:

javascript

```
await calendario.crearEvento({
  titulo: `Vacaciones - ${empleadoNombre}`,
  inicio: solicitud.inicio,
  fin: solicitud.fin,
  descripcion: `Vacaciones aprobadas. Contacto: ${solicitud.email}`
});
```

2. Actualizar Balance:

javascript

```
await db.query(`
  UPDATE balances_vacaciones
  SET dias_tomados = dias_tomados + $1,
      dias_disponibles = dias_disponibles - $1,
      ultima_actualizacion = NOW()
  WHERE empleado_id = $2
`, [solicitud.dias, empleadoid]);
```

3. Enviar Confirmación:

javascript

```
await email.enviar({
```

```
    to: solicitud.email,  
    subject: '✅ Vacaciones Aprobadas',  
    body: plantillas.vacacionesAprobadas(solicitud)  
  });
```

Si Rechazado:

```
javascript  
await email.enviar({  
  to: solicitud.email,  
  subject: '❌ Solicitud de Vacaciones Rechazada',  
  body: plantillas.vacacionesRechazadas(solicitud, errores)  
});
```

Estructura de Base de Datos

Tabla: solicitudes_vacaciones

```
sql  
CREATE TABLE solicitudes_vacaciones (  
  id SERIAL PRIMARY KEY,  
  empleado_id VARCHAR(50) NOT NULL,  
  fecha_solicitud TIMESTAMP DEFAULT NOW(),  
  fecha_inicio DATE NOT NULL,  
  fecha_fin DATE NOT NULL,  
  dias_solicitados INTEGER NOT NULL,  
  estado VARCHAR(20) DEFAULT 'pendiente', -- 'pendiente', 'aprobada', 'rechazada'  
  aprobador_id VARCHAR(50),  
  fecha_aprobacion TIMESTAMP,  
  motivo_rechazo TEXT,  
  errores JSONB,  
  metadata JSONB  
);
```

Tabla: balances_vacaciones

sql

```
CREATE TABLE balances_vacaciones (  
  empleado_id VARCHAR(50) PRIMARY KEY,  
  dias_totales INTEGER DEFAULT 22, -- Según política  
  dias_tomados INTEGER DEFAULT 0,  
  dias_pendientes INTEGER DEFAULT 0,  
  dias_disponibles INTEGER GENERATED ALWAYS AS (dias_totales - dias_tomados - dias_pendientes) STORED,  
  año INTEGER DEFAULT EXTRACT(YEAR FROM NOW()),  
  ultima_actualizacion TIMESTAMP DEFAULT NOW()  
);
```

💡 Reglas de Negocio Implementadas

Políticas Empresariales:

1. **Antelación mínima:** 7 días hábiles
2. **Aprobación automática:** ≤3 días, con 14+ días antelación
3. **Máximo concurrente:** Solo 1 empleado por departamento (excepto emergencias)
4. **Periodo de verano:** Límite del 30% del equipo simultáneamente
5. **Empleados nuevos:** Solo después de 6 meses (prorrateado)

Validaciones Adicionales:

javascript

// No solapamiento con otras solicitudes

```
async function verificarSolapamientos(empleadoId, inicio, fin) {  
  const solapamientos = await db.query(`  
    SELECT COUNT(*) as count  
    FROM solicitudes_vacaciones  
    WHERE empleado_id = $1  
    AND estado = 'aprobada'  
  `);
```



```

AND (
    (fecha_inicio BETWEEN $2 AND $3) OR
    (fecha_fin BETWEEN $2 AND $3) OR
    (fecha_inicio <= $2 AND fecha_fin >= $3)
)
`, [empleadold, inicio, fin]);

return solapamientos.rows[0].count > 0;
}

```

Integraciones del Sistema

1. Discord para RRHH

- **Canal:** #rrrh-vacaciones
- **Formato mensajes:**

```

text

📄 NUEVA SOLICITUD - #${id}
👤 ${empleado} (${departamento})
📅 ${fecha_inicio} → ${fecha_fin} (${dias} días)
⚠️ ${motivo_revision}
✅ /rechazar ${id} [motivo]

```
- **Webhook:** Respuestas automáticas actualizan estado

2. Google Calendar

- **Calendario compartido:** "Vacaciones Empresa"
- **Colores por departamento:**
 - Desarrollo: Azul
 - Ventas: Verde
 - Soporte: Naranja
 - Dirección: Rojo
- **Visibilidad:** Público interno

3. Sistema de Emails

- **Plantillas configurables:**
 - vacaciones_aprobadas.html
 - vacaciones_rechazadas.html
 - recordatorio_jefe.html
 - notificacion_rrhh.html

4. Base de Datos

- **PostgreSQL** con extensiones JSONB
- **Índices optimizados:**

sql

```
CREATE INDEX idx_vacaciones_empleado ON solicitudes_vacaciones(empleado_id, estado);
```

```
CREATE INDEX idx_vacaciones_fecha ON solicitudes_vacaciones(fecha_inicio, fecha_fin);
```

Métricas y Reporting

Dashboard de Vacaciones:

sql

-- Uso por departamento

SELECT

d.nombre as departamento,

COUNT(*) as solicitudes,

SUM(s.dias_solicitados) as dias_totales,

AVG(s.dias_solicitados) as promedio_dias,

ROUND(100.0 * COUNT(*) FILTER (WHERE s.estado = 'aprobada') / COUNT(*), 2) as tasa_a

probacion

FROM solicitudes_vacaciones s

JOIN empleados e ON s.empleado_id = e.id

JOIN departamentos d ON e.departamento_id = d.id

GROUP BY d.nombre;

Alertas Automáticas:

- **Pico de solicitudes:** > 10 solicitudes mismo día
 - **Déficit departamental:** < 2 empleados disponibles
 - **Patrones anómalos:** Múltiples solicitudes cortas
 - **Cumplimiento legal:** Reportes mensuales automáticos
-



Seguridad y Cumplimiento

Protección de Datos:

1. **GDPR Compliance:** Solo datos necesarios
2. **Encriptación:** SSL/TLS + encriptación en reposo
3. **Audit Trail:** Log de todas las aprobaciones
4. **Retención:** 5 años (requerimiento legal)

Controles de Acceso:

- **Empleados:** Solo sus propias solicitudes
 - **Jefes:** Su equipo + aprobaciones delegadas
 - **RRHH:** Todas las solicitudes + reporting
 - **Administradores:** Configuración del sistema
-



Mejoras Futuras

Fase 2 (Próxima):

- 1. **App móvil** para solicitudes
- 2. **Integración con nómina** (descuentos automáticos)
- 3. **Machine Learning** para predecir picos de demanda
- 4. **Chatbot** para consultas de saldo

Fase 3 (Avanzado):

- 1. **Intercambio de días** entre empleados
- 2. **Donación de días** en casos especiales
- 3. **Planificación avanzada** con visualización de equipo
- 4. **Integración con sistemas legacy** de RRHH

Beneficios del Sistema

Métrica	Antes	Después	Mejora
Tiempo procesamiento	3-5 días	2-48 horas	80%
Errores manuales	15-20%	<2%	90%
Satisfacción empleados	65%	92%	+27%
Costo administrativo	€12/solicitud	€1.50/solicitud	87.5%
Cumplimiento normativo	70%	99%	+29%

"Sistema completo de gestión de vacaciones que automatiza procesos, garantiza cumplimiento normativo y mejora la experiencia del empleado"