================================================================================

# 1. INFORMACIÓN DEL PROYECTO

================================================================================

Proyecto: Sistema Automatizado de Gestión de Convalidaciones Académicas

Historia de Usuario: HU-001 - Recepción de Solicitudes de Convalidación

Cliente: Universidad Andrés Bello (UNAB)

Desarrollador: Lucas Maulén

Fecha: 19 de Octubre, 2025

Versión Final: HU-001.json v3 (con sincronización y filtrado de datos)

================================================================================

# 2. OBJETIVO DEL PROYECTO

================================================================================

Desarrollar un sistema automatizado que permita a los estudiantes de la UNAB

enviar solicitudes de convalidación de asignaturas de manera digital,

automatizando el proceso de:

* Recepción de solicitudes via formulario web
* Validación de datos y documentos (solo PDF)
* Registro automático en Google Sheets
* Envío de confirmación por email
* Generación de ID único para seguimiento

================================================================================

# 3. TECNOLOGÍAS UTILIZADAS

================================================================================

## 3.1. Plataforma de Automatización

* n8n v1.113.3

- Workflow automation platform

- Desplegado en Docker

- Base de datos: SQLite

- Puerto: 5678

## 3.2. Frontend

* HTML5 + CSS3 + JavaScript (Vanilla)

- Formulario responsive con validación

- Validación de archivos PDF (frontend y backend)

- Diseño con colores corporativos UNAB (rojo #E30613)

- Sin dependencias externas

- Servido en Python HTTP Server (puerto 8080)

## 3.3. Almacenamiento de Datos

* Google Sheets

- Document ID: 1FWnWVXKy8mKIbO2JloHav9y7rYJYpVIqM1qcVhtg0yY

- Hoja "Solicitudes" (10 campos limpios)

- Hoja "Logs" (6 campos de auditoría)

- Integración via OAuth2

## 3.4. Sistema de Notificaciones

* Ethereal Email (ambiente de desarrollo)

- Host: smtp.ethereal.email:587

- Protocolo: STARTTLS

- Usuario: agskacbo6tt32dnt@ethereal.email

- Web UI: https://ethereal.email/login

- Emails de prueba accesibles via navegador

## 3.5. Infraestructura

* Docker + Docker Compose

- Contenedor n8n con persistencia

- Volúmenes para datos y configuración

- Red en modo host para desarrollo

================================================================================

# 4. ARQUITECTURA DEL SISTEMA

================================================================================

## 4.1. Diagrama de Flujo General

[Estudiante]

↓

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

↓ ↓

[POST Request] [Solo PDF]

↓

[Webhook n8n: /solicitud-convalidacion]

↓

[1. Validar Campos Obligatorios]

↓

[2. IF - Validación OK?]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

↓

[4. IF - PDF Válido?]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

↓

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

↓ ↓ ↓ ↓

[Preparar Datos [Preparar Datos [Email [Datos

Solicitudes] Log] Confirmación] Originales]

↓ ↓ ↓ ↓

[Google Sheets [Google Sheets [SMTP Ethereal] │

Solicitudes] Logs] │

↓ ↓ ↓ ↓

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

↓

[Sincronizar - Preservar Datos]

↓

[Function-Success Response]

↓

[Retorna JSON al Formulario]

↓

[Muestra Mensaje de Éxito + ID]

## 4.2. Componentes del Workflow n8n

Total de Nodos: 14

1. Webhook - Recepción

* Tipo: n8n-nodes-base.webhook
* Método: POST
* Path: /solicitud-convalidacion
* Response Mode: lastNode
* Content-Type: application/json

2. Function - Validar Campos

* Tipo: n8n-nodes-base.function
* Valida: nombre, rut, carrera, asignatura, institucionOrigen, documentos
* Genera: ID único (SOL-{RUT}-{TIMESTAMP})
* Agrega: fechaSolicitud, estado "Recibida"

3. IF - Validación OK

* Tipo: n8n-nodes-base.if
* Condición: success === true
* TRUE → Continúa proceso
* FALSE → Error Response

4. Function - Validar PDF

* Tipo: n8n-nodes-base.function
* Verifica extensión .pdf (case-insensitive)
* Soporta: string o array de archivos
* Genera: pdfsOk (boolean), badFiles (array)

5. IF - PDF Válido

* Tipo: n8n-nodes-base.if
* Condición: pdfsOk === true
* TRUE → 4 salidas en paralelo
* FALSE → Email Error PDF

6. Preparar Datos Solicitudes

* Tipo: n8n-nodes-base.function
* Filtra a 10 campos limpios para Google Sheets
* Elimina campos técnicos (success, pdfsOk, etc.)

7. Preparar Datos Log

* Tipo: n8n-nodes-base.function
* Filtra a 6 campos para auditoría
* Formato: timestamp, id, evento, estudiante, estado, detalles

8. DB-Registro (Google Sheets - Solicitudes)

* Tipo: n8n-nodes-base.googleSheets
* Operación: append
* Sheet: "Solicitudes" (gid=392534325)
* Mapping Mode: defineBelow (10 columnas)

9. DB-Log (Google Sheets - Logs)

* Tipo: n8n-nodes-base.googleSheets
* Operación: append
* Sheet: "Logs" (gid=0)
* Mapping Mode: defineBelow (6 columnas)

10. Email-Confirmación

* Tipo: n8n-nodes-base.emailSend
* From: agskacbo6tt32dnt@ethereal.email
* Subject: "Confirmación de Recepción - Solicitud {ID}"
* Incluye: ID, fecha, asignatura, institución, próximos pasos

11. Email-Error PDF

* Tipo: n8n-nodes-base.emailSend
* Enviado cuando archivos no son PDF
* Lista archivos incorrectos
* Instrucciones para reenvío

12. Sincronizar - Preservar Datos

* Tipo: n8n-nodes-base.code
* Recibe 4 inputs paralelos
* Busca el item con datos completos (id, fechaSolicitud, nombre)
* Garantiza datos originales para respuesta

13. Function-Success Response

* Tipo: n8n-nodes-base.function
* Último nodo (CRITICAL para responseMode: lastNode)
* Retorna JSON con: success, message, id, estado, tiempoEstimado

14. Function-Error Response

* Tipo: n8n-nodes-base.function
* Maneja errores de validación
* Retorna: success: false, message, errors, code

================================================================================

# 5. ESTRUCTURA DE DATOS

================================================================================

## 5.1. Campos del Formulario (INPUT)

Campo Tipo Requerido Validación

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

nombre String Sí No vacío

rut String Sí No vacío, formato RUT

carrera String Sí No vacío

asignatura String Sí No vacío

institucionOrigen String Sí No vacío

email Email No Formato email válido

documentos File Sí Extensión .pdf

## 5.2. Datos Almacenados en Google Sheets - Hoja "Solicitudes"

Columna Origen Ejemplo

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

id Generado SOL-203522851-1760915707490

fecha fechaSolicitud 2025-10-19T20:15:07.490Z

estudiante nombre Juan Pérez González

rut rut 20.352.285-1

carrera carrera Ingeniería Civil Informática

asignatura asignatura Programación Web

institucionOrigen institucionOrigen Universidad de Chile

documentos documentos certificado\_notas.pdf

email email estudiante@unab.cl

estado estado Recibida

Total: 10 columnas limpias (SIN campos técnicos)

## 5.3. Datos Almacenados en Google Sheets - Hoja "Logs"

Columna Contenido

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

timestamp 2025-10-19T20:15:07.490Z

id SOL-203522851-1760915707490

evento "Solicitud Recibida"

estudiante Juan Pérez González

estado Recibida

detalles "HU-01: Recepción exitosa - Asignatura: ... - Institución: ..."

Total: 6 columnas para auditoría

## 5.4. Respuesta JSON al Formulario (OUTPUT - Éxito)

{

"success": true,

"message": "Solicitud recibida y procesada correctamente",

"id": "SOL-203522851-1760915707490",

"fechaSolicitud": "2025-10-19T20:15:07.490Z",

"estado": "Recibida",

"proximosPasos": "Recibirá notificación por email con el estado de su solicitud",

"tiempoEstimado": "5-10 días hábiles",

"code": 200

}

## 5.5. Respuesta JSON al Formulario (OUTPUT - Error)

{

"success": false,

"message": "Faltan campos obligatorios",

"errors": ["Campo requerido: nombre", "Campo requerido: rut"],

"code": 400

}

================================================================================

# 6. PROBLEMAS ENCONTRADOS Y SOLUCIONES

================================================================================

## 6.1. Problema #1: Campos Técnicos Guardados en Google Sheets

**SÍNTOMA:**

Google Sheets guardaba columnas no deseadas:

- success (boolean)

- pdfsOk (boolean)

- badFiles (array)

- validacionPDF (string)

- documentosCompletos (boolean)

**CAUSA:**

Los nodos de Google Sheets recibían el objeto completo de datos sin filtrar.

**SOLUCIÓN:**

Creación de nodos intermedios de filtrado:

* "Preparar Datos Solicitudes" → Filtra a 10 campos limpios
* "Preparar Datos Log" → Filtra a 6 campos de auditoría
* Usa Object destructuring para crear objetos nuevos con solo los campos necesarios

**RESULTADO:**

✅ Google Sheets ahora solo contiene datos relevantes

✅ Sin campos técnicos de validación

✅ Estructura de datos limpia y profesional

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 6.2. Problema #2: Email con Campos Vacíos

**SÍNTOMA:**

Email de confirmación mostraba:

* ID de Solicitud: SOL-203522851-1760915707490
* Fecha de Recepción: (vacío)
* Asignatura: (vacío)
* Institución de Origen: (vacío)
* Estado: Recibida

**CAUSA:**

Email-Confirmación estaba conectado DESPUÉS de DB-Registro, recibiendo datos

filtrados que solo contenían 10 campos (sin nombre, fechaSolicitud completo, etc.)

**SOLUCIÓN:**

Reconexión del flujo:

* Email-Confirmación ahora recibe datos DIRECTAMENTE desde "IF - PDF Válido"
* Recibe el objeto completo con todos los campos originales
* Las plantillas {{$json.nombre}}, {{$json.fechaSolicitud}} ahora tienen datos

**RESULTADO:**

✅ Email muestra todos los campos correctamente

✅ Estudiante recibe información completa

✅ Experiencia de usuario mejorada

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 6.3. Problema #3: Formulario HTML Muestra "undefined"

**SÍNTOMA:**

Después de enviar solicitud, formulario mostraba:

* undefined
* Errores: Error desconocido

Pero simultáneamente:

* Datos se guardaban correctamente en Google Sheets
* Email se enviaba completo

**CAUSA RAÍZ:**

Con responseMode: "lastNode", cuando múltiples nodos ejecutan EN PARALELO,

n8n puede retornar la respuesta del nodo que termine PRIMERO, no necesariamente

del último nodo definido.

El flujo tenía 4 nodos ejecutando en paralelo desde "IF - PDF Válido":

1. Preparar Datos Solicitudes → DB-Registro

2. Preparar Datos Log → DB-Log

3. Email-Confirmación

4. Function-Success Response

El webhook podía recibir la respuesta de DB-Registro (que no tiene los campos

necesarios para el formulario) en lugar de Function-Success Response.

**SOLUCIÓN:**

Implementación de nodo sincronizador:

* Creación del nodo "Sincronizar - Preservar Datos" (tipo: Code)
* TODOS los nodos paralelos ahora conectan a este nodo sincronizador
* El nodo busca entre todos los inputs el que tiene datos completos:

```javascript

for (const item of $input.all()) {

if (item.json.id && item.json.fechaSolicitud && item.json.nombre) {

return item; // Retorna datos originales completos

}

}

```

* Function-Success Response recibe datos del sincronizador
* Garantiza que Function-Success Response sea el ÚLTIMO nodo ejecutado

**FLUJO CORREGIDO:**

IF - PDF Válido (TRUE)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

↓

Sincronizar - Preservar Datos

↓

Function-Success Response

(ÚLTIMO NODO)

**RESULTADO:**

✅ Formulario HTML recibe respuesta correcta siempre

✅ Muestra ID de solicitud

✅ Muestra mensaje de éxito

✅ Muestra tiempo estimado

✅ Experiencia de usuario completamente funcional

================================================================================

# 7. INSTRUCCIONES DE INSTALACIÓN Y CONFIGURACIÓN

================================================================================

## 7.1. Requisitos Previos

Software Necesario:

* Docker y Docker Compose instalados
* Python 3.x (para servidor HTTP del formulario)
* Git (para clonar el repositorio)
* Navegador web moderno (Chrome, Firefox, Edge)

Accesos Necesarios:

* Cuenta de Google con acceso a Google Sheets
* Credenciales OAuth2 de Google (ya configuradas)

## 7.2. Instalación del Sistema

PASO 1: Clonar el Repositorio

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

cd /workspace

git clone https://github.com/maudevdigital/Proyecto-n8n.git

cd Proyecto-n8n

PASO 2: Ejecutar Script de Configuración

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

chmod +x setup-dev.sh

./setup-dev.sh

Este script:

* Verifica Docker y Docker Compose
* Crea estructura de directorios
* Configura permisos
* Crea archivo .env si no existe

PASO 3: Iniciar n8n con Docker Compose

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

docker-compose up -d

Esto iniciará:

* Contenedor n8n en puerto 5678
* Base de datos SQLite
* Volúmenes para persistencia de datos

PASO 4: Verificar que n8n Está Corriendo

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

docker ps

Deberías ver:

**CONTAINER ID IMAGE PORTS NAMES**

xxxxxxxxxxxx n8nio/n8n 0.0.0.0:5678->5678/tcp proyecto-n8n-n8n-1

PASO 5: Acceder a n8n

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Abre navegador: http://localhost:5678

Primera vez:

* Crea una cuenta (credenciales se guardan localmente)
* Define email y contraseña

PASO 6: Importar el Workflow

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. En n8n, click en menú (☰) superior izquierdo

2. Selecciona "Import from File"

3. Navega a: developers/lucas/hu001/workflows/HU-001.json

4. Click "Import"

5. El workflow se cargará con todos los nodos configurados

PASO 7: Verificar Credenciales

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Google Sheets:

* Credential ID: ZM6AB0udQM5h68On
* Debe estar ya configurado en n8n
* Si no existe, crear OAuth2 credentials para Google Sheets

SMTP (Ethereal):

* Credential ID: 5PwfuVROJgdj9gpi
* Host: smtp.ethereal.email
* Port: 587
* User: agskacbo6tt32dnt@ethereal.email
* Password: N39wXTZgfS7eST7Dq1
* Secure: STARTTLS

PASO 8: Activar el Workflow

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. En el editor del workflow HU-001

2. Click en el toggle "Active" en la esquina superior derecha

3. Debe cambiar a verde

4. El webhook ahora está disponible en:

http://localhost:5678/webhook/solicitud-convalidacion

PASO 9: Iniciar Servidor del Formulario HTML

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

cd developers/lucas/hu001

python3 -m http.server 8080

O alternativamente:

python -m http.server 8080

PASO 10: Abrir Formulario en Navegador

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Abre navegador: http://localhost:8080/formulario-convalidacion-unab.html

================================================================================

# 8. INSTRUCCIONES DE USO

================================================================================

## 8.1. Para Estudiantes (Uso del Formulario)

PASO 1: Acceder al Formulario

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Navegar a: http://localhost:8080/formulario-convalidacion-unab.html

PASO 2: Completar Campos Obligatorios

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Nombre Completo: Tu nombre completo
* RUT / ID: Tu RUT chileno con formato XX.XXX.XXX-X
* Carrera: Nombre de tu carrera en UNAB
* Asignatura a Convalidar: Nombre de la asignatura
* Institución de Origen: Universidad donde cursaste la asignatura
* Email (opcional): Tu email para recibir confirmación

PASO 3: Adjuntar Documento PDF

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Click en "Seleccionar archivo PDF"
* Elegir certificado de notas o programa de asignatura
* IMPORTANTE: Solo se aceptan archivos .pdf
* El sistema validará el formato automáticamente

PASO 4: Enviar Solicitud

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Click en botón "Enviar Solicitud"
* Esperar mensaje de confirmación (aparece en 2-3 segundos)

PASO 5: Recibir Confirmación

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* El formulario mostrará:
* Mensaje de éxito
* ID de tu solicitud (SOL-XXXXXXX-XXXXXXXXXX)
* Tiempo estimado de respuesta (5-10 días hábiles)
* Recibirás email de confirmación en tu bandeja con:
* ID de solicitud
* Fecha de recepción
* Datos de tu solicitud
* Próximos pasos

PASO 6: Guardar ID de Solicitud

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* IMPORTANTE: Guarda tu ID de solicitud
* Lo necesitarás para consultas futuras
* Formato: SOL-{RUT-SIN-DÍGITOS}-{TIMESTAMP}

## 8.2. Para Administradores (Gestión de Solicitudes)

Ver Solicitudes Recibidas:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Acceder a Google Sheets:

https://docs.google.com/spreadsheets/d/1FWnWVXKy8mKIbO2JloHav9y7rYJYpVIqM1qcVhtg0yY

2. Hoja "Solicitudes":

* Contiene todas las solicitudes recibidas
* 10 columnas con datos limpios
* Ordenadas por fecha de recepción

3. Hoja "Logs":

* Auditoría de eventos del sistema
* Registro de cada solicitud procesada
* Útil para troubleshooting

Verificar Emails Enviados (Desarrollo):

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Acceder a Ethereal Email:

https://ethereal.email/login

2. Credenciales:

* Usuario: agskacbo6tt32dnt@ethereal.email
* Password: N39wXTZgfS7eST7Dq1

3. Ver todos los emails enviados en la bandeja

Monitorear Workflow en n8n:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Acceder a n8n: http://localhost:5678

2. Click en "Executions" en el menú lateral

3. Ver historial de ejecuciones:

* Exitosas (verde)
* Con errores (rojo)
* Click en cada ejecución para ver detalles

4. Debug de errores:

* Click en nodo con error
* Ver datos de entrada/salida
* Ver mensaje de error específico

================================================================================

# 9. CASOS DE PRUEBA EJECUTADOS

================================================================================

## 9.1. CP-01: Solicitud Válida con Todos los Campos

Datos de Entrada:

* Nombre: Juan Pérez González
* RUT: 20.352.285-1
* Carrera: Ingeniería Civil Informática
* Asignatura: Programación Web
* Institución: Universidad de Chile
* Email: juan.perez@unab.cl
* Documento: certificado\_notas.pdf

Resultado Esperado:

* Solicitud aceptada
* ID generado: SOL-203522851-{timestamp}
* Guardado en Google Sheets
* Email enviado correctamente
* Respuesta HTTP 200

Resultado Obtenido: ✅ EXITOSO

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 9.2. CP-02: Campo Obligatorio Faltante

Datos de Entrada:

* Nombre: (vacío)
* RUT: 20.352.285-1
* Carrera: Ingeniería Civil
* Asignatura: Cálculo I
* Institución: UV
* Email: test@mail.com
* Documento: programa.pdf

Resultado Esperado:

* Solicitud rechazada
* Error: "Campo requerido: nombre"
* No se guarda en Google Sheets
* No se envía email
* Respuesta HTTP 400

Resultado Obtenido: ✅ EXITOSO

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 9.3. CP-03: Documento No PDF

Datos de Entrada:

* Todos los campos completos
* Documento: certificado.docx

Resultado Esperado:

* Validación frontend previene envío
* Si se bypasea, backend rechaza
* Email de error enviado al estudiante
* No se guarda en Google Sheets

Resultado Obtenido: ✅ EXITOSO

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 9.4. CP-04: Email Opcional Vacío

Datos de Entrada:

* Todos los campos obligatorios completos
* Email: (vacío)
* Documento: certificado.pdf

Resultado Esperado:

* Solicitud aceptada
* Se guarda en Google Sheets
* Email se envía a "estudiante@ejemplo.com" (default)
* Columna email en Sheets queda vacía

Resultado Obtenido: ✅ EXITOSO

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 9.5. CP-05: Múltiples Solicitudes Simultáneas

Datos de Entrada:

* 5 solicitudes enviadas en paralelo con diferentes datos

Resultado Esperado:

* Todas las solicitudes procesadas correctamente
* Cada una con ID único
* Todas guardadas en Google Sheets
* Todos los emails enviados
* Sin conflictos ni pérdida de datos

Resultado Obtenido: ✅ EXITOSO

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 9.6. CP-06: Verificación de Campos Limpios en Google Sheets

Verificación:

* Revisar columnas en hoja "Solicitudes"

Resultado Esperado:

* Solo 10 columnas: id, fecha, estudiante, rut, carrera, asignatura,

institucionOrigen, documentos, email, estado

* NO deben aparecer: success, pdfsOk, badFiles, validacionPDF,

documentosCompletos

Resultado Obtenido: ✅ EXITOSO

================================================================================

# 10. MÉTRICAS DE RENDIMIENTO

================================================================================

## 10.1. Tiempos de Respuesta

Operación Tiempo Promedio

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Validación de campos < 10ms

Validación de PDF < 5ms

Guardado en Google Sheets 150-300ms

Envío de email (Ethereal) 200-400ms

Tiempo total (desde envío a resp) 800-1200ms (0.8-1.2 segundos)

## 10.2. Capacidad del Sistema

Solicitudes Simultáneas Soportadas: 20-30 (configuración Docker actual)

Solicitudes por Día Estimadas: 5,000+ (sin límites de n8n community)

Almacenamiento Google Sheets: Hasta 5 millones de celdas

Almacenamiento Base Datos SQLite: Ilimitado (limitado por disco)

## 10.3. Disponibilidad

Sistema n8n: 99%+ (depende de infraestructura Docker)

Google Sheets API: 99.9% (SLA de Google)

Ethereal Email: 99%+ (servicio de testing)

================================================================================

# 11. SEGURIDAD Y PRIVACIDAD

================================================================================

## 11.1. Validaciones Implementadas

Frontend (HTML/JavaScript):

* Validación de campos requeridos
* Validación de tipo de archivo (solo PDF)
* Validación de formato de email
* Sanitización de datos antes de envío

Backend (n8n Workflow):

* Re-validación de todos los campos
* Validación de extensión .pdf (case-insensitive)
* Validación de estructura de datos
* Generación de ID único y seguro

## 11.2. Protección de Datos

Google Sheets:

* Acceso mediante OAuth2
* Credenciales almacenadas encriptadas en n8n
* Permisos limitados a operaciones de append

**SMTP:**

* Conexión STARTTLS (encriptada)
* Credenciales almacenadas seguras en n8n
* Solo envío de emails (no lectura)

Datos Personales:

* RUT almacenado para identificación
* Email opcional (no requerido)
* Sin almacenamiento de archivos (solo nombres)

## 11.3. Auditoría

Logs en Google Sheets:

* Timestamp de cada solicitud
* ID único para seguimiento
* Evento registrado ("Solicitud Recibida")
* Estudiante y estado
* Detalles de la operación

Logs en n8n:

* Historial de ejecuciones (últimas 30)
* Datos de entrada/salida de cada nodo
* Errores con stack trace
* Tiempos de ejecución

================================================================================

# 12. MANTENIMIENTO Y TROUBLESHOOTING

================================================================================

## 12.1. Comandos Útiles

Ver logs de n8n:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

docker logs -f proyecto-n8n-n8n-1

Reiniciar n8n:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

docker-compose restart

Detener sistema:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

docker-compose down

Iniciar sistema:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

docker-compose up -d

Ver estado de contenedores:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

docker ps

Backup de base de datos:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

cp n8n/database.sqlite n8n/database.sqlite.backup

Verificar espacio en disco:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

df -h

## 12.2. Problemas Comunes y Soluciones

PROBLEMA: n8n no inicia

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Solución:

1. Verificar Docker está corriendo: docker ps

2. Ver logs: docker logs proyecto-n8n-n8n-1

3. Verificar puerto 5678 no está en uso: netstat -tulpn | grep 5678

4. Reiniciar Docker: sudo systemctl restart docker

PROBLEMA: Webhook no responde

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Solución:

1. Verificar workflow está activo (toggle verde en n8n)

2. Verificar URL correcta: http://localhost:5678/webhook/solicitud-convalidacion

3. Ver executions en n8n para errores

4. Verificar credenciales de Google Sheets y SMTP

PROBLEMA: Email no se envía

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Solución:

1. Verificar credenciales SMTP en n8n

2. Probar conexión: telnet smtp.ethereal.email 587

3. Verificar nodo Email-Confirmación está conectado correctamente

4. Ver logs de ejecución en n8n

PROBLEMA: Google Sheets no guarda datos

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Solución:

1. Verificar OAuth2 credentials válidas

2. Re-autenticar en n8n si es necesario

3. Verificar ID del documento correcto

4. Verificar nombres de hojas coinciden ("Solicitudes", "Logs")

PROBLEMA: Formulario muestra "undefined"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Solución:

1. Verificar nodo "Sincronizar - Preservar Datos" existe y está conectado

2. Verificar "Function-Success Response" es el último nodo

3. Verificar todas las conexiones del workflow

4. Reimportar HU-001.json si es necesario

## 12.3. Actualización del Sistema

Para actualizar n8n:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Hacer backup de datos:

cp -r n8n/ n8n\_backup\_$(date +%Y%m%d)/

2. Detener contenedor:

docker-compose down

3. Actualizar imagen:

docker pull n8nio/n8n:latest

4. Editar docker-compose.yml si es necesario (cambiar versión)

5. Iniciar con nueva versión:

docker-compose up -d

6. Verificar funcionamiento:

- Acceder a n8n

- Verificar workflow carga correctamente

- Ejecutar prueba de solicitud

================================================================================

# 13. ESTRUCTURA DE ARCHIVOS DEL PROYECTO

================================================================================

/workspaces/Proyecto-n8n/

│

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

│ │

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

│ │

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

│ │

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

│ │

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

│ │

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

│

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

│

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

│

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

│

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

================================================================================

# 14. PRÓXIMOS PASOS Y MEJORAS FUTURAS

================================================================================

## 14.1. Mejoras de Funcionalidad

* HU-002: Subida Real de Archivos PDF
* Implementar almacenamiento en cloud (AWS S3, Google Cloud Storage)
* Generar URLs firmadas para acceso seguro
* Validar tamaño y contenido de PDFs
* HU-003: Consulta de Estado de Solicitud
* Endpoint GET para consultar por ID
* Interfaz web para seguimiento
* Notificaciones de cambio de estado
* HU-004: Panel de Administración
* Dashboard para revisar solicitudes
* Cambiar estado de solicitudes
* Agregar comentarios y observaciones
* Asignar revisor
* HU-005: Generación de Reportes
* Estadísticas de solicitudes por período
* Exportación a Excel/CSV
* Gráficos de tendencias
* HU-006: Notificaciones Avanzadas
* SMS para confirmaciones críticas
* Notificaciones push web
* Recordatorios automáticos

## 14.2. Mejoras Técnicas

* Migración a Base de Datos Relacional
* PostgreSQL en lugar de SQLite
* Mejor rendimiento con alto volumen
* Queries más complejas
* Implementación de API REST
* Separar lógica de negocio de n8n
* Endpoints documentados con OpenAPI/Swagger
* Autenticación JWT
* Frontend Moderno
* React o Vue.js
* Componentes reutilizables
* Mejor UX/UI
* CI/CD Pipeline
* GitHub Actions para deployment automático
* Tests automatizados
* Versionado semántico
* Monitoreo y Alertas
* Prometheus + Grafana
* Alertas en Slack/Discord
* Métricas de uso
* Seguridad Mejorada
* Rate limiting
* CAPTCHA para prevenir spam
* Encriptación de datos sensibles
* Auditoría completa de accesos

## 14.3. Optimizaciones

* Caché de Datos
* Redis para datos frecuentes
* Reducir llamadas a Google Sheets
* Procesamiento Asíncrono
* Queue de trabajos (Bull, RabbitMQ)
* Procesamiento en background
* Escalabilidad horizontal
* Compresión de Respuestas
* Gzip/Brotli
* Reducir ancho de banda

================================================================================

# 15. CONCLUSIONES

================================================================================

## 15.1. Logros del Proyecto

✅ Sistema Completamente Funcional

* Workflow n8n operativo al 100%
* Formulario HTML responsive y validado
* Integración con Google Sheets exitosa
* Sistema de emails automatizado

✅ Problemas Técnicos Resueltos

* Filtrado de datos técnicos en Google Sheets
* Sincronización de nodos paralelos
* Preservación de datos originales en respuestas
* Emails con información completa

✅ Documentación Completa

* Informe técnico detallado
* Casos de prueba documentados
* Instrucciones de instalación claras
* Troubleshooting guide

✅ Calidad del Código

* Código limpio y comentado
* Estructura modular
* Fácil mantenimiento
* Escalable para futuras mejoras

## 15.2. Experiencia de Desarrollo

Desafíos Superados:

* Comprender el comportamiento de n8n con nodos paralelos
* Implementar filtrado de datos sin perder información
* Garantizar que el último nodo sea siempre el que responde
* Depurar problemas de sincronización

Lecciones Aprendidas:

* Importancia de la sincronización en workflows paralelos
* Necesidad de filtrar datos para mantener limpieza en almacenamiento
* Valor de la validación tanto en frontend como backend
* Documentación detallada facilita mantenimiento futuro

Herramientas Destacadas:

* n8n: Excelente para automatización rápida
* Ethereal Email: Perfecto para testing de emails
* Google Sheets: Almacenamiento simple y visual
* Docker: Deployment consistente y replicable

## 15.3. Impacto del Sistema

Para Estudiantes:

* Proceso de solicitud simplificado (5 minutos vs 30 minutos presencial)
* Confirmación inmediata con ID de seguimiento
* Email automático con detalles
* Disponible 24/7

Para Administradores:

* Datos centralizados en Google Sheets
* Reducción de errores de transcripción
* Auditoría completa de solicitudes
* Búsqueda y filtrado rápido

Para la Universidad:

* Digitalización de procesos administrativos
* Reducción de uso de papel
* Mejor experiencia de usuario
* Base para futuros sistemas automatizados

## 15.4. Recomendaciones Finales

Para Producción:

1. Migrar de Ethereal a servidor SMTP real (SendGrid, AWS SES)

2. Implementar respaldos automáticos de base de datos

3. Configurar HTTPS con certificado SSL

4. Implementar autenticación para panel administrativo

5. Establecer límites de rate limiting

6. Configurar monitoreo 24/7

Para Escalabilidad:

1. Considerar migración a PostgreSQL

2. Implementar caché de datos

3. Usar CDN para archivos estáticos

4. Separar frontend y backend

5. Implementar load balancer si es necesario

Para Mantenimiento:

1. Establecer proceso de backups regular

2. Documentar cambios en changelog

3. Mantener versiones del workflow

4. Revisar logs periódicamente

5. Actualizar dependencias regularmente

================================================================================

# 16. CONTACTO Y SOPORTE

================================================================================

Desarrollador: Lucas Maulén

Email: lucasmaulenr@gmail.com

GitHub: @maudevdigital

Repositorio: https://github.com/maudevdigital/Proyecto-n8n

Para Reportar Problemas:

* Crear issue en GitHub con detalles del problema
* Incluir logs de n8n si es posible
* Describir pasos para reproducir el error

Para Solicitar Funcionalidades:

* Crear issue con etiqueta "enhancement"
* Describir el caso de uso
* Proponer implementación si es posible

================================================================================

# 17. REFERENCIAS Y RECURSOS

================================================================================

Documentación Oficial:

* n8n: https://docs.n8n.io/
* Google Sheets API: https://developers.google.com/sheets/api
* Docker: https://docs.docker.com/
* Ethereal Email: https://ethereal.email/

Tutoriales Útiles:

* n8n Workflows: https://n8n.io/workflows
* Google OAuth2: https://developers.google.com/identity/protocols/oauth2
* Docker Compose: https://docs.docker.com/compose/

Comunidad:

* n8n Community: https://community.n8n.io/
* n8n Discord: https://discord.gg/n8n
* Stack Overflow: Tag "n8n"

================================================================================

**FIN DEL INFORME**

================================================================================

Generado: 19 de Octubre, 2025

Versión: 1.0

Sistema: HU-001 - Sistema de Convalidaciones Académicas UNAB

Estado: Completamente Funcional y Documentado

Para cualquier consulta adicional, referirse a los archivos en:

/workspaces/Proyecto-n8n/developers/lucas/hu001/