

Zoom → Make/n8n → Gamma

Resúmenes automáticos de webinars convertidos en presentaciones y enviados por email

STACK TÉCNICO	CASO DE USO	SALIDA FINAL	ENTREGA
Zoom Business + Make o n8n + Gamma API	Webinars / reuniones / clases online con resumen automático	Presentación en Gamma	Correo con link de descarga + link de acceso online

Objetivo del flujo

- Cuando termine un webinar en Zoom, Make/n8n enviará ese contenido a Gamma para crear automáticamente una presentación y, cuando esté lista, enviará por email la URL de la presentación online y/o la descarga del PDF.

Flujo completo

1	Zoom genera el resumen del webinar
2	Make/n8n recibe el webhook
3	Make filtra por host y título
4	Make envía el resumen a Gamma
5	Gamma crea la presentación
6	Make hace polling cada 30 segundos
7	Estado completed → email con URLs

- Ventajas: No hace falta pedir el resumen por API si el webhook ya trae summary_content. Se pueden filtrar solo determinados webinars, hosts o títulos. Se puede reutilizar un tema personalizado de Gamma. La salida puede terminar en email, SharePoint, Teams, Moodle o cualquier otro destino.

1

Preparación en Zoom

1.1 Activar el resumen de reuniones

- Entrar en la cuenta administradora de Zoom.
- Ir a **Admin** → **Account Settings** → **AI Companion**.
- Activar **Meeting Summary with AI Companion**.
- Opcional: activar el arranque automático del summary si interesa que no dependa del anfitrión.

1.2 Crear la app en Zoom App Marketplace

- Entrar en Zoom App Marketplace con la cuenta administradora.
- Ir a **Develop** → **Build App**.
- Elegir **Server-to-Server OAuth**.
- Asignar un nombre reconocible, por ejemplo: *Zoom Summary to Gamma*.

1.3 Features que hay que activar

- Activar **Event Subscriptions**.
- Pegar la URL del webhook de Make como **Event notification endpoint URL**. (aquí debemos empezar el workflow de Make/n8n)
- Validar el endpoint.
- Añadir el evento exacto: **Meeting summary has been completed**.

■ Este es el evento importante. Es el que dispara el flujo cuando Zoom ya ha terminado de generar el resumen. Si se elige otro evento parecido, puede que no llegue el campo `summary_content` y entonces el escenario ya no sería igual.

1.4 Scope que necesitamos

- En **Scopes**, añadir el permiso necesario para poder leer resúmenes:
`meeting_summary:read:admin`

Nota: dependiendo de la interfaz actual de Zoom, el nombre visible del scope puede variar ligeramente. Lo importante es que sea el permiso de lectura del summary a nivel admin. En algunas pantallas puede verse con un nombre equivalente dentro del grupo de Meeting summary.

1.5 Credenciales que hay que guardar

Credencial	Descripción / Uso
Account ID	Necesario para el endpoint OAuth de Zoom
Client ID	Usado como Username en Basic Auth
Client Secret	Usado como Password en Basic Auth
Secret Token del webhook	Para validar el endpoint en Zoom App Marketplace

Aunque en esta versión del flujo no necesitamos llamar a Zoom por API para pedir el summary (ya llega en el webhook), estas credenciales conviene guardarlas para futuras ampliaciones o para una versión con fallback por API.

2

Crear la automatización en Make/n8n

■ La mejor opción para ahorrar tiempo es pedirle a ChatGPT o Claude un blueprint JSON de Make / n8n con el escenario ya estructurado para importarlo directamente. Aun así, conviene revisar y editar después: webhook, filtros, API key de Gamma, email final y cualquier campo que dependa del entorno real.

2.1 Crear el escenario

- Crear un escenario nuevo en Make/n8n.
- Añadir un módulo **Webhooks** → **Custom webhook**. [Nodo 1]
- Copiar su URL y pegarla en la app de Zoom (esto ya deberíamos haberlo hecho cuando creamos la app de Zoom).
- Poner el escenario en **Run once** para capturar un ejemplo real del payload.

Campos importantes que deben llegarte en el webhook

Cuando el evento es correcto, el webhook debe incluir al menos estos campos:

```
payload.object.meeting_host_email
payload.object.meeting_topic
payload.object.summary_content
payload.object.meeting_start_time
payload.object.meeting_end_time
```

Filtro recomendado

Puedes utilizar filtros por si solo quieres crear presentaciones de determinadas reuniones o de un host concreto. El filtro se coloca entre el webhook y el nodo que envía el contenido a Gamma. Así evitamos generar presentaciones para reuniones internas, pruebas o sesiones que no interesan:

Filtro	Valor de ejemplo
meeting_host_email	correo electrónico del host
meeting_topic contains	WEBINAR (por ejemplo)

2.2 Nodo 2: OAuth a Zoom

Añadir un nodo **HTTP – Make a request** y configurarlo así:

Campo	Valor
Authentication type	Basic auth
Username (Credentials)	Client ID (guardado al crear la app en Zoom)
Password (Credentials)	Client Secret (guardado al crear la app en Zoom)
URL	https://zoom.us/oauth/token?grant_type=account_credentials&account_id=TU_ACCOUNT_ID
Method	POST

3 Nodo de Gamma: crear la presentación

3.1 Nodo 3: Configuración del módulo HTTP

El nodo que envía el contenido a Gamma es un módulo **HTTP** → **Make a request** configurado así:

Campo	Valor
Method	POST
URL	https://public-api.gamma.app/v1.0/generations
Header 1	X-API-KEY: tu API key de Gamma
Body type	Raw
Content type	application/json
Body input method	Data structure → MUY IMPORTANTE

■ Importante — usar Data structure y NO JSON string: cuando el contenido viene de Zoom, `summary_content` puede traer saltos de línea, markdown, comillas u otros caracteres que rompan el JSON. La solución correcta es usar Data structure para que Make/n8n genere el JSON de forma segura.

3.2 Campos que hay que crear en el Data structure

En Make, **Collection** equivale a un objeto JSON anidado. Por eso cardOptions se crea como Collection y dentro se añaden dimension y headerFooter.

Campo	Tipo	Valor esperado
inputText	Text	Texto principal que se envía a Gamma
additionalInstructions	Text	Instrucciones de estilo y enfoque
textMode	Text	condense
format	Text	presentation
numCards	Number	Número de slides <i>(o eliminar este campo para dejar que Gamma decida)</i>
exportAs	Text	pdf
themeId	Text	ID del tema personalizado de Gamma
cardOptions	Collection	
cardOptions.dimension	Text	16x9
Card.Option.headerFooter	Collection	
Card.Option.headerFooter.topRight	Collection	
Card.Option.headerFooter.topRight.type	Text	Image
Card.Option.headerFooter.topRight.source	Text	themeLogo
Card.Option.headerFooter.topRight.size	Text	xl

Ejemplo de inputText:

Título de la reunión: {{1.payload.object.meeting_topic}}

Resumen de la clase o webinar generado por Zoom: {{1.payload.object.summary_content}}

Ejemplo de additionalInstructions:

Crea una presentación didáctica, profesional y visual en español a partir de este resumen de una clase online o webinar de formación. La presentación debe servir para repasar el contenido tratado por los asistentes. Organiza las diapositivas con portada, contexto o introducción, desarrollo de conceptos clave, ideas principales, aprendizajes destacados, conclusiones y, si existen, próximos pasos o recomendaciones. Evita un tono ejecutivo; prioriza un enfoque formativo, claro y ordenado.

4

Polling: comprobar el estado en Gamma

Gamma no devuelve la presentación terminada en el mismo momento del POST. Primero devuelve un identificador de generación y luego hay que consultar su estado. Dependiendo del número de diapositivas puede tardar **1–3 minutos** en generar la presentación.

Nodo 4 — Sleep (60 segundos)

- Añadir un nodo **Sleep** con un delay de **60 segundos** para darle tiempo a Gamma a crear la presentación.

Nodo 5 — Comprobar el estado (GET)

Campo	Valor
Method	GET
URL	<code>https://public-api.gamma.app/v1.0/generations/{{generationId}}</code>
Header	X-API-KEY: tu API key de Gamma
Parse response	Yes

La URL debe usar el identificador devuelto por el nodo POST de Gamma. Si el ID no se mapea bien, la API responderá con error 404 porque estarás consultando solo /generations/ sin el id al final.

Estados esperados

Estado	Significado
pending	Gamma todavía está generando la presentación.
completed	La presentación ya está lista.

Nodo 6 — Router con dos ramas

Después del nodo de comprobación de estado, añadir un **Router** con dos ramas. En cada rama añadir un filtro:

Rama: `status = pending`

- Filtro: `status = pending`
- Añadir un nuevo **Sleep 60 seconds**.
- Duplicar o recrear el nodo de comprobación de estado.
- Volver a comprobar hasta que el estado pase a *completed*.

Rama: `status = completed`

- Filtro: `status = completed`
- Cuando el estado sea `completed` ya se pueden usar directamente los campos de salida del nodo de comprobación:

`{{gammaUrl}}`: link para ver la presentación online.

`{{exportUrl}}`: link para descargar la presentación en PDF.

5

Email final con el resultado

En la rama **completed**, añadir un módulo de email (Outlook o Gmail) y mapear directamente las URLs devueltas por Gamma (del nodo de comprobación de estado).

Campo del email	Ejemplo / Valor
To	Correo electrónico del receptor
Subject	Resumen del webinar: {{meeting_topic}}
Body	Incluye título, {{gammaUrl}} y {{exportUrl}} (ver ejemplo abajo)

Ejemplo de cuerpo del email:

```
La presentación del webinar ya está lista.  
  
Título: {{1.payload.object.meeting_topic}}  
Ver presentación online: {{5.data.gammaUrl}}  
Descargar PDF: {{5.data.exportUrl}}
```

■ Si queréis hacer el correo bonito, pasadle el cuerpo del mail a ChatGPT/Claude y pedirle que os haga el correo más bonito utilizando HTML.

6

Errores y problemas típicos

✗ Zoom envía el evento pero no llega summary_content

Revisar que el evento elegido sea exactamente "Meeting summary has been completed". Si se elige otro evento parecido, puede que no llegue el campo summary_content.

✗ Error al crear el JSON en Gamma

Usar Data structure en lugar de JSON string.

✗ Error con aspectRatio

En este endpoint no se usa aspectRatio; hay que usar cardOptions.dimensions = 16x9.

✗ Gamma devuelve 404 al comprobar estado

Suele significar que el generationId no se ha mapeado bien en la URL. Asegúrate de que la URL sea .../generations/{generationId} y no solo .../generations/

✗ No aparecen gammaUrl o exportUrl en Make

Esos campos solo estarán disponibles cuando el estado sea completed. No estarán presentes en el estado pending.

✗ El resumen sale raro o en otro idioma

Suele deberse a una transcripción pobre o fragmentada en esa reunión de prueba.

7

Checklist final

✓	Zoom Business con Meeting Summary activo.
✓	App de Zoom creada como Server-to-Server OAuth.
✓	Features: Event Subscriptions activo.
✓	Evento configurado: "Meeting summary has been completed".
✓	Scope configurado: meeting_summary:read:admin.
✓	Webhook de Make validado correctamente.
✓	Filtro en Make por host X y/o por la palabra X en el título.
✓	Nodo de Gamma configurado con Data structure.
✓	Polling cada 60 segundos con Router para pending/completed.
✓	Email final con {{gammaUrl}} y {{exportUrl}}.

Si se quiere acelerar el despliegue, se puede pedir a ChatGPT / Claude el blueprint JSON del escenario ya preparado para Make o n8n. Aun así, esta guía sigue siendo útil para entenderlo, revisarlo y mantenerlo.

