¡Vamos a por todo, Chema! Aquí tienes las **dos integraciones completas**:

**✅ 1. Descargar informe PDF desde el dashboard (Vue.js)**

**📁 ui/src/componentes/InformeResultado.vue**

<template>

<div>

<h2>📄 Informes disponibles</h2>

<ul>

<li v-for="informe in informes" :key="informe.nombre">

{{ informe.nombre }}

<a :href="informe.url" target="\_blank">📥 Descargar</a>

</li>

</ul>

</div>

</template>

<script>

export default {

name: "InformeResultado",

data() {

return {

informes: []

};

},

mounted() {

fetch("http://localhost:8000/informes/listar", {

headers: {

Authorization: `Bearer ${localStorage.getItem("token")}`

}

})

.then((res) => res.json())

.then((data) => {

this.informes = data;

});

}

};

</script>

**📁 api/app/rutas/resultados.py (añadir esta nueva ruta)**

from fastapi.responses import FileResponse

import os

@router.get("/listar-informes")

def listar\_informes(usuario=Depends(obtener\_usuario\_actual)):

base = "/informes"

archivos = []

for f in os.listdir(base):

if usuario["sub"] in f:

archivos.append({

"nombre": f,

"url": f"http://localhost:8000/informes/descargar/{f}"

})

return archivos

@router.get("/descargar/{archivo}")

def descargar\_informe(archivo: str, usuario=Depends(obtener\_usuario\_actual)):

path = os.path.join("/informes", archivo)

if usuario["sub"] not in archivo:

raise HTTPException(status\_code=403, detail="Acceso denegado.")

return FileResponse(path, filename=archivo, media\_type="application/pdf")

**✅ 2. Enviar PDF por WhatsApp automáticamente (vía Twilio)**

**📁 bot-whatsapp/utils.py (añadir esta función)**

def enviar\_pdf\_whatsapp(numero: str, ruta\_pdf: str):

media\_url = subir\_a\_un\_host\_publico(ruta\_pdf) # 💡 ver nota abajo

mensaje = "📎 Informe de escaneo generado."

client.messages.create(

body=mensaje,

from\_=FROM\_WA,

to=numero,

media\_url=[media\_url]

)

💡 **Importante**: Twilio solo admite URLs públicas (no archivos locales). Necesitas subir el .pdf temporalmente a:

* Un bucket S3 público
* Un servidor NGINX temporal
* O usar una URL firmada (pre-signed)

Ejemplo simple con Python y MinIO/S3:

import boto3

s3 = boto3.client("s3", aws\_access\_key\_id=..., aws\_secret\_access\_key=...)

s3.upload\_file(ruta\_pdf, "bucket", "informes/" + nombre)

media\_url = f"https://mi-bucket.s3.amazonaws.com/informes/{nombre}"

**¿Qué más puedo generar?**

* Ruta /informe/enviar/{scan\_id} para disparar el WhatsApp tras escaneo.
* Botón “🟢 Enviar por WhatsApp” en Vue.
* Dashboard completo tipo tabla (como ThreatGrid o Cuckoo).
* Generador de informes en Markdown, HTML y JSON.

¿Quieres que prepare el endpoint /informe/enviar o la tabla completa de escaneos por usuario?