ODISEIA4GOOD HACKATHON / SPRINT 3

Clara

CivicAid Voice

Traducimos la burocracia a tu lengua

Equipo: Robert Promes (PM), Marcos, Daniel, Andrea, Lucas
UDIT — Taller de Proyectos II — Dr. Gustavo Bermejo Martin
13 de febrero de 2026

3.2M	9.5M	4.5M	78%
inmigrantes	mayores 65	exclusion	WhatsApp

Indice con cruce de criterios

Seccion	Contenido	Criterio
01 Funcionalidades	Que hace Clara hoy	Innovacion (30%)
02 Arquitectura	Stack tecnico	Viabilidad (20%)
03 Procesos	Flujos paso a paso	Innovacion (30%)
04 Ventajas	Por que funciona	Impacto Social (30%)
05 Desventajas	Limitaciones honestas	Viabilidad (20%)
06 Prototipo	Lo que el jurado vera	Presentacion (20%)
07 Escalabilidad	Prototipo a producto	Viabilidad (20%)
08 Destacables	Lo diferente	Impacto Social (30%)
Criterios	Punto por punto	Todos (100%)
Scrum	Sprint review	Presentacion (20%)

01 FUNCIONALIDADES

Que hace Clara hoy

Clara es un asistente conversacional WhatsApp-first que guia a personas vulnerables a traves de tramites de servicios sociales en Espana.

Funcionalidades CORE (desplegadas)

#	Funcionalidad	Detalle	Estado
1	Chat multilingue	ES + FR, deteccion automatica	Operativo
2	Voz STT + TTS	Gemini Flash + gTTS	Operativo
3	Cache inteligente	8 respuestas, <2s	Operativo
4	Pipeline 11 skills	Orquestador modular	Operativo

Funcionalidades de CALIDAD (Sprint 3)

- Guardrails de seguridad — pre-check y post-check, 6 red team tests con 100% bloqueo
- Observabilidad — JSON logs estructurados con request_id y timings por stage
- Evaluaciones automatizadas — 16 casos en 4 sets con scripts/run_evals.py

Proxima fase (Sprint 4)

- Lector de documentos — analisis de imagenes con vision LLM
- Canal web — interfaz web en desarrollo, WhatsApp prioridad
- Elegibilidad proactiva — requiere persistencia de sesion

02 ARQUITECTURA

Stack 100% gratuito

Patron TwiML ACK (diferencial tecnico)

El patron TwiML ACK separa la respuesta HTTP (rapida) del envio del mensaje (asincrono):

TwiML ACK — 4 pasos

1. Twilio envia POST a /webhook
2. Flask responde HTTP 200 con TwiML vacio en <1 segundo
3. Hilo de fondo procesa: cache, LLM, audio
4. Resultado enviado via Twilio REST API

Garantiza que Twilio nunca recibe timeout (15s max).

Pipeline de 11 skills

#	Skill	Archivo	Funcion
1	detect_input	detect_input.py	Clasifica tipo entrada
2	fetch_media	fetch_media.py	Descarga multimedia
3	convert_audio	convert_audio.py	Convierte formatos audio
4	transcribe	transcribe.py	Transcribe via Gemini
5	detect_lang	detect_lang.py	Detecta idioma (ES/FR)
6	cache_match	cache_match.py	Busca en cache keywords
7	kb_lookup	kb_lookup.py	Busca en 3 KBs
8	llm_generate	llm_generate.py	Genera con Gemini Flash
9	verify_response	verify_response.py	Verifica vs KB
10	tts	tts.py	Texto a audio (gTTS)
11	send_response	send_response.py	Envia via Twilio REST

Software stack

Componente	Tecnologia	Coste
Lenguaje	Python 3.11	Gratis
Framework	Flask 3.1 + Gunicorn 21	Gratis
LLM + STT	Gemini 1.5 Flash	Gratis (free tier)
TTS	gTTS 2.5	Gratis
WhatsApp	Twilio 9 (sandbox)	Gratis
Hosting	Render.com (Frankfurt)	Gratis (free)
Container	Docker (Python 3.11-slim)	Gratis
Deteccion idioma	langdetect 1.0	Gratis
Validacion	Pydantic 2.x	Gratis

9 Feature Flags

Flag	Default	Efecto
DEMO_MODE	false	Cache-only, skip LLM tras miss
LLM_LIVE	true	Habilita Gemini
WHISPER_ON	true	Habilita transcripcion audio
LLM_TIMEOUT	6s	Timeout maximo Gemini
WHISPER_TIMEOUT	12s	Timeout transcripcion
GUARDRAILS_ON	true	Guardrails de contenido
STRUCTURED_OUTPUT_ON	false	Salida estructurada JSON
OBSERVABILITY_ON	true	Metricas y trazas
RAG_ENABLED	false	RAG (pendiente)

03 PROCESOS

Flujo paso a paso

Flujo A: Texto WhatsApp

6 pasos — Tiempo: <2s (cache) | 4-8s (LLM)

1. Usuario escribe mensaje en WhatsApp
2. Twilio envia POST a /webhook con Body, From, To
3. Flask valida firma X-Twilio-Signature (seguridad)
4. Flask responde TwiML vacio (ACK <1 segundo)
5. Hilo de fondo: guardrails -> detecta idioma -> cache -> KB -> LLM -> verify -> gTTS
6. Twilio REST envia respuesta al usuario

Flujo B: Audio WhatsApp

8 pasos — Tiempo: 6-12s

1. Usuario graba audio en WhatsApp
2. Twilio POST con MediaUrl0, MediaContentType0, NumMedia=1
3. Flask valida firma X-Twilio-Signature
4. Flask responde TwiML vacio (ACK <1 segundo)
5. Descarga audio + Gemini Flash transcripcion (STT)
6. Detecta idioma del texto transcrito
7. Cache match -> KB lookup -> Gemini -> Verify -> gTTS
8. Twilio REST envia respuesta texto + audio

04 VENTAJAS

Por que Clara funciona

- Accesible sin alfabetizacion digital — WhatsApp (78% de Espana) + mensajes de voz. No requiere instalar nada.
- Multilingue nativo — Deteccion automatica ES/FR. Extensible cambiando config.
- Coste cero de operacion — Render free + Gemini free + gTTS free + Twilio sandbox.
- Cache inteligente — 8 respuestas pre-calculadas en <2s con audio pre-generado.
- 96 tests automatizados — 85 unit + 7 integration + 4 e2e + evals + red team.
- Guardrails de seguridad — Pre-check y post-check. 6 red team tests: 100% bloqueo.
- Pipeline resiliente — TwiML ACK + cache-first + fallback + thread protection.

05 DESVENTAJAS

Limitaciones honestas

Limitacion	Impacto	Mitigacion
KB estatica (3 JSONs)	Solo 3 tramites	RAG flag preparado, extensible
Dependencia Gemini API	Sin Gemini no hay STT/LLM	Cache-first + fallback templates
Cold start Render	~30s primer request	Cron cada 14 min activo
Twilio Sandbox	Requiere join por usuario	Migracion trivial (1 variable)
Sin persistencia sesion	No recuerda previos	Redis/DB sin cambiar pipeline
Solo 2 idiomas probados	ES+FR, no EN/AR	Extensible por config
Canal web en desarrollo	Solo WhatsApp hoy	Web para Sprint 4

Nota: Todas las limitaciones son del prototipo, no de la arquitectura. El diseno modular permite superar cada una sin reescribir codigo.

Matriz de riesgos

Riesgo	Probabilidad	Impacto	Mitigacion
Gemini API down	Baja	Alto	Cache-first + fallback
Cold start en demo	Media	Alto	Cron 14 min
Twilio sandbox expira	Baja	Medio	Re-join 30s
Alucinacion LLM	Media	Alto	Guardrails + verify + KB
Audio no transcrito	Baja	Medio	Fallback idioma
Pico de trafico	Baja	Medio	Escalable con plan pago

06 PROTOTIPO

Lo que el jurado vera

URL de produccion

https://civicaid-voice.onrender.com

```
GET /health -> {"status":"healthy","cache_entries":8,"tramites_loaded":3,
  "twilio_configured":true,"demo_mode":true,"guardrails_on":true}
```

Tabla de funcionalidades

Funcionalidad	Disponible	Detalle
Chat texto ES	SI	3 tramites en espanol
Chat texto FR	SI	3 tramites en frances
Audio entrada	SI	Gemini Flash STT
Audio salida	SI	gTTS MP3
Cache inteligente	SI	8 respuestas + 6 MP3s
Guardrails	SI	Pre + post check
Canal web	NO	Sprint 4
Lector docs	NO	Sprint 4

3 casos de demo

- Maria (ES, 45) — IMV: escribe 'Como solicito el ingreso minimo vital?' -> requisitos + cuantias + telefonos
- Ahmed (FR, 32) — Empadronamiento: audio en frances -> Clara transcribe, detecta FR, responde en frances
- Laura (ES, 28) — Tarjeta sanitaria: 'Necesito la tarjeta sanitaria' -> documentos + donde ir + telefonos

07 ESCALABILIDAD

De prototipo a producto

Aspecto	Hoy (0 EUR)	Produccion (~200 EUR)
Servidor	Render free (512MB)	GCP Cloud Run
LLM	Gemini Flash free	Gemini Pro fine-tuned
Transcripcion	Gemini Flash	Whisper large / Gemini Pro
TTS	gTTS	Google Cloud TTS
KB	3 JSONs	RAG + vector DB
Idiomas	2 (ES, FR)	6+
Canal	WhatsApp sandbox	WA Business + Web + Telegram
Persistencia	Sin estado	Redis / PostgreSQL
Tests	96	200+ con CI/CD

Coste proyectado

Escala	Usuarios/mes	Coste
Prototipo	<100	0 EUR
Piloto ONG	1,000	~50 EUR
Municipio	10,000	~200 EUR
CCAA	100,000	~1,500 EUR

08 PUNTOS DESTACABLES

Lo que nos hace diferentes

Sprint 3 en numeros

0	16	96	11	9	36h	3	8
EUR/mes	commits	tests	skills	flags	dev	KBs	cache

3 personas reales

Maria (ES) — 45 anos, vulnerabilidad. No sabe navegar webs. Escribe a Clara por WhatsApp y recibe pasos exactos para IMV.
Ahmed (FR) — 32 anos, inmigrante francofono. Envia audio en frances, Clara responde en su idioma.
Laura (ES) — 28 anos, mudada. En 2 segundos tiene info completa de tarjeta sanitaria.

Datos INE

- 3.2 millones de inmigrantes en Espana
- 9.5 millones de mayores de 65 anos
- 4.5 millones en riesgo de exclusion social
- 78% de la poblacion usa WhatsApp a diario

CRITERIOS

Cruce con el jurado

Criterio (peso)	Evidencia clave
Innovacion (30%)	Pipeline 11 skills + TwiML ACK + guardrails + evals
Impacto Social (30%)	WhatsApp audio no-lectores + multilingue ES/FR
Viabilidad (20%)	Desplegado, 96 tests, 0 EUR, Docker, /health
Presentacion (20%)	Demo vivo WhatsApp + WOW texto + WOW audio FR

SCRUM

Sprint Review

Checkpoints

Sprint	Fecha	Objetivo	Estado
S1	30 Ene	Planificacion	COMPLETADO
S2	6 Feb	Doc tecnico + repo	COMPLETADO
S3	13 Feb	MVP + doc v2	EN CURSO (hoy)
S4	20 Feb	Demo final	PROXIMO

Cambios vs Sprint 2

Sprint 2 decia	Sprint 3 corrige	Razon
Web + WhatsApp	Solo WhatsApp	Web en Sprint 4
Whisper small	Gemini Flash STT	RAM Render free tier
HuggingFace Spaces	Render.com Docker	Decision de deploy
4 idiomas	2 probados (ES, FR)	Solo 2 con tests/evals
Mockup	URL real desplegada	Producto EXISTE
0 tests	96 tests	Desarrollo Sprint 3

Herramientas del equipo

Herramienta	Quien	Para que
Claude Code	Robert, Marcos	Dev, arquitectura, testing
Gemini 1.5 Flash	Producto	LLM + transcripcion audio
Perplexity	Lucas	Investigacion tramites
Figma	Andrea	Wireframes, slides
GitHub	Equipo	Repo + Issues (16 commits)
Docker	Robert, Marcos	Containerizacion
Render	Robert, Marcos	Deploy produccion
Notion	Andrea, equipo	Dashboard (81 entradas, 3 DBs)

Organizacion del equipo

Persona	Rol	Sprint 3
Robert	PM + Backend	Pipeline, deploy, demo presenter
Marcos	Routes + Twilio	Audio pipeline, webhook, Render
Lucas	KB + Testing	Investigacion, datos demo
Daniel	Web + Video	Canal web (Sprint 4), video backup
Andrea	Notion + Slides	Dashboard, presentacion, docs
