BARRIOLINK

David Zurita

2025

Contenido

[Abstract 1](#_Toc208862235)

[Resumen 1](#_Toc208862236)

[Descripción del proyecto APT 2](#_Toc208862237)

[Relación del proyecto APT con las competencias del perfil de egreso 3](#_Toc208862238)

[Relación del proyecto con tus intereses profesionales 3](#_Toc208862239)

[Argumento de factibilidad dentro de la asignatura 4](#_Toc208862240)

[Objetivos 5](#_Toc208862241)

[Objetivo general 5](#_Toc208862242)

[Objetivos específicos 5](#_Toc208862243)

[Propuesta metodológica de trabajo 5](#_Toc208862244)

[Principios 6](#_Toc208862245)

[Fases y criterios de salida 6](#_Toc208862246)

[Plan de trabajo para el proyecto APT 6](#_Toc208862247)

[Propuesta de evidencias de logro 7](#_Toc208862248)

[Criterios de verificación 7](#_Toc208862249)

[Formato de entrega 7](#_Toc208862250)

[Individual Conclusions 7](#_Toc208862251)

[Reflection 8](#_Toc208862252)

# Abstract

This report presents **BarrioLink**, a cloud-based information system that modernizes the management of neighborhood councils and their procedures and facilitates interaction between residents and their leadership. The Minimum Viable Product includes neighbor registration, residence certificates (PDF with email delivery), common space reservations with overlap control, project submissions, and news. The solution adopts modular monolith architecture, consisting of a Django + DRF API and a public Angular frontend, all deployed on Azure using services such as Static Web Apps, App Service, and PostgreSQL. A phase-based Waterfall methodology is used to ensure a predictable delivery by early December 2025, with formal evidence for each phase (requirements, architecture, testing, deployment). The results demonstrate feasibility within the academic context and alignment with graduation-profile competencies: requirements analysis, design, implementation, testing, ethics, and teamwork.

# Resumen

Este informe presenta **BarrioLink**, un sistema de información en la nube que moderniza la gestión de las juntas de vecinos y sus trámites, y facilita la interacción entre residentes y directiva. El Producto Mínimo Viable contempla: inscripción de vecinos, certificados de residencia (PDF + envío por email), reservas de espacios con control de solapamiento, postulaciones de proyectos y noticias. La solución utiliza un patrón de arquitectura monolito modular, la cual se compone de una API en Django + DRF (API Django + DRF y un frontend público Angular, todo esto desplegado en Azure y utilizando recursos como Static Web Apps, App Service y PostgreSQL. Se adopta una metodología de trabajo en Cascada por fases para asegurar una entrega predecible a inicios de diciembre de 2025, con evidencias formales por fase (requisitos, arquitectura, pruebas, despliegue). Los resultados demuestran factibilidad en el marco académico y alineación con competencias del perfil de egreso: análisis de requisitos, diseño, implementación, pruebas, ética y trabajo colaborativo.

# Descripción del proyecto APT

BarrioLink es una iniciativa para ordenar y transparentar la vida comunitaria. Busca simplificar trámites y mejorar la comunicación entre vecinos y directivas, de modo que cualquier persona pueda gestionar sus solicitudes desde un punto único, con estados y plazos claros, y que la directiva cuente con registros confiables para decidir y rendir cuentas.

El MVP prioriza cinco necesidades concretas: inscripción de vecinos; certificados de residencia con folio verificable y entrega por correo; reservas de espacios sin solapamientos; postulación de proyectos con seguimiento; y noticias/avisos en un canal oficial.

Nuestros resultados esperados son: reducción de tiempos de respuesta, trazabilidad de decisiones, cero dobles reservas, mayor alcance informativo y continuidad organizacional pese al recambio de directivas.

Cómo lo haremos:

* Arquitectura: monolito modular con una API REST única que aplica reglas y controla accesos.
* Canales: web pública para vecinos y back-office para directiva (aprobaciones y gestión).
* Datos y evidencias: base de datos con historial y folios; generación de PDF y notificaciones por email.
* Despliegue: en la nube (Azure) para disponibilidad y simplicidad operativa.
* Calidad: pruebas por flujo clave y documentación breve de uso y operación.

# Relación del proyecto APT con las competencias del perfil de egreso

Este proyecto se alinea con el perfil de egreso porque articula el ciclo completo de ingeniería de software orientado a un escenario real: desde la identificación de necesidades y el levantamiento de información con actores comunitarios, hasta la traducción de esas necesidades en requisitos claros, medibles y priorizados. La problemática social (trámites dispersos, baja trazabilidad y comunicación desigual) exige competencias de análisis, empatía con el usuario y pensamiento crítico para definir alcance, criterios de éxito y riesgos, siempre considerando principios éticos, resguardo de datos personales y comunicación clara.

En la dimensión técnica, el proyecto permite aplicar diseño y arquitectura de soluciones, modelado de datos, especificación de interfaces y verificación sistemática mediante planes de prueba (funcionales y de aceptación). Se promueve el uso responsable de tecnologías actuales y escalables, privilegiando buenas prácticas de calidad: control de versiones, documentación útil para usuarios y operación, gestión de incidencias y trazabilidad entre requisitos, casos de prueba y evidencias. Estas actividades fortalecen la capacidad de integrar conocimiento técnico con criterios de mantenibilidad, seguridad y experiencia de usuario.

Finalmente, el trabajo en equipo es central para el desarrollo profesional: la planificación por fases, la asignación de roles complementarios, las revisiones de pares y la toma de decisiones informada fomentan colaboración, liderazgo y comunicación efectiva. Cada idea se contrasta colectivamente con los objetivos del proyecto y con las restricciones de tiempo y recursos, reforzando la responsabilidad individual y el compromiso con el resultado colectivo. En conjunto, el proyecto ofrece un espacio auténtico para demostrar competencias de análisis, diseño, implementación, validación, ética y trabajo colaborativo propias del perfil de egreso.

# Relación del proyecto con tus intereses profesionales

Mi principal interés profesional es concretar soluciones tangibles que resuelvan problemas reales. Este proyecto me permite entregar un MVP operativo, desplegado y verificable por usuarios no técnicos, con métricas claras de éxito (tiempos de respuesta, trazabilidad de trámites, satisfacción). Ese resultado visible no solo valida mis capacidades técnicas, sino que también refuerza una mentalidad de producto: priorizar lo esencial, entregar valor pronto y mejorar de forma continua.

También me motiva que el trabajo tenga impacto social. Al ordenar trámites, transparentar decisiones y unificar la comunicación, el proyecto aporta a la vida comunitaria y a la confianza en la organización vecinal. Esto me permite ejercitar competencias de análisis con enfoque humano, comunicación clara y ética de datos personales, habilidades que considero fundamentales para mi desarrollo profesional.

Por último, busco que lo que construya sea escalable y sostenible, para seguir desarrollándolo y aprendiendo. El proyecto me da un marco para practicar diseño modular, pruebas y despliegue continuo, y para planificar un *roadmap* posterior (mejoras de accesibilidad, analítica, nuevas capacidades). Aprendo tanto de la dimensión técnica (diseño, arquitectura, calidad) como de la colaboración con *stakeholders* y el trabajo en equipo, preparando el terreno para evolucionar el sistema y mi propia trayectoria profesional.

# Argumento de factibilidad dentro de la asignatura

El proyecto es factible dentro de la asignatura porque el alcance está acotado a un MVP con cinco capacidades nucleares (inscripción, certificados en PDF con envío por email, reservas con anti-solapamiento, postulaciones y noticias), todas implementables con un monolito modular y back-office en Django Admin. Esta elección reduce drásticamente el esfuerzo de interfaz interna y nos permite concentrar el trabajo en reglas de negocio y trazabilidad de trámites, sin dispersarnos en múltiples frentes.

En términos de plazo y recursos, la planificación en Cascada por fases distribuye las actividades en hitos verificables hasta la primera semana de diciembre de 2025. El equipo de dos personas trabajará por verticales independientes (Certificados/Notificaciones; Reservas/Noticias/Proyectos), minimizando dependencias y permitiendo avances paralelos medibles en cada fase.

La infraestructura PaaS en Azure (Static Web Apps, App Service y PostgreSQL Flexible Server) y el uso de servicios gratuitos/educativos disminuyen la complejidad operativa, lo que libera tiempo para pruebas y documentación. Además, el enfoque de evidencias por fase se alinea con los criterios de evaluación de la asignatura y facilita el control del avance.

Los riesgos conocidos (alcance, carga de notificaciones, brecha entre organizaciones) están contenidos: mantenemos WhatsApp e integraciones externas fuera de v1, usamos email SMTP como canal único, y aseguramos separación por organización a nivel de aplicación. Con estas decisiones, un producto operable y demostrable en la nube —con métricas básicas de desempeño y aceptación— es alcanzable dentro del tiempo y las capacidades del equipo.

# Objetivos

## Objetivo general

Diseñar, construir y desplegar un MVP en la nube que digitalice trámites y comunicaciones de juntas de vecinos, mejorando tiempos de respuesta, trazabilidad y acceso para residentes y directivas antes de la primera semana de diciembre de 2025.

## Objetivos específicos

* Implementar autenticación por roles (Vecino/Directiva).
* Entregar certificados de residencia end-to-end: solicitud, aprobación, PDF, email.
* Habilitar reservas de espacios con validación de conflictos; 0 reservas solapadas aprobadas en pruebas.
* Publicar noticias/avisos con estado y fecha, visibles en la web pública.
* Registrar postulaciones de proyectos con estados y trazabilidad básica.
* Desplegar en Azure (Static Web Apps, App Service y PostgreSQL) con URLs públicas, health checks y respaldo básico de la base de datos.
* Asegurar calidad: plan SIT y UAT; al menos 1 prueba E2E por flujo y resolución de defectos críticos antes de la entrega.
* Mejorar usabilidad y accesibilidad básica: formularios validados, mensajes claros y compatibilidad móvil.
* Documentar: manual de usuario y técnico, diagrama de arquitectura, guía de datos de demostración.
* Generar evidencias formales por fase y cumplimiento de hitos en calendario.

# Propuesta metodológica de trabajo

Se adoptará Cascada por fases con hitos y criterios de salida definidos por fase, para asegurar trazabilidad, control de alcance y una entrega predecible.

## Principios

* Alcance acotado (MVP) y cambios controlados (solicitud, impacto, aprobación).
* Calidad integrada: criterios de aceptación por requisito, pruebas planificadas y revisión por pares.
* Evidencia por fase: cada fase cierra con artefactos verificables.

## Fases y criterios de salida

* Inicio y Requisitos: levantamiento, alcance, criterios de aceptación, riesgos.  
  *Criterio de salida*: Acta y especificación de requisitos aprobadas.
* Análisis y Diseño: modelo de dominio/ERD, arquitectura, GUI/Wireframes, plan de pruebas.  
  *Criterio de salida*: Diseño aprobado y base de datos modelada.
* Construcción: implementación del MVP por verticales (certificados; reservas/noticias/proyectos), datos de prueba.  
  *Criterio de salida*: Funcionalidades completas en entorno de prueba.
* Integración más Pruebas de Sistema (SIT): pruebas end-to-end y corrección de defectos críticos.  
  *Criterio de salida*: Reporte SIT aprobado sin críticos abiertos.
* Aceptación (UAT): ejecución de casos con usuarios objetivo y ajustes.  
  *Criterio de salida*: Acta UAT aprobada.
* Despliegue, Documentación y Presentación: publicación en la nube, manuales y demo.  
  *Criterio de salida*: URLs operativas, manuales listos y presentación final.

# Plan de trabajo para el proyecto APT

1. Hito Diseño aprobado: S5 (inicio septiembre).
2. Desarrollo de la aplicación: S5–S11 (sep–oct).
3. Pruebas de usabilidad y funcionalidad (SIT): S12–S15 (oct–nov).
4. Hito Pruebas aprobadas: S16 (nov).
5. Validación con la comunidad (UAT): S15–S16 (nov).
6. Documentación final: S16–S17 (nov–dic).
7. Entrega final: S18 (dic).
8. Levantamiento de requerimientos: S1–S2 (ago).
9. Diseño (diseño e implementación del diseño): S3–S4 (ago).

# Propuesta de evidencias de logro

* Inicio y Requisitos: Acta de proyecto; Especificación de requisitos (incluye objetivos, alcance v1, criterios de aceptación, riesgos); matriz de interesados.
* Análisis y Diseño: Diagrama de arquitectura (alto nivel y 4+1 lite); modelo de dominio/ERD; wireframes GUI; plan de pruebas (SIT/UAT).
* Construcción: Registro de avances (bitácora/commits); guía de datos semilla; evidencia de funcionalidades del MVP en entorno de prueba (capturas y breve guion de prueba).
* UAT: Casos de aceptación; acta de aprobación firmada; registro de observaciones y ajustes realizados.
* Despliegue, Documentación y Presentación: URLs operativas (front y back), credenciales de demo; manual de usuario; manual técnico (arquitectura, operación, respaldo); video/demo; presentación final.

## Criterios de verificación

* Certificado emitido en ≤ 2 min (de solicitud a PDF enviado).
* 0 reservas solapadas aprobadas en pruebas.
* ≥ 98% de correos entregados en entorno de pruebas.
* Todos los requisitos del MVP con evidencia (capturas/guiones) y trazabilidad a casos de prueba.

## Formato de entrega

* Un repositorio/documento con carpeta por fase y un índice que vincule cada evidencia.
* Documentos en PDF/URL institucional, con fecha y versión.

# Individual Conclusions

I believe a modular monolith on Azure strikes the right balance between speed and maintainability for a two-person team.

Formal phase gates (requirements, design, SIT, UAT) reduces rework and stabilizes the scope.

Generating PDFs and sending email notifications from the same service will simplify operations and user support. We considered WhatsApp notifications but dropped them early due to the tight timeline.

The project validates my skills in API design, database modeling, and deployment automation.

Next steps will focus on accessibility and telemetry to improve user experience and support. We believe the project could also benefit other communities—not only neighborhoods but also social and sports clubs and other large community groups.

# Reflection

Adopting a cloud PaaS stack and a modular monolith will help us keep cognitive load low while we practice professional engineering habits—version control, CI/CD, documentation, and testing. The main trade-off will be de-scoping advanced features (e.g., WhatsApp, multi-tenant RLS, rich calendars) to protect the schedule. We will articulate scope with measurable outcomes, design for change via a clean separation of API/domain/infra, and produce stakeholder-friendly artifacts (context, 4+1 lite, acceptance tests). Most importantly, the project will foreground ethical and social considerations by improving transparency and service quality for community members.