| **Definition of Done (DOD)** | | | | |
| --- | --- | --- | --- | --- |
| N° | Checklist | | | (Si/No) |
| 1 | El código generado en cada módulo cumple con los principios de modularidad, bajo acoplamiento y alta cohesión para asegurar la escalabilidad y mantenibilidad de Viaja Pro. | | | Si |
| 2 | Todos los miembros del equipo configuraron y trabajaron en entornos de desarrollo estándar, usando herramientas como Visual Studio Code, Git, y Angular CLI. | | | Si |
| 3 | Se utilizaron tecnologías definidas para el proyecto, incluyendo Angular para el frontend, Firebase para autenticación y base de datos en tiempo real, y Khipu para el sistema de pagos. | | | Si |
| 4 | El código cumple con los criterios de versionamiento mediante la utilización de Git, manteniendo un historial de commits claro y descriptivo. | | | Si |
| 5 | Se realizaron pruebas de calidad, incluyendo pruebas unitarias e integración, para cada módulo de *Viaja Pro*, utilizando frameworks como Jasmine y Karma para Angular. | | | Si |
| 6 | La arquitectura de la infraestructura (Angular + Firebase + Khipu + Google Maps API) está documentada, junto con los procedimientos de despliegue y configuraciones de red y seguridad. | | | Si |
| 7 | Se ha realizado una revisión de rendimiento para asegurar que las nuevas implementaciones no degraden el rendimiento general de *Viaja Pro*. | | | Si |
| 8 | Se han implementado las sugerencias y retroalimentación de los usuarios y stakeholders en cada entrega. | | | Si |
| 9 | La documentación del código, incluyendo comentarios actualizados, refleja de manera precisa el funcionamiento de las nuevas implementaciones. | | | Si |
| 10 | Se realizaron revisiones de código por al menos dos miembros del equipo antes de marcar una tarea como completada, usando GitHub para gestionar los Pull Requests. | | | Si |
| 11 | Se realizaron pruebas de seguridad en Firebase y Khipu, asegurando el manejo seguro de datos de autenticación y pagos. | | | Si |
| 12 | Las credenciales y claves API para servicios como Firebase, Khipu, y Google Maps están protegidas mediante prácticas de cifrado y almacenamiento seguro. | | | Si |
| 13 | Se revisó el cumplimiento de políticas de privacidad y protección de datos, asegurando conformidad con normativas aplicables, especialmente en la gestión de datos de Firebase y Khipu. | | | Si |
| 14 | Se verificó que la implementación cumple con todos los requisitos definidos en la documentación de requisitos y casos de uso. | | | Si |
| 15 | El código sigue las convenciones de codificación establecidas para *Viaja Pro*, asegurando consistencia en todo el equipo. | | | Si |
| 16 | Se realizaron pruebas de seguridad utilizando herramientas como OWASP ZAP y revisión de vulnerabilidades específicas en Firebase y Khipu. | | | Si |
| 17 | Se configuró redundancia en Firebase para garantizar la alta disponibilidad de los datos y servicios de la aplicación. | | | Si |
| 18 | La infraestructura de Firebase se configura automáticamente en el entorno de producción, utilizando prácticas de despliegue continuo. | | | Si |
| 19 | Se utilizaron herramientas y frameworks específicos, como Firebase Authentication, Firestore, Angular, Google Maps API, y Khipu. | | | Si |
| 20 | Se implementó monitoreo proactivo de rendimiento y errores en tiempo real usando Firebase Crashlytics y Analytics. | | | Si |
| 21 | Se configuró el autoescalado de Firebase para gestionar la demanda variable de usuarios. | | | Si |
| 22 | Se realizaron revisiones de código periódicas con el equipo utilizando herramientas como GitHub y revisiones manuales en los Pull Requests. | | | Si |
| 23 | Se aplicaron principios de diseño que permiten agregar nuevas funcionalidades o módulos sin comprometer la integridad del código existente. | | | Si |
| 24 | La gestión de dependencias se realizó mediante herramientas de Node.js (npm) y Firebase SDK, asegurando la correcta administración de librerías y versiones. | | | Si |