



# Metodología RUP



# Introducción

Es un enfoque de desarrollo de software basado en procesos iterativos. Se enfoca en planificar y gestionar el desarrollo a través de ciclos repetitivos (iteraciones), asegurando calidad y cumplimiento de requisitos. RUP organiza el trabajo en cuatro fases



# Índice de CONTENIDOS



---

**01. Inicio**

---

**02. Elaboración**

---

**Construcción**

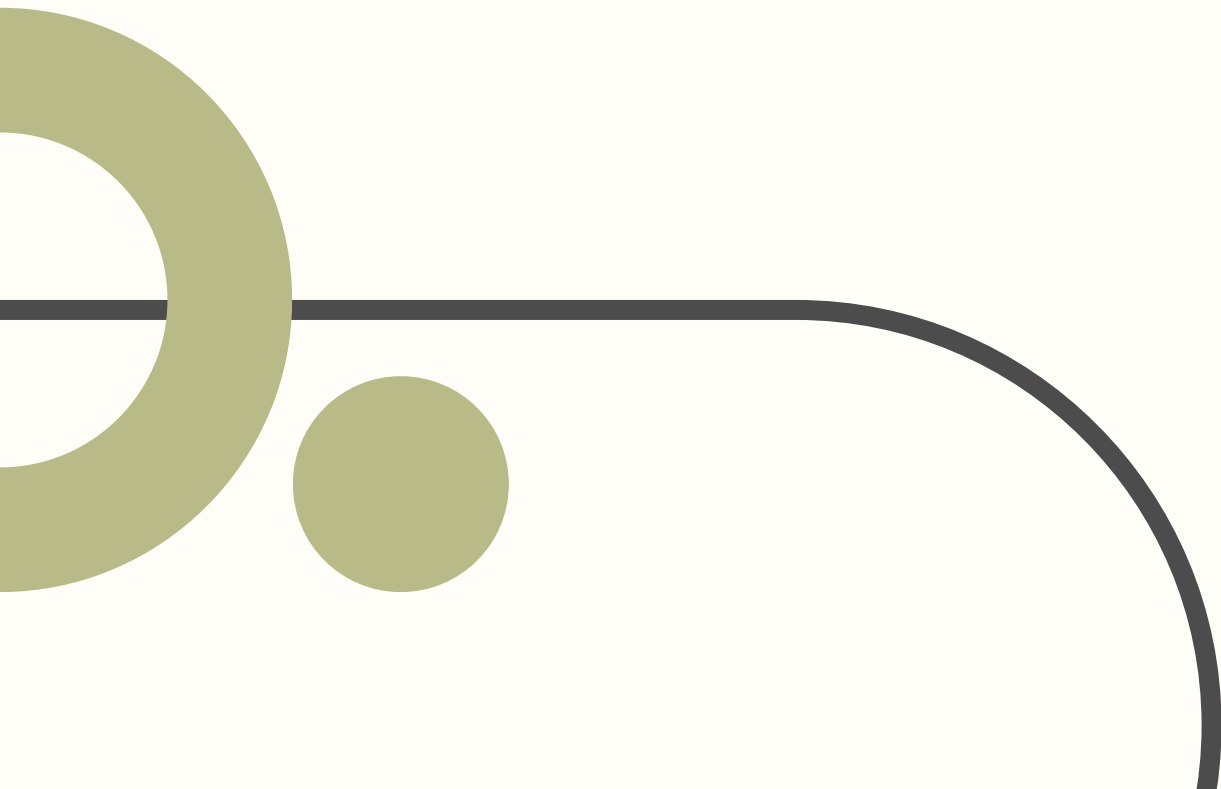
---

**04. Transición**

---

**05. Características principales**

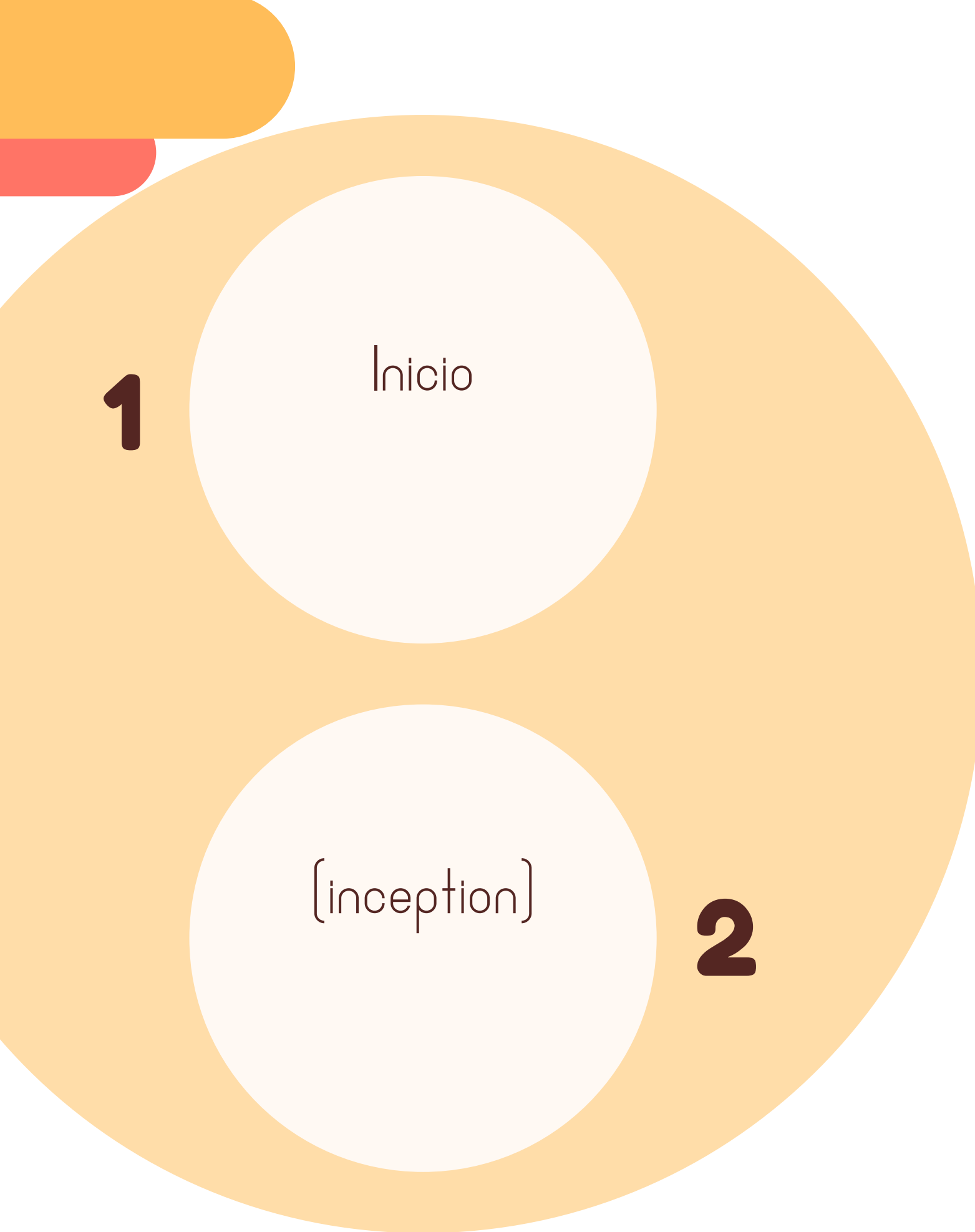
---



# Inicio

Esta fase establece las bases del proyecto. Se enfoca en definir los objetivos el alcance y la viabilidad del sistema a desarrollar

- Identificar a los interesados del proyecto (clientes, usuarios, equipo de desarrollo).
- Establecer los principales requisitos del sistema (funcionales y no funcionales).
- Evaluar riesgos iniciales y proponer estrategias para mitigarlos.
- Crear un caso de negocio para justificar la inversión.
- Elaborar una visión general del proyecto. Entregables típicos:
  - Documento de visión del proyecto.
  - Estimaciones iniciales de tiempo y costos.
  - Prototipos preliminares, si son necesarios para aclarar requisitos.






# Elaboración

Durante esta fase se profundiza en el análisis del sistema y se crea la arquitectura base para el proyecto. Es fundamental para mitigar riesgos técnicos y establecer la infraestructura necesaria.

Actividades clave:

- Analizar y documentar en mayor detalle los requisitos del sistema.
- Diseñar la arquitectura del software, identificando los componentes principales.
- Realizar pruebas iniciales en prototipos para evaluar la viabilidad de la arquitectura.
- Estimar riesgos técnicos y funcionales. Entregables típicos:
- Documento de diseño arquitectónico.
- Modelo detallado de casos de uso.
- Prototipo funcional o modelos de prueba.





Aquí se realiza la implementación y el desarrollo del sistema siguiendo las especificaciones y diseño establecidos. Se trabaja en ciclos iterativos para garantizar que se cumplan los requisitos.

Actividades clave:

- Codificar y construir los componentes del sistema.
- Realizar pruebas unitarias e integrales.
- Documentar el código y las funcionalidades desarrolladas.
- Gestionar configuraciones y cambios en el sistema. Entregables típicos:
- Sistema funcional (en desarrollo o completado, según la iteración).
- Planes de pruebas y resultados documentados.
- Manuales preliminares para usuarios y administradores.



# Construcción

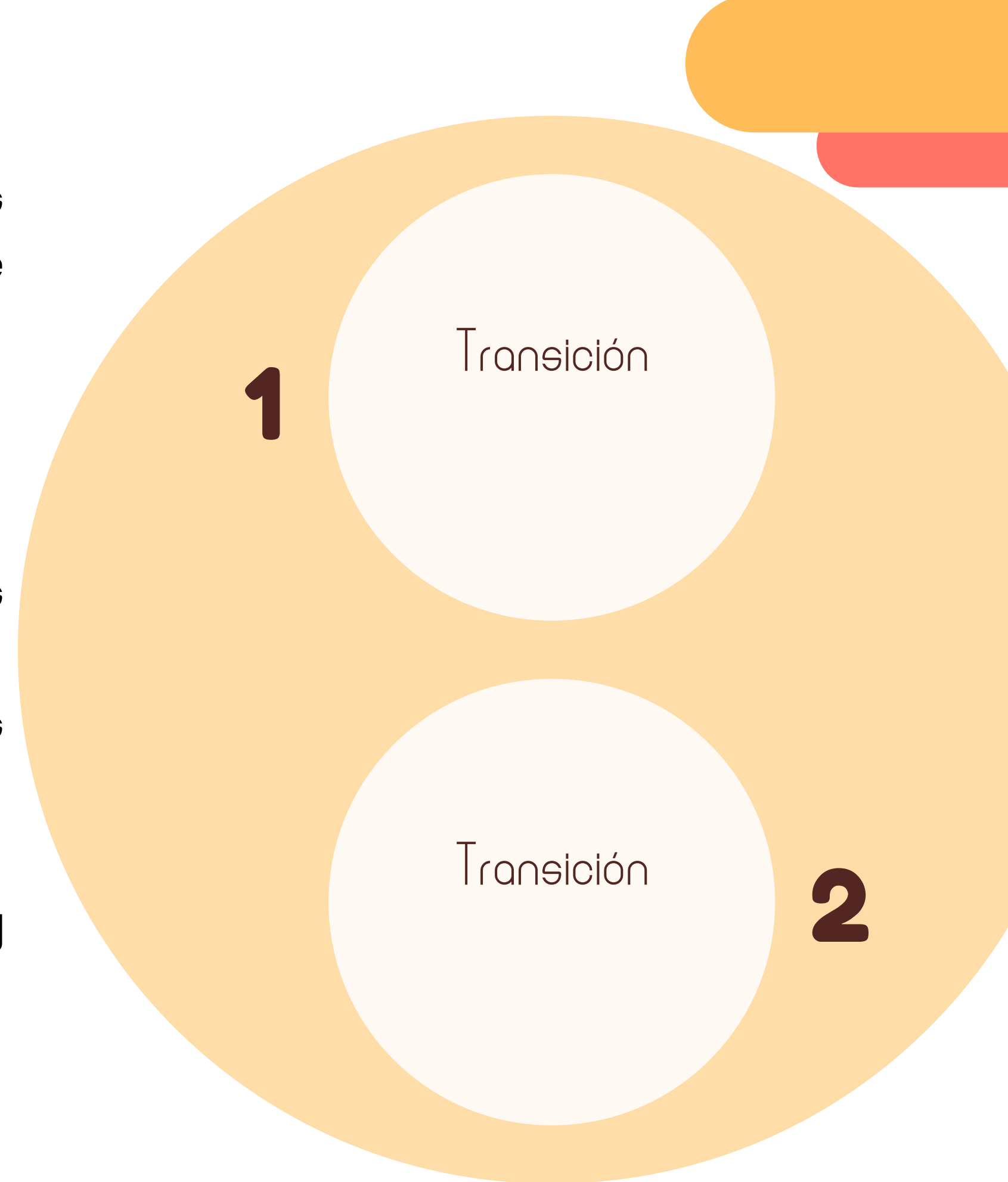


# Transición

La última fase se centra en la entrega del sistema a los usuarios finales. Se realizan ajustes finales basados en pruebas de aceptación y retroalimentación del cliente.

Actividades clave:

- Desplegar el sistema en el entorno de producción.
- Capacitar a los usuarios finales y administradores del sistema.
- Corregir errores y ajustar funcionalidades en función de las pruebas.
- Asegurar la documentación completa del sistema. Entregables típicos:
  - Sistema final en producción.
  - Documentación completa (manuales de usuario, administrador y técnico).
  - Informe de cierre del proyecto.



# Características

- Iterativo e incremental: El desarrollo se realiza en ciclos repetitivos, lo que permite entregar partes funcionales del sistema progresivamente.
- Basado en riesgos: Identifica y aborda los riesgos más críticos desde el inicio del proyecto.
- Enfoque en arquitectura: Define una arquitectura robusta desde las primeras fases.
- Uso de UML: Se utiliza el Lenguaje Unificado de Modelado (UML) para documentar los requisitos y el diseño del sistema.







**Gracias**