

# [SOP] Creación de proyectos de software (Aplicaciones y Servicios)

Procedimiento estándar para registrar, inicializar y dejar listo para desarrollo un nuevo proyecto de software (aplicación o servicio/API) del área.

- Contexto y lineamientos generales
  - Contexto y Alcance
  - Tipos de proyectos y Roles involucrados
  - Nomenclaturas y Convenciones
  - Consideraciones
- Inicialización técnica del proyecto
  - Inicialización técnica del proyecto

# Contexto y lineamientos generales

# Contexto y Alcance

## Propósito del procedimiento

Establecer el procedimiento estándar para la creación de nuevos proyectos de software (Aplicaciones o APIs/Servicios) en el área de desarrollo. Este procedimiento permite asegurar que cada proyecto queda correctamente registrado, identificado y inicializado antes de comenzar su desarrollo técnico.

El objetivo principal es mejorar la trazabilidad, estandarizar el arranque de proyectos y reducir ambigüedades en la fase inicial.

---

## Alcance del procedimiento

Este procedimiento aplica únicamente al desarrollo de software del área, en los siguientes casos:

- Desarrollo de **Aplicaciones**
- Desarrollo de **APIs/Servicios**

No aplica a prototipos, experimentos, pruebas de concepto ni a proyectos fuera del área de desarrollo.

---

## Responsable del registro

El registro del proyecto en la hoja de control administrativa es responsabilidad del **Líder de Proyecto**.

Este registro debe realizarse antes de crear el repositorio y antes de iniciar cualquier actividad técnica.

---

## Reinicio del consecutivo por año

El consecutivo interno se resetea cada año. Ejemplo:

- Año 2026 → inicia en `2026-0001`
- Año 2027 → inicia en `2027-0001`

# Tipos de proyectos y Roles involucrados

## Tipos de proyectos contemplados

### - Aplicaciones

Corresponden a sistemas con componentes UI (frontend) y backend, normalmente orientados a usuarios internos o externos.

### - APIs/Servicios

Corresponden a sistemas sin interfaz visual, orientados a exposición de servicios, integración con terceros o consumo interno.

---

## Roles involucrados en la fase inicial

Durante la fase de creación del proyecto intervienen los siguientes roles:

- **Líder de Proyecto:** responsable de iniciar el registro, asignar el nombre corto y seleccionar el template base.
- **Jefe de Área:** responsable de validar y autorizar la creación del proyecto cuando corresponda.

No se contempla participación de QA, usuarios funcionales o equipo de DevOps en esta fase.

# Nomenclaturas y Convenciones

## Nomenclatura de identificación del proyecto

Cada proyecto se identifica mediante un ID interno con la siguiente estructura:

Año-Consecutivo-Tipo-NombreCorto

Donde:

- **Año:** año de creación del proyecto (generalmente coincide con el año de solicitud).
- **Consecutivo:** número incremental que se reinicia cada año (ejemplo: `0001`).
- **Tipo:** categoría del proyecto. Especialmente cuando son servicios o apis.
- **NombreCorto:** identificador del proyecto en formato PascalCase.

Ejemplos por tipo:

- Aplicaciones:
  - 2026-0001-SeguimientoBecarios
- APIs/Servicios:
  - 2026-0001-Srv-NominaVehicular
  - 2026-0002-Srv-PagosCentral

---

## Convención sobre el NombreCorto

El campo **NombreCorto** debe cumplir:

- Formato **PascalCase**
- Representar el nombre lógico del sistema
- Evitar acentos, espacios y caracteres especiales
- Mantener consistencia respecto al dominio funcional del sistema

Ejemplo de conversión:

- Nombre funcional: Sistema de Seguimiento de Becarios
- NombreCorto: SeguimientoBecarios

# Convención para nombres de repositorios y contenedores

Para mantener la trazabilidad entre el registro administrativo y los artefactos técnicos, se establece la siguiente convención:

- El **nombre del repositorio Git** en GitHub debe ser exactamente igual al **ID del proyecto**.
- El **nombre del contenedor Docker** también debe ser exactamente igual al **ID del proyecto**.

Es decir, se reutiliza el mismo identificador en los tres niveles:

1. Registro administrativo (hoja de control)
2. Repositorio de código fuente
3. Contenedor en la plataforma de ejecución

## - Ejemplos

### Aplicación

- ID: `2026-0001-AplicacionSeguimientoBecarios`
- Repositorio GitHub: `2026-0001-AplicacionSeguimientoBecarios`
- Nombre de contenedor Docker: `2026-0001-AplicacionSeguimientoBecarios`

### API/Servicio

- ID: `2026-0002-Srv-PagosCentral`
- Repositorio GitHub: `2026-0002-Srv-PagosCentral`
- Nombre de contenedor Docker: `2026-0002-Srv-PagosCentral`

## - Consideraciones

- El **ID no debe alterarse** ni abreviarse al crear el repositorio o el contenedor.
- Si un desarrollador decide clonar el repositorio en una carpeta local con otro nombre, esto no afecta la convención oficial, pero se recomienda mantener el mismo nombre para evitar confusiones.
- Cualquier automatización futura (scripts internos, n8n, herramientas de generación de proyectos, etc.) debe asumir que el **ID es la fuente de verdad** para nombrar

repositorios y contenedores.

# Consideraciones

## Objetivo técnico de esta etapa

Al finalizar esta etapa el proyecto tiene:

1. Identificador único asignado
2. Tipo definido (Aplicación o API/Servicio)
3. NombreCorto definido
4. Año y consecutivo asignado
5. Responsable técnico asignado

Este es el estado mínimo requerido para pasar a la etapa de inicialización técnica.

# Inicialización técnica del proyecto

# Inicialización técnica del proyecto

## Objetivo de la fase

Dejar el proyecto listo para iniciar desarrollo técnico, con su registro, repositorio, entorno local y documentación mínima.

---

## Registro del proyecto en hoja de control

Antes de crear el repositorio se debe registrar el proyecto en la hoja de control del año en curso.

**Responsable:** Líder de Proyecto

**Estado esperado:**

Campos mínimos a completar:

- ID del proyecto
- Nombre funcional
- Nombre corto
- Tipo (Aplicación o Srv)
- Año
- Consecutivo
- Responsable técnico
- Fecha de registro
- Estado inicial

La url de la hoja de Google para el registro es: [SEGUIMIENTO DE PROYECTOS](#)

---

## Creación del repositorio

El repositorio debe crearse en [GitHub del Departamento](#) utilizando como **nombre del repositorio** el **ID del proyecto** asignado en el registro administrativo.

Ejemplo: `2026-0002-Srv-PagosCentral`

Se deberá establecer en modo **Privado** el repositorio y contener una **Descripción**.

---

## Selección y descarga de template base

Dependiendo del tipo de proyecto se debe seleccionar el template base correspondiente y utilizarlo como punto de partida del código:

- **Para Aplicaciones Normales:** <https://github.com/Desarrollo-Soluciones-Informaticas/AIIDTProyectoBase>
- **Para APIs:** <https://github.com/Desarrollo-Soluciones-Informaticas/AIIDTProyectoBase-API>

**Nota: El template base debe contener al menos:**

- Estructura inicial del proyecto
  - Archivos `.env.example`
  - `docker-compose.yml` básico
  - Dependencias mínimas
- 

## Configuración inicial del entorno local

Al crear el proyecto se debe:

- Generar o copiar archivo `.env` a partir de `.env.example`
  - Instalar dependencias (`composer install`, `npm install` o equivalente)
  - Confirmar que el proyecto levanta en modo local sin errores
- 

## Convenciones técnicas para nombres

El **nombre del contenedor Docker** debe ser igual al ID del proyecto.

Ejemplo:

**container\_name:** `2026-0002-Srv-PagosCentral`

Esto asegura trazabilidad entre registro, repositorio y runtime.

---

# Documentación inicial en el repositorio

Cada proyecto debe contar con un `README.md` inicial que incluya:

- Nombre del proyecto
- Descripción funcional corta
- Tipo (Aplicación o Srv)
- Stack tecnológico
- Cómo levantar el entorno local
- Cómo configurar variables de entorno
- Dependencias externas si aplica

Este README puede ser ampliado conforme avance el proyecto.

---

## Resultado de la fase

La fase de inicialización técnica se considera completa cuando el proyecto cumple:

- Proyecto registrado
- Repositorio creado
- Template aplicado
- Local levantando en contenedores
- README inicial existente

El proyecto está entonces **listo para desarrollo**.