**Planificación de Sprint 1 (duración sugerida: 2 semanas)**

**1) Objetivo medible del Sprint + Criterios de aceptación**

**Objetivo de Sprint.** Entregar un **baseline reproducible** del flujo *Ingesta → Preprocesamiento → Feature Engineering → Entrenamiento → Evaluación* con dataset inicial consolidado, *pipeline* encapsulado y reporte de métricas, listo para revisión institucional.

**Criterios de aceptación (DoD) — se consideran cumplidos cuando:**

1. **Repositorio** creado con ramas (*main*, *develop*), protección de *branch* y CI mínima (lint + tests).
2. **Esqueleto de *pipeline*** implementado (p. ej., sklearn.pipeline.Pipeline o TFX *skeleton*) que encadene: limpieza básica, vectorización/rasgos y estimador final; ejecutable con un solo comando (*Makefile* o *script*).
3. **Datos**: *schema* inferido y validado; *split* temporal **train/valid/test**; trazabilidad de origen (metadatos).
4. **Métricas**: reporte con **F1** (clase “riesgo”), **precision/recall** y **AUC-ROC** en *valid* + matriz de confusión; *card* de métricas versionada.
5. **Trazabilidad**: *README* con pasos de ejecución, dependencias, parámetros y decisiones; *issue* de “cerrado” que referencia *commit* y *metrics card*.

**Sustento metodológico.** El *time-box* y el *Incremento usable* responden a Scrum (*“los sprints son eventos de longitud fija… y contienen los demás eventos”*), mientras que el encadenamiento reproducible de pasos está formalizado por **Pipelines** (scikit-learn) y/o componentes **TFX**; la promoción del modelo exige *gates* de evaluación antes del *serving* (Schwaber & Sutherland, 2020; scikit-learn, s. f.; Baylor et al., 2017; Google Cloud, 2024).

**2) Asignación de tareas y responsables (RACI simplificado)**

| **ID** | **Tarea (entregable concreto)** | **Descripción técnica** | **Responsable (R)** | **Apoya (A)** | **Informa (I)** | **Aprueba (C)** |
| --- | --- | --- | --- | --- | --- | --- |
| T1 | **Repositorio y CI mínima** | Crear repo, ramas, *pre-commit*, black, flake8, *workflow* de lint/test | **Fernando García** | Hilario Aradiel | Asesor/UPG | Equipo |
| T2 | **Contrato de datos + *schema*** | Ingesta inicial (SEACE/SIAF/SIGA mock o muestreo); inferencia y validación de *schema*; metadatos | **Hilario Aradiel** | Fernando García | Asesor/UPG | Equipo |
| T3 | ***Pipeline* baseline** | Pipeline(prepro → features → modelo) con TF-IDF + Regresión Logística (o SVM) | **Fernando García** | Hilario Aradiel | Asesor/UPG | Equipo |
| T4 | **Partición y control de *leakage*** | *Split* temporal 70/15/15; transformaciones dentro del *pipeline* | **Hilario Aradiel** | Fernando García | Asesor/UPG | Equipo |
| T5 | **Reporte de métricas** | F1/precision/recall/AUC en *valid*; matriz de confusión; *metrics card* versionada | **Fernando García** | Hilario Aradiel | Asesor/UPG | Equipo |
| T6 | **Documentación y cierre** | README, *changelog*, bitácora de decisiones, *issue* de cierre con enlace a *commit* y reporte | **Hilario Aradiel** | Fernando García | Asesor/UPG | Equipo |

**Nota operativa.** El *pipeline* empaqueta preprocesos y modelo para evitar **fuga de información** (*“avoid leaking statistics from your test data…”*), y TFX/TFMA pueden incorporarse en sprints siguientes para *gates* de promoción (scikit-learn, s. f.; Baylor et al., 2017).

**3) Uso de tableros (GitHub Projects/Issues)**

**Estructura del tablero (kanban):** *Backlog* → *To do* → *In progress* → *Code Review* → *Validation* → *Done*.  
**Convenciones de Issues:**

* **Plantilla** con: *Objetivo*, *Criterios de aceptación (Given–When–Then)*, *Definición de Hecho*, *Riesgos*, *Referencias*.
* **Labels**: data, pipeline, metrics, docs, infra, blocked.
* **Milestone**: Sprint-1 (2w); automatizar paso a *Done* al cerrar *PR* con *checks* verdes.

**Ejemplo de *Issue* (T3 – Pipeline baseline)**

* *Objetivo*: Encapsular prepro → features → clasificador para tarea “riesgo/no riesgo”.
* *Criterios de aceptación*:
  + **Given** dataset v0.1 y *schema* validado, **when** ejecuto make train, **then** se guarda model.joblib y metrics.json en artifacts/.
  + **Given** *split* fijo, **when** ejecuto make evaluate, **then** obtengo F1 ≥ 0.70 en *valid* con reporte reproducible.
* *DoD*: *pipeline* serializado + *README* actualizado + *PR* aprobado.

**Justificación.** Scrum promueve objetivos claros por Sprint y transparencia del trabajo; los tableros e *issues* operacionalizan esa transparencia y la inspección/adaptación en ciclos cortos (Schwaber & Sutherland, 2020).

**4) Entregables del Sprint 1**

1. **Repositorio** con CI mínima y *hooks* de calidad.
2. **Datos iniciales** con *schema* y *split* documentados.
3. **Pipeline baseline** reproducible.
4. **Reporte de métricas** (F1, precision, recall, AUC; matriz de confusión) y **bitácora de decisiones**.
5. **Tablero** con *issues* cerrados y *PRs* aprobados.

**5) Métricas de seguimiento del Sprint**

* **De proceso**: *Throughput* (issues cerrados), *Lead time* por issue, % *PRs* aprobados al primer intento.
* **De producto**: F1 en *valid* (objetivo ≥ 0.70 en Sprint 1), varianza inter-fold, estabilidad por segmentos (sector/distrito).
* **De calidad**: cobertura mínima de tests en utilidades de datos; 0 *lint errors* en *main*.

**6) Riesgos y mitigaciones (Sprint 1)**

* **Datos heterogéneos o incompletos** → *Mitigación*: contrato de *schema* y validaciones tempranas; *mock datasets* para desbloquear *pipeline*.
* **Deriva de requisitos** → *Mitigación*: *review* de mitad de Sprint con *stakeholders* y ajuste controlado del *scope*.
* **Deuda técnica** → *Mitigación*: *pre-commit* + CI obligatoria; definición explícita de DoD.

**Cómo insertar en su documento**

**Ubicación sugerida:** Cap. 4.1.5 (*Integración de la metodología*) o **Anexo Operativo — Sprint 1**.  
**Referencia cruzada:** vincular con 4.2.1 (*Diseño del Pipeline de Datos*) y 3.3.4 (*Métricas*).

Plantilla\_Plan\_Tesis\_IA-Aradiel…

**Referencias (APA 7)**

Baylor, D., Breck, E., Cheng, H.-T., Fiedel, N., Polyzotis, N., Zinkevich, M., & Whang, S. (2017). TFX: A TensorFlow-Based Production-Scale Machine Learning Platform. *KDD ’17*. [https://dl.acm.org/doi/10.1145/3097983.3098021](https://dl.acm.org/doi/10.1145/3097983.3098021?utm_source=chatgpt.com)

Google Cloud. (2024, agosto 28). *MLOps: Continuous delivery and automation pipelines in machine learning*. [https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning](https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning?utm_source=chatgpt.com)

Schwaber, K., & Sutherland, J. (2020). *La Guía Scrum: La guía definitiva de Scrum* (v2020). [https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf](https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf?utm_source=chatgpt.com)

scikit-learn. (s. f.). *7.1. Pipelines and composite estimators*. [https://scikit-learn.org/stable/modules/compose.html](https://scikit-learn.org/stable/modules/compose.html?utm_source=chatgpt.com)

**¿Desea que le entregue esto como un bloque editable para su Word (con tablas y estilos) o que lo convierta en una lista de *issues* iniciales (*CSV/Markdown*) para importar a GitHub Projects?**

Fuentes

Principio del formulario

Pensamiento ampliado

Final del formulario

ChatGPT puede cometer errores. Consider