

Funcionalidades														Proyecto			
Nombre	Distribución del procesamiento	Distribución del modelo o plan de entrenamiento	Uso de los datos	Gobierno y permissionado	Privacidad	Descubrimiento de recursos	Otras funcionalidades	Web	Repositorio	Documentación	Última versión	Sponsors	Partners	Inicio del proyecto			
	Mecanismos que permitan distribuir la ejecución (cómputo) del Plan de entrenamiento entre los Silos de datos.	Mecanismos que permitan distribuir la definición del propio Plan de entrenamiento (código) entre los silos de datos.	Mecanismos implementados para permitir el acceso y uso de los datos de un Silo de datos por parte del Data Scientist.	Mecanismos de aprobación y denegación o creación políticas de acceso a cualquier recurso que participe de la federación (silos, capacidad de cómputo, conjuntos de datos, datos, ...).	Técnicas o mecanismos que aseguren la privacidad de los datos de los Silos de datos, evitando posibles fugas de información por parte del Data Scientist.	Mecanismos que permitan a un Data Scientist descubrir nuevos Silos de datos o datasets.	Aspectos o funcionalidades destacables que no se hayan contemplado en otros parámetros.										
PySyft	Sí, el entrenamiento se ejecuta en cada Silo de datos, orquestado por PySyft.	Sí, el framework se encarga de distribuir el modelo planteado por el Data Scientist entre los silos de los datos que dicho modelo requiere.	El DataScientist hace uso de los datos por medio de referencias a nivel de datos a los mismos (punteros o vectores) que permiten restringir su acceso en claro.	Permite al DataOwner limitar los datos que el DataScientist puede ver en claro y permissionar el resultado del entrenamiento. Permite el gobierno de la red a nivel de Silo de datos.	Permite utilizar privacidad diferencial para visualizar datos privados (agregando ruido). Utiliza MPC para agregar los resultados de los entrenamientos de los silos y para permitir el cifrado del modelo durante el entrenamiento.	Permite descubrir otros silos y datasets que ya se encuentren en la misma red. Sin embargo, el punto de acceso a la red debe conocerse de manera previa.		Web	Repositorio	Documentación	0.6.0	OpenMined	Twitter, Google	2017			
Fate	Sí, el entrenamiento se ejecuta en cada Silo de datos, orquestado por el subproyecto FATE-Flow.	Sí, el framework se encarga de distribuir el modelo planteado por el Data Scientist entre los silos de los datos que dicho modelo requiere mediante el subproyecto FATE-Flow.	El DataScientist hace uso de los datos por medio de referencias a nivel de dataset a los mismos.	Disponibiliza un servicio llamado FATE-Cloud que permite gestionar los Silos de datos, gestionar la colaboración y seguridad entre Data Scientist y Data Owners. Se divide en dos subproyectos que permiten la gestión a distintos niveles: Federated Cloud (para la gestión general de la red de colaboración) y Federated Site (para la gestión local de cada Silo de datos).	Implementa múltiples algoritmos y técnicas PET que permiten realizar estadísticos, transformaciones y otras operaciones sobre los datos manteniendo la privacidad de los datos. Puedes consultar la lista completa aquí .	Permite descubrir otros Silos de datos disponibles en la red y obtener información sobre datasets conocidos, pero no ofrece descubrimiento de los propios datasets.	Disponibilizan una herramienta (FATE-Board) para monitorizar el proceso de entrenamiento de los modelos y visualizar resultados parciales del propio entrenamiento.	Web	Repositorio	Documentación	1.7.1.1	WeBank	VMWare	2019			
OpenFL	Sí, el entrenamiento se ejecuta en cada Silo de datos (dentro del nodo <i>Collaborator</i>) orquestado por el framework a través de las políticas definidas por el rol de <i>Director Manager</i> (Data Scientist).	Sí, el framework se encarga de transferir las tareas definidas por el Data Scientist (mediante el <i>Experiment Manager</i>) a los nodos <i>Collaborator</i> del Silo de datos previa aprobación del Data Owner (mediante el <i>Collaborator Manager</i>).	El DataScientist hace uso de los datos por medio de referencias a nivel de dataset a los mismos.	El identifiación entre los nodos se realiza mediante claves, pero OpenFL no se encarga de la generación, gestión o distribución de dichas claves. Existen tres roles definidos en su arquitectura, asignables en la ejecución de las operaciones y TLS cifrar las comunicaciones del entorno.	Utilizan entornos seguro de ejecución (Trusted Execution Environments) SGX para securizar la ejecución de las operaciones y TLS cifrar las comunicaciones del entorno.	Los nodos de cualquier tipo (Data Scientist o Silos de datos) participantes en la federación deben conocerse de forma previa a la definición. El Data Scientist si dispone de mecanismos para descubrir los datos y características que los Data Owners disponibilizand de sus Silos de datos.	Incluye el refinamiento de modelos locales a partir del proceso de agregación de los resultados parciales en un modelo nuevo, por lo que parece más orientado a fine-tuning de un modelo general para cada Silo de datos que al entrenamiento de un modelo común más robusto.	-	Repositorio	Documentación	1.2.1	Intel	-	2019			
Flower.dev	Sí, el entrenamiento se ejecuta en cada Silo de datos, orquestado por Flower.	No, el framework no realiza la distribución del modelo o plan de entrenamiento entre los silos o nodos de datos.	No provee mecanismos de acceso a los datos, ya que no abstrae la distribución del plan de entrenamiento que hace uso de los mismos, por tanto el Data Scientist debe conocer los datos de cada Silo de datos de forma previa.	No tiene ningún tipo de capa de gobierno o permissionado.	Recientemente han incorporado el soporte para utilizar Opacus, un framework compatible con PyTorch para entrenar modelos utilizando Diferencial Privacy.	El framework no cuenta con mecanismos de descubrimiento de silos de datos o datasets, solo se centra en facilitar la federación del entrenamiento.	Permite establecer estrategias de agregación personalizadas.	Web	Repositorio	Documentación	0.18.0	-	Samsung, Nokia	2020			
Tensor Flow Federated	No, orientado a la simulación de la federación en un mismo entorno.	No, orientado a la simulación de la federación en un mismo entorno, aunque ya tienen implementados mecanismos de serialización para distribuir los planes de entrenamiento en un futuro.	No provee mecanismos de acceso a los datos.	No provee mecanismos de gobierno ni permissionado.	No implementa ningún mecanismo de privacidad más allá de los algoritmos de agregación de parámetros más básicos, aunque en la versión <i>nightly</i> si incluye el uso de Privacidad Diferencial.	No provee mecanismos de descubrimiento de ningún tipo de recursos.	-	Web	Repositorio	Documentación	0.19.0	Google	-	2018			
Paddle FL	Sí, el entrenamiento se ejecuta en cada Silo de datos, orquestado por PaddleFL.	Sí, el plan de entrenamiento se distribuye a cada Silo de datos, orquestado por Flower.	El DataScientist hace uso de los datos por medio de referencias a los mismos.	No provee mecanismos de gobierno ni permissionado.	Implementa Multy Party Computation para las tareas de entrenamiento e inferencia y para tareas previas como realizar joins entre datasets federados sin revelar información.	No provee mecanismos de descubrimiento de ningún tipo de recursos.	-	-	Repositorio	Documentación	1.2.0	-	-	2019			
FEDn & STACKn	Sí, el entrenamiento se ejecuta en cada Silo de datos, orquestado por STACKn. y federado por FEDn	Sí, el plan de entrenamiento se distribuye a cada Silo de forma automática.	El DataScientist hace uso de los datos por medio de referencias a los mismos.	STACKn permite gestionar todos los Silos de datos, recursos de computación, los entornos de desarrollo, o los datos en si. Sin embargo no provee sistema de roles ni permissionado a bajo nivel, entiende que el entorno será gestionado de forma centralizada.	No, aunque en futuras iteraciones tienen planificado incluir mecanismos de Privacidad Diferencial y Multy Party Computation.	STACKn permite visibilizar todos los Silos de datos, recursos de computación, los entornos de desarrollo, o los datos en si.	STACKn disponibiliza entornos de desarrollo on-premise que soportan numerosos IDE's y entornos de ML, así como permite monitorizar a muy bajo nivel los procesos de entrenamiento e inferencia (tiene integración con Grafana y Prometheus por ejemplo).	Web	Repositorio (FEDn)	Documentación (FEDn)	0.3.1	ScaleOut Systems	Swiss, Saab, AstraZeneca	2020			
									Repositorio (STACKn)	Documentación (STACKn)	0.5.0						
FLARE	Sí, el entrenamiento se ejecuta en cada silo de datos, orquestado por Flare. Permite también evaluar el resultado del proceso de forma local y general.	Sí, el framework se encarga de distribuir el plan de entrenamiento definido por el Data Scientist entre los silos.	El DataScientist hace uso de los datos por medio de referencias (Shareable) que provee el framework similares a punteros a datos remotos.	Permite implementar políticas de acceso y administrar los Silos de datos (organizaciones) que se encuentran en la red. Sin embargo está muy orientado a generar un entorno gobernado por un único ente (Data Scientist + Data Owner) aunque sin acceso a los datos.	Facilitan el uso de algunos algoritmos de Privacidad diferencial de forma previa a compartir información entre partes. De igual forma, permite proteger datos con HE para que un tercero pueda operar con ellos sin ver el resultado.	No, su objetivo no es enfrentar este reto, solo se centra en facilitar la federación del entrenamiento y la agregación del resultado.	Permite ejecutar un subconjunto acotado de comandos de consola de forma remota, desde el DataScientist al Silo de datos. Soporta un ciclo de agregación entre Silos de datos, eliminando la necesidad de que sea el nodo Data Scientist quien realice la agregación.	Web	Repositorio	Documentación	2.0.14	NVIDIA	Azure, Roche	2021			