



PLATFORM DOCUMENTATION

CorePlexML

Versión 0.1.0

Rodrigo Henriquez M.

Generated on 09 de marzo de 2026

Índice general

1. 1. Cuenta y Onboarding	3
1.1. Objetivo	3
1.2. Para quien es	3
1.3. Precondiciones	3
1.4. Paso a paso	3
1.5. Checklist de validacion funcional	4
1.6. Resultado esperado	4
1.7. Errores comunes y recuperacion	4
1.8. Imagen	4
2. 2. Navegacion y Workspace	7
2.1. Objetivo	7
2.2. Semantica del workspace	7
2.3. Paso a paso	7
2.4. Checklist de validacion funcional	7
2.5. Resultado esperado	8
2.6. Errores comunes y recuperacion	8
2.7. Imagen	8
3. 3. Proyectos y Datasets	9
3.1. Objetivo	9
3.2. Precondiciones	9
3.3. Crear proyecto	9
3.4. Subir dataset	9
3.5. Validar calidad de datos	10
3.6. Checklist de validacion funcional	10
3.7. Resultado esperado	10
3.8. Errores comunes y recuperacion	10
3.9. Imagenes	10
4. 4. Dataset Builder AI	13
4.1. Objetivo	13
4.2. Que debe cumplir este modulo	13
4.3. Flujo conversacional realista	13
4.4. Si el resultado no es el esperado	14
4.5. Checklist de validacion funcional	14
4.6. Resultado esperado	14

4.7.	Errores comunes y recuperacion	14
4.8.	Imagen	14
5.	5. Experimentos, Modelos, Predicciones	17
5.1.	Objetivo	17
5.2.	Crear experimento	17
5.3.	Revisar resultados y modelos	17
5.4.	Ejecutar predicciones	17
5.5.	Checklist de validacion funcional	18
5.6.	Resultado esperado	18
5.7.	Errores comunes y recuperacion	18
5.8.	Imagenes	18
6.	6. MLOps y AB Testing	21
6.1.	Objetivo	21
6.2.	Crear deployment	21
6.3.	Monitorear deployment	21
6.4.	Ejecutar AB test	21
6.5.	Checklist de validacion funcional	22
6.6.	Resultado esperado	22
6.7.	Errores comunes y recuperacion	22
6.8.	Imagenes	22
7.	7. Privacy y SynthGen	25
7.1.	Objetivo	25
7.2.	Flujo de Privacy	25
7.3.	Flujo de SynthGen	25
7.4.	Checks de calidad y paridad	26
7.5.	Checklist de validacion funcional	26
7.6.	Resultado esperado	26
7.7.	Errores comunes y recuperacion	26
7.8.	Imagenes	26
8.	8. Teams, Billing, Admin y Troubleshooting	29
8.1.	Objetivo	29
8.2.	Gestion de equipos	29
8.3.	Billing y ciclo de plan	29
8.4.	Controles Admin	30
8.5.	Checklist de validacion funcional	30
8.6.	Resultado esperado	30
8.7.	Playbook rapido de recuperacion	30
8.8.	Imagen	31

Audiencia: usuarios de la plataforma web CorePlexML.

CAPÍTULO 1

1. Cuenta y Onboarding

1.1 Objetivo

Crear una cuenta valida, completar primer login y confirmar acceso al workspace.

1.2 Para quien es

1. Usuarios nuevos en planes Free, Pro o Team.
2. Miembros invitados por owner/admin.

1.3 Precondiciones

1. URL web de CorePlexML disponible.
2. Email valido con acceso a bandeja.
3. Navegador con cookies y JavaScript habilitados.

1.4 Paso a paso

1. Abrir `/register`.
2. Completar campos obligatorios: - Nombre completo. - Email de trabajo. - Password segun politica. - Aceptacion de terminos.
3. Enviar registro.
4. Si aplica verificacion de email, abrir correo y confirmar.
5. Abrir `/login` e iniciar sesion.

6. Completar perfil inicial si el sistema lo solicita: - Nombre visible. - Zona horaria. - Workspace/proyecto por defecto.
7. Confirmar que el dashboard carga sin error de autenticacion.

1.5 Checklist de validacion funcional

1. Sesion activa despues de login.
2. Menu de usuario muestra email correcto.
3. Widgets del dashboard cargan con respuestas correctas.
4. Logout cierra sesion y redirige a login.
5. Re-login recupera acceso normal.

1.6 Resultado esperado

1. Usuario autenticado con navegacion habilitada por rol.
2. Estado de sesion estable al refrescar pagina.

1.7 Errores comunes y recuperacion

1. `Email not verified`: - Reenviar verificacion. - Confirmar que el link corresponde al mismo ambiente.
2. `Invalid credentials`: - Usar `/auth/forgot-password`.
3. Logout inmediato despues de login: - Limpiar cookies del dominio. - Reintentar login.

1.8 Imagen

The screenshot displays the CorePlexML Web Interface dashboard. The top navigation bar includes a home icon, the text 'CorePlexML Web Interface', and a 'UI Audit' dropdown menu. The main content area features a 'Dashboard' header with a welcome message and a summary of key metrics: 1 Project, 3 Datasets, 3 Models, and 1 Prediction. Below this, there are two tables: 'Recent Projects' and 'Recent Models'. The 'Recent Projects' table lists a project named 'UI Audit Project dbdf56' created on 02/10/2026 at 03:04:40 AM, with an 'Active' status. The 'Recent Models' table lists three models: 'gbm', 'xgboost', and 'stackedensemble', all trained on 02/10/2026 at 03:08:23 AM, 03:08:12 AM, and 03:08:03 AM respectively, with RMSE values of 0.5610, 0.1912, and 0.3481. A sidebar on the left contains navigation links for Dashboard, Projects, Datasets, Dataset Builder AI, Experiments, Models, Predictions, MLOps, Model Registry, and Alerts. The footer includes copyright information for Intellicore and GitHub, a BUSL-1.1 license, and mentions of FastAPI and H2O.ai.

Name	Created	Status
UI Audit Project dbdf56	02/10/2026, 03:04:40 AM	Active

Model Type	Trained	Metrics
gbm	02/10/2026, 03:08:23 AM	RMSE: 0.5610
xgboost	02/10/2026, 03:08:12 AM	RMSE: 0.1912
stackedensemble	02/10/2026, 03:08:03 AM	RMSE: 0.3481

Figura 1: Dashboard visible tras registro y login exitoso.

CAPÍTULO 2

2. Navegacion y Workspace

2.1 Objetivo

Entender navegacion global, contexto de proyecto activo y transicion segura entre modulos.

2.2 Semantica del workspace

1. Sidebar izquierdo controla navegacion por modulo.
2. Topbar controla busqueda, perfil y acciones rapidas.
3. Proyecto activo define donde se crean datasets, experimentos y modelos.

2.3 Paso a paso

1. Abrir dashboard.
2. Identificar zonas principales: - Sidebar: Datasets, Builder, Experiments, Models, MLOps, Privacy, SynthGen. - Topbar: busqueda, notificaciones, perfil/team.
3. Cambiar de modulo y confirmar cambio de titulo/URL.
4. Cambiar proyecto activo (si existe selector).
5. Volver al modulo anterior y confirmar que el contexto se conserva.

2.4 Checklist de validacion funcional

1. Cada item de menu abre el modulo correcto.
2. Back/forward del navegador mantiene estado esperado.

3. Proyecto activo se mantiene consistente entre modulos.
4. Modulos restringidos se ocultan o bloquean por rol.
5. Filtros de tabla y busqueda se comportan segun diseno del producto.

2.5 Resultado esperado

1. Navegacion estable sin perdida de contexto.
2. Permisos visibles y aplicados segun rol y plan.

2.6 Errores comunes y recuperacion

1. Datos de proyecto incorrecto: - Revisar selector de proyecto activo. - Refrescar y reingresar.
2. Opcion de menu faltante: - Verificar rol y plan de suscripcion.
3. Estado visual obsoleto: - Limpiar storage local y volver a iniciar sesion.

2.7 Imagen

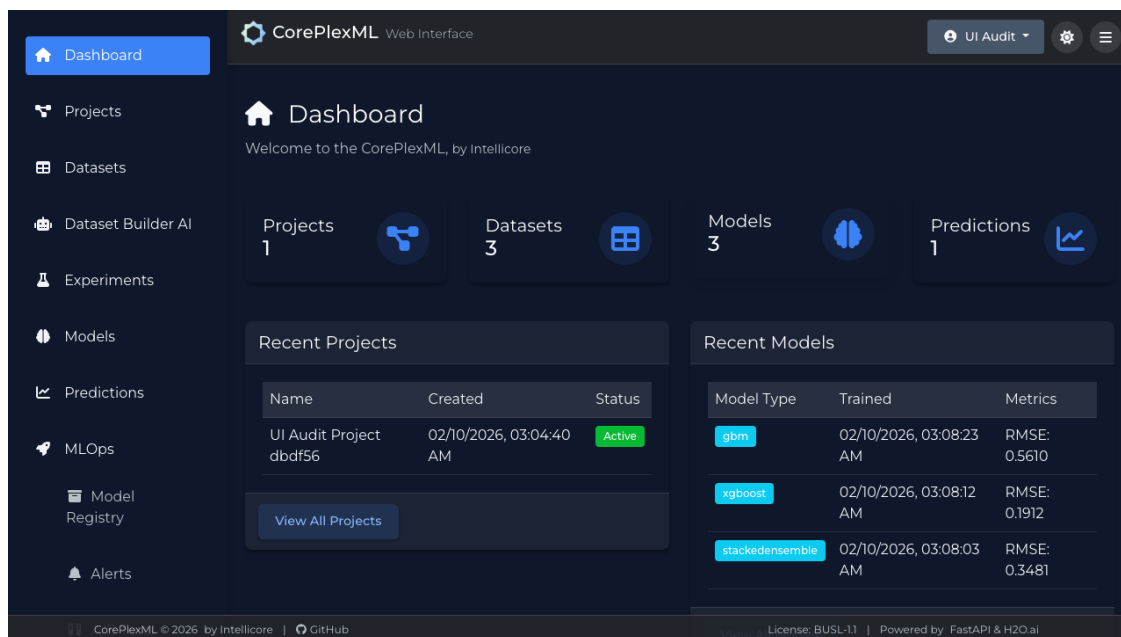


Figura 1: Navegacion principal con accesos de modulo y perfil.

CAPÍTULO 3

3. Proyectos y Datasets

3.1 Objetivo

Crear proyectos, subir datasets y validar integridad de datos antes del modelado.

3.2 Precondiciones

1. Usuario con permisos de crear proyecto y subir datasets.
2. Archivo fuente en formato CSV o Parquet.

3.3 Crear proyecto

1. Abrir Projects.
2. Click en Create Project.
3. Definir: - Nombre del proyecto. - Descripción opcional. - Alcance de visibilidad/team si aplica.
4. Guardar y confirmar que aparece en el listado.

3.4 Subir dataset

1. Abrir Datasets dentro del proyecto objetivo.
2. Click en Upload Dataset.
3. Seleccionar archivo y metadata opcional.
4. Esperar estado de ingesta completado.

5. Abrir detalle del dataset.

3.5 Validar calidad de datos

1. Revisar inferencia de schema: - Nombres de columnas. - Tipos de datos. - Nulos por columna.
2. Revisar conteo de filas y duplicados.
3. Revisar preview para errores de parseo.
4. Confirmar que fechas y delimitadores se interpretaron bien.

3.6 Checklist de validacion funcional

1. Artifact de dataset creado y visible.
2. Row count reportado coincide con lo esperado.
3. Columnas criticas mantienen tipo correcto.
4. Preview no presenta corrimiento/truncado.
5. Reabrir detalle muestra metadata consistente.

3.7 Resultado esperado

1. Proyecto y dataset listos para Builder y Experiments.
2. Version de dataset visible y seleccionable.

3.8 Errores comunes y recuperacion

1. Upload en estado infinito: - Reintentar con muestra menor. - Revisar logs de parser.
2. Tipos mal detectados: - Re-subir archivo con headers/fechas normalizadas.
3. Diferencia de filas: - Validar delimitador, comillas y filas mal formadas.

3.9 Imagenes

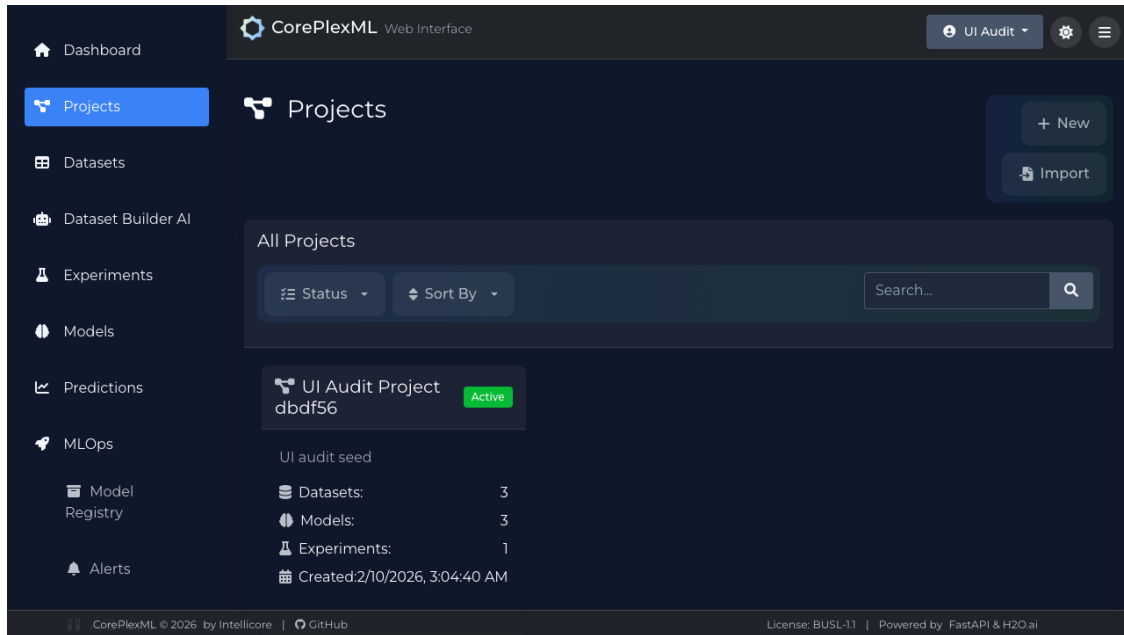


Figura 1: Modulo de proyectos con creacion y seleccion.

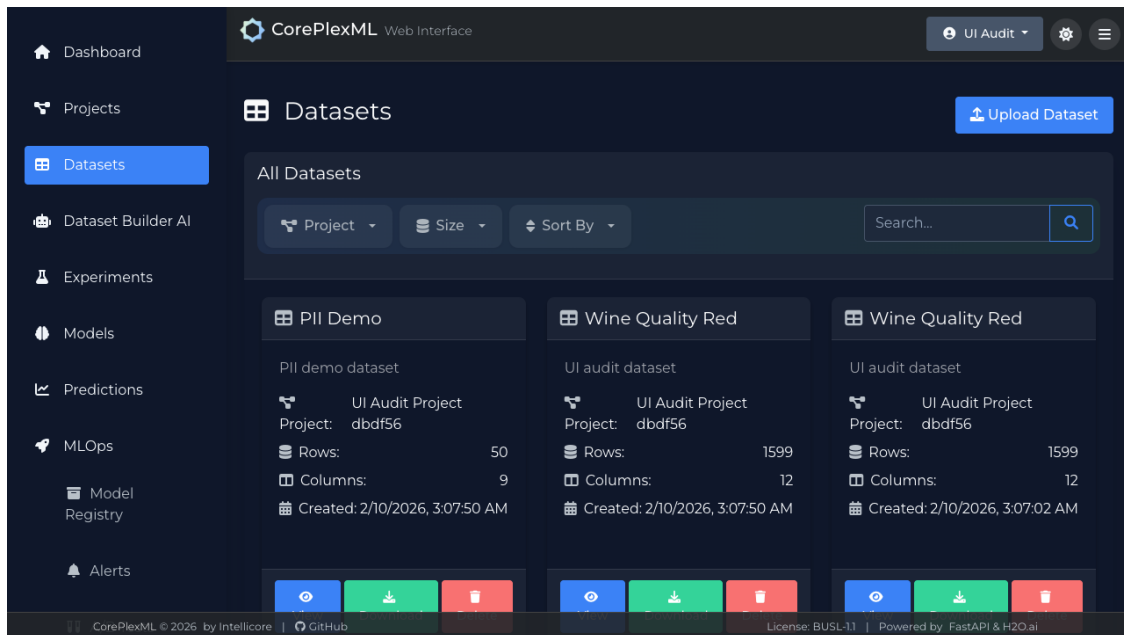


Figura 2: Registro de datasets con estado y metadata.

CAPÍTULO 4

4. Dataset Builder AI

4.1 Objetivo

Generar un dataset transformado mediante flujo conversacional guiado y trazable.

4.2 Que debe cumplir este modulo

1. Hacer preguntas relevantes por etapa.
2. Generar y ejecutar transformaciones.
3. Mostrar preview con impacto en filas/columnas.
4. Permitir confirmar, reintentar o corregir.

4.3 Flujo conversacional realista

1. Abrir Dataset Builder AI y seleccionar version de dataset.
2. Leer analisis inicial.
3. Responder preguntas con interaccion humana real: - Pedir aclaraciones. - Cambiar una decision. - Rechazar resultado y pedir nuevo intento.
4. Continuar hasta ver preview de salida.
5. Confirmar solo si la salida cumple objetivo.

4.4 Si el resultado no es el esperado

1. Pedir aclaracion antes de confirmar la etapa.
2. Ajustar decisiones previas y ejecutar retry.
3. Si el row count baja de forma inesperada, reintentar con parametros mas conservadores.
4. Si la salida queda en cero filas, usar retry y revisar opciones de transformacion.

4.5 Checklist de validacion funcional

1. Builder crea artifact y nueva version de dataset.
2. Row count final se reporta explicitamente.
3. Preview refleja las transformaciones elegidas.
4. Estado de conversacion persiste tras refresh/reingreso.
5. Retry restaura dataset previo valido cuando corresponde.

4.6 Resultado esperado

1. Flujo completo desde analisis hasta artifact final sin perdida de estado.
2. Dataset resultante utilizable en Experiments.

4.7 Errores comunes y recuperacion

1. Zero-row output: - Ejecutar retry y ajustar estrategia.
2. Resultado no esperado: - Solicitar rollback/retry en la conversacion.
3. Interrupcion de sesion: - Reabrir sesion y continuar desde checkpoint.

4.8 Imagen

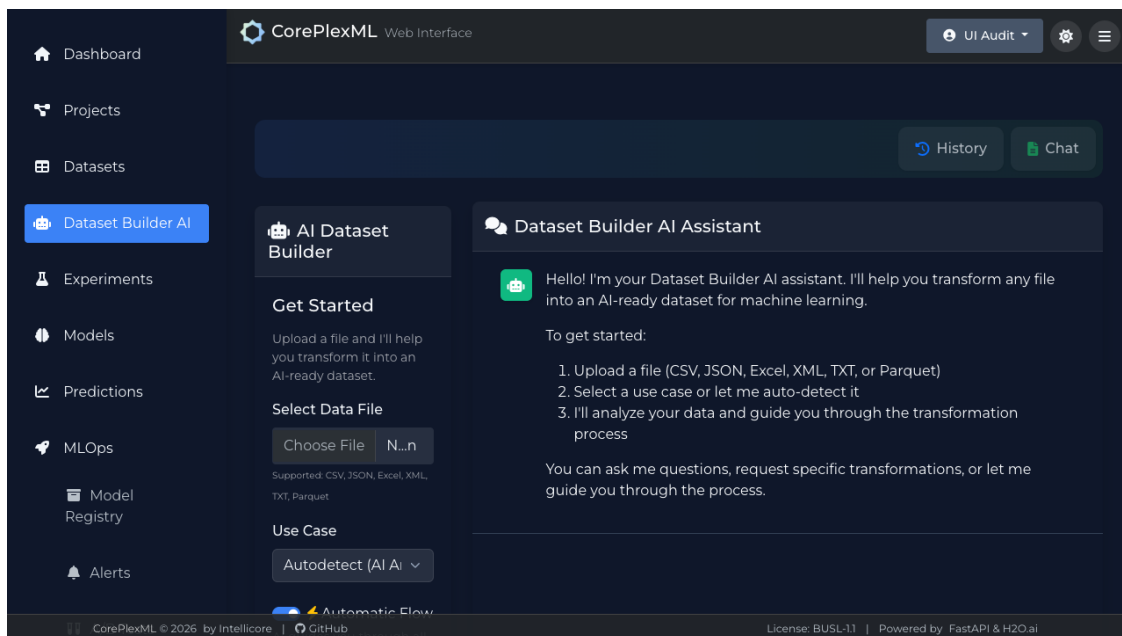


Figura 1: Flujo conversacional con preview de transformacion.

CAPÍTULO 5

5. Experimentos, Modelos, Predicciones

5.1 Objetivo

Entrenar modelos, comparar resultados, registrar el mejor candidato y ejecutar predicciones.

5.2 Crear experimento

1. Abrir Experiments.
2. Click en New Experiment.
3. Configurar: - Dataset/version. - Columna target. - Tipo de problema. - Estrategia de validacion y metrica objetivo.
4. Iniciar corrida y monitorear estado.

5.3 Revisar resultados y modelos

1. Abrir leaderboard al completar corrida.
2. Validar metrica primaria y ranking.
3. Abrir detalle del mejor modelo y revisar: - Metricas. - Explainability/importancia de features si aplica. - Metadata de artifact/version.
4. Registrar o fijar modelo segun flujo interno.

5.4 Ejecutar predicciones

1. Abrir Predictions.
2. Ejecutar prediccion individual desde formulario.

3. Ejecutar prediccion batch por archivo si esta disponible.
4. Validar campos de salida: - Valor/clase predicha. - Confianza/probabilidad si aplica. - Timestamp y referencia de corrida.

5.5 Checklist de validacion funcional

1. Experimento llega a estado terminal sin falla silenciosa.
2. Metricas de detalle coinciden con leaderboard.
3. Esquema de salida de prediccion es estable.
4. Batch conserva paridad de filas input/output.
5. Errores de prediccion entregan mensaje accionable.

5.6 Resultado esperado

1. Al menos un modelo listo para despliegue.
2. Predicciones consistentes y trazables.

5.7 Errores comunes y recuperacion

1. Falla temprana de experimento: - Revisar target y manejo de nulos.
2. Metrica inconsistente: - Confirmar mismo split/seed.
3. Input rechazado en prediccion: - Alinear nombres/tipos con schema del modelo.

5.8 Imagenes

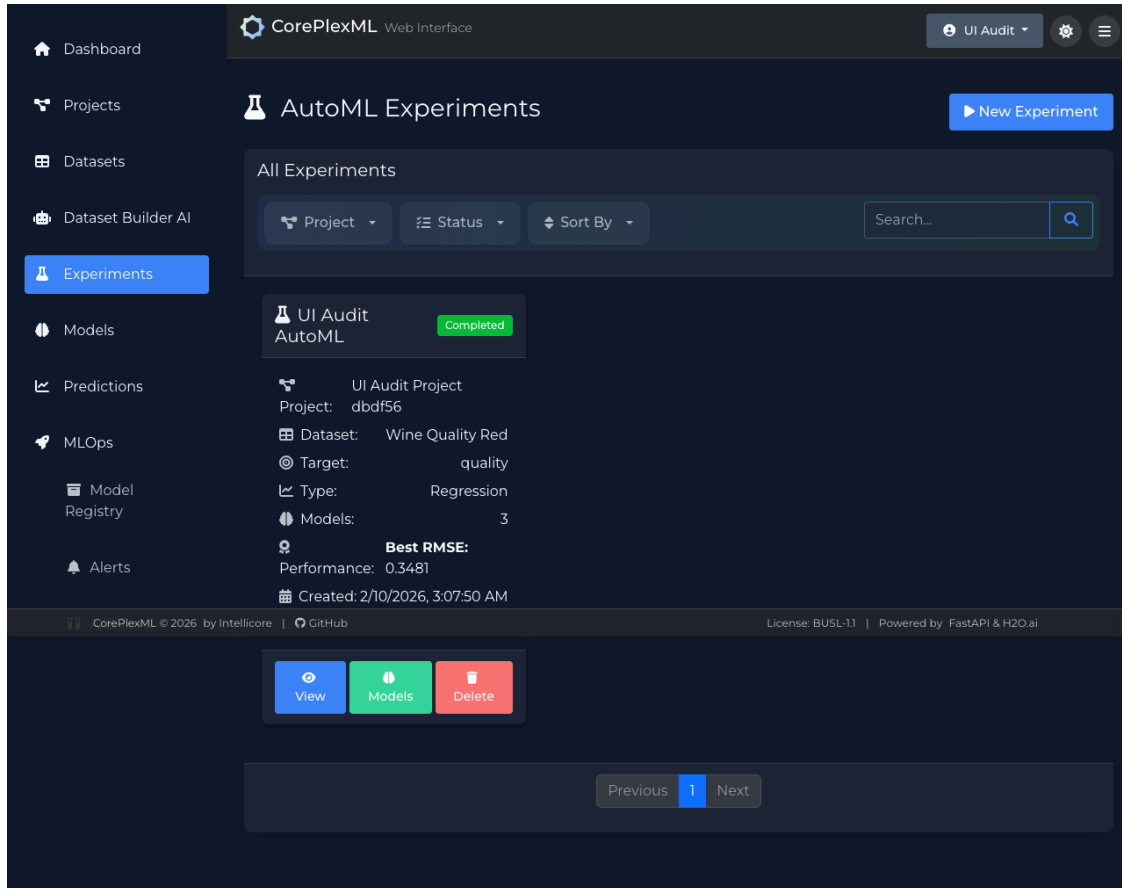


Figura 1: Ejecucion de experimentos y monitoreo.

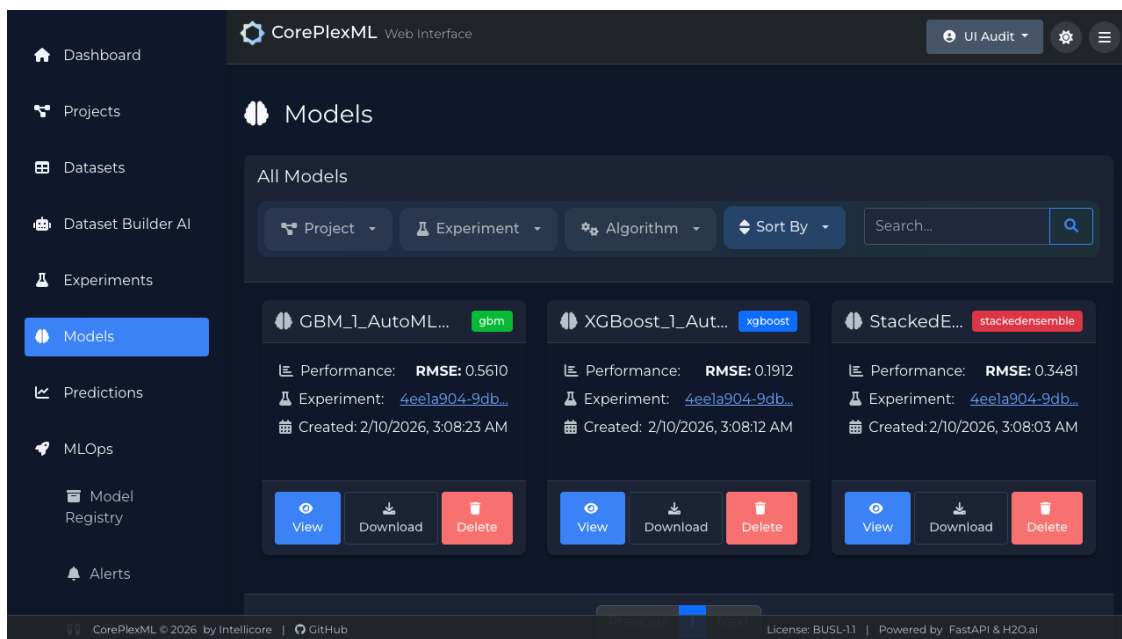


Figura 2: Registro de modelos con metricas y metadata.

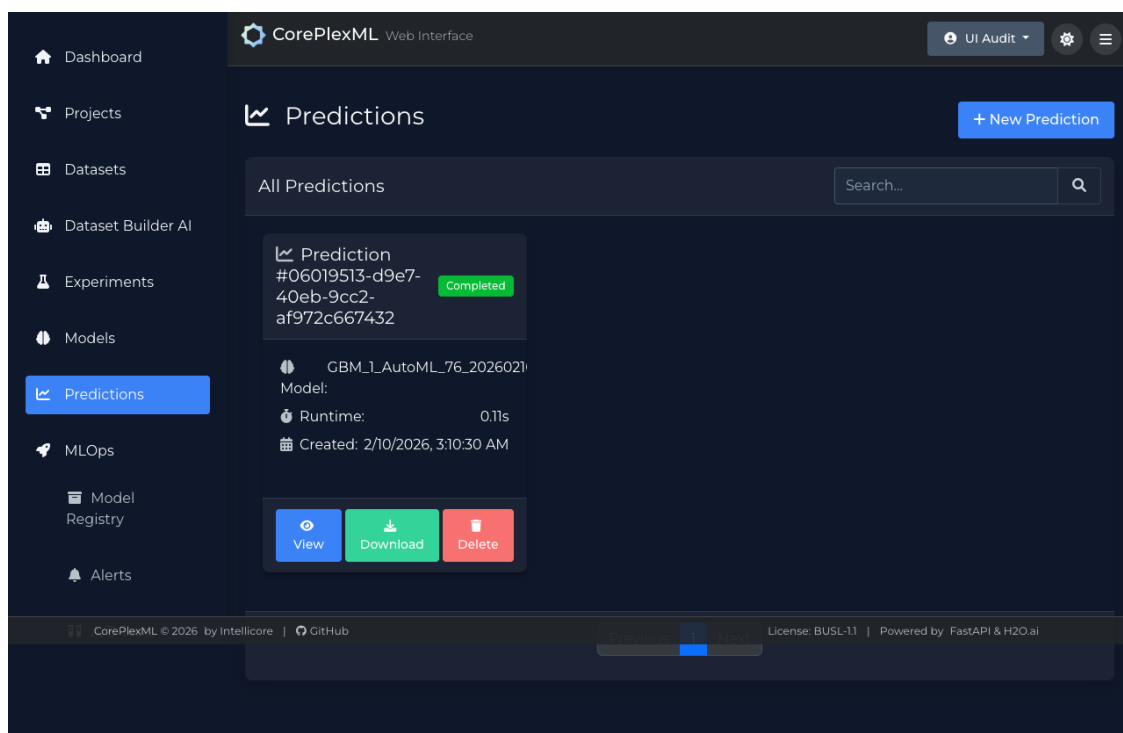


Figura 3: Prediccion individual y batch desde UI.

CAPÍTULO 6

6. MLOps y AB Testing

6.1 Objetivo

Desplegar modelos con seguridad, monitorear runtime y comparar variantes con AB testing.

6.2 Crear deployment

1. Abrir MLOps > Deployments.
2. Seleccionar version de modelo.
3. Configurar: - Ambiente. - Perfil de replicas/compute. - Estrategia de rollback.
4. Ejecutar despliegue y esperar estado active.

6.3 Monitorear deployment

1. Validar indicadores de salud: - Uptime/health. - Error rate. - Latencia.
2. Ejecutar inferencia de prueba y validar respuesta.
3. Revisar logs recientes para excepciones.

6.4 Ejecutar AB test

1. Abrir MLOps > A/B Tests.
2. Crear test con: - Modelo baseline (A). - Modelo candidato (B). - Split de trafico. - Metrica principal. - Tamano minimo de muestra y duracion.
3. Iniciar test y monitorear asignacion.

4. Evaluar ganador cuando se cumpla criterio.

6.5 Checklist de validacion funcional

1. Deployment activo responde sin downtime.
2. Salud y metricas actualizan en tiempo operativo.
3. Split AB se refleja en conteo real de requests.
4. Ganador reportado coincide con metrica definida.
5. Rollback disponible ante degradacion.

6.6 Resultado esperado

1. Ruta productiva estable y observable.
2. Promocion de modelos basada en datos.

6.7 Errores comunes y recuperacion

1. Deployment en pending: - Revisar capacidad y artifact del modelo.
2. Error rate alto post-release: - Ejecutar rollback inmediato.
3. Test AB inconcluso: - Aumentar tiempo o muestra.

6.8 Imagenes

CorePlexML Web Interface

MLOps Deployments

+ New Deployment

Project: UI Audit Project dbdf56

ID	Model	Stage	Active	Created	Actions
3aa7b8af...	0906745b...	staging	Yes	2026-02-10T03:10:32.738118-03:00	Detail

CorePlexML © 2026 by Intellicore | GitHub License: BUSL-1.1 | Powered by FastAPI & H2O.ai

Figura 1: Lista de deployments y estado runtime.

CorePlexML Web Interface

A/B Tests

+ New A/B Test

Project: UI Audit Project dbdf56 Status: All statuses Refresh

Name	Project	Status	Traffic Split	Observations	Winner	Created	Actions
UI Audit A/B	UI Audit Project dbdf56	draft	50% / 50%	A: 0, B: 0	-	2/10/2026	
UI Audit A/B	UI Audit Project dbdf56	draft	50% / 50%	A: 0, B: 0	-	2/10/2026	

CorePlexML © 2026 by Intellicore | GitHub License: BUSL-1.1 | Powered by FastAPI & H2O.ai

Figura 2: Configuración y comparativa en AB testing.

CAPÍTULO 7

7. Privacy y SynthGen

7.1 Objetivo

Proteger datos sensibles con politicas de privacidad y generar datasets sinteticos utiles.

7.2 Flujo de Privacy

1. Abrir Privacy.
2. Crear o seleccionar politica.
3. Asociar politica a dataset/sesion.
4. Ejecutar deteccion y revisar entidades PII detectadas.
5. Configurar accion por regla (mask/redact/hash/encrypt/generalize/suppress).
6. Ejecutar transformacion y exportar dataset protegido.

7.3 Flujo de SynthGen

1. Abrir SynthGen.
2. Seleccionar dataset fuente y perfil de generacion.
3. Configurar tamano objetivo y restricciones opcionales.
4. Iniciar job de generacion.
5. Descargar artifact sintetico al completar.

7.4 Checks de calidad y paridad

1. Paridad de schema: - Conteo de columnas y nombres esperados.
2. Paridad de privacidad: - Valores sensibles originales no expuestos.
3. Paridad de utilidad: - Tendencias de distribucion utiles para modelado.
4. Integridad: - Archivo legible y row count segun configuracion.

7.5 Checklist de validacion funcional

1. Deteccion privacy devuelve hallazgos cuando hay PII.
2. Transformacion genera artifact descargable.
3. SynthGen termina en estado `completed`.
4. Dataset generado puede cargarse en Datasets/Experiments.
5. Estados de error entregan diagnostico util.

7.6 Resultado esperado

1. Datos sensibles protegidos segun politica.
2. Datos sinteticos usables para prototipos y pruebas.

7.7 Errores comunes y recuperacion

1. Sin hallazgos PII cuando se esperan: - Revisar politica/reglas y columnas fuente.
2. Falla de transformacion: - Reintentar con alcance de reglas mas acotado.
3. Timeout de SynthGen: - Reducir tamano y reejecutar.

7.8 Imagenes

The screenshot displays the 'Privacy Suite' web interface. The top navigation bar includes 'Dashboard', 'Projects', 'Datasets', 'Dataset Builder AI', 'Experiments', 'Models', 'Predictions', 'MLOps', 'Model Registry', and 'Alerts'. The main content area is titled 'Privacy Suite' and features a 'Scan Dataset' button and a '+ New Policy' button. Below this, a 'Project' dropdown is set to 'UI Audit Project dbdf56', showing '2 Policies' and '0 Sessions'. A summary section displays four status cards: 'Active Policies' (1), 'Compliant' (0), 'Pending Review' (0), and 'Non-Compliant' (1). A 'Privacy Policies' table lists one policy: 'UI Audit Privacy' with a 'HIPAA' profile, '0' rules, 'active' status, and '0%' compliance. Below this, a 'Recent Sessions' table shows one session: 'Qed12364...' with 'UI Audit Privacy' policy, 'PII Demo' dataset, 'detected' status, and '9' PII found.

Name	Profile	Rules	Status	Compliance	Created	Actions
UI Audit Privacy	HIPAA	0	active	0%	2/10/2026, 3:13:18 AM	View Delete

Session ID	Policy	Dataset	Status	PII Found	Created	Actions
Qed12364...	UI Audit Privacy	PII Demo	detected	9	2/10/2026, 3:13:18 AM	View

Figura 1: Modulo Privacy con ejecucion de politicas.

The screenshot displays the CorePlexML Web Interface for SynthGen. The interface includes a sidebar with navigation options: Dashboard, Projects, Datasets, Dataset Builder AI, Experiments, Models, Predictions, MLOps, Model Registry, and Alerts. The main content area is titled "SynthGen - Synthetic Data Generation" and features a "Project" dropdown set to "UI Audit Project dbdf56", showing "1 Models" and "1 Jobs".

Key metrics are shown in four colored boxes:

- Total Models: 1
- Active Jobs: 0
- Completed: 1
- Failed: 0

Below these metrics is a table titled "Synthetic Data Models" with columns: Name, Synthesizer, Status, Dataset, Quality Score, Created, and Actions. The table contains one entry:

Name	Synthesizer	Status	Dataset	Quality Score	Created	Actions
UI Audit SynthGen	CTGAN	ready	Wine Quality Red (v1)	N/A	2/10/2026, 3:13:18 AM	View Delete

At the bottom, there is a "Recent Jobs" section with a "View All" button and a table with columns: Job ID, Model, Type, Status, Rows, Created, and Actions. The table contains one entry:

Job ID	Model	Type	Status	Rows	Created	Actions
c09fea70...	UI Audit SynthGen	Training	completed	-	2/10/2026, 3:13:18 AM	View

The footer of the interface includes: CorePlexML © 2026 by Intellicore | GitHub, License: BUSL-1.1 | Powered by FastAPI & H2O.ai.

Figura 2: Flujo de generacion y salida de SynthGen.

CAPÍTULO 8

8. Teams, Billing, Admin y Troubleshooting

8.1 Objetivo

Operar colaboracion de equipos, ciclo de suscripcion y controles admin con resultados predecibles.

8.2 Gestion de equipos

1. Abrir Teams.
2. Crear team o abrir team existente.
3. Invitar miembros por email.
4. Asignar rol por miembro (owner/admin/member segun politica).
5. Validar que cada usuario ve solo modulos autorizados.

8.3 Billing y ciclo de plan

1. Abrir seccion de perfil/billing.
2. Validar estado de plan actual (Free, Pro, Team).
3. Ejecutar upgrade cuando aplique: - Free -> Pro. - Pro -> Team.
4. Confirmar actualizacion de estado despues del checkout.
5. Validar que permisos Team quedan habilitados.

8.4 Controles Admin

1. Abrir Admin (solo roles autorizados).
2. Revisar: - Audit logs. - Estado operativo. - Controles tenant/team.
3. Aplicar cambios solo con ticket/aprobacion.

8.5 Checklist de validacion funcional

1. Invitacion, aceptacion y rol de team completan correctamente.
2. Upgrade de plan refleja permisos y estado en UI.
3. Estado de suscripcion persiste tras logout/login y refresh.
4. Paginas admin no son accesibles por roles no autorizados.
5. Errores de billing son explicitos y sin estado ambiguo.

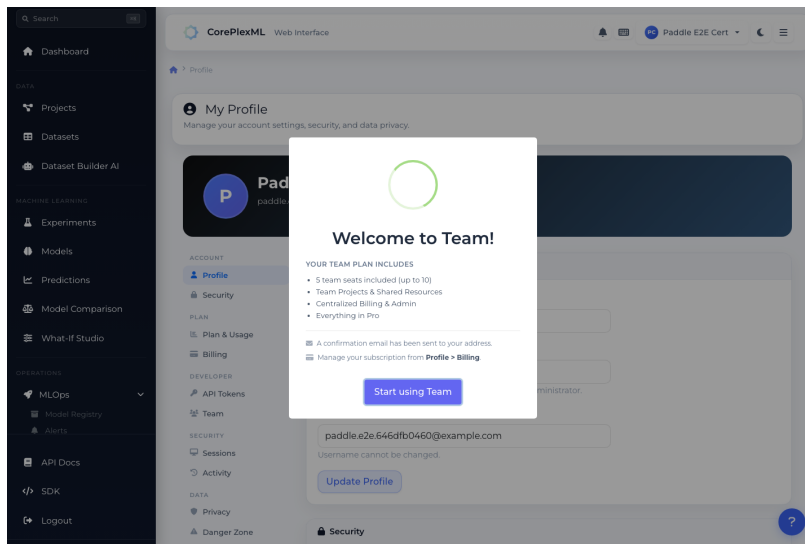
8.6 Resultado esperado

1. Colaboracion y billing funcionan de punta a punta.
2. Admin puede diagnosticar incidencias sin corromper datos.

8.7 Playbook rapido de recuperacion

1. Problema de sesion: - Logout/login y validar estado de cuenta.
2. Modulo/dato faltante: - Validar proyecto activo y permisos.
3. Billing inconsistente: - Verificar provider de suscripcion y owner del team.
4. UI obsoleta: - Hard refresh y reintentar una vez.

8.8 Imagen



Controles de team y billing desde perfil/workspace.