



PLATFORM DOCUMENTATION

CorePlexML

Release 0.1.0

Rodrigo Henriquez M.

Generated on 09 mar. 2026

Sumário

1	1. Conta e Onboarding	3
1.1	Objetivo	3
1.2	Para quem e	3
1.3	Precondicoes	3
1.4	Passo a passo	3
1.5	Checklist de validacao funcional	4
1.6	Resultado esperado	4
1.7	Erros comuns e recuperacao	4
1.8	Imagem	4
2	2. Navegacao e Workspace	5
2.1	Objetivo	5
2.2	Semantica do workspace	5
2.3	Passo a passo	5
2.4	Checklist de validacao funcional	5
2.5	Resultado esperado	6
2.6	Erros comuns e recuperacao	6
2.7	Imagem	6
3	3. Projetos e Datasets	7
3.1	Objetivo	7
3.2	Precondicoes	7
3.3	Criar projeto	7
3.4	Upload de dataset	7
3.5	Validar qualidade de dados	8
3.6	Checklist de validacao funcional	8
3.7	Resultado esperado	8
3.8	Erros comuns e recuperacao	8
3.9	Imagens	8
4	4. Dataset Builder AI	11
4.1	Objetivo	11
4.2	O que este modulo deve cumprir	11
4.3	Fluxo conversacional realista	11
4.4	Se o resultado nao for o esperado	11
4.5	Checklist de validacao funcional	12
4.6	Resultado esperado	12

4.7	Erros comuns e recuperacao	12
4.8	Imagem	12
5	5. Experimentos, Modelos e Predicoes	13
5.1	Objetivo	13
5.2	Criar experimento	13
5.3	Revisar resultados e registro de modelos	13
5.4	Executar predicoes	13
5.5	Checklist de validacao funcional	14
5.6	Resultado esperado	14
5.7	Erros comuns e recuperacao	14
5.8	Imagens	14
6	6. MLOps e AB Testing	17
6.1	Objetivo	17
6.2	Criar deployment	17
6.3	Monitorar deployment	17
6.4	Executar AB test	17
6.5	Checklist de validacao funcional	18
6.6	Resultado esperado	18
6.7	Erros comuns e recuperacao	18
6.8	Imagens	18
7	7. Privacy e SynthGen	21
7.1	Objetivo	21
7.2	Fluxo de Privacy	21
7.3	Fluxo de SynthGen	21
7.4	Checks de qualidade e paridade	22
7.5	Checklist de validacao funcional	22
7.6	Resultado esperado	22
7.7	Erros comuns e recuperacao	22
7.8	Imagens	22
8	8. Teams, Billing, Admin e Troubleshooting	25
8.1	Objetivo	25
8.2	Gestao de times	25
8.3	Billing e ciclo de plano	25
8.4	Controles Admin	25
8.5	Checklist de validacao funcional	26
8.6	Resultado esperado	26
8.7	Playbook rapido de recuperacao	26
8.8	Imagem	26

Publico: usuarios da plataforma web CorePlexML.

CAPÍTULO 1

1. Conta e Onboarding

1.1 Objetivo

Criar conta valida, concluir primeiro login e confirmar acesso ao workspace.

1.2 Para quem e

1. Usuarios novos nos planos Free, Pro e Team.
2. Membros convidados por owner/admin.

1.3 Precondicoes

1. URL web da CorePlexML acessivel.
2. Email valido com acesso a caixa de entrada.
3. Navegador com cookies e JavaScript habilitados.

1.4 Passo a passo

1. Abrir `/register`.
2. Preencher campos obrigatorios: - Nome completo. - Email de trabalho. - Senha conforme politica. - Aceite de termos.
3. Enviar cadastro.
4. Se houver verificacao de email, abrir mensagem e confirmar.
5. Abrir `/login` e autenticar.
6. Completar perfil inicial se solicitado: - Nome de exibicao. - Fuso horario. - Workspace/projeto padrao.
7. Confirmar carregamento do dashboard sem erro de autenticacao.

1.5 Checklist de validacao funcional

1. Sessao ativa apos login.
2. Menu de usuario mostra email correto.
3. Widgets do dashboard carregam normalmente.
4. Logout encerra sessao e redireciona para login.
5. Novo login restaura acesso normal.

1.6 Resultado esperado

1. Usuario autenticado com navegacao liberada pelo papel.
2. Sessao estavel apos refresh.

1.7 Erros comuns e recuperacao

1. Email not verified: - Reenviar verificacao. - Confirmar dominio correto do ambiente.
2. Invalid credentials: - Usar /auth/forgot-password.
3. Logout imediato apos login: - Limpar cookies do dominio. - Tentar login novamente.

1.8 Imagem

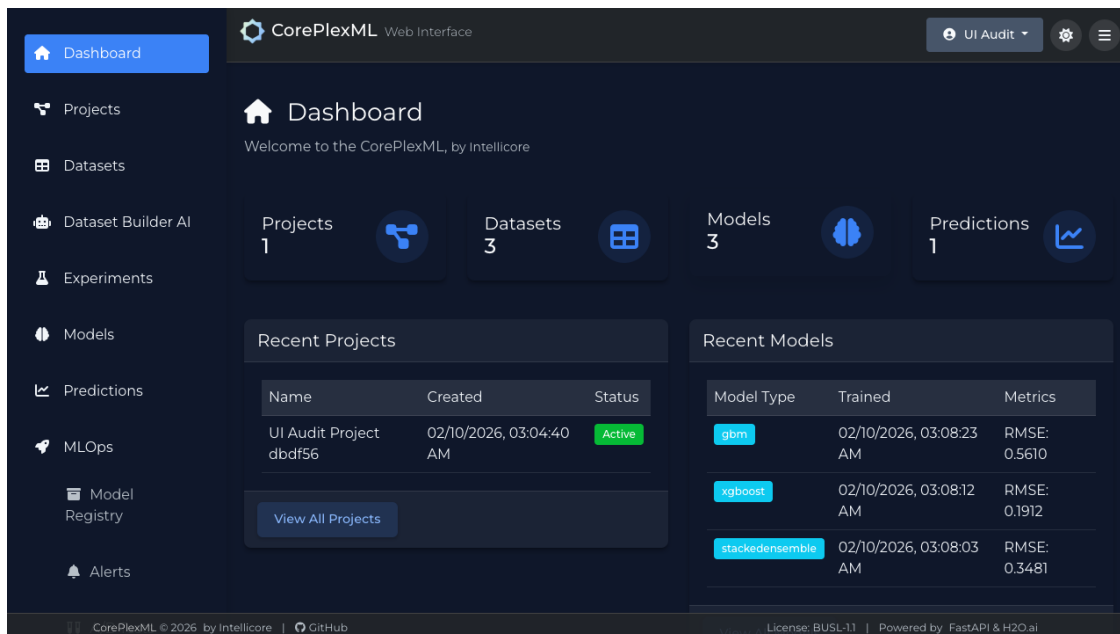


Fig. 1: Dashboard disponivel apos cadastro e login.

CAPÍTULO 2

2. Navegacao e Workspace

2.1 Objetivo

Entender navegacao global, contexto de projeto ativo e transicao segura entre modulos.

2.2 Semantica do workspace

1. Sidebar esquerdo controla navegacao por modulo.
2. Barra superior controla busca, perfil e acoes rapidas.
3. Projeto ativo define onde datasets e execucoes sao criados.

2.3 Passo a passo

1. Abrir dashboard.
2. Identificar areas principais: - Sidebar: Datasets, Builder, Experiments, Models, MLOps, Privacy, SynthGen. - Topbar: busca, notificacoes, perfil/team.
3. Trocar modulo e confirmar titulo/URL.
4. Trocar projeto ativo (se houver seletor).
5. Voltar ao modulo anterior e confirmar contexto preservado.

2.4 Checklist de validacao funcional

1. Cada item do menu abre a URL/modulo correto.
2. Back/forward do navegador mantem estado esperado.
3. Projeto ativo permanece consistente entre modulos.
4. Modulos restritos ficam ocultos ou bloqueados por papel.

5. Filtros e busca funcionam conforme comportamento esperado.

2.5 Resultado esperado

1. Navegacao estavel sem perda de contexto.
2. Permissoes aplicadas conforme papel e plano.

2.6 Erros comuns e recuperacao

1. Dados de projeto errado: - Revisar seletor de projeto ativo. - Atualizar pagina e reentrar.
2. Opcao de menu ausente: - Verificar papel e plano de assinatura.
3. Estado visual desatualizado: - Limpar storage local e refazer login.

2.7 Imagem

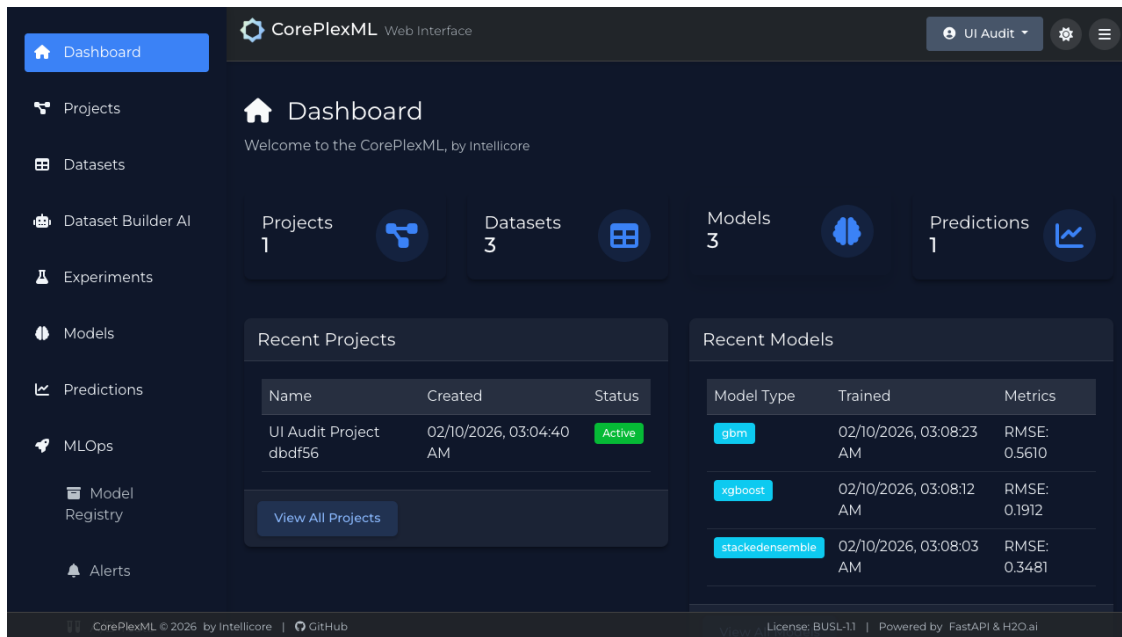


Fig. 1: Navegacao principal com acesso a modulos e perfil.

CAPÍTULO 3

3. Projetos e Datasets

3.1 Objetivo

Criar projetos, subir datasets e validar integridade dos dados antes de modelagem.

3.2 Precondicoes

1. Usuario com permissao de criar projeto e upload.
2. Arquivo fonte em CSV ou Parquet.

3.3 Criar projeto

1. Abrir Projects.
2. Clicar em Create Project.
3. Definir: - Nome do projeto. - Descricao opcional. - Escopo de visibilidade/team se aplicavel.
4. Salvar e confirmar item na lista.

3.4 Upload de dataset

1. Abrir Datasets dentro do projeto alvo.
2. Clicar em Upload Dataset.
3. Seleccionar arquivo e metadata opcional.
4. Aguardar status de ingestao finalizado.
5. Abrir detalhe do dataset.

3.5 Validar qualidade de dados

1. Revisar inferencia de schema: - Nomes de colunas. - Tipos de dados. - Nulos por coluna.
2. Revisar contagem de linhas e duplicados.
3. Revisar preview para erro de parsing.
4. Confirmar interpretacao correta de datas/delimitadores.

3.6 Checklist de validacao funcional

1. Artifact de dataset criado e visivel.
2. Row count reportado confere com expectativa.
3. Colunas criticas preservam tipo correto.
4. Preview sem deslocamento/truncamento indevido.
5. Reabrir detalhe retorna metadata consistente.

3.7 Resultado esperado

1. Projeto e dataset prontos para Builder e Experiments.
2. Versao de dataset disponivel para selecao.

3.8 Erros comuns e recuperacao

1. Upload travado: - Repetir com amostra menor. - Revisar logs de parser.
2. Tipagem incorreta: - Reenviar arquivo com cabecalhos/formatos normalizados.
3. Divergencia de linhas: - Validar delimitador, aspas e linhas mal formadas.

3.9 Imagens

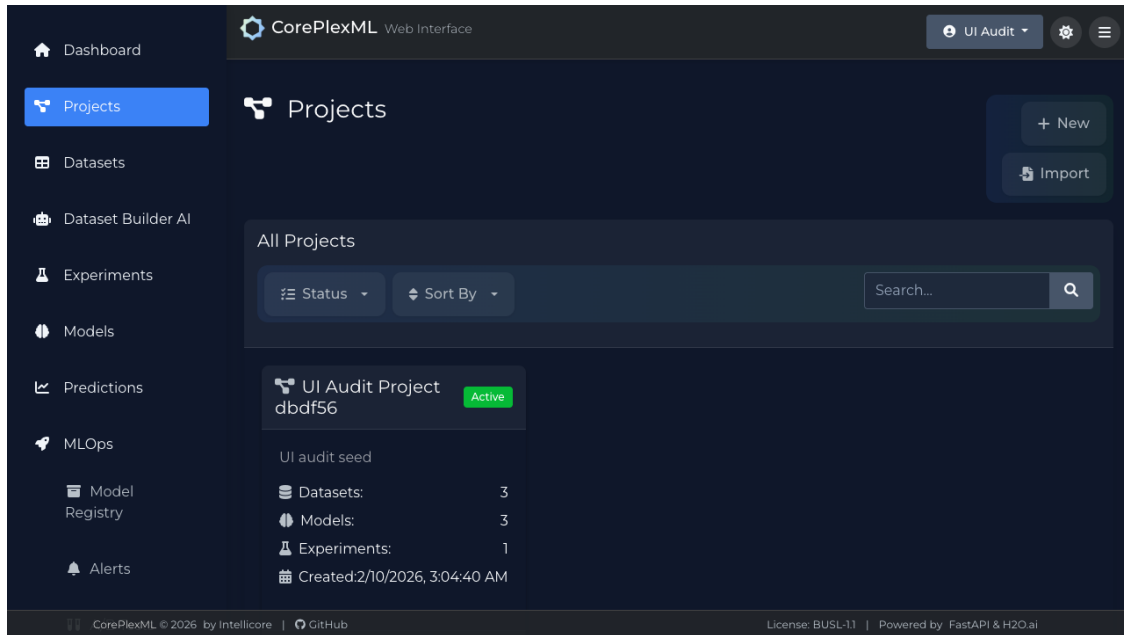


Fig. 1: Modulo Projects com criacao e selecao.

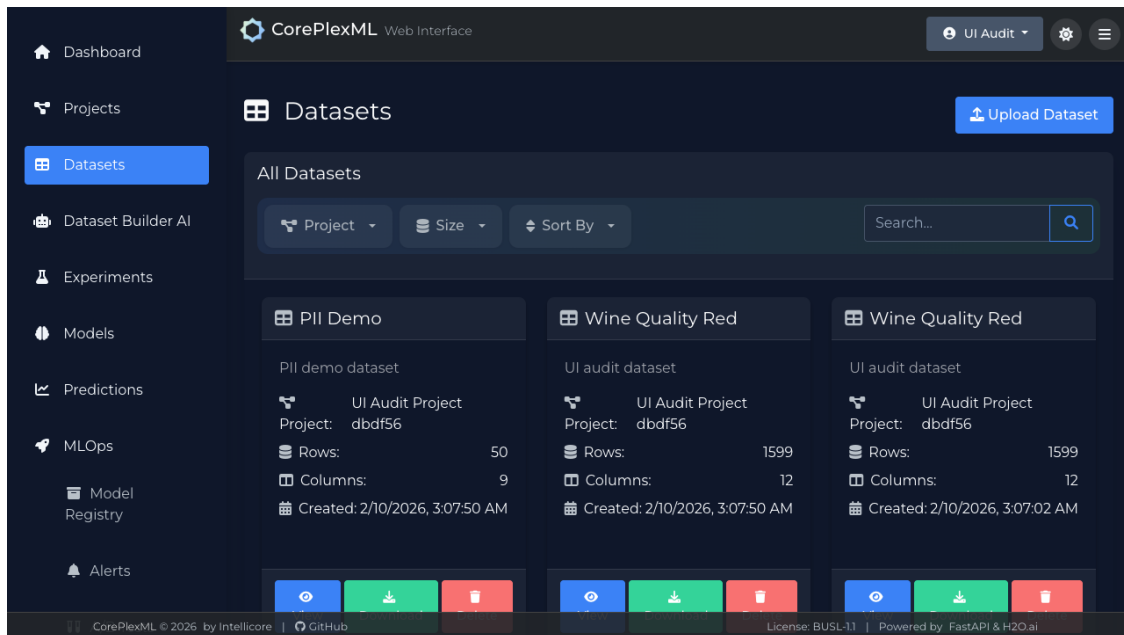


Fig. 2: Registro de datasets com status e metadata.

CAPÍTULO 4

4. Dataset Builder AI

4.1 Objetivo

Gerar dataset transformado com fluxo conversacional guiado e rastreavel.

4.2 O que este modulo deve cumprir

1. Fazer perguntas relevantes por etapa.
2. Gerar e executar transformacoes.
3. Exibir preview com impacto em linhas/colunas.
4. Permitir confirmar, repetir ou corrigir.

4.3 Fluxo conversacional realista

1. Abrir Dataset Builder AI e selecionar versao do dataset.
2. Ler analise inicial.
3. Responder perguntas com interacao humana real: - Pedir esclarecimentos. - Mudar decisao no meio do fluxo. - Rejeitar resultado e pedir nova tentativa.
4. Continuar ate aparecer preview final.
5. Confirmar apenas quando a saida cumprir o objetivo.

4.4 Se o resultado nao for o esperado

1. Pedir esclarecimento antes de confirmar a etapa.
2. Ajustar decisoes anteriores e executar retry.
3. Se o row count cair de forma inesperada, repetir com parametros mais conservadores.

4. Se a saída ficar com zero linhas, usar retry e revisar opções de transformação.

4.5 Checklist de validação funcional

1. Builder cria artifact e nova versão de dataset.
2. Row count final e reportado explicitamente.
3. Preview reflete transformações escolhidas.
4. Estado da conversa persiste após refresh/reentrada.
5. Retry restaura dataset válido anterior quando solicitado.

4.6 Resultado esperado

1. Fluxo completo de análise até artifact final sem perda de estado.
2. Dataset final utilizável em Experiments.

4.7 Erros comuns e recuperação

1. Zero-row output: - Executar retry e ajustar estratégia.
2. Resultado inesperado: - Solicitar rollback/retry na conversa.
3. Interrupção de sessão: - Reabrir sessão e continuar do checkpoint.

4.8 Imagem

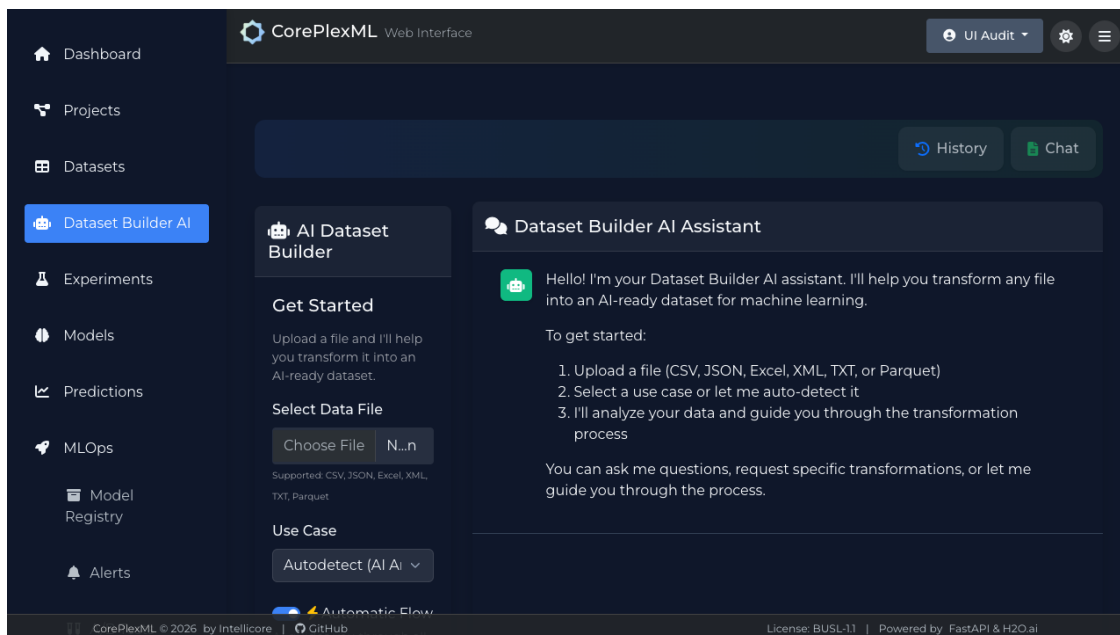


Fig. 1: Fluxo conversacional com preview de transformação.

CAPÍTULO 5

5. Experimentos, Modelos e Predicoes

5.1 Objetivo

Treinar modelos, comparar resultados, registrar melhor candidato e executar predicoes.

5.2 Criar experimento

1. Abrir Experiments.
2. Clicar em New Experiment.
3. Configurar: - Dataset/versao. - Coluna target. - Tipo de problema. - Estrategia de validacao e metrica objetivo.
4. Iniciar execucao e monitorar status.

5.3 Revisar resultados e registro de modelos

1. Abrir leaderboard apos conclusao.
2. Validar metrica principal e ordenacao.
3. Abrir detalhe do melhor modelo e revisar: - Metrics. - Explainability/importancia de features se disponivel. - Metadata de artifact/versao.
4. Registrar ou fixar modelo conforme fluxo.

5.4 Executar predicoes

1. Abrir Predictions.
2. Executar predicao unitria via formulario.
3. Executar predicao em lote via upload, se disponivel.

4. Validar campos de saída: - Valor/classe prevista. - Confiança/probabilidade quando aplicável. - Timestamp e referência da execução.

5.5 Checklist de validação funcional

1. Experimento chega a estado terminal sem falha silenciosa.
2. Métricas do detalhe batem com leaderboard.
3. Schema de saída de predição e estavel.
4. Predição em lote preserva paridade de linhas entrada/saída.
5. Erros de predição retornam mensagem acionável.

5.6 Resultado esperado

1. Pelo menos um modelo pronto para deploy.
2. Predições consistentes e rastreáveis.

5.7 Erros comuns e recuperação

1. Falha inicial de experimento: - Revisar target e tratamento de nulos.
2. Métrica inconsistente: - Confirmar mesmo split/seed.
3. Entrada rejeitada em predição: - Alinhar nomes/tipos com schema de entrada do modelo.

5.8 Imagens

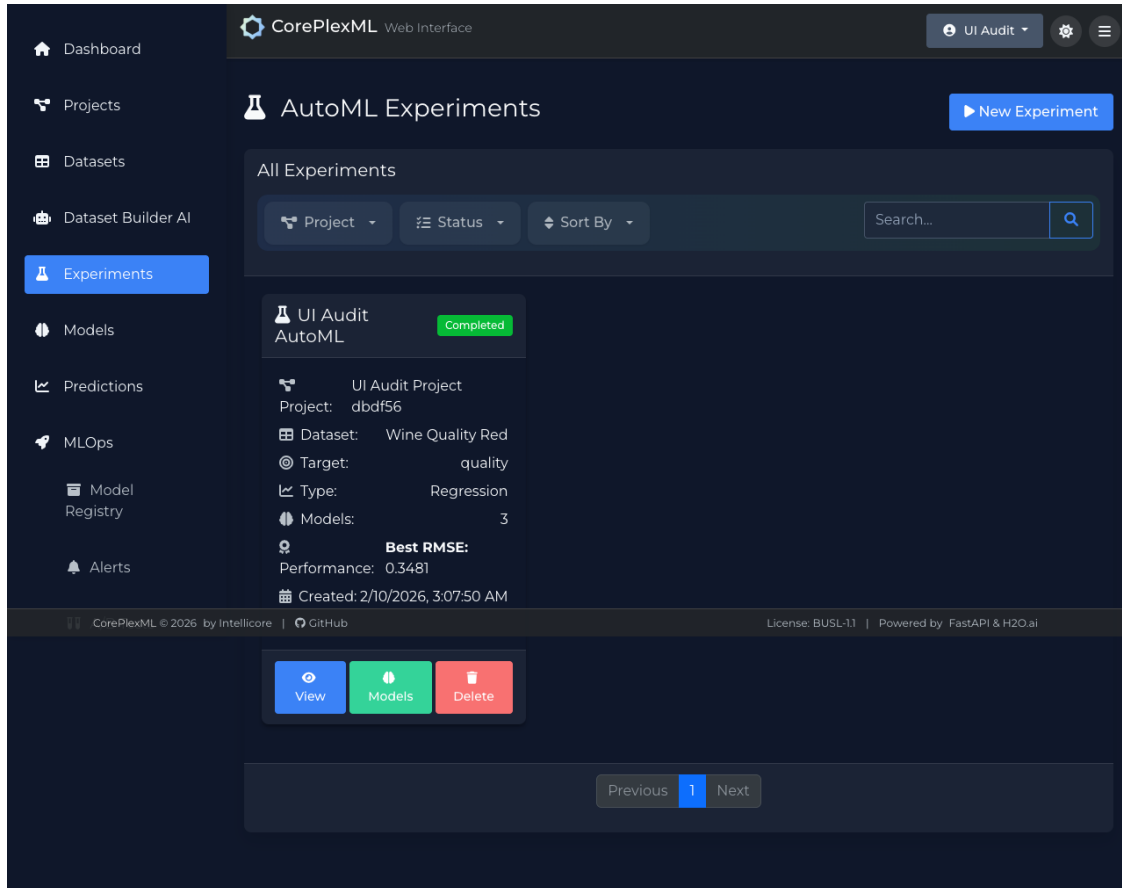


Fig. 1: Execucao de experimentos e monitoramento.

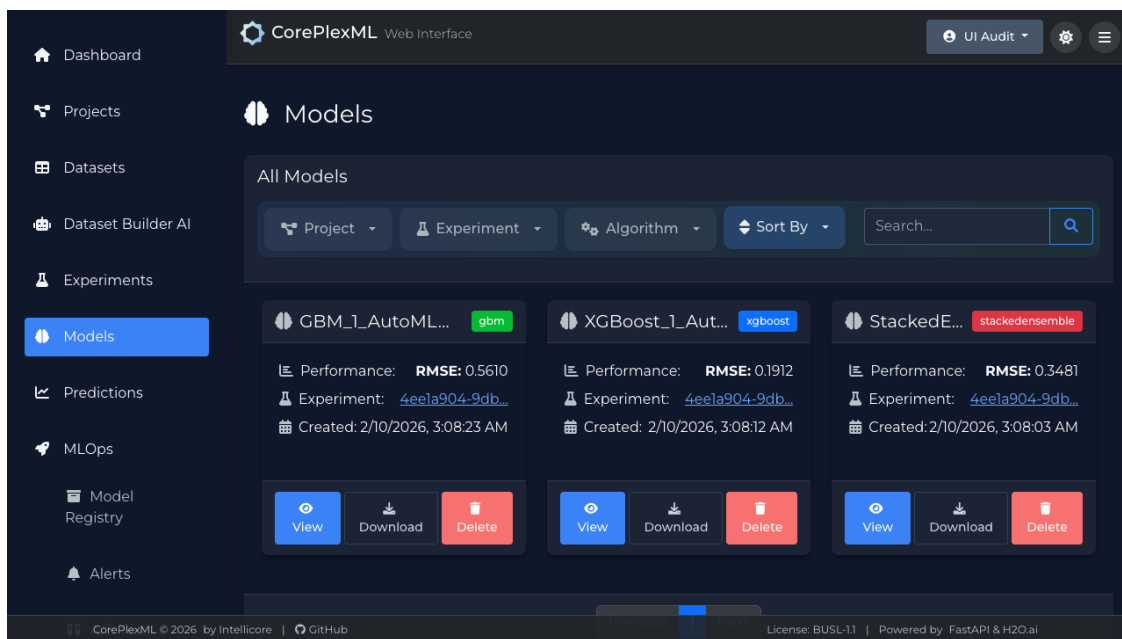


Fig. 2: Registro de modelos com metricas e metadata.

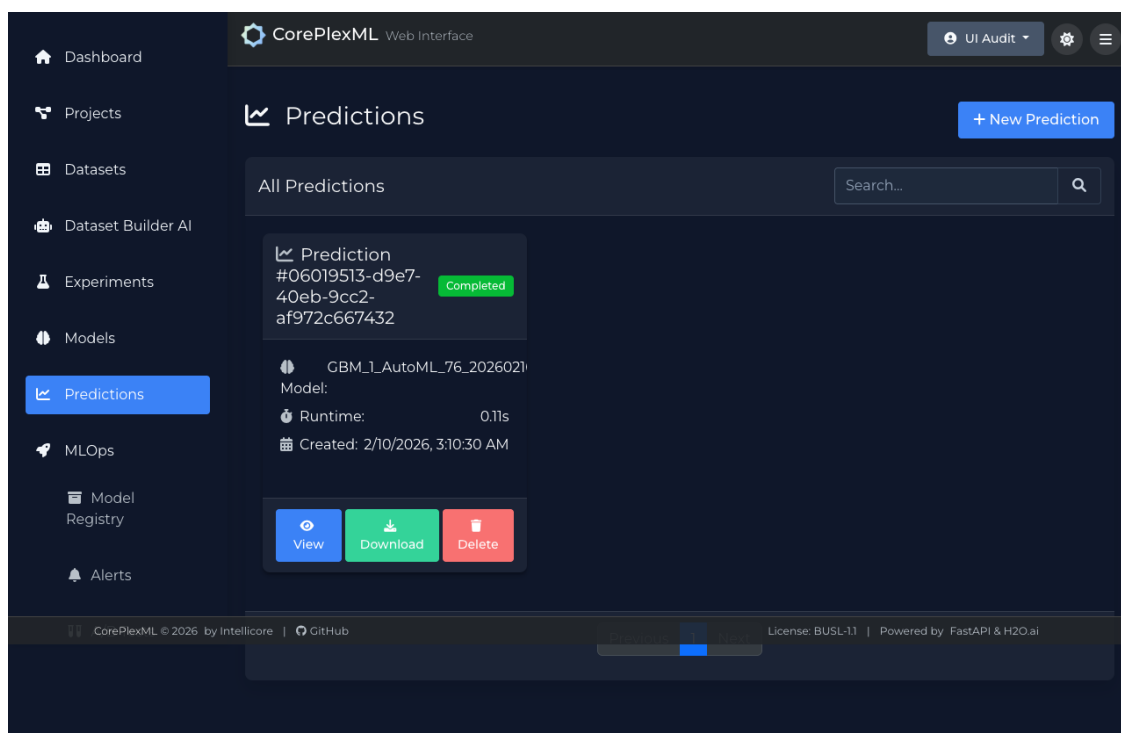


Fig. 3: Predicao unitria e em lote na UI.

CAPÍTULO 6

6. MLOps e AB Testing

6.1 Objetivo

Publicar modelos com segurança, monitorar runtime e comparar variantes com AB testing.

6.2 Criar deployment

1. Abrir MLOps > Deployments.
2. Selecionar versão de modelo.
3. Configurar: - Ambiente. - Perfil de réplicas/compute. - Estratégia de rollback.
4. Publicar e aguardar status active.

6.3 Monitorar deployment

1. Validar indicadores de saúde: - Uptime/health. - Error rate. - Latência.
2. Executar inferência de teste e validar resposta.
3. Revisar logs recentes para exceções.

6.4 Executar AB test

1. Abrir MLOps > A/B Tests.
2. Criar teste com: - Modelo baseline (A). - Modelo candidato (B). - Split de tráfego. - Métrica principal. - Amostra mínima/duração.
3. Iniciar teste e monitorar alocação.
4. Avaliar vencedor quando critério for atingido.

6.5 Checklist de validacao funcional

1. Deployment ativo responde sem downtime.
2. Saude e metricas atualizam em tempo operacional.
3. Split AB aparece no volume real de requests.
4. Vencedor reportado respeita metrica configurada.
5. Rollback disponivel em caso de degradacao.

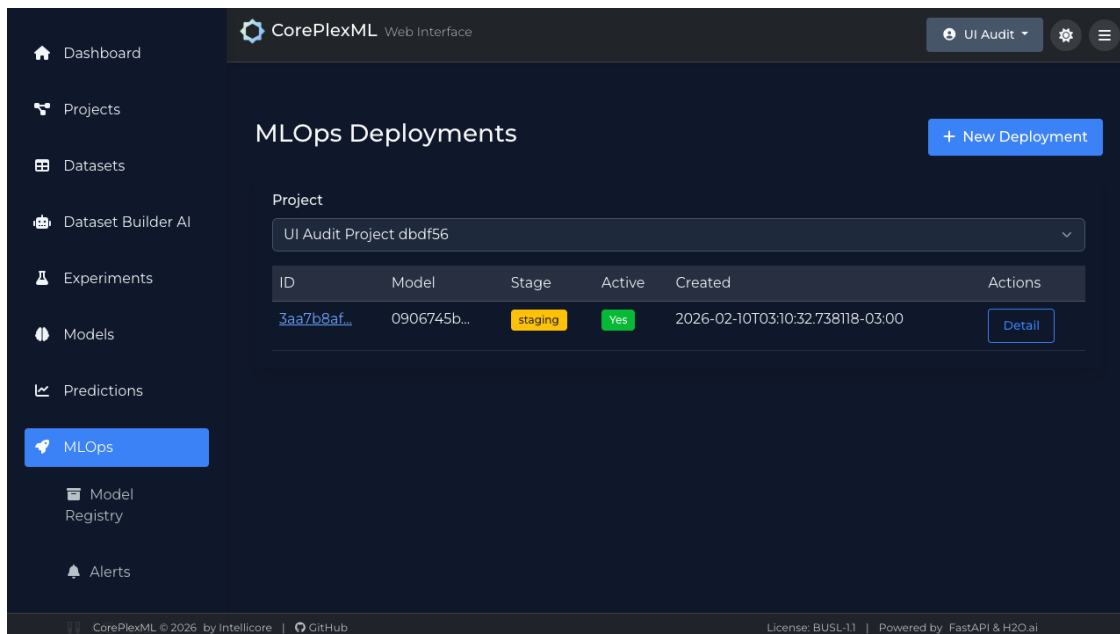
6.6 Resultado esperado

1. Rota produtiva estavel e observavel.
2. Promocao de modelos baseada em dados.

6.7 Erros comuns e recuperacao

1. Deployment preso em pending: - Verificar capacidade e artifact do modelo.
2. Error rate alto apos release: - Executar rollback imediato.
3. AB test inconclusivo: - Aumentar duracao ou amostra.

6.8 Imagens



CorePlexML Web Interface

MLOps Deployments

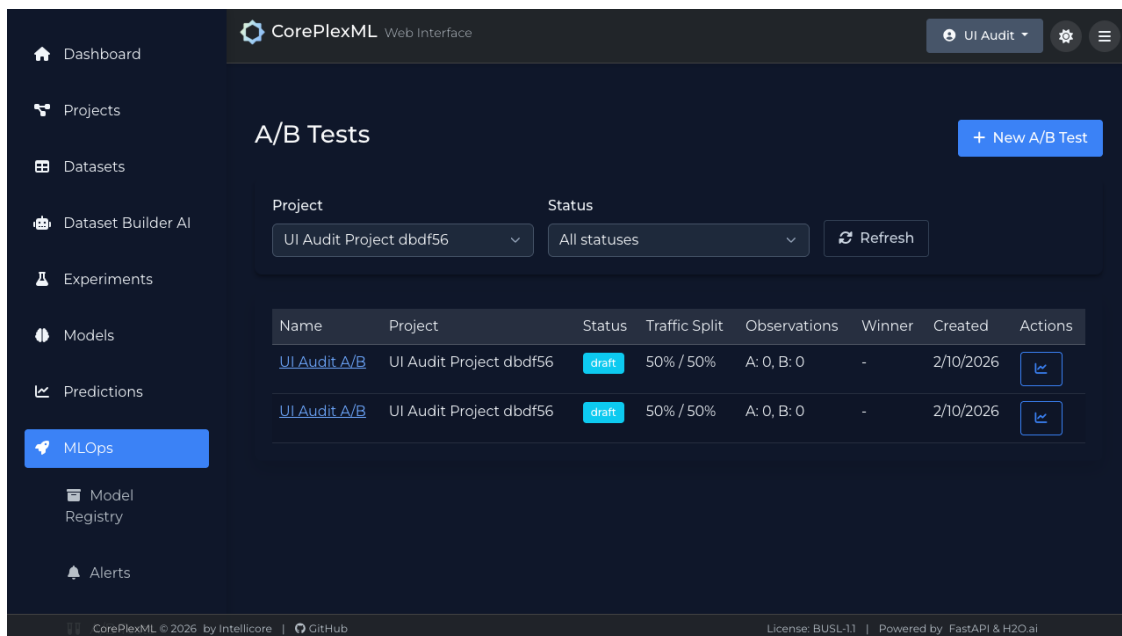
+ New Deployment

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

Fig. 1: Lista de deployments e status runtime.



The screenshot displays the CorePlexML Web Interface for A/B Testing. The interface includes a sidebar with navigation options: Dashboard, Projects, Datasets, Dataset Builder AI, Experiments, Models, Predictions, MLOps (highlighted), Model Registry, and Alerts. The main content area is titled "A/B Tests" and features a "+ New A/B Test" button. Below this, there are filters for "Project" (set to "UI Audit Project dbdf56") and "Status" (set to "All statuses"), along with a "Refresh" button. A table lists the active A/B tests:

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	

At the bottom of the interface, the footer contains the text: "CorePlexML © 2026 by Intellicore | GitHub License: BUSL-1.1 | Powered by FastAPI & H2O.ai".

Fig. 2: Configuracao e comparacao no AB testing.

CAPÍTULO 7

7. Privacy e SynthGen

7.1 Objetivo

Proteger dados sensíveis com políticas de privacidade e gerar datasets sintéticos com utilidade prática.

7.2 Fluxo de Privacy

1. Abrir Privacy.
2. Criar ou selecionar política.
3. Associar política ao dataset/sessão.
4. Rodar detecção e revisar entidades PII encontradas.
5. Configurar ação por regra (mask/redact/hash/encrypt/generalize/suppress).
6. Executar transformação e exportar dataset protegido.

7.3 Fluxo de SynthGen

1. Abrir SynthGen.
2. Selecionar dataset fonte e perfil de geração.
3. Configurar tamanho alvo e restrições opcionais.
4. Iniciar job de geração.
5. Baixar artefato sintético após conclusão.

7.4 Checks de qualidade e paridade

1. Paridade de schema: - Quantidade de colunas e nomes esperados.
2. Paridade de privacidade: - Valores sensíveis originais não aparecem na saída.
3. Paridade de utilidade: - Tendências de distribuição permanecem úteis.
4. Integridade: - Arquivo legível e row count conforme configuração.

7.5 Checklist de validação funcional

1. Detecção de PII retorna resultados quando existe PII.
2. Transformação gera artefato baixável.
3. SynthGen termina em estado `completed`.
4. Dataset sintético pode ser carregado em `Datasets/Experiments`.
5. Erros retornam diagnóstico acionável.

7.6 Resultado esperado

1. Dados sensíveis protegidos conforme política.
2. Dados sintéticos aptos para protótipo e testes.

7.7 Erros comuns e recuperação

1. Nenhum resultado PII quando esperado: - Revisar política/regras e colunas fonte.
2. Falha de transformação: - Reexecutar com escopo menor de regras.
3. Timeout no SynthGen: - Reduzir tamanho e tentar novamente.

7.8 Imagens

The screenshot displays the 'Privacy Suite' interface in CorePlexML. The sidebar on the left contains navigation items: Dashboard, Projects, Datasets, Dataset Builder AI, Experiments, Models, Predictions, MLOps, Model Registry, and Alerts. The main content area is titled 'Privacy Suite' and includes a 'Scan Dataset' button and a '+ New Policy' button. The current project is 'UI Audit Project dbdf56', showing 2 Policies and 0 Sessions. Summary cards indicate: 1 Active Policy, 0 Compliant, 0 Pending Review, and 1 Non-Compliant. A table of Privacy Policies shows one policy named 'UI Audit Privacy' with a HIPAA profile, 0 rules, active status, 0% compliance, and a creation time of 2/10/2026, 3:13:18 AM. Below this is a 'Recent Sessions' table with one session: 'Qed12364...' under the 'UI Audit Privacy' policy, 'PII Demo' dataset, 'detected' status, 9 PII found, and a creation time of 2/10/2026, 3:13:18 AM.

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

Fig. 1: Modulo Privacy com execucao de politicas.

The screenshot shows the CorePlexML Web Interface for the SynthGen Synthetic Data Generation project. The interface is dark-themed and includes a sidebar with navigation options: Dashboard, Projects, Datasets, Dataset Builder AI, Experiments, Models, Predictions, MLOps, Model Registry, and Alerts. The main content area features the SynthGen logo and title, with buttons for 'Train Model' and 'Generate Data'. Below this, a 'Project' dropdown menu is set to 'UI Audit Project dbdf56', showing '1 Models' and '1 Jobs'. A summary section displays four metrics: Total Models (1), Active Jobs (0), Completed (1), and Failed (0). A table titled 'Synthetic Data Models' lists one model: 'UI Audit SynthGen' using the 'CTGAN' synthesizer, with a 'ready' status, 'Wine Quality Red (v1)' dataset, and a 'N/A' quality score. The 'Recent Jobs' section shows a single job with ID 'c09fea70...', model 'UI Audit SynthGen', type 'Training', status 'completed', and creation time '2/10/2026, 3:13:18 AM'. The footer contains copyright information for Intellicore, GitHub, and FastAPI & H2O.ai.

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

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

Fig. 2: Fluxo de geracao e resultado do SynthGen.

CAPÍTULO 8

8. Teams, Billing, Admin e Troubleshooting

8.1 Objetivo

Operar colaboracao de times, ciclo de assinatura e controles admin com resultado previsivel.

8.2 Gestao de times

1. Abrir Teams.
2. Criar time ou abrir time existente.
3. Convidar membros por email.
4. Definir papel por membro (owner/admin/member conforme politica).
5. Validar que cada usuario acessa apenas modulos autorizados.

8.3 Billing e ciclo de plano

1. Abrir secao de perfil/billing.
2. Validar estado de plano atual (Free, Pro, Team).
3. Executar upgrade quando necessario: - Free -> Pro. - Pro -> Team.
4. Confirmar atualizacao de status apos checkout.
5. Validar habilitacao dos recursos de Team.

8.4 Controles Admin

1. Abrir Admin (somente papeis autorizados).
2. Revisar: - Audit logs. - Estado operacional. - Controles de tenant/team.
3. Aplicar mudancas apenas com ticket/aprovacao.

8.5 Checklist de validacao funcional

1. Convite, aceite e papel de time concluem com sucesso.
2. Upgrade de plano reflete no estado e permissoes da UI.
3. Status de assinatura persiste apos logout/login e refresh.
4. Paginas admin ficam bloqueadas para papeis nao autorizados.
5. Erros de billing sao explicitos e sem estado ambiguo.

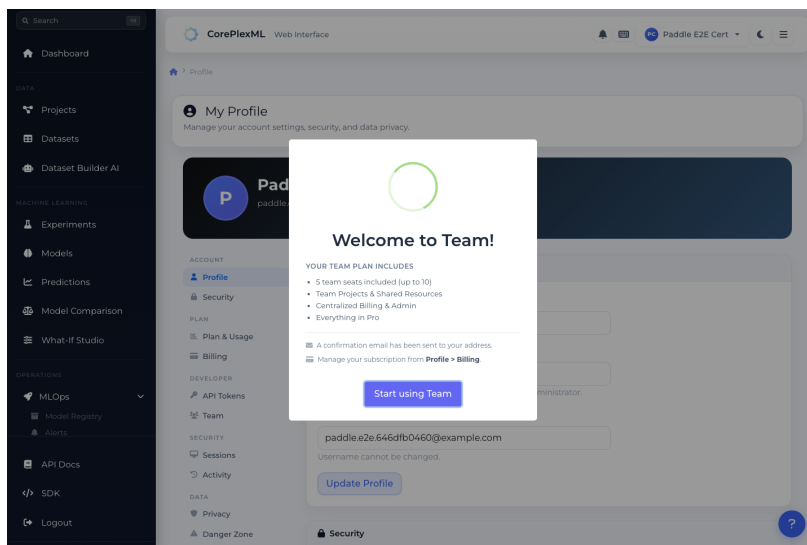
8.6 Resultado esperado

1. Colaboracao e billing funcionam ponta a ponta.
2. Admin diagnostica incidentes sem corromper dados.

8.7 Playbook rapido de recuperacao

1. Problema de sessao: - Logout/login e validar estado da conta.
2. Modulo/dado ausente: - Validar projeto ativo e permissoes.
3. Billing inconsistente: - Verificar provider da assinatura e owner do time.
4. UI desatualizada: - Hard refresh e repetir ultima acao.

8.8 Imagem



Controles de time e billing no perfil/workspace.