

PROGETTO ID: DIDATTIX-2025-001

“Registro Elettronico Didattico Integrato – DIDATTIX”

Piano Operativo Definitivo – Project Brief (D03)

## 1. Stato del documento (B01)

### 1.1 Informazioni generali

- Progetto: DIDATTIX-2025-001 – Registro Elettronico Didattico Integrato – DIDATTIX
- Documento: D03 – Piano Operativo Definitivo (Project Brief)
- Versione: 1.0
- Data emissione: 10 Dicembre 2025
- Stato rilascio: In approvazione
- Contributori: Sara Tantucci (PM), Daniele Strino, Filippo Hinceanu, Nicolò Fiorucci

### 1.2 Scopo e campo di applicazione

Il presente Piano Operativo Definitivo disciplina in modo puntuale l’assetto operativo e funzionale del progetto “Registro Elettronico Didattico Integrato – DIDATTIX”, definendo requisiti, architettura, modalità di erogazione del servizio e criteri di accettazione dei deliverable.

Il documento costituisce riferimento esecutivo per il team di progetto e per gli stakeholder scolastici, stabilendo le condizioni di conformità, le metriche di qualità e i presupposti per il collaudo formale della soluzione.

### 1.3 Approvazione e distribuzione

#### Approvazione

- Project Manager – Alterra Studios: Sara Tantucci – Data: 10 Dicembre 2025 – Firma: \_\_\_\_\_
- Utente Senior Scolastico – Istituto Beneficiario: [Nome Referente Scolastico] – Data: //\_\_\_\_ – Firma: \_\_\_\_\_
- Executive del Progetto – Istituto Beneficiario: [Nome Dirigente Scolastico] – Data: //\_\_\_\_ – Firma: \_\_\_\_\_

#### Distribuzione

Il presente documento è destinato ai seguenti soggetti: Taskforce Agenda Digitale in ASC, Umbria Digitale in ASC, Comitato di Progetto e P3O (se applicabile), Team Alterra Studios (development team).

#### Documenti di riferimento

Il D03 è da considerarsi coerente e coordinato con: D02 Business Case Preliminare, D04 Piano di Progetto Esecutivo (PID), disciplina regionale e linee guida per l’innovazione didattica digitale, nonché con gli Allegati A–D (specifica tecnica, piano di testing, architettura infrastrutturale, risk register dettagliato).

## 2. Descrizione dei prodotti specialistici (B05)

### 2.1 Piattaforma web DIDATTIX – Moduli core

La piattaforma DIDATTIX è un sistema informativo web-based modulare, destinato alla gestione integrata di presenze, valutazioni, ammonizioni disciplinari, utenze e reporting in ambito scolastico.

Ogni macro-funzione è implementata come modulo applicativo specializzato, con interfacce distinte per docenti, studenti, genitori/tutori e amministratori, su un’unica base dati centralizzata.

#### 2.1.1 Modulo Presenze

Il Modulo Presenze fornisce funzioni per la registrazione strutturata di presenze/assenze, l’analisi dei trend di frequenza e la notifica automatica alle famiglie in caso di situazioni critiche.

Principali funzionalità:

- Registrazione giornaliera presenze/assenze per classe e studente, con interfaccia ottimizzata per l’uso in aula, timestamp automatico e tracciamento del docente che effettua l’operazione.
- Visualizzazione cronologica delle presenze per studente su archi temporali configurabili (giorno, settimana, mese, trimestre, anno), con filtri per classe e periodo.
- Calcolo automatico di indicatori di frequenza (totale assenze, assenze consecutive, percentuale frequenza, giornate frequentate) e identificazione di pattern di rischio sulla base di soglie configurabili.
- Invio automatico di notifiche e-mail/SMS ai genitori/tutori al superamento di determinate soglie di assenza, con template personalizzabili e registro di consegna.

- Generazione di report di sintesi per docente, classe e istituto, con opzioni di export in formati standard (Excel, PDF).

#### 2.1.2 Modulo Valutazioni

Il Modulo Valutazioni gestisce l'intero ciclo di registrazione, aggregazione e consultazione dei voti e dei giudizi, supportando differenti scale di valutazione.

Principali funzionalità:

- Inserimento voti per materia, periodo (trimestre/quadrimestre/anno) e anno scolastico, con supporto a scale numeriche (1–10, 1–100), letterali (A–F) e descrittive, incluse scale internazionali quali ECTS.
- Calcolo automatico di medie per studente e per classe (aritmetiche e ponderate), con parametri di peso configurabili lato amministrazione.
- Generazione di pagelle in formato PDF, comprensive di dati anagrafici, voti, medie e giudizi, con possibilità di firma digitale e pubblicazione sul portale di studenti e genitori.
- Monitoraggio dei trend di rendimento per materia e studente, con evidenziazione di materie critiche, cali significativi e confronto con le medie di classe.
- Esportazione dei dati di valutazione in formati interoperabili (Excel, XML, CSV) a supporto di eventuali sistemi legacy o richieste amministrative.

#### 2.1.3 Modulo Ammonizioni

Il Modulo Ammonizioni consente la registrazione, la tracciabilità e l'analisi delle annotazioni disciplinari, a supporto della funzione educativa e della gestione dei casi critici.

Principali funzionalità:

- Creazione di annotazioni disciplinari con descrizione, data, livello di gravità, docente responsabile, studente/classe e tipologia di infrazione.
- Visualizzazione della storia disciplinare di ciascuno studente, con timeline delle ammonizioni e dettaglio delle azioni intraprese (colloqui, sospensioni, comunicazioni alle famiglie).
- Invio di notifiche automatiche via e-mail ai genitori/tutori al momento della registrazione di una nuova ammonizione, con possibilità di allegare comunicazioni ufficiali.
- Tracciamento dei follow-up (data colloquio, decisioni, misure correttive, esito), per consentire una valutazione storica dei percorsi educativi.
- Reportistica disciplinare aggregata per classe, periodo e tipologia di infrazione, con individuazione di pattern ricorrenti e studenti a rischio.

#### 2.1.4 Gestione Utenti, Ruoli e Portali

La gestione dell'accesso è basata su un modello combinato RBAC/ABAC, che differenzia in modo netto i profili (admin, docente, studente, genitore/tutore, viewer) e limita ciascun utente al proprio perimetro di dati.

Elementi principali:

- Creazione e gestione massiva degli account (anche via import CSV), con associazione a classi, materie e strutture organizzative.
- Autenticazione robusta tramite credenziali con hashing sicuro (bcrypt/Argon2) e password policy configurabile, con opzione di 2FA basata su TOTP.
- Predisposizione per integrazione con sistemi di identità federata (es. SPID/FedUmbria), con Single Sign-On e gestione centralizzata di sessione e logout.
- Portale Docente, Portale Studente e Portale Genitore/Tutore con viste dedicate su presenze, voti, ammonizioni, calendario, notifiche e comunicazioni.

### 2.2 Stack infrastrutturale (VM, Docker, Networking)

Lo stack infrastrutturale assicura un'erogazione stabile, sicura e monitorata della piattaforma, basata su una VM Linux e su un'architettura containerizzata.

Componenti principali:

- Virtual Machine con sistema operativo Ubuntu 22.04 LTS, 8 vCPU, 16 GB RAM, 80 GB SSD, IP statico in modalità bridged e politiche di snapshot per backup incrementali.
- Hardening dell'accesso amministrativo tramite autenticazione SSH esclusivamente a chiave, disabilitazione del login diretto root, timeout di sessione e logging esteso.

- Firewall UFW con politica “deny by default” in ingresso, apertura selettiva delle sole porte necessarie (SSH, HTTP/HTTPS, servizi di monitoring e SMTP locale).
- Docker Engine CE e Docker Compose v2 come piattaforma di containerizzazione, con rete bridge dedicata, volumi persistenti per dati di database, metriche, log e configurazioni.
- Container per frontend (nginx), backend PHP-FPM, MySQL, Prometheus, Grafana, Loki, Promtail e Node Exporter, configurati per l’auto-start e con health-check definiti.

## 2.3 Stack di monitoraggio e logging avanzato

Lo stack di osservabilità garantisce visibilità in tempo quasi reale sullo stato dell’infrastruttura, dell’applicazione e dei servizi di supporto.

- Prometheus come motore di raccolta, archiviazione e query di metriche time-series, con regole di alerting basate su soglie e condizioni di anomalia (CPU, RAM, latenza, error rate, ecc.).
- Grafana come layer di visualizzazione, con dashboard preconfigurate per hardware, container e applicazione, e meccanismi di alerting verso canali standard (email, Slack, webhook).
- Loki e Promtail per la centralizzazione e l’analisi dei log di sistema, dei container, dei servizi applicativi e di sicurezza, con ricerche avanzate tramite linguaggio LogQL.
- Node Exporter per l’esposizione delle metriche hardware di basso livello (CPU, memoria, disco, rete, uptime), integrato in Prometheus.

## 2.4 Automazioni, backup e health-check

Sono previste automazioni schedule per garantire la continuità operativa, la salvaguardia dei dati e l’individuazione precoce delle anomalie.

- Backup giornaliero del database MySQL tramite mysqldump, con compressione, retention minima di 30 giorni, politiche di cleaning e notifica email al termine di ogni esecuzione.
- Report giornaliero di sistema in formato HTML, contenente indicatori di sistema, servizi, applicazione, database e sicurezza, inviato automaticamente all’amministratore IT.
- Script di health-check eseguito con frequenza tipica di 5 minuti, con verifiche su frontend, API backend, database, stack di monitoring, risorse disco e memoria, e invio di email di allerta in caso di failure.
- Analisi automatizzata dei log su base giornaliera, con individuazione di pattern anomali (errori ricorrenti, tentativi di intrusione, errori applicativi) e produzione di un report sintetico per l’admin.

## 2.5 Documentazione tecnica e formazione

Il progetto prevede un set strutturato di documentazione tecnica e attività di formazione per garantire autonomia d’uso e manutenibilità nel medio-lungo periodo.

- Diagrammi di architettura applicativa, topologia di rete, deployment e schema ER del database, inclusi nei documenti tecnici di dettaglio.
- Repository GitHub privato con branch strategy definita, struttura delle directory standardizzata e README per ogni componente (frontend, backend, database, docker, script, docs).
- Documentazione API in formato OpenAPI/Swagger, con descrizione puntuale di endpoint, parametri, formati di richiesta/risposta e codici di errore.
- Runbook di deployment e di gestione operativa (provisioning VM, installazione prerequisiti, configurazione, startup container, check post-deploy).
- Piano di Disaster Recovery con RTO pari a 4 ore e RPO pari a 1 giorno, test periodici di restore e possibilità di failover su seconda VM.
- Workshop dedicati a docenti, amministratori IT, studenti/famiglie e manuali utente specifici per ciascun profilo.

## 3. Requisiti funzionali e criteri di accettazione (B11)

Per ciascun modulo sono definiti requisiti identificati da un codice univoco, categoria (es. Functional Suitability, Performance Efficiency, Operability), descrizione, criterio di accettazione, metodo di verifica e responsabile dell’accettazione.

### 3.1 Modulo Presenze

Esempi rappresentativi:

- Il sistema deve consentire al docente di registrare presenze/assenze per tutti gli studenti delle proprie classi tramite interfaccia web dedicata, con salvataggio persistente su database. Criterio: registrazioni correttamente salvate e recuperabili; verifica tramite UAT e test automatici.
- Ogni registrazione deve includere data, stato presenza/assenza, docente responsabile e timestamp dell'operazione; verifica tramite ispezione schema e log di audit.
- La consultazione cronologica delle presenze deve essere disponibile su intervalli temporali configurabili, con possibilità di export in formato Excel.

### 3.2 Modulo Valutazioni

Esempi rappresentativi:

- Il sistema deve consentire l'inserimento di voti per materia e periodo con supporto a più scale di valutazione; il docente deve poter selezionare la scala prevista per la materia.
- Devono essere generate automaticamente medie e pagelle in PDF, consultabili da studenti e genitori secondo le autorizzazioni di ruolo.
- Lo storico delle valutazioni per studente deve essere completamente consultabile, con vista tabellare e, ove previsto, grafico di trend.

### 3.3 Modulo Ammonizioni

Esempi rappresentativi:

- Il sistema deve consentire al docente la registrazione di ammonizioni disciplinari con tutti gli attributi richiesti (descrizione, data, gravità, studente, classe, tipologia).
- Deve essere disponibile la storia disciplinare completa dello studente, con cronologia delle annotazioni e degli eventuali follow-up.
- Deve essere inviata una comunicazione automatica ai genitori/tutori alla creazione di nuove ammonizioni, con registrazione degli esiti di consegna.

### 3.4 Gestione Utenti e Sicurezza

Esempi rappresentativi:

- Il sistema deve implementare un modello di controllo accessi basato su ruoli e attributi, limitando la visibilità dei dati ai soli soggetti legittimati.
- Le password devono essere memorizzate esclusivamente in forma di hash sicuro, con policy di complessità configurabile e possibilità di reset tramite canale email controllato.
- È richiesta la tracciatura di tutte le operazioni rilevanti ai fini di audit (accessi, modifiche, operazioni amministrative) con conservazione dei log per un periodo definito.

### 3.5 Requisiti non funzionali

Esempi rappresentativi:

- Il sistema deve garantire un tempo di risposta P95 inferiore a 500 ms per le principali operazioni, con carico di almeno 100 utenti concorrenti.
- L'uptime dell'applicazione in orario di servizio scolastico deve essere almeno pari al 99%, salvo manutenzioni programmate comunicate.
- Tutte le comunicazioni tra client e server devono avvenire esclusivamente tramite HTTPS con TLS 1.2 o superiore e configurazione sicura dei cipher.
- L'interfaccia utente deve rispettare i principi di accessibilità conformi al livello WCAG 2.1 AA.

## 4. Analisi dei rischi e strategie di mitigazione (B07)

Il progetto è supportato da un risk register strutturato, che classifica i rischi secondo probabilità, impatto e priorità, associando per ciascuno misure preventive, piani di contingency e owner di riferimento.

Esempi di rischi:

- Rischi di slittamento delle tempistiche di sviluppo e integrazione, mitigati da approccio iterativo per sprint, gestione dello scope e monitoraggio delle dipendenze critiche.
- Rischi di sicurezza (esfiltrazione o perdita di dati), mitigati da misure di hardening, backup, cifratura dei canali e controlli di sicurezza applicativa.

- Rischi infrastrutturali (guasti VM, indisponibilità servizi), mitigati da procedure di backup/restore, monitoraggio proattivo e opzione di failover su seconda VM.

## 5. Pianificazione temporale della fase di esecuzione (B13)

La realizzazione della soluzione DIDATTIX è articolata in sprint di durata tipica pari a 2–3 settimane, con milestone definite e tracciabilità dei deliverable.

- Sprint iniziali focalizzati su predisposizione infrastrutturale (VM, Docker, monitoring), repository e strutture base di codice.
- Sprint successivi dedicati allo sviluppo dei moduli core (Presenze, Valutazioni, Ammonizioni), integrazione frontend–backend e affinamento delle funzionalità.
- Fase finale centrata su testing avanzato (performance, sicurezza), UAT con stakeholder scolastici, formazione utenti e predisposizione documentazione definitiva.

Le dipendenze tra attività sono gestite per evitare colli di bottiglia sul percorso critico, in particolare in corrispondenza di testing e UAT pre-go-live.

## 6. Gestione della qualità e testing (B21)

La strategia di qualità si basa su un insieme coordinato di attività di verifica e validazione, condotte sia a livello tecnico sia a livello utente finale.

- Test di unità e integrazione su backend e frontend, con coverage minima concordata e integrazione in pipeline di build.
- Test di carico e stress per validare performance e stabilità dell'applicazione rispetto agli obiettivi prefissati.
- Test funzionali e UAT condotti con un campione rappresentativo di docenti, amministratori, studenti e genitori, sulla base di scenari d'uso predefiniti.
- Controlli di sicurezza (es. scansioni di vulnerabilità, verifica configurazione TLS, controlli su gestione credenziali e log).

L'esito positivo delle attività di testing, unitamente al rispetto dei criteri di accettazione definiti per i requisiti critici, costituisce condizione necessaria per il collaudo e l'entrata in esercizio della piattaforma DIDATTIX.