📑 Checklist Progetto Padel – Tabella

Fase	Attività principali	Done when
0 – Setup	- Crea progetto Spring Boot- Inizializza repo Git- Configura JaCoCo (≥80%)- Profili `dev/test` in `application.yml`- Struttura pacchetti `controller/service/model/repository/pattern`	Build `mvn test` passa e genera report JaCoCo
1 – Modello dati	- Entità `User`, `Match`, `Registration`, `Feedback`- Enum `Level`, `MatchType`, `MatchStatus`, `RegistrationStatus`- Vincoli: max 4 iscritti, 1 feedback per utente- Seeder con utenti e partite demo	Avvio app crea schema H2 e dati iniziali
2 – Auth & Profilo	- Registrazione/login/logout- Livello dichiarato in registrazione- Pagina profilo: livello dichiarato, percepito, partite giocate- Validazioni su email/password	Posso registrarmi, loggarmi e vedere il profilo
3 – Partite & Iscrizioni	- Creazione partite proposte (utente)- (Opz) Partite fisse (admin/tool)- Join/leave partite con limite 4- Stato `WAITING` → `CONFIRMED` al 4° iscritto	Posso creare/listare partite, join/leave con vincoli corretti

4 – Observer (Pattern)	- `DomainEventPublisher` + `EventListener`- Eventi: `MatchConfirmedEvent`, `MatchFinishedEvent`- Listener: `NotificationListener`, `StatsListener`- Trigger al 4° join → conferma e notifica	Su conferma match compaiono notifiche/log
5 – Stato FINISHED	- Marcare partita come `FINISHED` (manuale o automatica)- Su evento: incrementa `matchesPlayed` per ogni iscritto- Notifica utenti per feedback	Finishing match aggiorna statistiche e invia notifica
6 – Feedback & livello percepito	- Feedback post-partita con livello suggerito + commento- Calcolo `perceivedLevel` (media o moda, documentare scelta)- Vincolo: 1 feedback per coppia utente-partita	Dopo feedback il profilo mostra livello percepito aggiornato
7 – Strategy (Pattern)	- Interfaccia `ListingStrategy`- Strategie: `ByDate`, `ByPopularity`, `ByLevel`- Integrazione in `GET /matches?sort=...`	Cambiando parametro sort la lista cambia ordinamento
8 – Singleton (Pattern)	- `NotificationService` come Singleton- Unica istanza usata da listener/servizi	Tutte le notifiche passano da un unico servizio
9 – View & UX	- Lista partite (filtri + sort)- Dettaglio partita (iscritti, join/leave)- Profilo utente (statistiche)- Pagina feedback post-match- Sezione notifiche/log	Scenario demo end-to-end percorribile da UI
10 – Test & Coverage	- Unit test service layer (match, feedback, user)- Controller test principali (MockMvc)- Report JaCoCo ≥80%	Coverage ≥80% e report salvato

11 – UML	- Use Case (registrazione, match, join/leave, feedback)- Class Diagram (model + pattern)- 2 Sequence Diagram (match confermato, feedback post-match)	Diagrammi UML pronti in PNG/SVG
12 – Documentazione	- Relazione con: Introduzione, Requisiti, Architettura MVC, Pattern, UML, Implementazione, Test, Conclusioni- Inserire coverage e decisioni progettuali- ≤40 pagine	PDF finale in `doc/RelazioneProgetto.pdf`
13 – Demo & Consegna	- Script demo completo- README con istruzioni e credenziali demo- Tag `v1.0` + accesso repo docente	Demo funziona dall'inizio alla fine