



Università degli Studi di Padova

Laurea: Informatica

Corso: Ingegneria del Software

Anno Accademico: 2025/2026



**Gruppo RubberDuck**

email: GroupRubberDuck@gmail.com

## **Valutazione capitolati**

<b>Stato</b>	Validato
<b>Versione</b>	2.0
<b>Autori</b>	Aldo Bettega Davide Lorenzon Felician Mario Neculescu, Draghici Ana Maria
<b>Revisori</b>	Filippo Guerra Davide Testolin Aldo Bettega
<b>Validatori</b>	Filippo Guerra Davide Testolin
<b>Uso</b>	Esterno
<b>Destinatari</b>	Professor Tullio Vardanega Professor Riccardo Cardin

Versione	Data	Descrizione	Autore	Revisore	Validatore
2.0	2025-11-04	Riscrittura per nuova candidatura	Ana Maria Draghici, Felician Mario Necsulescu	Aldo Bettega	Davide Testolin
1.0	2025-10-29	Prima stesura	Aldo Bettega, Davide Lorenzon, Felician Mario Necsulescu	Davide Testolin	Filippo Guerra

## Indice

1	Introduzione .....	1
2	Valutazione del capitolato selezionato .....	1
2.1	Capitolato 1: Automated EN18031 Compliance Verification .....	1
3	Valutazione dei capitolati preferiti .....	2
3.1	Capitolato 7: Sistema di acquisizione dati da sensori .....	2
3.2	Capitolato 8: Smart Order .....	4
3.3	Capitolato 5: Nexum .....	5
4	Valutazione degli altri capitolati .....	6
4.1	Capitolato 9: View4Life .....	6
4.2	Capitolato 3: DIPReader .....	8
4.3	Capitolato 2: Code Guardian .....	8
4.4	Capitolato 6: Second Brain .....	9
4.5	Capitolato 4: L'app che Protegge e Trasforma .....	10

## 1 Introduzione

In questo documento viene presentata la valutazione dei capitolati del gruppo RubberDuck al fine di esporre le nostre motivazioni di scelta e le considerazioni riguardo ai progetti proposti dai proponenti. Sarà presentata una valutazione dettagliata di “Automated EN18031 Compliance Verification” di Bluewind, seguita da una panoramica dei capitolati preferiti. Saranno inoltre discusse brevemente le ragioni per cui altri capitolati non sono stati scelti.

## 2 Valutazione del capitolato selezionato

### 2.1 Capitolato 1: Automated EN18031 Compliance Verification

#### 2.1.1 Descrizione

Il capitolato prevede lo sviluppo di un’applicazione per automatizzare la verifica di conformità allo standard EN18031, norma armonizzata alla Direttiva RED, applicabile ai dispositivi radio commercializzati nel mercato europeo.

#### 2.1.2 Dominio applicativo

Il progetto si colloca nell’ambito della certificazione dei dispositivi radio, un contesto in cui la verifica manuale dei requisiti risulta lenta e soggetta a errori. L’applicazione proposta intende automatizzare il processo attraverso l’utilizzo di Decision Tree, che guidano l’utente nella compilazione delle verifiche e producono un report di conformità chiaro e tracciabile.

#### 2.1.3 Dominio tecnologico

Il progetto si colloca nell’ambito dello sviluppo software per la verifica automatizzata della conformità alla norma EN18031. L’applicazione dovrà integrare funzionalità di importazione, esecuzione e visualizzazione dei Decision Tree, oltre alla gestione delle dipendenze gerarchiche tra requisiti.

- L’azienda ha suggerito pertanto l’utilizzo di **Python 3.x** per la parte di **backend**, in quanto linguaggio versatile e adatto alla gestione della logica applicativa e dell’elaborazione dei dati.
- Per quanto riguarda il **frontend**, l’azienda lascia libertà di scelta sulle tecnologie da adottare, potendo utilizzare Python stesso o soluzioni web più orientate all’interfaccia grafica, a discrezione del gruppo di sviluppo.

Con supporto e consiglio, l’azienda si rende disponibile per fornire indicazioni e confronto tecnico durante le fasi di implementazione e scelta delle tecnologie più opportune.

#### 2.1.4 Motivazione della scelta

La scelta del capitolato è stata ampiamente condivisa dal gruppo per diversi motivi:

- **Disponibilità e supporto dell'azienda** : Bluewind si distingue per un approccio altamente collaborativo, offrendo 1-2 incontri settimanali con il team aziendale. Questa frequenza di interazione rappresenta un valore aggiunto significativo rispetto ad altri capitolati, garantendo un feedback immediato durante tutto il ciclo di sviluppo e riducendo drasticamente i rischi tipici dei progetti software.
- **Fattibilità del progetto** : il capitolato dimostra una notevole maturità progettuale concentrandosi strategicamente su un gruppo ristretto di requisiti della norma EN 18031-1, piuttosto che sull'intero standard che comprende centinaia di requisiti complessi. Questa delimitazione razionale dello scope presenta vantaggi concreti per il successo del progetto.
- **Settore in crescita** : la norma EN 18031 è diventata obbligatoria il 1° agosto 2025, solo pochi mesi fa, e l'intero ecosistema dei produttori di dispositivi wireless sta ancora adattandosi a questa nuova realtà normativa. Non si tratta dunque di lavorare su problemi consolidati, ma su una normativa appena entrata in vigore in un settore in rapidissima evoluzione.
- **Libertà tecnologica** : viene lasciata ampia libertà in materia, consentendo al gruppo di scegliere tecnologie con cui ha più familiarità.
- **Applicazione immediata** : L'azienda fornisce inoltre un caso studio concreto basato su una macchina del caffè IoT con connettività Wi-Fi e protocollo MQTT, permettendo di testare immediatamente la soluzione su uno scenario realistico e tangibile.

## 2.1.5 Conclusioni

Il gruppo non ha riscontrato criticità né relative al capitolato né riguardanti l'azienda proponente. Il capitolato è stato valutato molto positivamente in quanto affronta il tema dell'automazione dei processi, argomento di grande rilevanza nel contesto tecnologico contemporaneo. Si evidenziano inoltre la disponibilità e la professionalità dimostrate dall'azienda, che hanno contribuito in modo significativo alla decisione finale. Per tali ragioni, dopo anche un'attenta valutazione della competizione sugli altri capitolati di interesse, il gruppo ha scelto questo capitolato come prima opzione.

## 3 Valutazione dei capitolati preferiti

### 3.1 Capitolato 7: Sistema di acquisizione dati da sensori

#### 3.1.1 Descrizione

Il capitolato descrive lo sviluppo di una piattaforma cloud per la raccolta, gestione e visualizzazione di dati provenienti da sensori Bluetooth Low Energy (BLE).

### 3.1.2 Dominio applicativo

Il progetto si colloca nell'ambito dell'Internet of Things, con particolare riferimento ai sistemi di monitoraggio distribuito dei parametri vitali di una persona. Deve essere presente un'interfaccia web che visualizza in tempo reale le informazioni derivate dall'elaborazione dei dati acquisiti dai sensori.

### 3.1.3 Dominio tecnologico

Viene richiesto di sviluppare un simulatore di gateway-BLE in grado di inviare dati ad una piattaforma centrale che si occupa della gestione sicura di dati e li rende disponibili tramite API e un'interfaccia web. L'azienda ha suggerito l'utilizzo delle seguenti tecnologie:

- **Node.js** e **Nest.js** con **TypeScript** per lo sviluppo dei microservizi.
- **Go** per eventuali componenti ad alte prestazioni.
- **NATS** o **Apache Kafka** per la gestione di messaggi distribuiti.
- **Google Cloud Platform** e **Kubernetes** per una gestione centralizzata e servizi cloud affidabili e scalabili.
- **MongoDB** e **PostgreSQL** per la persistenza dati.
- **Redis** per migliorare le prestazioni del sistema.
- **Angular** per l'interfaccia web.

### 3.1.4 Aspetti positivi

- Il progetto offre l'occasione di affacciarsi ad un gran numero di tecnologie moderne e applicabili in progetti futuri.
- È di comune interesse la possibilità di approfondire tematiche avanzate legate ai sistemi distribuiti e alle architetture cloud.
- M31 ha lasciato un'ottima impressione in quanto disponibilità e flessibilità, in particolare durante l'incontro zoom si è dimostrata molto organizzata e puntuale nella risposta delle nostre domande.
- Pensiamo che il progetto possa avere un impatto concreto nella vita delle persone.

### 3.1.5 Aspetti negativi

- Competizione e grande interesse da parte di altri gruppi.

### 3.1.6 Conclusioni

Il progetto ha riscosso il consenso unanime del team. Tuttavia, a seguito dell'analisi della competitività per questo capitolo e della valutazione della disponibilità di posti per il capitolo C1 - Bluewind, il quale è risultato altrettanto soddisfacente per il gruppo, si è optato per quest'ultimo.

## 3.2 Capitolato 8: Smart Order

### 3.2.1 Descrizione

Il capitolato “Smart Order” ha come oggetto la gestione degli ordini di acquisto da parte dei clienti di un’azienda in contesti in cui gli ordini possono assumere diverse forme.

### 3.2.2 Dominio applicativo

L’obiettivo del progetto è automatizzare la ricezione di questi ordini, interpretarli correttamente e trasformarli in ordini cliente strutturati e pronti per l’inserimento nel database aziendale. Il progetto SmartOrder consiste in una piattaforma intelligente in grado di analizzare input multimodali (testo, audio e immagini) e convertirli automaticamente in ordini strutturati.

### 3.2.3 Dominio tecnologico

Per la realizzazione del progetto è previsto l’uso delle seguenti tecniche e tecnologie:  
L’uso di database per interfacciarsi con i sistemi aziendali  
Entrando più nello specifico:

- Database relazionali, per interfacciarsi con i sistemi aziendali.
- Modelli di Linguaggio e NLP.
- Visione Computazionale e OCR.
- Riconoscimento Vocale e Trascrizione.
- API REST.
- Comunicazione da/per il database.
- Interfaccia utente.

Dal punto di vista tecnologico sono state fornite molte indicazioni o suggerimenti ma senza porre vincoli stringenti sulle tecnologie specifiche da adottare.

### 3.2.4 Aspetti positivi

- Ha suscitato l’interesse del gruppo.
- L’azienda si è mostrata molto disponibile.
- Problema concreto: automatizzare la gestione ordini è un’esigenza reale delle aziende.
- Flessibilità tecnologica: ampia scelta tra tecnologie suggerite.

### 3.2.5 Aspetti negativi

- Complessità: richiede competenze in NLP, il tema non è chiaro a tutti i membri del gruppo.

### 3.2.6 Conclusioni

Non sono emersi aspetti critici o problematiche sostanziali che abbiano precluso la fattibilità del progetto. Tuttavia, dopo un’attenta valutazione comparativa dei

capitolati disponibili, il gruppo ha ritenuto che il capitolato C7 - M31 rispondesse in misura maggiore agli interessi del team.

### 3.3 Capitolato 5: Nexum

#### 3.3.1 Descrizione

Il Capitolato C5 consiste nell'estensione dell'ecosistema NEXUM con lo sviluppo di nuove funzionalità intelligenti che migliorino la gestione HR, il dialogo con gli studi dei Consulenti del Lavoro e l'esperienza digitale dei dipendenti.

#### 3.3.2 Dominio applicativo

L'attuale applicativo Nexum include già alcune funzioni. Il progetto mira a introdurre funzionalità intelligenti, 2 AI assistant:

- Un AI Assistant per la creazione di contenuti in grado di adeguare stile e tono comunicativo a quello dell'azienda.
- Un AI Copilot per i consulenti del lavoro (CdL).

Oltre al potenziamento di alcuni dei moduli già presenti.

#### 3.3.3 Dominio tecnologico

L'impresa proponente è stata molto specifica sulle tecnologie da utilizzare, in quanto il prodotto finale deve essere facilmente integrabile nei sistemi già esistenti.

- Front-end
  - Dashboard amministrativa (Angular).
  - PWA utenti finali (Next.js): hosting su AWS Amplify Hosting oppure alternativa S3+CloudFront (se full-static).
- API & Backend
  - Ruby on Rails.
  - Background jobs: Sidekiq su service Fargate dedicato + SQS come coda.
  - Cache/sessioni: ElastiCache for Redis.
  - Database: Amazon RDS for PostgreSQL.
  - Storage documenti: S3.
  - Sicurezza dati: KMS per chiavi gestite.
  - Segreti: Secrets Manager.
  - Identity/Access: Amazon Cognito (pool utenti/identity) oppure identity provider esterno.
- Integrazioni e-mail/notify: SES (email), SNS (notifiche push/eventi).
- Rete & Sicurezza
  - VPC con subnet public (ALB, NAT Gateway) e private (ECS/RDS/Redis).
  - Security Groups a "minimo privilegio".
  - WAF + AWS Shield su ALB/CloudFront.

- IAM policy granulari per task role ECS.
- Observability & Ops -CloudWatch Logs/Metrics/Alarms, X-Ray (tracing). -AWS Config + GuardDuty (compliance & threat detection).

### 3.3.4 Aspetti positivi

- Esperienza pratica reale: lavoro su un prodotto che andrà effettivamente sul mercato, non solo un progetto accademico
- Stack tecnologico moderno: esperienza con Ruby on Rails, Angular, AWS, AI generativa - competenze molto richieste.
- Collaborazione aziendale: networking con una software house e possibili opportunità di stage/lavoro successive.

### 3.3.5 Aspetti negativi

- Carico di lavoro elevato: progetto ambizioso con molti casi d'uso complessi.
- Complessità dell'integrazione: dover lavorare con un'architettura esistente (NEXUM Core).

### 3.3.6 Conclusioni

Benché il capitolato presenti aspetti tecnologici interessanti, l'ambiguità riscontrata nella definizione dei casi d'uso ha costituito un elemento di criticità, portando il gruppo a non sceglierlo come capitolato principale.

## 4 Valutazione degli altri capitolati

### 4.1 Capitolo 9: View4Life

#### 4.1.1 Descrizione

Nel capitolato si propone di progettare una piattaforma per la gestione degli impianti Smart nelle residenze protette.

#### 4.1.2 Dominio applicativo

Il capitolato si concentra nell'ambito dell'IoT, in particolare sulle residenze protette e sulla loro gestione. Si mira a creare un sistema che migliori la qualità di vita della persona, aiutandola nella gestione di un'illuminazione e temperatura adeguata e fornendo un supporto immediato in caso di disagi. L'applicazione intende anche efficientizzare l'uso di energia elettrica grazie a sensori e monitoraggio continuo dei consumi.

#### 4.1.3 Dominio tecnologico

Vengono posti dei vincoli tecnologici:

- Docker per l'infrastruttura cloud (oppure Terraform, AWS CDK V2 o Ansible).

- **KNX IoT 3rd party API** (un'interfaccia di programmazione per l'integrazione con il sistema KNX, lo standard internazionale per la domotica e l'automazione degli edifici) con l'utilizzo di OAuth2 per l'autenticazione.
- **AWS LightSail** o **AWS EC2** per l'infrastruttura cloud container. In alternativa si può usare un'architettura ibrida (serverless + container), sono suggeriti diverse tecnologie **AWS**:
  - **AWS SimpleNotificationService**.
  - **AWS Lambda**.
  - **AWS Kinesis**.
  - **AWS Athena**.
  - **AWS CloudWatch**.
  - **AWS S3**.
- **MySQL** e **PostgreSQL** per l'applicativo web responsive.
- **TimescaleDB** o **InfluxDB** per le analytics.
- **Angular**, **React** o **Flask** per il Frontend, con **Bootstrap** o **Tailwind** per lo stile.
- Per il backend sono proposte come soluzioni:
  - **NodeJS + Express**.
  - **Java + Spring**.
  - **Python + Flask** o **Fast API**.
- Build automatico con **GitHub Runners**.

#### 4.1.4 Aspetti positivi

- Contenuti: il tema del capitolato ha riscontrato un grande interesse comune da parte del gruppo.
- Tecnologie: il documento risulta molto dettagliato e lineare nelle tecnologie usate.
- Impatto: il capitolato intende rendere più efficiente un sistema di gestione di alloggi protetti, il che andrebbe a migliorare la qualità della vita di soggetti più fragili. Per questo il gruppo pensa che il progetto sia di alto valore sociale.

#### 4.1.5 Aspetti negativi

- Disponibilità: è stato riscontrato da diversi gruppi che l'azienda ha avuto difficoltà nel presentarsi agli incontri, cosa che non fa ben pensare in quanto a disponibilità.

#### 4.1.6 Conclusioni

Il capitolato per la sua complessità risulta essere una sfida stimolante che può generare dei risvolti positivi in ambito sociale, inoltre la forma del documento e la puntualità delle tecnologie utilizzate fanno trasparire una grande professionalità da parte dell'azienda. Tuttavia rimane il dubbio che l'azienda, rispetto ad altre, sia meno disponibile nel seguire gli studenti.

## 4.2 Capitolato 3: DIPReader

### 4.2.1 Descrizione

Il progetto DIPReader si colloca nell'ambito della gestione documentale di sistemi di archiviazione digitale.

### 4.2.2 Dominio applicativo

L'obiettivo è facilitare la gestione di archivi digitali massivi a fini gestionali, legali, ecc. Si mira a garantire l'autenticità, l'integrità, l'affidabilità, la leggibilità e la reperibilità dei documenti informatici, e a tale scopo è necessario il supporto alla ricerca tramite linguaggio naturale.

### 4.2.3 Dominio tecnologico

Le tecnologie consigliate sono state le seguenti:

- **SQLite** per il database.
- **FAISS** per la ricerca di documenti.
- **Angular o React** per il Frontend.

### 4.2.4 Aspetti positivi

- Impatto a livello concreto.
- Requisiti funzionali estremamente chiari e misurabili: rappresentazione semplificata pacchetto DIP, ricerca per metadati, anteprima documenti.
- React o Angular per frontend, tecnologie web standard con ampia documentazione e community.

### 4.2.5 Aspetti negativi

- Tecnologie: rispetto ad altri capitolati sembra che vengano usate poche tecnologie.
- Conservazione digitale è un settore molto specifico, poco conosciuto agli studenti e con terminologia tecnica complessa.
- Progetto più tecnico-documentale, meno accattivante rispetto ad altri.

### 4.2.6 Conclusioni

Il progetto non ha destato l'interesse del gruppo se non per qualche componente.

## 4.3 Capitolato 2: Code Guardian

### 4.3.1 Descrizione

L'obiettivo del progetto è la realizzazione di una piattaforma web, basata su un sistema ad agenti, per l'analisi di repository GitHub.

### 4.3.2 Dominio applicativo

Questo capitolato mira alla creazione di uno strumento per monitorare e migliorare la qualità del codice versionato tramite repository GitHub tramite analisi, valutazioni di vario tipo, reportistica automatica su test, documentazione e sicurezza e la

capacità di suggerire soluzioni in caso di lacune o altri problemi. Il progetto prevede l'uso di un sistema ad agenti con orchestratore.

#### 4.3.3 Dominio tecnologico

- **Node.js** e **Python** per il Backend.
- **React.js** per il Frontend.
- **MongoDB** o **PostgreSQL** per il database.
- **Github Actions** per la CI/CD e Integrazioni.
- **AWS** per l'architettura cloud.

#### 4.3.4 Aspetti positivi

- Tecnologie: le tecnologie usate sono molto richieste dal mercato e con ecosistemi molto ricchi.
- Impatto: uno strumento automatico che controlli la qualità del codice risulta utile e soddisfa un bisogno attuale.
- Comprensione profonda delle vulnerabilità comuni (SQL Injection, XSS, CSRF) spendibile in qualsiasi progetto futuro.
- Nessuna necessità di apprendere domini complessi, focus su codice e best practices.

#### 4.3.5 Aspetti negativi

- Interesse: non ha suscitato l'interesse di tutti i membri del gruppo.

#### 4.3.6 Conclusioni

Il progetto nel complesso risulta buono, per tecnologie usate e temi che si andrebbero ad affrontare, tuttavia non ha avuto un riscontro positivo da parte di tutti i membri del gruppo.

### 4.4 Capitolato 6: Second Brain

#### 4.4.1 Descrizione

L'obiettivo del capitolato è testare le potenzialità dell'intelligenza artificiale tramite un editor di testo minimale, basato sul linguaggio di markup MarkDown. Questo editor di testo deve supportare diverse funzioni basate sull'intelligenza artificiale.

#### 4.4.2 Dominio applicativo

L'editor di testo deve implementare operazioni come riscrittura del testo con un diverso stile, creazione di riassunti, analisi e distant writing.

#### 4.4.3 Dominio tecnologico

Il capitolato intende sviluppare un'applicazione web con le seguenti tecnologie:

- **HTML** per l'interfaccia web.
- **Markdown** come linguaggio di markup per la formattazione, con apposite librerie di rendering per la visualizzazione.

- **API OpenAI-compatibili** per l’interazione con gli LLM.
- **Gemini, Mistral e Gemma** come LLM.
- **Java o Python** per la parte server-side in caso di estensione database (che richiederà il conseguente uso di tecnologie apposite alla persistenza dati).
- **HTTP** come protocollo per la gestione delle chiamate tra client e server.

#### 4.4.4 Aspetti positivi

- Azienda: si tratta di una grande azienda da cui ci si aspetta una grande professionalità, inoltre risulta esperta nel collaborare con studenti avendo anni di esperienza nella gestione di progetti di questo tipo.
- Focus su Large Language Models (LLM) e capacità di prompt engineering, competenze molto richieste al momento.
- Tutti conoscono e usano editor di testo quotidianamente, dunque interfaccia intuitiva.

#### 4.4.5 Aspetti negativi

- Contenuti: non è di comune interesse trattare tematiche come gli LLM e lavorare con questo tipo di tecnologie.
- “Sei cappelli per pensare” applicati a LLM è un esperimento, successo non assicurato.
- Meno opportunità di apprendere competenze architetturali complesse rispetto ad altri capitoli.

#### 4.4.6 Conclusioni

Il progetto proposto da Zucchetti presenta indubbiamente aspetti interessanti e attuali, con un focus su tecnologie all'avanguardia come i Large Language Models e competenze emergenti nel campo del prompt engineering. Tuttavia, dopo un'attenta analisi e discussione interna, il capitolo non ha raccolto il consenso unanime necessario all'interno del gruppo.

### 4.5 Capitolo 4: L'app che Protegge e Trasforma

#### 4.5.1 Descrizione

Il presente capitolo ha per oggetto l'affidamento dei servizi di progettazione, sviluppo e implementazione di un'applicazione mobile innovativa denominata “L'app che Protegge e Trasforma”, finalizzata alla prevenzione e supporto delle vittime di violenza di genere.

#### 4.5.2 Dominio applicativo

Il progetto deve fornire un'applicazioni mobile per iOS e Android con un'interfaccia utente intuitiva e accessibile che offre diverse diverse funzionalità tra cui:

- rilevamento con AI di situazioni pericolose.
- alert ad una rete di contatti di aiuto.

- formazione degli utenti e prevenzione della violenza di genere.
- uno spazio sicuro per la condivisione di esperienze.

#### 4.5.3 Dominio tecnologico

- **Flutter** per lo sviluppo della parte mobile.
- Viene consigliato **AWS** per ospitare l'infrastruttura di Backend.
- **NoSQL** per db non relazionali.

#### 4.5.4 Aspetti positivi

- Progetto con forte valore etico e che permette la possibilità di contribuire concretamente a un problema sociale rilevante.
- Architettura serverless su AWS, che rappresenta lo standard moderno per applicazioni scalabili.
- Esperienza full-stack completa: mobile (Flutter), backend (AWS Lambda), AI/ML, sicurezza, database NoSQL e relazionali.
- Formazione iniziale specifica sul tema della violenza di genere per comprendere a fondo il contesto e le esigenze degli utenti.

#### 4.5.5 Aspetti negativi

- Necessità di apprendere simultaneamente Flutter, AWS serverless, Amazon Bedrock/SageMaker e tutta una serie di tecnologie avanzate e diverse tra loro.
- Scope: il documento non è chiarissimo e risulta quasi dispersivo, il numero di requisiti sembra molto alto rispetto ad altri capitoli.
- Necessità di comprendere a fondo le dinamiche della violenza di genere per evitare funzionalità che potrebbero mettere a rischio le utenti.

#### 4.5.6 Conclusioni

Il progetto, oltre ad avere una complessità elevata per quanto riguarda la sensibilità del tema, non è stato approvato in modo unanime dal gruppo perché i requisiti non sembrano essere precisi e puntuali, essendo state elencate molte macrofunzionalità.