



*ByteOps.swe@gmail.com*

## Valutazione Capitolati

### Informazioni documento

---

<b>Data</b>	26/10/2023
<b>Redattori</b>	A. Barutta R. Smanio N. Preto
<b>Verificatori</b>	E. Hysa L. Skenderi D. Diotto
<b>Amministratore</b>	F. Pozza
<b>Destinatari</b>	T. Vardanega R. Cardin

# Indice

ByteOps

February 12, 2024

## Contents

<b>1</b>	<b>Valutazione del capitolato scelto</b>	<b>3</b>
1.1	<b>Capitolato C6</b> - SyncCity: Smart city monitoring platform . . . . .	3
<b>2</b>	<b>Valutazione sui capitolati rimanenti</b>	<b>4</b>
2.1	<b>Capitolato C1</b> - Knowledge management AI . . . . .	4
2.2	<b>Capitolato C2</b> - Sistemi di raccomandazione . . . . .	5
2.3	<b>Capitolato C3</b> - Easy meal . . . . .	7
2.4	<b>Capitolato C4</b> - A ChatGPT plugin with Nuvolaris . . . . .	8
2.5	<b>Capitolato C5</b> - WMS3: Warehouse Management 3D . . . . .	9
2.6	<b>Capitolato C7</b> - Chat GPT vs Bedrock developer analysis . . . . .	10
2.7	<b>Capitolato C8</b> - JMAP: il nuovo protocollo per la posta elettronica . . . . .	11
2.8	<b>Capitolato C9</b> - ChatSQL: creare frasi SQL da linguaggio naturale . . . . .	12
<b>3</b>	<b>Conclusioni</b>	<b>13</b>

# 1 Valutazione del capitolato scelto

## 1.1 Capitolato C6 - SyncCity: Smart city monitoring platform

**Proponente:** Sync Lab

**Obiettivo:** Sviluppare una *piattaforma<sub>G</sub>* di monitoraggio di una "Smart City" che consenta di avere sotto controllo lo stato di salute della città in modo tale da prendere decisioni veloci, efficaci ed analizzare poi gli effetti conseguenti.

A tale scopo il *proponente<sub>G</sub>* richiede di simulare dei sensori posti in diverse aree per reperire informazioni relative alle condizioni della città come, ad esempio, temperatura, umidità, polveri sottili nell'aria, traffico, livelli di acqua, stato di riempimento delle isole ecologiche, guasti elettrici e molto altro.

I dati trasmessi in tempo reale dai sensori devono poter essere memorizzati in un *database<sub>G</sub>* in modo tale da renderli disponibili per la visualizzazione tramite una *dashboard<sub>G</sub>*, composta anche da *widget<sub>G</sub>* e grafici, per una visione d'insieme delle condizioni della città in tempo reale.

L'applicativo potrà consentire alle autorità locali di prendere decisioni informate e tempestive sulla gestione delle risorse e sull'implementazione di servizi e, inoltre, si potrebbe rivelare uno strumento essenziale per coinvolgere i cittadini nella gestione e nel miglioramento della città.

**Tecnologie suggerite:** L'azienda ha fortemente suggerito l'utilizzo delle seguenti tecnologie, ritenute ideali per la gestione del *data streaming processing<sub>G</sub>*:

- **Python** (*Faker o simili*): Per la simulazione delle informazioni provenienti dai sensori.
- **Apache Kafka**: Broker per *disaccoppiare<sub>G</sub>* lo *stream<sub>G</sub>* di informazioni provenienti dai simulatori dei sensori.
- **ClickHouse**: *database<sub>G</sub>* OLAP per mantenere i numerosi dati provenienti dai sensori.
- **Grafana**: *piattaforma<sub>G</sub>* di Data Visualization per permettere il monitoraggio della città e la visualizzazione delle informazioni raccolte dai sensori.

### Considerazioni

- Interessante approfondimento sulla gestione dei Big Data nell'ambito dell'IoT.
- Tema e obiettivi affascinanti e futuristici.
- Ogni membro del gruppo ha ritenuto molto utile ed interessante l'apprendimento delle tecnologie proposte.
- Disponibilità dell'azienda con affiancamento di professionisti e incontri di supporto.
- Formazione sulle nuove tecnologie da parte di Sync Lab.

## 2 Valutazione sui capitolati rimanenti

### 2.1 Capitolato C1 - Knowledge management AI

**Proponente:** AzzurroDigitale

**Obiettivo:** L'obiettivo del progetto è sviluppare una *piattaforma<sub>G</sub>* di *Knowledge Management* che, sfruttando le capacità dell'AI basate sull'apprendimento automatico e algoritmi avanzati di ricerca e comprensione del linguaggio naturale, fornisca un'interfaccia chat per rendere la documentazione aziendale più accessibile ai dipendenti. Inoltre la *piattaforma<sub>G</sub>* Web deve permettere la gestione dei documenti come il caricamento, la consultazione e l'eliminazione degli stessi. L'obiettivo finale è consentire ai dipendenti di poter fare domande mirate e ricevere risposte immediate in modo tale da migliorare l'assistenza e la formazione del personale.

**Tecnologie suggerite:**

- **OpenAI:** Motore per le funzionalità di Natural Language Processing, ovvero di comprensione del testo e di generazione delle risposte.
- **Langchain:** Progetto *open-source<sub>G</sub>* che permette di integrare modelli AI senza conoscerne i dettagli interni.
- **NodeJS, Angular:** Per lo sviluppo della *piattaforma<sub>G</sub>* Web.

**Considerazioni:**

- Il progetto propone una soluzione per affrontare problematiche esistenti nel contesto lavorativo, concentrandosi in particolare sulla questione della sicurezza sul lavoro.
- Impiego dell'intelligenza artificiale per migliorare ed automatizzare *processi<sub>G</sub>* aziendali.
- Interessante approfondimento sull' apprendimento automatico e sul NLP (*Natural Language Processing*)
- Possibilità di modellare le funzionalità di *OpenAI* per scopi specifici.
- Relativamente alle tecnologie suggerite, è disponibile un'ampia documentazione.

## 2.2 Capitolato C2 - Sistemi di raccomandazione

**Proponente:** Ergon

**Obiettivo:** L'obiettivo del capitolato è creare un *sistema<sub>G</sub>* di raccomandazione che permetta ad aziende nel settore delle vendite multiprodotto di effettuare proposte commerciali mirate a clienti target.

L'implementazione di questi sistemi può avvenire in diverse modalità, le più diffuse sono:

- **Collaborative Filtering:** Si elabora un grado di similarità tra utenti e in base a questo, vengono proposti dei prodotti per uno specifico utente in base al comportamento degli altri utenti.
- **Content-Based Filtering:** Si elabora un grado di similarità tra prodotti e in base a questo, vengono proposti dei prodotti per uno specifico utente in base al suo comportamento.

Il *proponente<sub>G</sub>* richiede quindi di realizzare le seguenti componenti:

- Un *database<sub>G</sub>* contenente dati relativi al comportamento dei clienti.
- Un *sistema<sub>G</sub>* di raccomandazione che dovrà reperire i dati per l'apprendimento dal *database<sub>G</sub>*, effettuare una fase di pre-processing dei dati ed infine elaborare i dati per sviluppare, grazie ad algoritmi di machine learning, un modello che consenta di effettuare le raccomandazioni richieste.
- Un'interfaccia (Desktop-based oppure Web-based) per la visualizzazione dei risultati e la gestione dei feedback degli utenti.

**Tecnologie suggerite:** Per la realizzazione del *sistema<sub>G</sub>* di raccomandazione è consigliato uno tra:

- **ML.NET:** Un *framework<sub>G</sub>* che permette di sviluppare applicazioni di Machine Learning che utilizza il linguaggio C#.
- **Surprise:** Una libreria in ambito *Python* utilizzata per la creazione e l'analisi di sistemi di raccomandazione.

Per l'interazione con il Database, che deve essere di tipo relazionale, vengono consigliati diversi approcci:

- **Entity Framework:** Nel caso si utilizzi *ML.NET*.
- Una fonte dati **ODBC** (*Open Database Connectivity*) nel caso si utilizza la libreria *Surprise*.

- Implementazione un middleware che gestisca la comunicazione tra i componenti (es. *JSON*) che renda indipendente il *sistema<sub>G</sub>* di raccomandazione dal *database<sub>G</sub>*.

**Considerazioni:**

- La libreria Surprise offre un ambiente solido e completo per la progettazione di sistemi di raccomandazione, consentendo l'*integrazione<sub>G</sub>* con le altre numerose *librerie<sub>G</sub>* Python.
- L'azienda è disponibile a fornire un consistente set di dati da utilizzare per l'apprendimento del modello.
- Approfondimento di notevole interesse riguardante i sistemi di raccomandazione, i quali sono onnipresenti nella nostra esperienza quotidiana di navigazione e fruizione di contenuti sul Web.

## 2.3 Capitolato C3 - Easy meal

**Proponente:** Imola Informatica

**Obiettivo:** Sviluppare un'applicazione *Web Responsive* che consentirà agli utenti di gestire agevolmente le prenotazioni e le ordinazioni presso i ristoranti affiliati al *servizio<sub>G</sub>*.

L'applicazione metterà a disposizione degli utenti la possibilità di prenotare un tavolo presso il ristorante di loro scelta, oltre a gestire le ordinazioni: ogni commensale al tavolo avrà la facoltà di selezionare autonomamente le pietanze dal menù, con la comodità di monitorare le ordinazioni degli altri partecipanti e comunicare eventuali allergie o richieste speciali relative ai piatti scelti.

Inoltre, grazie all'*integrazione<sub>G</sub>* di una chat diretta con il personale del ristorante, gli utenti potranno ottenere informazioni rapide e dettagliate sui piatti offerti e ricevere assistenza in tempo reale. Per quanto riguarda il pagamento del conto, l'applicazione offrirà la possibilità di effettuare transazioni, consentendo ai clienti di dividere il conto tra i partecipanti al tavolo e persino di coprire le spese per altri membri del gruppo.

Infine, gli utenti saranno in grado di condividere le proprie opinioni e feedback sui ristoranti e sui piatti, contribuendo così alla creazione di una raccolta di recensioni affidabili che beneficeranno tutti gli utenti iscritti alla *piattaforma<sub>G</sub>*.

**Tecnologie suggerite:** Per la realizzazione del progetto l'azienda non impone nessun vincolo particolare relativo alle tecnologie da utilizzare. In generale l'applicazione dovrà essere Web responsive (PC, IOS e Android) e dovrà avere una copertura di *test<sub>G</sub>* maggiore o uguale all'80%.

### Considerazioni

- Nonostante lo scopo del progetto non risulti particolarmente innovativo, rappresenta comunque una eccellente risorsa di apprendimento, specialmente perché molte delle funzionalità richieste sono comuni nell'ambito dello sviluppo Web.
- Flessibilità nella scelta delle tecnologie da utilizzare e possibilità di ampliare il bagaglio di tecnologie conosciute per lo sviluppo di applicazioni Web.
- Opportunità di estendere in modo significativo il progetto grazie ai requisiti opzionali, i quali risultano particolarmente interessanti.

## 2.4 Capitolato C4 - A ChatGPT plugin with Nuvolaris

**Proponente:** Nuvolaris

**Obiettivo:** Il progetto si propone di sviluppare un plugin di *ChatGPT* dedicato alla creazione di categorie di applicazioni tramite Nuvolaris.

Il caso d'uso tipico consiste nell'utente che richiede a *ChatGPT* di generare un'applicazione. Il plugin interpreta questa richiesta e, a partire da un template, crea ed effettua il deployment dell'App.

In seguito, l'utente, tramite *ChatGPT*, può richiedere la personalizzazione dell'applicazione appena creata. Queste richieste di personalizzazione vengono anch'esse interpretate dal plugin, il quale, tramite delle funzioni, apporterà le modifiche necessarie al file di configurazione dell'applicazione, eseguirà il re-build ed effettuerà il deployment.

**Tecnologie suggerite:**

- **Nuvolaris:** Piattaforma che permette lo sviluppo serverless (senza la gestione complessa del server) di applicazioni cloud-native che siano altamente scalabili, altamente resilienti e semplici da realizzare.
- **ChatGPT Pro:** Viene fornito un account professionale con accesso illimitato a tutte le funzionalità.

**Considerazioni**

- Idea innovativa che unisce il potenziale di ChatGPT con lo sviluppo di applicazioni cloud-native, le quali stanno guadagnando sempre maggiore rilevanza nell'ambito dell'IT.
- Nonostante uno dei punti di forza di Nuvolaris sia la sua semplicità d'uso, nessun membro del gruppo ne è familiare, di conseguenza, il processo di formazione richiesto potrebbe comportare un notevole dispendio in termini di tempo.
- Data la compatibilità di Nuvolaris con molti dei più comuni linguaggi di programmazione c'è ampia flessibilità nella scelta del linguaggio da utilizzare.
- La presentazione del capitolato da parte del *proponente<sub>G</sub>* non è stata ritenuta convincente.



## 2.5 Capitolato C5 - WMS3: Warehouse Management 3D

**Proponente:** Sanmarco Informatica

**Obiettivo:** Lo scopo del progetto è la realizzazione di un WMS3 (*Warehouse Management 3D*) ovvero un *software<sub>G</sub>* per la simulazione di un magazzino 3D che permette di monitorare e ottimizzare le *attività<sub>G</sub>* di stoccaggio, movimentazione e distribuzione dei prodotti all'interno di un magazzino.

Rispetto alla gestione tradizionale di un magazzino emergono i seguenti vantaggi:

- **Maggiore comprensione spaziale:** Visione più accurata delle proporzioni e delle relazioni spaziali tra gli oggetti. Questo è particolarmente utile quando si progetta la struttura dei magazzini.
- **Migliore visualizzazione dei dati complessi:** Facilita la comprensione del livello di occupazione delle celle (Bin) di magazzino così da consentire di eseguire azioni di ottimizzazione degli spazi.
- **Visualizzazione e presentazione:** Identificare il corretto stoccaggio dei materiali in relazione alla classe della cella e alla classe di movimentazione dei codici stoccati.
- **Esecuzione di simulazioni:** Al fine di capire come modifiche strutturali di magazzino o del flusso dei materiali possano aumentare l'efficienza del *servizio<sub>G</sub>*.

**Tecnologie suggerite:**

- **Three.js:** Framework cross-platform per la realizzazione di contenuti 3D per il Web.

In alternativa:

- **Unity:** Programma per la realizzazione di contenuti 3D, basato su linguaggio C++.
- **Unreal Engine:** Programma per la realizzazione di contenuti 3D, basato su linguaggio C#.

**Considerazioni**

- Particolarmente stimolante l'apprendimento delle tecnologie proposte, ciascuna delle quali rientra nell'ambito del rendering 3D. Nonostante ciò il processo di formazione potrebbe richiedere un notevole investimento di tempo.
- Lo scopo del progetto è considerato innovativo, altamente utile e comporta benefici tangibili.
- La quantità di funzionalità da implementare non è ben definita.

## 2.6 Capitolato C7 - Chat GPT vs Bedrock developer analysis

**Proponente:** Zero12

**Obiettivo:** L'obiettivo del capitolato è quello di sfruttare l'intelligenza artificiale per supportare i *processi<sub>G</sub>* di sviluppo *software<sub>G</sub>*.

Viene richiesto di realizzare un Middleware (sotto forma di interfaccia Web) che riceva in input dei requisiti di business e produca epic e user stories associate a quest'ultimi tramite *Chat GPT* e *AWS BedRock*.

Inoltre, è richiesto di comparare la capacità di ChatGPT e quella di BedRock nell'interpretare il codice sorgente ed associarlo alle user stories generate con l'obiettivo di verificare se vengono soddisfatti i criteri di accettazione. Questa funzionalità sarà resa disponibile mediante lo sviluppo di un plugin per *Visual Studio Code* e *Apple Xcode*.

Al fine di ottimizzare i risultati futuri è necessario che, dopo il processo di interpretazione del codice sorgente e verifica dei criteri di accettazione, vengano elaborati dei *report<sub>G</sub>* che permettano di fornire feedback al *sistema<sub>G</sub>*.

**Tecnologie suggerite:**

- **Amazon Web Services** (*AWS Fargate*) : Servizio serverless per gestione a *container<sub>G</sub>*.
- **MongoDB**: Database documentale per lo *storage<sub>G</sub>* delle epic ed user stories e dei *report<sub>G</sub>*.
- **NodeJS**: ideale per lo sviluppo di *API<sub>G</sub>* Restful JSON a supporto dell'applicativo.
- **Python**: Linguaggio di programmazione per lo sviluppo del plugin per Xcode.
- **Typescript**: Linguaggio di programmazione per lo sviluppo del plugin per Visual Studio Code.

**Considerazioni**

- Opportunità di confrontare *ChatGPT* e *Amazon BedRock* e di sviluppare un'applicazione in grado di combinare i punti di forza di entrambi.
- Interessante l'idea di sfruttare la potenza dell'intelligenza artificiale per automatizzare alcuni *processi<sub>G</sub>* di sviluppo *software<sub>G</sub>*.
- Opportunità di studio di tecnologie ampiamente adottate, come *Node.js*, *Amazon Web Services* e *MongoDB*.

## 2.7 Capitolato C8 - JMAP: il nuovo protocollo per la posta elettronica

**Proponente:** Zextras

**Obiettivo:** Valutazione di fattibilità e prestazioni del protocollo di posta elettronica JMAP all'interno dell'ecosistema Carbonio.

È richiesto lo sviluppo di una *demo<sub>G</sub>* testabile per poter effettuare una valutazione approfondita delle prestazioni, della manutenibilità e della completezza del protocollo *JMAP* rispetto ai protocolli attualmente in uso nella suite Carbonio, mantenendo gli *standard<sub>G</sub>* precedenti per garantire la compatibilità con i client esistenti.

Il fine è quello di verificare se sia vantaggioso espandere le funzionalità offerte ai client di nuova generazione offrendo uno *standard<sub>G</sub>* di implementazione alla versione Open Source di Carbonio.

### **Tecnologie suggerite:**

- **Java:** Linguaggio principale dello *stack tecnologico<sub>G</sub>* di Carbonio.
- **iNPUTmice/jmap:** Per l'implementazione del protocollo JMAP.
- **Docker:** Piattaforma progettata per aiutare gli sviluppatori a creare, condividere ed eseguire applicazioni in *container<sub>G</sub>*.

### **Considerazioni**

- Notevole interesse nell'apprendimento di *Docker* e nell'approfondimento delle conoscenze legate a *Java*.
- Richiede un'analisi approfondita della documentazione della suite Carbonio e dei protocolli IMAP e JMAP per ottenere una visione più completa e concreta del progetto.
- Nonostante la finalità del progetto risulti particolarmente utile e di grande rilevanza, il progetto non ha riscosso un interesse significativo tra i vari membri del team.

## 2.8 Capitolato C9 - ChatSQL: creare frasi SQL da linguaggio naturale

**Proponente:** Zucchetti

**Obiettivo:** Sviluppare un'applicazione che traduca in linguaggio SQL delle query formulate con il linguaggio naturale.

Per ottenere ciò è necessaria la realizzazione di un generatore di prompt che, a partire da una richiesta in linguaggio naturale e un dizionario dati (descrizione della struttura del *database<sub>G</sub>* in cui verranno eseguite le query), produca in output un prompt da sottoporre ad un LLM quale ChatGPT, Palm, LLaMa o altri più specializzati.

La modalità d'uso prevede che un operatore al corrente della struttura del *database<sub>G</sub>* carichi il dizionario dati così da permettere a chi vuole ottenere la query SQL di proporre la sua frase in linguaggio naturale e il generatore creerà il prompt da sottoporre al *sistema<sub>G</sub>* di AI.

**Tecnologie suggerite:** L'azienda ha lasciato libera scelta per quanto riguarda le tecnologie, tuttavia ha fornito numerosi siti dove reperire informazioni ed esempi come:

- **HuggingFace:** La più grande raccolta di modelli LLM.
- **Catalyzex:** Raccolta di paper scientifici sull'uso degli LLM per produrre frasi SQL.

### Considerazioni

- Obiettivo del progetto molto stimolante e con importanti risvolti pratici.
- Interessante l'approfondimento su LLM e prompt engineering.
- Possibilità di ampliare il progetto e di renderlo più completo grazie ai numerosi requisiti opzionali

### 3 Conclusioni

Dopo un'attenta valutazione di tutti i capitolati proposti, ogni membro del gruppo ha espresso le proprie preferenze ed è emerso che il capitolato C6 è quello che ha suscitato un interesse unanime.

Con la scelta di questo capitolato si vuole ampliare il proprio bagaglio di conoscenze nell'ambito *Big Data* e *IoT*, usando tecnologie moderne e di grande interesse per tutto il gruppo.

L'implementazione di una città monitorata da sensori rappresenta un approccio promettente nell'ottica di ottimizzare l'efficienza e la qualità della vita urbana. Tale *sistema<sub>G</sub>* può consentire una raccolta continua di dati e informazioni cruciali, fornendo una base solida per l'ottimizzazione dei servizi pubblici, la gestione del traffico, la sicurezza e la sostenibilità ambientale.

Guidati dalla profonda attrazione per l'argomento proposto, intendiamo sviluppare un prodotto professionale che sia all'altezza delle sfide del futuro e in grado di soddisfare le sue crescenti esigenze.