|  |  |
| --- | --- |
| **PROJECT WORK**  CdS Informatica per le aziende digitali (L-31)  ***Settori Scientifici Disciplinari*** Informatica (INF/01), Sistemi di elaborazione delle informazioni  (ING-INF/05), Impianti meccanici industriali(ING-IND/17) | |
| **Cognome e Nome:** | **Andrea D’Orazio** |
| **Numero di Matricola**: | **0312300107** |
| **Corso di Studio:** | Barrare la casella riferita al proprio corso di studio |
| ◊   L-5 Filosofia ed Etica |
| ◊   L-22 Scienze Motorie |
| * L-31 Informatica per le Aziende Digitali |
|  |
| **Tema n:** | 1 |
| **Titolo del tema:** | **La digitalizzazione dell’impresa** |
| **Traccia del PW n:** | **1.1** |
| **Titolo della traccia:** | **Lotto Economico di Ordinazione (EOQ) per materiali a domanda**  **indipendente** |
| **Titolo dell’elaborato:** | Attribuire un titolo al proprio elaborato progettuale |
| **PARTE PRIMA – DESCRIZIONE DEL PROCESSO** | |
| **Utilizzo delle conoscenze e abilità derivate dal percorso di studio:** | Descrivere quali conoscenze e abilità apprese durante il percorso di studio sono state utilizzate per la redazione dell’elaborato, facendo eventualmente riferimento agli insegnamenti che hanno contribuito a maturarle  Collegare alle richieste alle nozioni apprese, non fare elenco, argomentare motivando gli argomenti, abbastanza generico ma discorsivo.  Per la realizzazione del progetto, le competenze e i concetti acquisiti nei corsi di Programmazione e Tecnologie Web, insieme al corso di Programmazione Distribuita, sono stati di fondamentale importanza. Questi corsi hanno fornito le competenze necessarie per la progettazione e l'implementazione del progetto. L'applicativo web sviluppato offre agli utenti uno strumento intuitivo per il calcolo del Lotto Economico di Ordinazione (EOQ), la sua realizzazione è stata possibile con l’utilizzo delle tecniche di programmazione e delle librerie più diffuse, apprese durante il corso di Tecnologie Web.  Per la parte teorica, sono stati utili gli argomenti di pianificazione e gestione delle scorte affrontati nel corso di 'Organizzazione aziendale e marketing', tuttavia, poiché l'argomento del calcolo del Lotto Economico di Ordinazione non era esplicitamente trattato, sono stati necessari ulteriori approfondimenti per acquisire le competenze necessarie [1][3][4]. Le nozioni apprese nel corso di Statistica sono state utilizzate invece per la realizzazione di un semplice esempio di previsione di serie storiche, integrato nell’applicativo. |
| **Fasi di lavoro e relativi tempi di implementazione per la predisposizione dell’elaborato:** | Descrivere le attività svolte in corrispondenza di ciascuna fase di redazione dell'elaborato. Indicare il tempo dedicato alla realizzazione di ciascuna fase, le difficoltà incontrate e come sono state superate.  Tabella di marcia, individuare macrofasi e fasi e il tempo dedicato  Ricerca informazioni, raccolta dati, fase di strutturazione, come sviluppare l’elaborato..poi fasi di implementazione… schematico ma argomentando  La realizzazione del project work è stata possibile pianificando una serie di fasi con relative tempistiche più o meno rispettate durante lo svolgimento dell’elaborato.  Di seguito le fasi pianificate dopo la scelta del tema e relativa traccia :   * analisi preliminare della scheda del project work; * approfondimento e studio del tema inerente al modello economico di acquisto (EOQ - Economic Order Quantity) con relativa selezione delle fonti; * approfondimento e utilizzo delle nozioni apprese dagli insegnamenti del Corso di Laurea necessari alla realizzazione del project work ; * selezione e scelta del tipo di architettura per la realizzazione dell’applicativo; * progettazione dell’algoritmo di calcolo (input, output e passi di elaborazione); * realizzazione dei diagrammi di caso d’uso per la realizzazione del software; * scelta degli ambienti di sviluppo e dei linguaggi di programmazione più idonei allo sviluppo dell’applicativo; * implementazione dell’algoritmo di calcolo e della logica di business; * realizzazione dell’interfaccia utente per il calcolo del Lotto Economico di Ordinazione.   Sintetizzando ho individuato quindi tre macrofasi con i rispettivi tempi di esecuzione:   * Fase 1, analisi, studio e approfondimento, selezione del materiale e delle fonti – circa tre settimane. * Fase 2, scelta dell’architettura software, dei framework e dei linguaggi di programmazione, progettazione dell’algoritmo di elaborazione e del software – due settimane. Un aspetto significativo di questa fase è stato il corretto dimensionamento del progetto, bilanciando la risposta alla traccia del project work con l'implementazione di aspetti aggiuntivi.   Questa fase si è parzialmente sovrapposta con la seguente (fase 3) in quanto è stato necessario valutare la fattibilità di alcune scelte operative, come ad esempio l’aggiunta di strumenti previsionali.   * Fase 3 realizzazione dell’applicativo (algoritmo, logica di business e interfaccia utente) – tre settimane. In questa fase è stato sviluppato tutto il codice per la realizzazione del software |
|  |
|  |
|  |
|  |
| **Risorse e strumenti impiegati:** | Descrivere quali risorse (bibliografia, banche dati, ecc.) e strumenti (software, modelli teorici, ecc.) sono stati individuati ed utilizzati per la redazione dell’elaborato. Si descrivano inoltre  - i motivi che hanno orientato la scelta delle risorse e degli strumenti;  -la modalità di individuazione e reperimento delle risorse e degli strumenti;  indicare difficoltà e motivi di scelta, come reperire i dati e come è stato risolto  **Risorse utilizzate per l’approfondimento teorico.**  L’individuazione delle risorse necessarie alla realizzazione del project work è iniziata dalla ricerca del primo articolo riguardante il calcolo del Lotto Economico di Acquisto [1] e una sua analisi postuma [2], in seguito sono stati selezionati testi utilizzati come riferimento per l’analisi e la gestione delle scorte [3], [4].  Per contestualizzare il tema da un punto di vista attuale sono stati inoltre utilizzati due articoli [5][7] per sottolineare i limiti del modello classico del Lotto Economico di Ordinazione.  **Risorse utilizzate per la realizzazione del software**  Il software richiesto dalla traccia di questo project work è stato realizzato  utilizzando un’architettura **client-server** di tipo *web application* che ha previsto l’utilizzo del linguaggio HyperText Markup Language (HTML)[8] e il linguaggio Javascript [9] per l’implementazione dell’interfaccia utente. Per un layout funzionale è stato selezionato il framework Bootstrap [10] che prevede sia funzioni Javascript che fogli di stile (*Cascade Style Sheet – CSS*) [11]. Per la parte server è stato scelta la piattaforma *open-source* *NodeJS* [12] che consente l’esecuzione a *runtime* di codice Javascript.  La disponibilità, inoltre, di numerosi moduli – *Node Package Manager – NPM* da possibilità di creare agevolmente il proprio web-server per la realizzazione di applicazioni facilmente scalabili e performanti. Il modulo Express[13] è stato concepito proprio per l’implementazione di applicazioni web minimali e flessibili, facilmente integrabili con eventuali sistemi aziendali già in uso.  Il web server utilizza uno stile architetturale di tipo *REST (Representational state tranfer)*[14]molto utilizzato in sistemi distribuiti, che sfrutta il protocollo HTTP (HyperText Transfer Protocol) per lo scambio di risorse, ad esempio dati strutturati nel formato standard de-facto JSON [15].  Per estendere le funzionalità dell’applicazione è stato utilizzato anche il linguaggio *Python*[16] per implementare l’algoritmo di calcolo, linguaggio versatile e ad oggi il più utilizzato [17]. La scelta di utilizzare Python è fondata su due motivi, il primo è che è dotato di una molteplicità di *packages* che ne estendono i campi di applicazione, il secondo è che può essere richiamato direttamente da NodeJS grazie al modulo *child\_process* che permette di istanziare sottoprocessi e l’interazione diretta con i risultati delle elaborazioni.  Tra i vari packages ho individuato StatsForecast[18] che fornisce diversi modelli di previsionem inclusi alcuni modelli automatici come AutoARIMA[22], utilizzato in questo progetto come esempio di possibile previsione per una delle componenti di calcolo , la domanda.  **Altri strumenti utilizzati**  Sono stati necessari durante la stesura del project work, di ulteriori strumenti come draw.io[19] per la realizzazione di grafici e diagrammi, il portale W3School[20], come risorsa per le tecnologie web implementate nel lavoro e GitHub[21] come repository per la gestione e diffusione del codice sviluppato. Come strumento editor per la scrittura del codice ho utilizzato il prodotto Microsoft, Visual Studio Code[22].  La ricerca di strumenti e risorse ha impiegato principalmente la fase iniziale dell’elaborato, ed è stata condotta sia con ricerche online, sia nella consultazione di testi. Il reperimento delle informazioni non è stato complicato ma alcune difficoltà sono emerse nell’armonizzare le varie fonti per produrre una sintesi esaustiva per la sezione teorica del progetto, legata al calcolo dell’EOQ.  Per la parte di sviluppo dell’applicativo web invece il materiale individuato è stato esclusivamente online, ogni framework o linguaggio utilizzato infatti è sempre dotato di una ricca documentazione corredata di esempi esaustivi. Anche le librerie Python per la parte di calcolo statistico sono ampiamente documentate online nelle pagine web a supporto dei relativi progetti.  **Bibliografia:**  [1] Harris, F.W. ,1913, *How many parts to make at once*. Factory, The Magazine of Management, Volume 10, N° 2.  [2] Erlenkotter D. ,2014, *Ford Whitman Harris’s Economical Lot Size Model*. International Journal of Production Economics.  [3] G. Urgelletti Tinarelli ,1981, *La gestione delle scorte*, Etas Libri.  [4] Brandolesi A., Pozzetti A., Sianesi A. ,1991, *Gestione della produzione industriale*. Hoepli.  [5] *50 Years of EOQ: A Critical Review,* Journal of the Operational Research Society  [6] *Data Science e Machine Learning*, Michele di Nuzzo  **Sitografia:**  [7] *La crisi da Covid-19 - Dalla crisi sanitaria alla crisi economica* <https://www.consob.it/web/investor-education/crisi-sanitaria-economica>. |
| [8] WHATWG, “*HTML”, 2024.* [Online]. Disponibile: <https://html.spec.whatwg.org/multipage/> [Accesso: 10-dic-2024].  [9] Mozilla Developer Network, *JavaScript, 2024.* [Online]. Disponibile: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> [Accesso: 10-dic-2024].  [10] Bootstrap, *Documentazione di Bootstrap*. [Online]. Disponibile: <https://getbootstrap.com>. [Accesso: 10-dic-2024]  [11] Mozilla Developer Network (MDN*), CSS: Cascading Style Sheets*. [Online]. Disponibile: <https://developer.mozilla.org/it/docs/Web/CSS>. [Accesso: 10-dic-2024].  [12] Node.js Foundation, Node.js. [Online]. Disponibile: <https://nodejs.org>. [Accesso: 10-dic-2024].  [13] Express.js Foundation, *Express - Fast, unopinionated, minimalist web framework for Node.js*. [Online]. Disponibile: <https://expressjs.com>. [Accesso: 10-dic-2024].  [14] Restfulapi.net, *Guida alle API RESTful*. [Online]. Disponibile: <https://restfulapi.net>. [Accesso: 10-dic-2024].  [15] ECMA International, JSON. [Online].Disponibile: . <https://www.json.org/json-en.html> [Accesso: 10-dic-2024].  [16] Python Software Foundation, *Python*. [Online]. Disponibile: <https://www.python.org>. [Accesso: 10-dic-2024].  [17] Tiobe, Index for December 2024.[Online]. Disponibile:  <https://www.tiobe.com/tiobe-index/>. [Accesso: 10-dic-2024].  [18] Nixtla. *Statsforecast: A fast and scalable forecasting library*. [Online]. Disponibile: <https://github.com/Nixtla/statsforecast>. [Accesso: 10-dic-2024].  **Strumenti**  [19] JGraph Ltd. *draw.io: Diagramming tool*. [Online]. Disponibile: https://app.diagrams.net. [Accesso: 10-dic-2024].  [20] W3Schools. *W3Schools Online Web Tutorials*. [Online]. Disponibile: <https://www.w3schools.com>. [Accesso: 10-dic-2024].  [21] GitHub, Inc. *GitHub*. [Online]. Disponibile: <https://github.com>. [Accesso: 10-dic-2024].  [22] Microsoft Visual Studio Code. Disponibile: <https://code.visualstudio.com/>  [Accesso: 10-dic-2024] |
| **PARTE SECONDA – PREDISPOSIZIONE DELL’ELABORATO** | |
| **Obiettivi dell’elaborato/progetto/artefatto:** | Specificare gli obiettivi raggiunti dall’elaborato, indicando in che modo esso risponde a quanto richiesto dalla traccia del Project Work.  Collegare l’output prodotto con le richieste, sviluppato codice in grado di …ed effettivamente risponde alla richiesta…ho ottenuto i risultati  La realizzazione dell’elaborato è stata approcciata per proporre uno strumento, benché si tratti di un prototipo, semplice e pratico per il calcolo del Lotto economico di ordinazione (EOQ), pensato per una reale integrazione nel contesto aziendale. La scelta di sviluppare una *web application* è stata motivata dall'esigenza di garantire semplicità, performance e scalabilità del sistema.  L'architettura software, basata sul framework Node.js e l’integrazione di Python, ha permesso di realizzare un applicativo con struttura modulare che unita all'utilizzo di tecnologie web standard lo rendono facilmente integrabile con eventuali software aziendali. L'interfaccia utente, sviluppata con HTML e librerie JavaScript, ne semplifica l’utilizzo.  Per rendere l’utilizzo ancora più realistico è stata inserita la possibilità di lavorare quantità di dati maggiori, consentendo il caricamento di file strutturati, permettendo così elaborazioni in serie storica per l’analisi dell’andamento sia del valore dell’EOQ che delle componenti di costo.  L’implementazione di funzionalità di analisi predittiva della domanda, estendono le potenzialità dello strumento oltre il semplice calcolo del lotto economico.  L’output che si ottiene utilizzando il software vuole offrire un quadro seppure sintetico, che illustri in modo chiaro l’andamento delle spese e degli ordini negli anni, avvalendosi anche di supporto grafico. |
| **Contestualizzazione:** | Elaborare una breve descrizione del contesto teorico-applicativo nell’ambito del quale è stato sviluppato l’elaborato.  Descrivere il contesto teorico applicativo parlare anche dell’output del project work, come può essere uilizzato realmene..questo srumento può essere utile per ….  L’obiettivo di efficienza è oggi più che mai fondamentale durante i processi di produzione e la corretta gestione del magazzino e delle scorte rappresentano un elemento di importanza vitale per la sopravvivenza di un’azienda. Il modello economico di ordinazione (EOQ - Economic Order Quantity) a domanda indipendente nasce come modello per la corretta gestione delle scorte, con l’obiettivo di definire la quantità ottimale di acquisto necessaria per minimizzare il costo totale di acquisto ottenuto tenendo in considerazione sia i costi di approvvigionamento sia i costi di mantenimento delle scorte in magazzino.  Ideato e proposto da Ford W.Harris nel 1913 nell’articolo “How many parts to make at once” [1], il modello è stato ampiamente utilizzato, discusso e rielaborato negli anni [2]. Fondamentali gli aggiornamenti di R.H. Wilson (1935) che lo hanno fatto divenire un valido strumento, ancora attuale, per il controllo e la gestione delle scorte.  La gestione delle scorte [3] può essere vista come la ricerca di una risposta bilanciata a due interrogativi:   * *quanto*? Il volume di prodotti da acquistare ad ogni ordine * *quando*? Con quale frequenza effettuare un ordine   La sfida è infatti nel rispondere correttamente a queste due domande, che costituiscono il punto sul quale fanno leva i costi delle scorte. Entrano in gioco molte variabili che influenzano drasticamente e in maniera significativa le scelte adottate su quantità e tempistica degli ordini di acquisto. Ad esempio, lotti di ordinazione di maggiori dimensioni diminuiscono i costi relativi all’emissione degli ordini, oltre l’opportunità di ottenere prezzi d’acquisto più vantaggiosi, d’altro canto però i costi di capitale saranno più alti e si aggiungeranno anche i costi di immagazzinamento, oltre a rischi legati ad obsolescenza e deperimento dei materiali.  Il modello del lotto economico di acquisto semplifica notevolmente le variabili in gioco, infatti, nella sua versione base così come concepita da Harris, prevede delle ipotesi iniziali abbastanza stringenti [3]:   * La domanda ***D*** totale annua viene considerata nota e costante; * I costi di gestione ***S*** degli ordini sono costanti; * Il tasso del costo di mantenimento ***I*** del singolo bene è costante; * Il costo di acquisto ***C*** per unità è costante e indipendente dalla quantità acquistata; * I tempi di consegna del lotto sono nulli; * Il lotto viene consegnato per intero; * Il tempo che intercorre tra due ordini (tempo di riordino) è definito e costante.   Secondo il modello, il dimensionamento degli ordini sarà influenzato da due tipologie di costi che costituiscono i costi totali [4].  Costi di *approvvigionamento* che rappresentano tutti i costi sostenuti per l’emissione di un ordine, vale a dire il momento in cui si decide di riordinare. Rappresentano la somma del costo di acquisto, costi amministrativi di emissione ordine, di trasporto, movimentazione interna al magazzino, verifica e controllo dell’ordine, dei materiali ricevuti. I costi di ordinazione dipendono dal numero di ordini emessi nel periodo, come mostrato in Figura 1 aumentando la quantità di prodotti acquistati ***Q*** per ogni ordine, visto che il numero di prodotti necessari è costante nel periodo (corrisponde alla domanda ***D***), il numero di ordini diminuisce con conseguente diminuzione del costo di approvvigionamento.  Definiamo quindi il Costo totale di approvvigionamento nel seguente modo:  ***Ca = S \* (D/Q)***  Dove (D/Q) rappresenta il numero di ordini nel periodo.    Figura 1 - Grafico andamento costo unitario di approvvigionamento - Fonte [3]  L’altra componente riguarda i costi di *immagazzinamento*, relativi alle spese per attrezzature, spazio di stoccaggio, assicurazioni per i locali, spese del personale di magazzino, eventuali danni e obsolescenza dei prodotti oltre ai costi riferiti al capitale immobilizzato nel tempo.  All’aumentare dei costi di immagazzinamento verrà privilegiato l’acquisto di lotti più piccoli con conseguente riduzione della giacenza in magazzino.  In Figura 2 è visibile l’andamento dei costi proporzionale alla quantità di prodotti in magazzino.    Figura 2 - Grafico andamento costi di immagazzinamento – Fonte [3]  I costi totali di immagazzinamento come:  ***Ci= I \* C \* Q/2***  Harris, definisce ***I*** come il tasso di interesse da calcolare sul prezzo unitario del prodotto come deprezzamento del valore, abbiamo quindi  ***I\*C*** che equivale al *costo unitario di mantenimento in magazzino* mentre ***Q/2*** rappresenta la *giacenza media* presente in magazzino, ottenuta ipotizzando il consumo completo dei prodotti a magazzino prima di un nuovo ordine.  Il costo totale è ottenuto sommando le componenti:  È necessario quindi un *trade-off* tra le componenti dei costi evidenziati, per ottenere il minimo è sufficiente derivare rispetto alla quantità *Q:*  L’ultima equazione corrisponde alla formula di Harris per il calcolo del quantitativo da ordinare.  Il grafico in Figura 3 mostra l’andamento della funzione dei costi totali (al netto dei costi di acquisto) rispetto ai costi di immagazzinamento e approvvigionamento. L’incrocio tra la retta e la curva dei due costi indica sull’asse delle ascisse la dimensione del Lotto Economico di Ordinazione, che corrisponde al livello che minimizza i costi totali (derivata prima uguale a zero).    Figura 3 - Grafico andamento costi totali – Fonte [1]  In figura 4 è visibile l’andamento della giacenza in magazzino nel tempo. Individuiamo il *livello di riordino* che indica la quantità minima in giacenza per cui si effettua un nuovo ordine di materiale e la distanza tra due punti di riordino rappresenta il *tempo di approvvigionamento*. Nel caso di esaurimento completo dei prodotti il livello di riordino è uguale a zero. La quantità *Q* di prodotti per ogni nuovo ordine corrisponde al Lotto Economico di Ordinazione - EOQ.    Figura 4 - Grafico punto di riordino  Il modello fin qui presentato nella sua formulazione originaria è, seppur efficace, estremamente semplicistico e difficilmente aderente alla realtà. Dalla sua nascita sono stati formulati diversi modelli [5] che migliorano l’impianto iniziale, nei quali le altre variabili in gioco non sono considerate costanti. Esempi come l’EOQ con *sconti di quantità* dove il prezzo di acquisto del prodotto è funzione della quantità acquistata, il modello con *tempo di approvvigionamento variabile* nel quale il tempo di consegna non è costante, oppure il modello con *deterioramento,* dove i prodotti immagazzinati subiscono una perdita di valore e sono soggetti a un loro ciclo vita, rappresentano un’evoluzione del modello iniziale.  In questo lavoro oltre al calcolo base dell’EOQ viene proposta anche l’implementazione del modello con *domanda variabile*, condizione che, specie in questi ultimi anni, ha evidenziato come la domanda di un bene possa variare influenzata da condizioni sociali ed economiche, esogene a qualsiasi legge di mercato, come sperimentato durante l’ultima pandemia e le crisi internazionali ancora in atto.  L’utilizzo di tecniche di *forecasting* rappresenta un valido strumento per la previsione dell’andamento della domanda di beni richiesti *D*, così da poter ottimizzare la gestione delle scorte e l’emissione di nuovi ordini.  Le previsioni si basano sull’analisi delle serie temporali [6], che rappresentano sequenze di osservazioni in un periodo ovvero la sequenza di valori che una dimensione, nel nostro caso la domanda, assume nel tempo.  Tra i modelli più semplici per prevedere i valori di una serie temporale è possibile utilizzare i *modelli autoregressivi AR*. Questi modelli suppongono che il valore *y* di una serie al tempo *t* sia composto dalla somma di due componenti: una che dipende dai valori precedenti ottenuto applicando una *regressione lineare* e l’altra da un valore aggiuntivo detto *rumore bianco* (white noise)  il valore di *y* al tempo *t* è dato dalla somma di , che rappresenta la regressione lineare applicata al periodo precedente più che rappresenta il valore del rumore.  Il modello ARIMA (*Autoregressivo a media mobile integrato*) [6] costruito dalla combinazione di modelli autoregressivi AR e il *modello a media mobile* MA, è quello che ho selezionato per un esercizio di applicazione.  Il modello ARIMA è indicato per serie temporali stazionarie che presentano media costante e varianza costante, condizioni non aderenti alla domanda di un bene nel mondo reale, ma apportano una semplificazione accettabile se rapportate alle condizioni inziali del calcolo del Lotto Economico di Ordinazione, dove la domanda è considerata costante. |
| **Descrizione dei principali aspetti progettuali:** | Sviluppare l’elaborato richiesto dalla traccia prescelta.  Cuore dell’elaborato, i vari output e sviluppo elaboratp, non necessario inserire tutto il codice,ma inserire il link del progetto github in github realizzare il readme per aiutare nel test.  Per la realizzazione dell’applicativo si è ipotizzato di implementare il prototipo di un piccolo sistema, che risponda alle richieste del progetto di calcolo del Lotto Economico di Ordinazione (EOQ), una piattaforma modulare che offra scalabilità ed estensibilità, in termini di funzionalità, e che sia in grado di adattarsi alle esigenze in evoluzione di un’ipotetica azienda che necessita di dotarsi di un’applicazione in grado di interfacciarsi con eventuali sistemi preesistenti.  L’applicazione deve consentire le seguenti operazioni di base:   * Inserimento dati da parte dell’utente; * Calcolo del Lotto Economico di Ordinazione e dei costi totali per diversi anni; * Offrire un’interfaccia *user-friendly* e intuitiva per l’inserimento dei dati di input e la lettura dell’output restituito.   L’utilizzo dell’applicazione è rappresentato dal diagramma UML del caso d’uso in figura 5.    Figura 5 - Caso d'uso calcolo EOQ  L’applicazione si presenta con una pagina iniziale con le informazioni base per l’utilizzo con un menù orizzontale (Figura 5)    Figure 6 - Home page dell'applicazione  Selezionando dal menù la voce ***EOQ*** l’utente ha due possibilità di inserimento dati (Figura 7), manuale compilando un modulo (*form – Figura 8*) o effettuando il caricamento di un file esterno, di tipo *Comma Separated Value*, per agevolare l’inserimento in caso di dati riguardanti più anni.    Figure 7 - Pagina principale per l'inserimento dei dati e visualizzazione    Figure 8 - Form inserimento dati modalità manuale  In entrambe le modalità di inserimento dati informazioni da fornire sono le seguenti:   * Periodo, di tipo numerico che rappresenta il periodo di riferimento dei dati, generalmente l’anno; * Costo unitario (C), costo unitario di acquisto del prodotto; * Costo di gestione dell’ordine (S), è il costo di emissione di un ordine, considerato per definizione costante nel periodo; * Costo mantenimento (H), è il costo unitario per il mantenimento di un prodotto nel periodo; * Domanda (D), è il valore della domanda di prodotto espressa come media del periodo.   I dati inseriti vengono mostrati nella tabella in Figura 9    Figure 9 - Tabella elaborazione e gestione dati  La tabella è predisposta per lavorare con dati in serie storica e sono presenti le colonne per contenere i seguenti dati di output relativi ai parametri di ogni riga:   * EOQ, valore del lotto economico di ordinazione; * Numero lotti, numero dei lotti da ordinare nel periodo * Costo totale, costo totale di acquisto calcolato sommando i costi di approvvigionamento, mantenimento e acquisto secondo la formula   In testata di tabella sono presenti le seguenti funzioni:   * Carica dati, modalità inserimento dati da file; * Aggiungi dati, modalità inserimento dati manuale; * Svuota tabella, per ripulire la tabella e inserire nuovi dati; * Selettore di tipo *switch*, per abilitare il calcolo dell’EOQ per un periodo ulteriore con stima del valore della domanda; * Calcola EOQ, avvio della procedura di calcolo del lotto economico di ordinazione per tutti i periodi presenti in tabella.   L’output finale è stato integrato anche con il dettaglio della composizione del costo totale e i grafici a linea per evidenziare l’andamento in serie storica del lotto di ordinazione e della domanda annua.    Figure 10 - Output di elaborazione  \*\*\*\*dettagli di implementazione, vantaggi app-web, struttura e poi codice  L’idea di base per la realizzazione del software è stata quella di implementare un’applicazione web,  L’applicazione è composta da tre componenti principali:   * livello di presentazione (Presentation Layer), relativo al client, quindi alla parte front-end, corrisponde al browser utilizzato dall’utente per collegarsi al sistema e include tutte le componenti responsabili dell’interfaccia utente e le interazioni con l’utente finale; * livello logica di business (Business Logic), corrisponde all’applicazione in back-end che ha il compito di gestire le richieste del client, veicolare gli input per restituire gli output; * livello di elaborazione dati (Data Processing Layer), corrisponde al modulo specifico che si occupa dell’elaborazione dei dati, potenzialmente con tecniche evolute, scalabili e indipendenti dal layer che le richiama.   I vantaggi tangibili di un’applicazione di questo tipo sono principalmente, la concorrenza, ovvero il server web è in grado di gestire agevolmente richieste multiple e concorrenti, l’indipendenza dei livelli che consente la scalabilità orizzontale (*scale out*) del sistema con la possibilità di aumentarne la dimensione aggiungendo più computer (*host*). Altro elemento a vantaggio di architetture di questo tipo è l’affidabilità, è infatti agevole isolare eventuali problemi e il ripristino dei componenti avviene senza eccessive interruzioni dell’intero sistema.  Le applicazioni web per natura sono fruibili via rete ed utilizzano linguaggi e protocolli di comunicazioni tipici della rete Internet, garantendo una facilità d’implementazione e gestione, grazie ai numerosi linguaggi e framework disponibili.  In quest’ottica il software realizzato per questo progetto, seppure semplice, vuole essere una base idealmente ampliabile con funzioni di calcolo via via più complesse con la possibilità di ospitare i livelli su più host, migliorando l’efficienza e permettendo una gestione ottimale delle risorse disponibili con possibilità di distribuzione geografica, offrendo quindi un ideale strumento per aziende di diverse dimensioni e strutture.  Il livello di presentazione è stato implementato, trattandosi di un’applicazione web, con il linguaggio HTML, l’utilizzo di fogli di stile CSS e codice Javascript per gestirne la parte dinamica. La tipologia è di tipo Single Page Application (SPA), quindi una pagina singola che cambia contenuto dinamicamente in risposta alle interazioni dell’utente.  Per realizzare l’interfaccia con un aspetto pratico e funzionale ho utilizzato la libreria Bootstrap [10], composta sia da codice CSS che Javascript, che offre numerosi strumenti, stili e componenti. In particolare, ho utilizzato il *Grid system* che realizza layout di tipo *responsive* che si adattano automaticamente alla dimensione dello schermo (desktop, laptop o smartphone) che sta utilizzando l’utente.  La pagina principale (file index.html) è composta da sezioni definite con tag <div> con un container *container-fluid*  di Bootstrap definisce un contenitore che si adatta alla larghezza del dispositivo e racchiude il navigatore, il contenuto principale dinamico e un footer.    FigBBBBBBBBBB\*\*\*\*\*\*ura 5 – Definizione container e menu navigazione  La sezione con id *main* contiene la sezione principale aggiornata dinamicamente    Figura 6 - Definizione contenuto principale    Figura 7 - Definizione footer    **app.js** |
| **Campi di applicazione:** | Descrivere gli ambiti di applicazione dell’elaborato progettuale e i vantaggi derivanti della sua applicazione.  Possibili campi di applicazione e vantaggi dal particolare al generale anche in altri ambiti  Il progetto è stato sviluppato con l’obiettivo di realizzare un’applicazione che possa essere adottata in realtà aziendali medio piccole, che necessitino di uno strumento per l’analisi dei costi di gestione dalle scorte di magazzino e la corretta valutazione dell’acquisto di prodotti. La tecnologia utilizzata per l’implementazione dell’applicativo, è stata selezionata per consentirne l’esecuzione sia su dispositivi in locale (personal computer) che in remoto (server).  Questo grazie alla flessibilità della configurazione e modularità del codice, ad esempio--- |
| **Valutazione dei risultati (potenzialità e criticità):** | Descrivere le criticità e i limiti ai quali i risultati dell’elaborato sono potenzialmente esposti.  Limiti e criticità e sviluppi futuri |