

PROJECT WORK

Membri del gruppo: Giulio Angelo Cusinato, Alessandro Monteleone , Lorenzo Segalla, Samuele Antonio Barbiera

SPIEGAZIONE PROGETTO

Il seguente progetto, prevede un'automazione che, grazie ad un'applicazione permette ad un PC situato in un ufficio di inviare dati di commesse specifiche ad un PLC situato in reparto produzione.

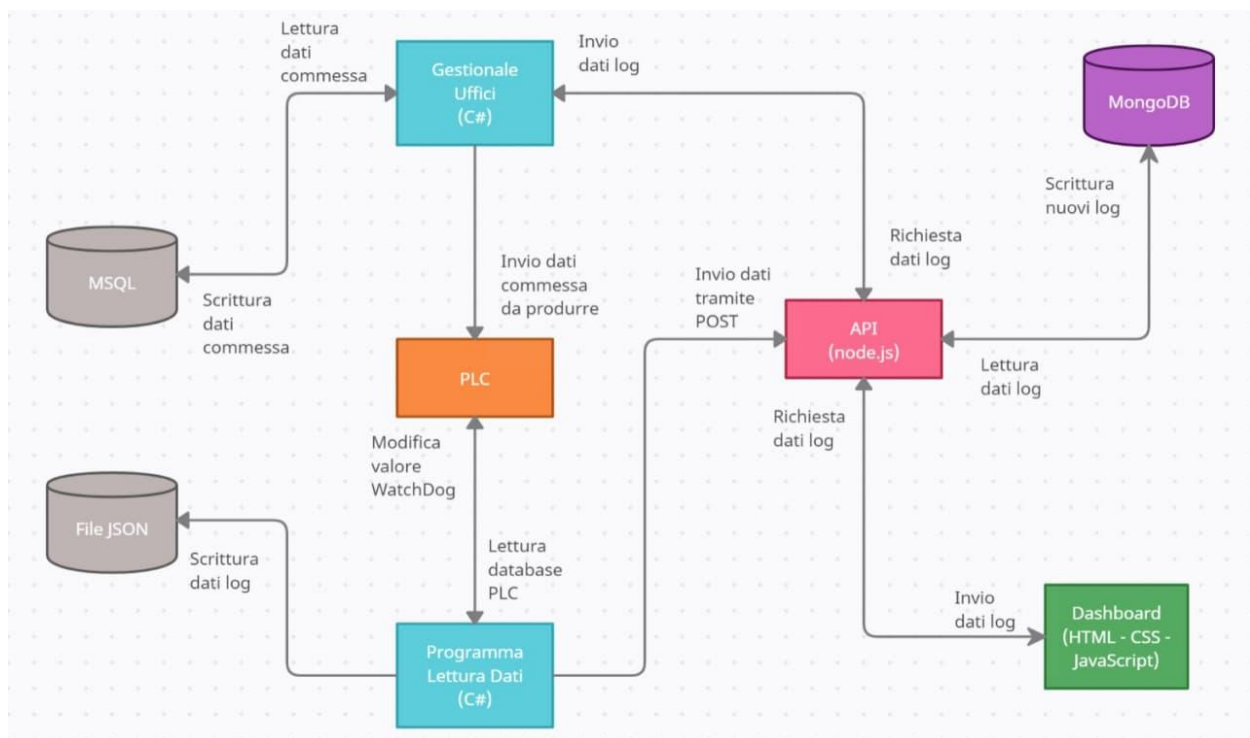
All'arrivo dei dati da due possibilità di avvio, la prima è un avvio virtuale, ovvero attraverso l'applicazione da ufficio si può avviare la produzione , l'altro invece è lo start fisico da pulsantiera collegata al PLC.

Una volta iniziata la produzione, essa si fermerà solo in condizione di emergenza della macchina, la quale mostrerà sul display HMI l'errore specifico, che verrà riportato poi sull'applicazione per ufficio e sulla dashboard web.

Man mano che la macchina stamperà le etichette, il valore dei pezzi prodotti aumenterà in tempo reale sia sul PLC che sul sito web , in modo da monitorare in modo efficiente tale produzione.

Sia dalla applicazione riservata agli uffici che dalla pagina web si potrà vedere lo storico delle commesse eseguite, con relativi codici, articoli, pezzi da produrre, data di inizio e data di consegna relative a tutte le commesse portate a termine.

Verrà inoltre controllato continuamente che la connessione tra PC, API , PLC e database , sia sempre attiva e, in caso contrario , l'applicazione console darà come risultato l'errore di collegamento , mostrando quale parte di sistema sarà disconnessa.



Schema logico funzionamento del progetto

Riassumendo avremo:

- una parte di controllo delle connessioni ad API, PC, PLC e DB tramite un'applicazione console che invierà anche i dati all'API che , in caso di mancato collegamento segnali un errore specifico;

- una Windows Form per l'immissione dei dati della commessa , la simulazione degli errori , il blocco, il rallentamento da ufficio e l'avvio forzato da ufficio .
- un programma PLC che si occupa della produzione della commessa attraverso una ciclica automatica ma che dia anche la possibilità di commandistica manuale da parte dell'operatore e che, in caso di errori, guasti o caduta della connessione tra ufficio e produzione , segnali l'errore che si presenta.;
- una pagina web , disponibile a degli utenti tramite login, attraverso il quale si può osservare la produzione in corso e lo storico di commesse portate a termine con i dati specifici relativi a tali commesse.

PLC

Programma attraverso TIA Portal v16 per PLC Siemens s7-1200

Blocchi di programma, di cui i principali, suddivisi per le seguenti specifiche:

-**ALLARMI**: blocco adibito alle condizioni di avvio del programma automatico e al controllo della pinza.

Solo in caso di chiave girata su "Automatico" , emergenza impostata ad "ok" e watch dog attivo o disabilitato , l'autoritenuta manterrà alto il valore dell'"EMERGENZA OK" che darà condizione di avvio al ciclo.

-**AUTOMATICO**: blocco attraverso il quale , il programma automatico può svolgersi.

Blocco che crea un ciclo che viene eseguito in circostanze specifiche , come l'assenza di allarmi , presenza di pezzi al completamento della commessa e la continua affluenza di materiale nel nastro adibito all'input di materiale per la creazione di fialette.

Avendo "EMERGENZA OK" e premendo il pulsante di START, il ciclo automatico inizierà , fermandosi soltanto in caso di guasti/allarmi o a commessa finita.

-**MANUALE**: blocco collegato direttamente con lo schermo HMI , attraverso il quale , l'operatore può avere un controllo libero e manuale del braccio antropomorfo.

(Tutte le azioni, i controlli e il conteggio pezzi vengono disabilitati all'inserimento del comando manuale nel PLC).

Impostazioni quindi , di comandi manuali e liberi , che avrà la precedenza sull'automatico , ma solo in condizione di chiave girata su "MANUALE".

-**PEZZI**: blocco esclusivamente utilizzato per le operazioni matematiche tra i pezzi (conteggio pezzi totali e azzeramento pezzi della commessa).

-**COMUNICAZIONE HMI**: in questo blocco vengono , attraverso specifiche, attivati bit per la visualizzazione su schermo HMI , relativi agli allarmi e allo stato della macchina.

-**WATCH DOG/ UFFICIO-OPERATORE**: controllo del watch dog (mantenimento del collegamento tra PLC e PC di ufficio produzione) per la continua trasmissione di informazioni tra ufficio e produzione relative alla commessa in esecuzione.

Altra funzione del blocco è quella del controllo del processo di produzione da lato ufficio, ovvero forzatura di start, blocco forzato per macchina in manutenzione o per altri motivi.

In caso di watch dog caduto , la macchina e il ciclo si fermeranno all'istante e la lampada d'allarme inizierà a lampeggiare velocemente per segnalare la caduta del collegamento , riprendendo SOLTANTO quando verrà stabilito o bypassato il collegamento.

-**USCITE**: blocco adibito al solo eccitamento delle uscite (elettrovalvole e lampade) tramite input del manuale e attivazione di Merker dall'automatico.

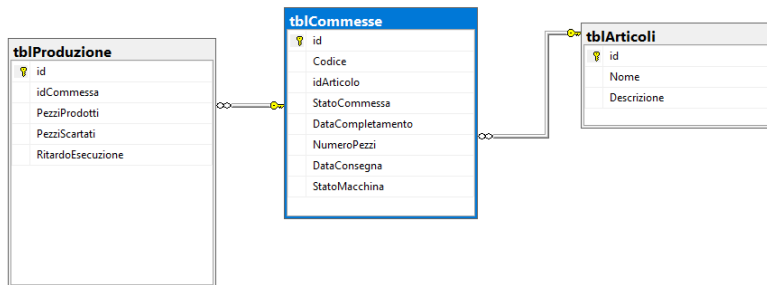
-HMI

Creazione di pagine per il controllo manuale , le specifiche della commessa (codice , articolo , pezzi da produrre e data di scadenza della commessa), pagina adibita alla sola comunicazione tra ufficio e reparto produzione, pagina per il monitoraggio di allarmi e stato macchina e infine pagina per il controllo della produzione (pezzi buoni , riutilizzabili , scarto, ecc).

DATABASE LOCALE SQL-SERVER

Il database è stato creato secondo il principio della normalizzazione.

Qui in figura si può visualizzare il diagramma ER del database con le varie relazioni e dipendenze



Schema logico tabelle del database Microsoft SQL Server

Il database locale sql è composto di 3 tabelle distinte:

-tblArticoli: Contiene il nome dell'articolo e una sua descrizione.

-tblCommesse: Contiene i dati relativi ad una commessa ed all'articolo associato.

-tblProduzione: Contiene i oltre ad un riferimento alla commessa anche le informazioni che vengono spesso modificate durante la produzione della commessa come il delay per la produzione del pezzo, pezzi prodotti attualmente e pezzi scartati.

Il database locale sql è provvisto anche di un'unica vista:

-viewTotale: la quale serve per rendere espliciti e chiari i vari riferimenti tra le tabelle ed a visualizzare tutti i dati di una commessa in produzione gestiti dal sistema.

PROGRAMMA PER LA SCRITTURA DEI DATI:

Programma Windows Form:

Form Main: questa form è composta da due tabelle che visualizzano le commesse inattive e la commessa attiva. Questa è la form principale la quale mi permette di navigare all'interno delle diverse form del programma.

Il bottone principale di questa form è il bottone di avvia la commessa attiva il quale mi permette di selezionare una commessa attiva in arresto macchina che richiamerà i metodi di Scrittura PC PLC

Il bottone modifica commessa attiva permetterà accedere alla form di modifica commessa

Form Mod: con questa form inviamo le modifiche alla commessa attualmente attiva

Queste modifiche possono essere inviate tramite tre bottoni:

-Con l'icona di salvataggio il programma invia il nuovo delay che il plc dovrà attendere per simulare una differenza di velocità

-Con l'icona dello stop il programma permetterà di simulare un guasto da ufficio

-Con l'icona del cestino il programma cancellerà il messaggio dell'operatore per ufficio dal database del PLC

Home

Dati Commessa Attiva

	Id Commessa	Codice Commessa	Stato Commessa	Stato Macchina
▶	25	ado21	Attiva	Macchina in STA...

Disattiva Commessa

Inserisci messaggio per operatore

Dati Commesse In Attesa

Aggiungi la commessa in coda

	Id Commessa	Codice Commessa	Stato	Nome Articolo
▶	21	ADO12	Inattiva	AziendaB
	22	ADo15	Inattiva	AziendaB
	23	Ado18	Inattiva	AziendaB
	24	Ado19	Inattiva	AziendaA

Avvia la Commessa Attiva

▶

Modifica la commessa Attiva

i

Messaggio dell'operatore

Aggiungi Nuova Commessa

Commesse Completate

Esempio di schermata form

Scrittura dati su PLC

Scrittura_PC_PLC: E' la parte del programma incaricata di avviare la connessione e di inserire i dati all'interno del plc e di avviare la macchina

Lettura e Sincronizzazione tramite API

Lettura Dati Plc: Il programma ad aggiornamento della form acquisirà i dati della commessa corrente dal API e sincronizzerà i dati del database nel cloud con il database locale

PROGRAMMA PER LA LETTURA DEI DATI:

E' un programma con interfaccia a console che si occupa di leggere i dati prodotti dal PLC, di controllare se il collegamento al PLC rimane stabile, di archiviare e inviare i log della macchina ad un server API.

All'accensione il programma tenta il collegamento al PLC.

Se il collegamento non riesce apparirà un messaggio di fallimento, il programma si bloccherà e attenderà la chiusura da parte dell'utente tramite il pulsante ESC o la X in alto a destra.

Se il collegamento riesce apparirà un messaggio di successo e il programma tramite timer interno comincerà a dialogare con il PLC.

"Gestione del WATCHDOG"

E' la sezione dedicata allo stato di connessione con il PLC.

Essa è gestita da una funzione chiamata da un timer ogni secondo.

La funzione modifica uno specifico dato nel PLC chiamato CONTROLLO: legge il dato, ne controlla il valore, lo modifica e lo scrive all'interno del PLC. Il PLC provvederà poi a cambiare il dato che una volta modificato verrà riletto dal programma. Il continuo cambio di valore di quel dato indica che la connessione

al PLC è stabile. Nel caso invece il dato non venga modificato o il un valore non cambi per un certo periodo di tempo la connessione sarà segnalata come fallita e il PLC smetterà di funzionare.

“Lettura del database”

E’ la sezione dedicata alla lettura dei dati del PLC, alla creazione dei log, al loro salvataggio su file e invio all’API.

E’ gestita da un timer con un tempo pari a 2 secondi.

La funzione legge dati dal database del PLC, li elabora traducendo i codici o valori particolari in stringhe di testo comprensibili all’uomo, le scrive all’interno di una stringa in formato JSON che viene salvata in un file e inviata tramite POST al server API.

E’ presente un controllo che segnala il mancato collegamento al server API e un controllo che segnala se la scrittura del file è avvenuta correttamente.

La schermata console viene aggiornata grazie ad un timer dedicato che la aggiorna ogni secondo.

```
##### Lettura del database #####
Connessione al Database PLC: SUCCESSO
Scrittura log sul Ado19.json
Scrittura locale SUCCESSO
Tentativo di invio dati in POST all'API: SUCCESSO
#####

##### Gestione del WATCHDOG #####
Lettura del WatchDog: SUCCESSO
Scrittura del WatchDog: SUCCESSO
#####
Press ESC to stop
```

Esempio schermata console

FULL-STACK

(APIs/BackEnd, FrontEnd)

BACKEND

l’api riceve delle chiamate dei dati con delle richieste HTTP dalla console di c# (collegato al plc) , fornendo un file json che contiene i risultati della produzione per ogni commessa , dopo aver ricevuto i dati l’api entra in gioco con dei middleware gestendo la creazione dei dati da inserire nel DataBase MongoDB, oltre alla funzione di popolazione del database, vengono aggiornati ogni secondo i dati che verranno mostrati sul Front End tramite un caricamento effettuato prima di ogni pagina in modo da ricevere i dati necessari prima di caricare il contenuto.

FRONTEND

la prima pagina di accesso è formata da un login , per rendere i dati dell'azienda/e sicuri e accessibili solo da esse, il tutto è formato tramite diversi framework che si appoggiano ad express per gestire le richieste di accesso tramite HTTP(handlebars come estensione html per il passaggio di variabili al server e l'uso di condizioni direttamente da html , bcrypt per crittografare la password dell'utente admin che non è neanche visibile sul database), se l'utente non è registrato non può accedere a nessuna pagina del sito , neanche tramite un url specifico ,venendo reindirizzato automaticamente alla pagina di login
Successivamente se l'utente ha eseguito l'accesso si trova davanti a tre pagine indicate sulla navbar:

HOME

avendo presente l'ultimo log commessa linkato alla sua pagine , e stessa cosa con il suo stato, la foto della fialetta prodotta, la foto per lo stato della macchina; un grafico variabile in base alla quantità di pezzi prodotti e una tabella della produzione

DB

presenta una tabella di tutti i logs aggiornati con la possibilità di filtrare il risultato di essa tramite un textbox di ricerca per il codice della commessa e un data set di inizio e uno di fine

LAST LOGS

presenta una tabella dell'ultimo log per ogni commessa, ordinati in base alla data di produzione e il codice della commessa, all'interno di essa ci sono due link href ai lati, collegati alla pagina dei dettagli della specifica commessa

CLOUD

Per lo sviluppo del servizio cloud ho proceduto per prima cosa con la creazione di un'istanza contenente un sistema operativo linux (tramite AWS), con IP ELASTICO fisso in modo da avere lo stesso indirizzo all'avvio della macchina.

Inserendo delle porte di entrata per le richieste tramite API (HTTP,HTTPS,SSH,TCP,MONGODB,PORTA DELL'INDEX.JS).

Cercando di implementare la versione più recente di tutti i servizi anche del S.O. stesso, per rendere il tutto più sicuro dall'esterno e performante anche se la guida era datata.

Installando un "framework" M.E.N. (MongoDB come database per contenere i dati ricevuti tramite richiesta post http dall'api, Express come framework che agisce su node , NodeJS per gestire i dati dell'api) tutto nella stessa istanza, cercando di attuare l'api presente nel codice del backend, il tutto appoggiato su software Nginx per ospitare la pagina, ho cercato di usare Nginx al posto di Apache per la sua migliore efficienza e per una gestione migliore delle richieste HTTP tramite nodejs, fondamentale per tutta la mole di dati inviati dal plc al server ogni secondo.

Caricando il progetto su github con una apposita repository ho effettuato un git pull di essa all'interno della cartella /var/www/ per allocare il progetto e appoggiarlo sul servizio di Nginx.

Per rendere la pagina sempre attiva in background occupando la porta 80 ho usato un servizio pm2 a differenza del node start che va a annullarsi se l'istanza viene riavviata o spenta.

ANALISI DEI COSTI

Il tempo impiegato per la realizzazione del progetto è pari a 70 ore durante le quali tutti i componenti del gruppo hanno partecipato attivamente per il completamento del progetto. Il costo orario per il lavoro di una persona è pari a 60€. Considerando solamente le ore di lavoro il costo totale del progetto è pari a 16800€.

Per quanto riguarda il costo dell'utilizzo della produzione si suddivide in 3 parti

- CLOUD AWS = all'incirca 0,50 € ogni 24 ore con la tariffa base da 1 core e 1gb di ram
- PLC = PLC Siemens (programmable logic controller) e il display HMI per visualizzare 500 euro
- Sql microsoft DB locale = 300 euro per la versione standard

Link a presentazione per altri dettagli:

https://docs.google.com/presentation/d/1eVktWLclhZWDMOUd_ozMRBgTJg4m2EWHDpYG7FjnUiE/edit?usp=sharing