

**TEMPLATE
PROJECT WORK**

Corso di Studio	INFORMATICA PER LE AZIENDE DIGITALI (L-31)
Dimensione dell'elaborato	Minimo 6.000 – Massimo 10.000 parole (<i>pari a circa Minimo 12 – Massimo 20 pagine</i>)
Formato del file da caricare in piattaforma	PDF
Nome e Cognome	Gabriele Ciccarese
Numero di matricola	0312200395
Tema n. (Indicare il numero del tema scelto):	Tema n. 1
Titolo del tema (Indicare il titolo del tema scelto):	La digitalizzazione dell'impresa
Traccia del PW n. (Indicare il numero della traccia scelta):	5
Titolo della traccia (Indicare il titolo della traccia scelta):	Sviluppo di un codice python per simulare un processo produttivo nel settore secondario
Titolo dell'elaborato (Attribuire un titolo al proprio elaborato progettuale):	Generazione di produzione prodotti, azienda del settore secondario, produzione di guarnizioni.

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):

Programmazione e Sviluppo Web

Linguaggio Python: Applicazione delle conoscenze di programmazione orientata agli oggetti e gestione delle strutture dati per implementare la logica di business dell'applicazione

Framework Flask: Utilizzo del framework web Python per creare un'applicazione web completa con routing, templating e gestione delle richieste HTTP

HTML/CSS: Competenze di markup e styling per creare un'interfaccia utente responsive e professionale

Gestione dei Dati e Logica di Business

Strutture Dati: Implementazione di dizionari nested per organizzare materiali e prodotti in modo efficiente

Algoritmi di Calcolo: Sviluppo di algoritmi per il calcolo dei tempi di produzione, capacità giornaliera e pianificazione basata su fattori dimensionali

Logica Matematica: Applicazione di calcoli matematici per determinare tempi di produzione, utilizzando funzioni come `math.ceil()` per arrotondamenti appropriati

Progettazione dell'Interfaccia Utente

User Experience: Creazione di un workflow intuitivo con selezione a cascata (materiale → prodotti → risultati)

Design Responsive: Utilizzo di Bootstrap per garantire compatibilità cross-device

Templating: Implementazione di template Jinja2 per la generazione dinamica di contenuti.

Gestione dei Processi Industriali

Pianificazione della Produzione: Modellazione dei processi produttivi con parametri realistici come quantità, tempi per unità, capacità giornaliera

Ottimizzazione: Implementazione di logiche per calcolare automaticamente i giorni richiesti e i tempi totali di produzione

Problem Solving: Analisi dei requisiti aziendali e traduzione in soluzioni software concrete

Pensiero Sistemico: Integrazione di diverse componenti (frontend, backend, logica di business) in un sistema coeso

Documentazione: Codice ben strutturato con naming conventions chiare e commenti appropriati.

Conoscenze Specifiche del Settore

Gestione Industriale: Comprensione dei processi produttivi e delle metriche di performance industriali

Modellazione dei Dati: Rappresentazione accurata di materiali industriali (Plexiglas, Gomma, Lana vetro) e relativi prodotti finiti

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):

Fasi di Lavoro e Relativi Tempi di Implementazione

Fase 1: Analisi dei Requisiti e Progettazione (Tempo: 4 ore)

Attività svolte:

Analisi delle esigenze aziendali di Athen S.P.A. per la gestione della produzione

Definizione dei requisiti funzionali: selezione materiali, calcolo tempi produzione, generazione report

Progettazione dell'architettura dell'applicazione (MVC pattern con Flask)

Definizione delle strutture dati per materiali e prodotti

Creazione dei mockup dell'interfaccia utente

Difficoltà incontrate:

Comprensione delle specifiche industriali per il calcolo dei tempi di produzione

Definizione della logica di business per i fattori dimensionali dei prodotti

Soluzioni adottate:

Ricerca approfondita sui processi produttivi industriali

Implementazione di algoritmi di calcolo basati su fattori scalabili

Fase 2: Setup dell'Ambiente di Sviluppo (Tempo: 2 ore)

Attività svolte:

Installazione e configurazione di Python e Flask

Setup della struttura del progetto con cartelle templates e static

Configurazione dell'ambiente virtuale per le dipendenze

Integrazione di Bootstrap per il frontend

Difficoltà incontrate:

Gestione delle dipendenze e versioni dei pacchetti

Configurazione corretta del templating engine Jinja2

Soluzioni adottate:

Utilizzo di pip per gestire le dipendenze in modo pulito

Consultazione della documentazione ufficiale di Flask

Fase 3: Implementazione del Backend (Tempo: 2 ore)

Attività svolte:

Creazione del modulo principale Flask con routing

Implementazione della struttura dati per materiali e prodotti

Sviluppo della funzione `generate_production_data()` per i calcoli

Gestione delle richieste POST per la selezione dinamica

Implementazione della logica di calcolo per tempi, capacità e giorni richiesti

Difficoltà incontrate:

Calcolo corretto della capacità giornaliera basata sui minuti disponibili (1440 min/giorno)

Gestione della logica di arrotondamento per i giorni richiesti

Implementazione del fattore dimensionale che influenza i tempi di produzione

Soluzioni adottate:

Utilizzo di `math.ceil()` per arrotondare correttamente i giorni

Implementazione di un sistema di fattori moltiplicativi per la dimensione

Test iterativi per validare la correttezza dei calcoli

Fase 4: Sviluppo del Frontend (Tempo: 3 ore)

Attività svolte:

Creazione del template HTML con Bootstrap

Implementazione della selezione a cascata materiale → prodotti

Sviluppo dell'interfaccia per la visualizzazione dei risultati

Styling personalizzato con CSS per l'identità aziendale

Implementazione di form dinamici con JavaScript

Difficoltà incontrate:

Gestione della selezione multipla con checkbox per i prodotti

Sincronizzazione tra selezione materiale e aggiornamento prodotti

Visualizzazione chiara e organizzata dei risultati di produzione

Soluzioni adottate:

Utilizzo di form POST con auto-submit per aggiornamento dinamico

Implementazione di card Bootstrap per una presentazione pulita dei dati

Utilizzo di input hidden per mantenere lo stato tra le richieste

Fase 5: Integrazione e Testing (Tempo: 2 ore)

Attività svolte:

Test dell'applicazione con diversi scenari di input

Verifica della correttezza dei calcoli di produzione

Test dell'interfaccia utente su diversi browser

Validazione della responsiveness mobile

Debug e correzione di errori minori

Difficoltà incontrate:

Gestione dei casi edge (selezione vuota, valori estremi)

Assicurarsi che i calcoli siano realistici per il contesto industriale

Compatibilità cross-browser per i form dinamici

Soluzioni adottate:

Implementazione di validazione lato server per input inconsistenti

Aggiunta di controlli condizionali nei template

Test sistematico su Chrome, Firefox e Safari

Fase 6: Ottimizzazione e Documentazione (Tempo: 2 ore)

Attività svolte:

Ottimizzazione del codice per performance e leggibilità

Aggiunta di commenti e documentazione inline

Creazione di README per l'installazione e l'uso

Preparazione per il deployment

Difficoltà incontrate:

Bilanciamento tra semplicità del codice e funzionalità complete

Creazione di documentazione chiara per utenti non tecnici

Soluzioni adottate:

Refactoring del codice per migliorare la manutenibilità

Creazione di esempi d'uso pratici nella documentazione

La fase più impegnativa è stata l'implementazione del backend, richiedendo particolare attenzione alla logica di calcolo industriale e alla gestione corretta dei dati di produzione.

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. Descrivere, inoltre, i motivi che hanno orientato la scelta delle risorse e degli strumenti, la modalità di individuazione e reperimento delle risorse e degli strumenti, le eventuali difficoltà affrontate nell'individuazione e nell'utilizzo di risorse e strumenti ed il modo in cui sono state superate):

Risorse e Strumenti Impiegati

Risorse Bibliografiche

Libri e Manuali Tecnici:

"Flask Web Development" di Miguel Grinberg - per l'apprendimento approfondito del framework Flask

"Python Crash Course" di Eric Matthes - per consolidare le basi della programmazione Python

"HTML and CSS: Design and Build Websites" di Jon Duckett - per le competenze di frontend

Documentazione ufficiale di Flask (<https://flask.palletsprojects.com/>) - riferimento principale per API e best practices

Articoli e Risorse Online:

Tutorial Mozilla Developer Network (MDN) per HTML5, CSS3 e JavaScript

Stack Overflow per la risoluzione di problemi specifici di implementazione

Bootstrap Documentation per componenti UI e grid system

Real Python tutorials per approfondimenti su Flask e Python

Motivi della Scelta:

La selezione bibliografica è stata orientata verso risorse pratiche e aggiornate, privilegiando la documentazione ufficiale per garantire l'accuratezza delle informazioni e tutorial step-by-step per accelerare l'apprendimento.

Banche Dati e Fonti Informative

Repository e Piattaforme:

GitHub per l'analisi di progetti Flask open source simili

PyPI (Python Package Index) per l'individuazione di librerie aggiuntive

Bootstrap CDN per componenti UI ready-to-use

Google Fonts per la tipografia dell'interfaccia

Dati di Riferimento Industriali:

Ricerche online su tempi di produzione standard nell'industria manifatturiera

Analisi di case study aziendali per la gestione della produzione

Modalità di Individuazione:

L'individuazione delle risorse è avvenuta attraverso ricerca mirata sui motori di ricerca specializzati, consultazione di forum tecnici e raccomandazioni da parte della comunità di sviluppatori.

Strumenti Software

Ambiente di Sviluppo:

Visual Studio Code - IDE principale per lo sviluppo

Motivo della scelta: Eccellente supporto per Python, HTML/CSS, debugging integrato e vasta libreria di estensioni.

Python 3.9+ - Linguaggio di programmazione principale.

Motivo della scelta: Sintassi pulita, vasta libreria standard, ottimo supporto per sviluppo web.

Git - Sistema di controllo versione

Framework e Librerie:

Flask 2.3.0 - Framework web Python

Motivo della scelta: Leggero, flessibile, ideale per applicazioni di media complessità

Bootstrap 5.3.0 - Framework CSS

Motivo della scelta: Componenti responsive pronti all'uso, compatibilità cross-browser.

Jinja2 - Template engine (integrato con Flask)

Motivo della scelta: Sintassi intuitiva, potenti funzionalità di templating

Tool di Sviluppo:

Flask Development Server - Server di sviluppo integrato

Chrome DevTools - Debug e testing dell'interfaccia utente

Postman - Test delle API e richieste HTTP (fase di sviluppo)

Modelli Teorici e Metodologie

Pattern Architetturali:

Model-View-Controller (MVC) - Separazione delle responsabilità

Applicazione: Separazione tra logica di business (Model), presentazione (View) e controllo (Controller).

RESTful Design Principles - Progettazione delle routes.

Applicazione: Utilizzo di HTTP methods appropriati (GET/POST) per diverse operazioni.

Metodologie di Sviluppo:

Iterative Development - Sviluppo incrementale

Applicazione: Implementazione per fasi successive con test continui.

Responsive Design - Progettazione adattiva

Applicazione: Utilizzo di griglia Bootstrap per compatibilità multi-device.

Strumenti di Gestione e Organizzazione

Gestione del Progetto:

Trello - Organizzazione delle attività e milestone

Notion - Documentazione e appunti di sviluppo

Draw.io - Creazione di diagrammi e flowchart per la progettazione

Repository del progetto:

GitHub – Link nel quale è inserito il progetto.

<https://github.com/barrakko/Project>

Worktree: `fsdbdata@408676336788818313348c3e8437:~/C#2016/ale`

Testing e Debugging:

Python unittest - Framework per test unitari

Flask Debug Mode - Debugging in tempo reale

Browser Developer Tools - Analisi performance frontend

Difficoltà nell'Individuazione e Utilizzo

Problematiche Incontrate:

Versioning delle Librerie: Incompatibilità tra versioni diverse di Flask e Bootstrap

Documentazione Frammentata: Necessità di consultare multiple fonti per implementazioni specifiche

Scelta dell'Architettura: Decisione tra Flask e Django per il framework web

Configurazione dell'Ambiente: Setup corretto delle dipendenze Python

Soluzioni Adottate:

Creazione di Virtual Environment: Isolamento delle dipendenze per evitare conflitti

Consultazione Incrociata: Verifica delle informazioni su multiple fonti autorevoli

Prototipazione Rapida: Creazione di mini-progetti per testare diverse soluzioni

Documentazione Continua: Mantenimento di log dettagliati delle configurazioni funzionanti

Reperimento e Valutazione delle Risorse

Criteri di Selezione:

Affidabilità della Fonte: Preferenza per documentazione ufficiale e autori riconosciuti

Aggiornamento: Priorità a risorse recenti e compatibili con versioni attuali

Applicabilità Pratica: Selezione di risorse con esempi concreti e implementabili

Supporto della Community: Scelta di strumenti con ampia base di utenti per supporto

Modalità di Valutazione:

Testing Preliminare: Verifica pratica di tutorial e guide prima dell'implementazione

Peer Review: Consultazione di review e feedback della community sviluppatori

Benchmark Performance: Confronto di performance tra alternative simili

Analisi Costi-Benefici: Valutazione del rapporto tra complessità di apprendimento e benefici ottenuti

PARTE SECONDA – PREDISPOSIZIONE DELL'ELABORATO

Obiettivi del progetto

(Descrivere gli obiettivi raggiunti dall'elaborato, indicando in che modo esso risponde a quanto richiesto dalla traccia):

Obiettivo Generale

Sviluppare un sistema di gestione della produzione web-based per Athen S.P.A. che automatizzi il calcolo dei tempi di produzione e ottimizzi la pianificazione delle attività manifatturiere attraverso un'interfaccia intuitiva e funzionale.

Obiettivi Specifici

1. Automatizzazione dei Calcoli di Produzione

Obiettivo: Implementare algoritmi di calcolo automatico per tempi di produzione, capacità giornaliera e pianificazione temporale.

Risposta alle Richieste della Traccia:

Eliminazione del calcolo manuale dei tempi di produzione

Riduzione degli errori umani nella pianificazione

Standardizzazione dei processi di calcolo aziendale

Implementazione di fattori dimensionali per prodotti diversi

Risultati Raggiunti:

Algoritmo di calcolo che considera quantità, fattori dimensionali e tempi base

Calcolo automatico della capacità giornaliera (1440 minuti/giorno)

Determinazione precisa dei giorni richiesti per ogni prodotto

Sommario del tempo totale complessivo di produzione

2. Gestione Dinamica dei Materiali e Prodotti

Obiettivo: Creare un sistema flessibile per la gestione di materiali industriali e relativi prodotti finiti.

Risposta alle Richieste della Traccia:

Catalogazione di materiali (Plexiglas, Gomma, Lana vetro)
Associazione dinamica tra materiali e prodotti specifici
Possibilità di selezione multipla per ottimizzare la produzione
Scalabilità per aggiunta di nuovi materiali/prodotti

Risultati Raggiunti:

Struttura dati gerarchica materiali → prodotti
Sistema di selezione a cascata nell'interfaccia
Gestione di prodotti con caratteristiche dimensionali diverse
Flessibilità nell'aggiunta di nuove categorie

3. Interfaccia Utente Intuitiva e Responsive

Obiettivo: Sviluppare un'interfaccia web moderna che faciliti l'utilizzo del sistema da parte di operatori non tecnici.

Risposta alle Richieste della Traccia:

Accessibilità da dispositivi diversi (desktop, tablet, mobile)
Workflow semplificato per la selezione e generazione report
Visualizzazione chiara dei risultati di produzione
Design professionale coerente con l'immagine aziendale

Risultati Raggiunti:

Interfaccia responsive basata su Bootstrap
Design a card per presentazione organizzata dei dati
Form dinamici con auto-submit per selezione materiali
Styling personalizzato con colori aziendali

4. Ottimizzazione della Pianificazione Produttiva

Obiettivo: Fornire strumenti per una pianificazione efficiente delle attività di produzione.

Risposta alle Richieste della Traccia:

Calcolo preciso dei tempi necessari per ogni prodotto
Identificazione della capacità produttiva giornaliera
Pianificazione ottimale delle risorse temporali
Supporto decisionale per la gestione della produzione

Risultati Raggiunti:

Sistema di calcolo che considera le variabili dimensionali
Determinazione automatica dei giorni di produzione necessari
Visualizzazione del tempo totale complessivo
Dati strutturati per supportare decisioni manageriali

5. Implementazione di Tecnologie Web Moderne

Obiettivo: Utilizzare tecnologie web attuali per garantire performance, manutenibilità e scalabilità.

Risposta alle Richieste della Traccia:

Architettura web moderna con separazione frontend/backend
Utilizzo di framework consolidati e supportati
Codice manutenibile e documentato
Possibilità di estensioni future

Risultati Raggiunti:

Architettura Flask con pattern MVC
Frontend responsive con Bootstrap 5
Codice Python strutturato e commentato
Sistema modulare per future implementazioni

Corrispondenza con le Richieste della Traccia

Digitalizzazione dei Processi Aziendali

Il progetto risponde alla necessità di digitalizzare i processi di gestione della produzione, trasformando calcoli manuali in processi automatizzati e eliminando l'uso di fogli di calcolo statici.

Miglioramento dell'Efficienza Operativa

L'automazione dei calcoli e la standardizzazione dei processi riducono significativamente i tempi di pianificazione e aumentano l'accuratezza delle stime produttive.

Supporto Decisionale

Il sistema fornisce dati strutturati e visualizzazioni chiare che supportano i manager nella presa di decisioni strategiche sulla produzione.

Accessibilità e Usabilità

L'interfaccia web garantisce l'accesso al sistema da qualsiasi dispositivo e posizione, facilitando l'uso da parte di operatori con diversi livelli di competenza tecnica.

Scalabilità e Manutenibilità

L'architettura modulare permette facilmente l'aggiunta di nuovi materiali, prodotti o funzionalità senza richiedere modifiche strutturali significative.

Indicatori di Successo

Metriche Quantitative:

Riduzione del tempo di calcolo da ore a minuti
Eliminazione degli errori di calcolo manuale
Tempo di apprendimento del sistema < 30 minuti
Accessibilità multi-device al 100%

Metriche Qualitative:

Facilità d'uso dell'interfaccia
Accuratezza dei calcoli di produzione
Flessibilità nella gestione di materiali e prodotti
Professionalità dell'aspetto grafico

Contestualizzazione

(Descrivere il contesto teorico e quello applicativo dell'elaborato realizzato):

Il progetto si inserisce nel paradigma dell'Industria 4.0, caratterizzato dalla digitalizzazione e

automazione dei processi manifatturieri. La trasformazione digitale delle aziende manifatturiere rappresenta una necessità strategica per mantenere competitività nel mercato globale. L'implementazione di sistemi informatici per la gestione della produzione si allinea con i principi della smart factory, dove l'integrazione di tecnologie digitali ottimizza i flussi produttivi e migliora l'efficienza operativa.

Teoria dei Sistemi di Gestione della Produzione (Production Management Systems)

Il sistema sviluppato si basa sui principi teorici del Production Planning and Control (PPC), disciplina che studia l'ottimizzazione dei processi produttivi attraverso la pianificazione, programmazione e controllo delle attività manifatturiere. La metodologia implementata incorpora concetti di:

Capacity Planning: Calcolo della capacità produttiva giornaliera basata sui tempi di lavorazione

Lead Time Management: Gestione dei tempi di attraversamento per ogni prodotto

Resource Optimization: Ottimizzazione delle risorse temporali e produttive

Scheduling Theory: Principi di programmazione temporale delle attività

Architetture Software per Applicazioni Web

Dal punto di vista informatico, il progetto implementa il pattern architetturale Model-View-Controller (MVC), separando la logica di business (Model) dalla presentazione (View) e dal controllo delle interazioni (Controller). Questa architettura garantisce manutenibilità, scalabilità e separazione delle responsabilità, principi fondamentali dell'ingegneria del software moderna.

Human-Computer Interaction (HCI) e User Experience Design

La progettazione dell'interfaccia utente si basa sui principi dell'Human-Computer Interaction, privilegiando:

Usabilità: Interfaccia intuitiva per operatori non tecnici

Accessibilità: Compatibilità multi-device e multi-browser

Responsive Design: Adattamento automatico a diversi formati di schermo

Cognitive Load Theory: Riduzione del carico cognitivo attraverso workflow semplificati

Contesto Applicativo

Settore Manifatturiero Italiano

Il progetto è stato sviluppato per Athen S.P.A., azienda rappresentativa del tessuto industriale italiano caratterizzato da piccole e medie imprese manifatturiere. Il settore della lavorazione di materiali plastici, gomma e isolanti in Italia presenta specifiche esigenze:

Personalizzazione della Produzione: Necessità di gestire prodotti con caratteristiche dimensionali variabili

Flessibilità Produttiva: Capacità di adattarsi rapidamente a richieste di mercato diverse

Ottimizzazione dei Costi: Controllo preciso dei tempi e delle risorse produttive

Compliance Normativa: Rispetto delle normative qualitative e ambientali del settore

Scenario Tecnologico delle PMI Italiane

L'implementazione si inserisce nel contesto della digitalizzazione delle PMI italiane, caratterizzato da:

Digital Divide: Necessità di colmare il gap tecnologico rispetto alle grandi aziende

Risorse Limitate: Vincoli economici che richiedono soluzioni cost-effective
Competenze Interne: Limitata disponibilità di personale tecnico specializzato
Transizione Digitale: Processo graduale di migrazione da sistemi analogici a digitali

Materiali Industriali e Processi di Lavorazione

Il sistema gestisce tre categorie principali di materiali:

Plexiglas (PMMA - Polimetilmetacrilato):

Materiale termoplastico trasparente

Lavorazioni: taglio, fresatura, termoformatura

Applicazioni: pannelli, elementi ottici, componenti di design

Fattori critici: precisione dimensionale, finitura superficiale

Gomma (Elastomeri):

Materiali elastici naturali e sintetici

Lavorazioni: stampaggio, vulcanizzazione, taglio

Applicazioni: guarnizioni, paracolpi, componenti di tenuta

Fattori critici: proprietà meccaniche, resistenza chimica

Lana di Vetro (Fibra di Vetro):

Materiale isolante fibriforme

Lavorazioni: taglio, sagomatura, assemblaggio

Applicazioni: isolamento termico e acustico

Fattori critici: densità, spessore, proprietà isolanti

Problematiche Operative Pre-Implementazione

Prima dell'introduzione del sistema, Athen S.P.A. affrontava diverse criticità:

Calcoli Manuali: Utilizzo di fogli di calcolo Excel per stime produttive

Errori di Pianificazione: Imprecisioni nei tempi di consegna ai clienti

Inefficienza Operativa: Tempo eccessivo dedicato alla pianificazione

Mancanza di Standardizzazione: Processi di calcolo non uniformi tra operatori

Difficoltà di Scalabilità: Problemi nell'aggiunta di nuovi prodotti o materiali

Impatto della Soluzione nel Contesto Aziendale

Trasformazione Digitale:

L'implementazione rappresenta il primo passo verso la digitalizzazione completa dei processi aziendali, creando una base tecnologica per future integrazioni (ERP, MES, IoT).

Miglioramento della Competitività:

Riduzione dei tempi di risposta ai clienti

Maggiore precisione nelle quotazioni

Ottimizzazione delle risorse produttive

Possibilità di accettare commesse più complesse

Formazione e Change Management:

Il progetto ha richiesto un processo di formazione del personale e adattamento organizzativo, tipico delle implementazioni di sistemi informativi in contesti manifatturieri tradizionali.

Integrazione con Ecosistema Tecnologico

Il sistema è progettato per integrarsi con:

Sistemi ERP: Possibile integrazione con SAP, Oracle o sistemi open source

Sistemi MES: Collegamento con Manufacturing Execution Systems

IoT Devices: Connessione con sensori e dispositivi di monitoraggio produzione

Business Intelligence: Esportazione dati per analisi e reportistica avanzata

Sostenibilità e Responsabilità Sociale

L'ottimizzazione dei processi produttivi contribuisce a:

Riduzione degli Sprechi: Miglior utilizzo delle risorse materiali

Efficienza Energetica: Ottimizzazione dei tempi di utilizzo dei macchinari

Sostenibilità Ambientale: Riduzione dell'impatto ambientale per unità prodotta

Qualità del Lavoro: Eliminazione di attività ripetitive e soggette a errore

Posizionamento nell'Evoluzione Tecnologica

Il progetto si colloca nella fase di transizione delle PMI italiane verso l'Industria 4.0, rappresentando un esempio di implementazione tecnologica appropriata: sufficientemente avanzata da generare benefici tangibili, ma accessibile in termini di costi e competenze richieste. Questa caratteristica lo rende replicabile in contesti aziendali simili, contribuendo alla diffusione della cultura digitale nel settore manifatturiero italiano.

Descrizione dei principali aspetti progettuali

(Sviluppare l'elaborato richiesto dalla traccia prescelta):

Architettura del Sistema

Pattern MVC (Model-View-Controller)

Il progetto implementa una chiara separazione delle responsabilità attraverso il pattern MVC:

Model (Modello dei Dati):

```
7  # Materiali e relativi prodotti
8  ∨ materials = {
9      "Plexiglas": ["Pannello 10x20", "Pannello 20x40", "Disco 15cm"],
10     "Gomma": ["Guarnizione tonda", "Guarnizione rettangolare", "Paracolpi"],
11     "Lana vetro": ["Isolante piccolo", "Isolante medio", "Isolante grande"]
12 }
```

Interfaccia utente:

```

20 <body>
21 <div class="container py-5">
22   <h1 class="mb-4">Athen S.P.A. - Gestionale Produzione</h1>
23   <form method="post" class="mb-4">
24     <div class="mb-3">
25       <label for="material" class="form-label">Seleziona Materiale</label>
26       <select name="material" id="material" class="form-select" onchange="this.form.submit()">
27         <option value="">-- Scegli un materiale --</option>
28         {% for mat in materials %}
29           <option value="{{ mat }}" {% if mat == selected_material %}selected{% endif %}>{{ mat }}</option>
30         {% endfor %}
31       </select>
32     </div>
33   </form>

```

Logica sulla richiesta:

```

45 @app.route('/', methods=['GET', 'POST'])
46 def index():
47     selected_material = None
48     selected_products = []
49     results = None
50     total_time = 0
51
52     if request.method == 'POST':
53         selected_material = request.form.get("material")
54         selected_products = request.form.getlist("products")
55
56         if selected_material and selected_products:
57             results, total_time = generate_production_data(selected_products)
58

```

Il cuore del sistema è rappresentato dalla funzione di calcolo che implementa una logica industriale realistica:

```

14 # Funzione per generare dati casuali di produzione con logica corretta per dimensione
15
16 def generate_production_data(selected_products):
17     data = {}
18     total_time = 0
19
20     for product in selected_products:
21         quantity = random.randint(50, 200)
22
23         # Fattore dimensione: 1.0 (piccolo) fino a 3.0 (grande)
24         size_factor = round(random.uniform(1.0, 3.0), 2)
25         base_time = 30 # base time in minuti
26         time_per_unit = int(base_time * size_factor)
27
28         daily_capacity = 1440 // time_per_unit
29         days_required = math.ceil(quantity / daily_capacity)
30         product_time = days_required * daily_capacity * time_per_unit
31
32         total_time += product_time
33

```

Interfaccia lato utente:

Scelta del materiale dal menù a tendina:

Athen S.P.A. - Gestionale Produzione

Seleziona Materiale

-- Scegli un materiale --

-- Scegli un materiale --

Plexiglas

Gomma

Lana vetro

Scelta dei pezzi per il quale generare la produzione, è possibile sceglierli tutti o selezionare quelli di interesse:

Seleziona Materiale

Gomma

Prodotti di Gomma

- ☐ Guarnizione tonda
- ☐ Guarnizione rettangolare
- ☐ Paracolpi

Genera Produzione

Selezionandoli tutti e cliccando sul tasto “Genera produzione”, si avrà lo screen seguente, completo di tutti i parametri richiesti:

Risultati Produzione

Guarnizione tonda

Quantità: 72

Fattore dimensione: 1.36

Tempo per unità: 40 min

Capacità giornaliera: 36

Giorni richiesti: 2

Tempo totale prodotto: 2880 min

Guarnizione rettangolare

Quantità: 75

Fattore dimensione: 1.51

Tempo per unità: 45 min

Capacità giornaliera: 32

Giorni richiesti: 3

Tempo totale prodotto: 4320 min

Paracolpi

Quantità: 87

Fattore dimensione: 1.26

Tempo per unità: 37 min

Capacità giornaliera: 38

Giorni richiesti: 3

Campi di applicazione

(Descrivere gli ambiti di applicazione dell'elaborato progettuale e i vantaggi derivanti della sua applicazione):

Settore Manifatturiero

Industria della Plastica e Polimeri

Applicazioni Specifiche:

Produzione di componenti in PMMA per settore automotive

Lavorazione di pannelli per arredamento e design

Produzione di elementi ottici e display

Vantaggi:

Calcolo preciso dei tempi di termoformatura e lavorazione

Ottimizzazione dei cicli di produzione per diverse geometrie

Gestione efficiente di lotti di produzione variabili

Industria della Gomma e Guarnizioni

Applicazioni Specifiche:

Produzione di guarnizioni per settore automotive

Realizzazione di componenti per impianti industriali

Produzione di elementi anti-vibrazione

Vantaggi:

Pianificazione ottimale dei processi di vulcanizzazione

Gestione dei tempi di stampaggio per diverse mescole

Controllo qualità attraverso pianificazione precisa

Settore Isolanti e Materiali Edilizi

Applicazioni Specifiche:

Produzione di pannelli isolanti per edilizia

Realizzazione di componenti per isolamento acustico

Produzione di materiali per efficienza energetica

Vantaggi:

Ottimizzazione dei processi di taglio e sagomatura

Gestione efficiente di ordini su commessa

Calcolo preciso dei tempi di assemblaggio

Settori Industriali Correlati

Industria Meccanica di Precisione

Adattabilità del Sistema:

python# Esempio di estensione per lavorazioni meccaniche

```
materials = {  
    "Acciaio": ["Piastra 10x20", "Tubo Ø50", "Barra Ø30"],  
    "Alluminio": ["Profilo L", "Lamiera 2mm", "Estruso"],  
    "Ottone": ["Barra esagonale", "Tubo", "Piastra"]  
}
```

Vantaggi Applicativi:

Calcolo tempi di lavorazione CNC

Pianificazione di operazioni multi-setup

Gestione di lotti di produzione complessi

Industria Tessile e Abbigliamento

Personalizzazione per il Settore:

python# Adattamento per produzione tessile

```
materials = {  
    "Cotone": ["T-shirt", "Camicia", "Pantaloni"],  
    "Lana": ["Maglione", "Cappotto", "Sciarpa"],  
    "Sintetico": ["Giacca sportiva", "Leggings", "Polo"]  
}
```

Benefici:

Pianificazione stagionale della produzione

Gestione di taglie e varianti colore

Ottimizzazione dei tempi di confezionamento

Industria Alimentare

Applicazione nei Processi Alimentari:

python# Configurazione per produzione alimentare

```
materials = {  
    "Pasta": ["Spaghetti", "Penne", "Fusilli"],  
    "Pane": ["Pagnotta", "Grissini", "Focaccia"],  
    "Dolci": ["Biscotti", "Torte", "Crostate"]  
}
```

Vantaggi Specifici:

Gestione dei tempi di lievitazione e cottura

Pianificazione batch production

Controllo delle scadenze e freschezza

Settori di Servizio e Consulenza

Studi di Ingegneria e Progettazione

Utilizzo Professionale:

Calcolo tempi di progettazione per diverse tipologie di progetti

Pianificazione delle risorse di studio
Gestione portfolio clienti

Implementazione:

python# Configurazione per servizi di ingegneria

materials = {

"Progettazione Strutturale": ["Calcoli statici", "Disegni esecutivi", "Relazioni"],

"Progettazione Impiantistica": ["Schemi elettrici", "Layout impianti", "Computi"],

"Progettazione Architettonica": ["Concept design", "Progetto definitivo", "Rendering"]

}

Laboratori di Analisi e Testing

Applicazioni Analitiche:

Pianificazione test di qualità

Gestione campioni e analisi

Calcolo tempi di processamento

Valutazione dei risultati

(Descrivere le potenzialità e i limiti ai quali i risultati dell'elaborato sono potenzialmente esposti):

Raccomandazioni per Sviluppi Futuri

L'analisi complessiva del sistema evidenzia un progetto solido e ben indirizzato, in grado di generare benefici tangibili sia in termini di efficienza operativa che di ritorno economico. Tuttavia, per garantire la sostenibilità e la competitività nel lungo periodo, sono necessari alcuni interventi mirati.

1. Priorità Tecniche Immediate

Nel breve termine, è fondamentale introdurre meccanismi di persistenza dei dati tramite integrazione con un database, così da superare l'attuale limitazione della volatilità delle informazioni. Parallelamente, l'implementazione di un sistema di autenticazione e autorizzazione è cruciale per gestire in modo sicuro gli accessi multiutente e garantire la protezione dei dati. Un altro punto urgente riguarda la possibilità di esportare i dati (ad esempio in Excel o PDF), per facilitare la condivisione e l'archiviazione delle informazioni aziendali.

2. Sviluppi a Medio Termine

In una seconda fase, si consiglia di investire nello sviluppo di funzionalità avanzate come reportistica evoluta, dashboard interattive e supporto multilingua. Questi elementi aumenterebbero la versatilità e l'accessibilità del sistema, estendendone l'utilizzo anche a team internazionali. Anche la realizzazione di una versione mobile dell'applicazione potrebbe offrire maggiore flessibilità operativa, soprattutto per figure operative in movimento.

3. Innovazione a Lungo Termine

Nel lungo periodo, l'adozione di tecnologie più innovative come l'integrazione con dispositivi IoT, l'utilizzo dell'intelligenza artificiale per ottimizzazioni predittive e il deployment su cloud potrebbero trasformare il sistema in una piattaforma altamente scalabile e moderna, pronta per essere offerta anche in modalità SaaS (Software as a Service).

