



Università degli Studi di Padova

Laurea: Informatica

Corso: Ingegneria del Software

Anno Accademico: 2024/2025



Gruppo: SWEg Labs

Email: gruppo.sweg@gmail.com

Glossario

Versione 2.0.0

Stato	Approvato
Redazione	Federica Bolognini Michael Fantinato Giacomo Loat Filippo Righetto Riccardo Stefani Davide Verzotto
Verifica	Federica Bolognini Michael Fantinato Giacomo Loat Filippo Righetto Riccardo Stefani Davide Verzotto
Proprietario	Riccardo Stefani
Uso	Interno
Destinatari	Prof. Tullio Vardanega Prof. Riccardo Cardin Gruppo <i>SWEg Labs</i>

Registro delle modifiche

Versione	Data	Descrizione	Autore	Verificatore
2.0.0	...	Modifiche a	Riccardo Stefani
...	...	Modifiche a
...
1.1.5	18-03-25	Aggiunto il termine <u>Manuale Sviluppatore</u>	Federica Bolognini	Michael Fantinato
1.1.4	17-03-25	Aggiunto il termine <u>Data binding</u>	Riccardo Stefani	Michael Fantinato
1.1.3	11-03-25	Aggiunti i termini <u>Container Docker</u> , <u>Docker Desktop</u> , <u>Dockerfile</u> e <u>Immagine Docker</u>	Riccardo Stefani	Filippo Righetto
1.1.2	09-03-25	Aggiunti i termini <u>SQL</u> e <u>Chunk</u>	Riccardo Stefani	Federica Bolognini

Versione	Data	Descrizione	Autore	Verificatore
1.1.1	08-03-25	Aggiunti i termini <u>Facade</u> , <u>Dependency Injection</u> , <u>Model (MVVM)</u> , <u>View (MVVM)</u> , <u>ViewModel (MVVM)</u>	Riccardo Stefani	Giacomo Loat
1.1.0	08-03-25	Migliorata l'introduzione del documento per segnalare più chiaramente che il documento è interno, e quindi è utile a noi membri del gruppo	Riccardo Stefani	Federica Bolognini
1.0.14	07-03-25	Aggiunti i termini <u>Architettura logica</u> , <u>Architettura di deployment</u> e <u>Token</u>	Riccardo Stefani	Michael Fantinato
1.0.13	06-03-25	Aggiunti i termini <u>API Key</u> , <u>Polling REST</u> e <u>WebSocket</u>	Riccardo Stefani	Filippo Righetto
1.0.12	05-03-25	Aggiunti i termini <u>Design pattern</u> e <u>Coverage</u>	Davide Verzotto	Riccardo Stefani
1.0.11	01-03-25	Aggiunti i termini <u>Requisiti software</u> e <u>TDD</u>	Federica Bolognini	Riccardo Stefani
1.0.10	27-02-25	Aggiunti i termini <u>Use case</u> , <u>Port</u> , <u>Adapter (Architettura esagonale)</u> e <u>Repository (Architettura Esagonale)</u>	Davide Verzotto	Riccardo Stefani
1.0.9	25-02-25	Aggiunti i termini <u>Application Logic</u> , <u>DTO</u> , <u>Entity</u> e <u>Persistent Logic</u>	Riccardo Stefani	Davide Verzotto
1.0.8	25-02-25	Aggiunto il termine <u>Docker Compose</u>	Filippo Righetto	Riccardo Stefani
1.0.7	24-02-25	Aggiunti i termini <u>Adapter</u> , <u>Business Logic</u> , <u>Componente software</u> , <u>Domain business model</u> , <u>Manutenibilità</u> , <u>Mock</u> , <u>Scalabilità</u> , <u>Stub</u>	Davide Verzotto	Riccardo Stefani
1.0.6	19-02-25	Aggiunti i termini <u>Singleton</u> , <u>Diagramma UML di sequenza</u> , <u>Service</u> , <u>Controller</u> , <u>Test di integrazione</u> , <u>Test di unità</u> , <u>Progettazione logica</u> , <u>Progettazione di dettaglio</u> e <u>Architettura monolitica</u>	Riccardo Stefani	Federica Bolognini

Versione	Data	Descrizione	Autore	Verificatore
1.0.5	11-02-25	Aggiunto il termine Architettura esagonale	Riccardo Stefani	Michael Fantinato
1.0.4	10-02-25	Aggiunti i termini PyTest e Jasmine	Riccardo Stefani	Giacomo Loat
1.0.3	10-02-25	Aggiunto il termine Alpine Linux	Michael Fantinato	Federica Bolognini
1.0.2	07-02-25	Aggiunto il termine Committente	Giacomo Loat	Davide Verzotto
1.0.1	06-02-25	Aggiunto il termine JavaScript	Riccardo Stefani	Federica Bolognini
1.0.0	23-01-25	Approvazione del documento	Riccardo Stefani	Riccardo Stefani
0.4.0	23-01-25	Verifica del documento	Federica Bolognini	Federica Bolognini
0.3.9	23-01-25	Aggiunti i termini: Pascal Case , Kebab Case , Camel Case , Accessibilità	Federica Bolognini	Davide Verzotto
0.3.8	22-01-25	Aggiunti i termini: SonarQube for IDE , ESLint , PyLint	Michael Fantinato	Riccardo Stefani
0.3.7	18-01-25	Aggiunto il termine Scheduler	Riccardo Stefani	Michael Fantinato
0.3.6	17-01-25	Aggiunto il termine Implementazione	Riccardo Stefani	Giacomo Loat
0.3.5	16-01-25	Aggiunto il termine Specifica Tecnica	Riccardo Stefani	Michael Fantinato
0.3.4	09-01-25	Aggiunti i termini CSS , File .env , HTML , Logging e Refactoring	Riccardo Stefani	Michael Fantinato
0.3.3	03-01-25	Aggiunti i termini analisi statica e analisi dinamica	Filippo Righetto	Riccardo Stefani
0.3.2	03-01-24	Aggiunti i nuovi termini Docker , FastAPI e Flask	Riccardo Stefani	Filippo Righetto
0.3.1	28-12-24	Aggiunto il termine Google Chrome	Filippo Righetto	Riccardo Stefani
0.3.0	20-12-24	Verifica del documento allo stato attuale	Riccardo Stefani	Riccardo Stefani
0.2.4	20-12-24	Aggiunti i termini Consuntivo , Cron , Python Crontab , Trigger , Similarità , Web Server , Docstring , Git Flow e LangChain	Riccardo Stefani	Michael Fantinato

Versione	Data	Descrizione	Autore	Verifica
0.2.3	16-12-24	Aggiunti i termini <u>Manuale Utente</u> , <u>Modello di Question answering</u> e <u>Question answering (Modello di)</u>	Riccardo Stefani	Davide Verzotto
0.2.2	12-12-24	Aggiunti i termini <u>Bot</u> , <u>Prompt</u> , <u>MVC</u> e <u>MVVM</u>	Federica Bolognini	Davide Verzotto
0.2.1	09-12-24	Aggiunti i termini <u>Database relazionale</u> e <u>Scenario alternativo</u>	Riccardo Stefani	Giacomo Loat
0.2.0	06-12-24	Verifica del documento allo stato attuale	Riccardo Stefani	Riccardo Stefani
0.1.16	06-12-24	Aggiunti i termini <u>ChatGPT</u> , <u>Embedding (modello di)</u> , <u>Modello di embedding</u> , <u>OpenAI</u> , <u>Postgres (PostgreSQL)</u> , <u>Qdrant</u> , <u>Sottocaso d'uso</u> , <u>Textai</u> e <u>TypeScript</u>	Riccardo Stefani	Giacomo Loat
0.1.15	04-12-24	Aggiunti i termini <u>Chroma</u> , <u>Pinecone</u> , <u>Streamlit</u> , <u>Supabase</u> e <u>UML</u>	Riccardo Stefani	Federica Bolognini
0.1.14	29-11-24	Aggiunti i termini <u>Whisper.ai</u> , <u>Front-end</u> e <u>Back-end</u>	Giacomo Loat	Riccardo Stefani
0.1.13	20-11-24	Aggiunti i termini <u>Controllo di versione</u> , <u>Database vettoriale</u> , <u>Framework</u> , <u>Funzionalità</u> , <u>Hosting</u> , <u>LLM</u> , <u>Markup</u> , <u>Next.js</u> , <u>Piattaforma</u> , <u>Python</u> e <u>Stakeholder</u>	Riccardo Stefani	Davide Verzotto
0.1.12	18-11-24	Aggiunto il termine <u>Progettazione</u>	Filippo Righetto	Federica Bolognini
0.1.11	17-11-24	Aggiunti i termini <u>Specifiche funzionali</u> , <u>Specifiche tecniche</u> , <u>Contesto applicativo</u> , <u>Architettura</u> , <u>Accoppiamento</u> , <u>Pattern architetturale</u> , <u>Pull request</u> , <u>Build</u> , <u>Feedback</u> , <u>Google Meet</u> e <u>Issue Tracking System (ITS)</u>	Giacomo Loat	Federica Bolognini

Versione	Data	Descrizione	Autore	Verifica
0.1.10	16-11-24	Aggiunti i termini <u>Codifica</u> , <u>ISO/IEC 9126</u> , <u>Efficienza</u> , <u>Efficacia</u> , <u>Prodotto software</u> e <u>Test</u>	Riccardo Stefani	Federica Bolognini
0.1.9	16-11-24	Aggiunti i termini <u>Angular</u> , <u>NestJS</u> , <u>Node.js</u> , <u>Spring</u> , <u>Spring Boot</u> e <u>Telegram</u>	Riccardo Stefani	Michael Fantinato
0.1.8	13-11-24	Aggiunti i termini <u>Metrica</u>	Michael Fantinato	Filippo Righetto
0.1.9	13-11-24	Aggiunto il termine <u>ISO/IEC/IEEE 12207</u>	Federica Bolognini	Filippo Righetto
0.1.8	13-11-24	Aggiunti i termini <u>Demo</u> e <u>Snippet</u>	Riccardo Stefani	Filippo Righetto
0.1.7	12-11-24	Modificato il termine <u>Processo</u>	Giacomo Loat	Riccardo Stefani
0.1.6	12-11-24	Aggiunti i termini <u>Diario di Bordo</u> , <u>Agile (modello di sviluppo)</u> , <u>Incrementale (modello di sviluppo)</u> , <u>Burndown (diagramma di)</u> , <u>Preventivo</u>	Giacomo Loat	Riccardo Stefani
0.1.5	11-11-24	Aggiunti i termini <u>Capitolato</u> , <u>Caso d'uso</u> , <u>Attore</u> , <u>Scenario d'uso</u> , <u>Intelligenza artificiale</u> , <u>IA</u> , <u>API</u> , <u>Applicazione web</u> , <u>Browser</u> , <u>Confluence</u> , <u>Slack</u> e <u>Onboarding</u>	Filippo Righetto	Riccardo Stefani
0.1.4	11-11-24	Aggiunto il termine <u>ISO/IEC 31000:2018</u>	Federica Bolognini	Michael Fantinato
0.1.3	10-11-24	Aggiunti i termini <u>Baseline</u> , <u>Branch</u> , <u>Feature</u> , <u>Git</u> , <u>IDE</u> , <u>Product Baseline (PB)</u> , <u>Proponente</u> , <u>Proof of Concept (PoC)</u> , <u>Merge</u> , <u>Minimum Viable Product (MVP)</u> , <u>Snake Case</u> e <u>Visual Studio Code (VSC)</u>	Riccardo Stefani	Michael Fantinato
0.1.2	09-11-24	Aggiunti i termini <u>Ciclo di vita del software</u> , <u>Processo</u> , <u>Verifica</u> e <u>Way of Working</u>	Riccardo Stefani	Giacomo Loat
0.1.1	08-11-24	Inserimento dei primi termini, tratti dai primi due verbali interni della RTB	Riccardo Stefani	Filippo Righetto
0.1.0	05-11-24	Creazione del documento	Riccardo Stefani	Filippo Righetto

Tabella 1: Registro delle modifiche

Indice

1	Introduzione	1
1.1	Obiettivo del documento	1
1.2	Struttura del documento	1
A		2
B		5
C		7
D		10
E		13
F		14
G		16
H		18
I		19
J		21
K		22
L		23
M		24
N		26
O		27
P		28
Q		31
R		32
S		33
T		37
U		39
V		40
W		41
X		42
Y		43
Z		44

1 Introduzione

1.1 Obiettivo del documento

Questo documento è stato creato con l'intento di raccogliere tutti i termini che possano risultare sconosciuti ai membri del team o che possano creare ambiguità. Lo scopo è quello di creare una base omogenea di conoscenza specifica.

1.2 Struttura del documento

Il documento è suddiviso in sezioni rappresentanti le lettere dell'alfabeto; all'interno di queste si troveranno i termini ambigui o specifici del progetto che sono stati contrassegnati nel resto della documentazione in *corsivo* e con un pedice_G alla fine della parola.

A

Accoppiamento

In informatica, l'accoppiamento è il grado di dipendenza tra due o più componenti di un sistema software. Un accoppiamento elevato indica una forte interdipendenza tra le componenti, mentre un accoppiamento basso indica una minore dipendenza. Un basso accoppiamento è generalmente preferibile, poiché rende il sistema più flessibile, modulare e facile da mantenere.

Accessibilità

Capacità di rendere le informazioni sono più facilmente fruibili, condivisibili e adattabili alle esigenze di ciascun utente.

Adapter

Design pattern strutturale che consente di convertire l'interfaccia di una classe in un'altra interfaccia che il client si aspetta. L'adapter permette a classi con interfacce incompatibili di lavorare insieme, facilitando l'integrazione di componenti software eterogenee.

Adapter (Architettura esagonale)

Nell'architettura esagonale, l'adapter è un componente che si occupa di adattare le richieste provenienti dall'esterno del sistema alle interfacce interne, e viceversa. L'adapter permette di isolare il core del sistema dalle dipendenze esterne, garantendo una maggiore flessibilità e facilità di manutenzione. In particolare, esso converte i tipi di dato di business in oggetti Entity, e li fornisce alla classe Repository per l'interazione con il database o con lo strumento di persistenza dei dati utilizzato. In caso di recupero di dati dal database, l'adapter converte gli oggetti Entity che arrivano dal repository in tipi di dato di business, e li fornisce alla business logic per l'elaborazione.

Agile, modello di sviluppo

Approccio alla gestione dei progetti e allo sviluppo del software che enfatizza la flessibilità, la collaborazione e il miglioramento continuo. Ha come principi fondamentali consegna continua, collaborazione stretta tra sviluppatori e clienti, adattabilità ai cambiamenti, comunicazione diretta tra le parti e valutazione e miglioramento continuo del processo e del prodotto.

Alpine Linux

Alpine Linux è una distribuzione Linux minimalista, progettata per garantire un'elevata efficienza delle risorse e una robusta sicurezza. Offre un ambiente operativo leggero e performante, ideale per l'utilizzo in container e microservizi. La sua impronta ridotta consente tempi di avvio rapidi e una minore superficie di attacco, mentre il design minimalista favorisce una gestione semplificata e una maggiore affidabilità, rendendola una scelta privilegiata in ambienti di produzione dove le prestazioni e la sicurezza sono fondamentali.

Analisi dei Requisiti

Processo fondamentale dello sviluppo di un prodotto software che si concentra sulla raccolta, analisi e definizione delle necessità e delle aspettative degli utenti finali, degli stakeholder e del sistema nel suo complesso. Questo processo mira a comprendere e documentare in modo chiaro e completo le esigenze, le funzionalità, le prestazioni e i vincoli che il sistema deve soddisfare. L'obiettivo principale dell'analisi dei requisiti è fornire una base solida per tutte le fasi successive dello sviluppo del software, assicurando che il prodotto finale soddisfi le esigenze degli utenti e raggiunga gli obiettivi del progetto.

Analisi dei Rischi

L'analisi dei rischi è il processo di identificazione, valutazione e prioritizzazione dei rischi in un progetto, sistema o attività, al fine di ridurre o gestire il loro impatto potenziale. Viene utilizzata per prevedere gli eventi negativi che potrebbero influenzare il successo di un progetto e per determinare le azioni preventive o correttive da intraprendere.

Analisi dinamica

Esecuzione del software per verificarne il comportamento, identificare malfunzionamenti e valutare il rispetto dei requisiti in un ambiente reale o simulato.

Analisi statica

Controlli eseguiti senza avviare il software, utilizzati per verificare la qualità del codice, individuare errori e garantire conformità agli standard.

Analogico

Angular

Framework open-source sviluppato da Google per creare applicazioni web dinamiche e scalabili. Basato su TypeScript, utilizza un'architettura a componenti, il data binding bidirezionale e strumenti integrati per lo sviluppo di interfacce utente robuste e performanti.

API

Acronimo di Application Programming Interface, è un insieme di regole e protocolli che consente a diverse applicazioni software di comunicare tra loro per scambiare dati, caratteristiche e funzionalità. Le API fungono da intermediari, permettendo lo scambio di informazioni tra software diversi, semplificando e accelerando lo sviluppo di applicazioni.

API Key

Chiave di accesso univoca utilizzata per autenticare e autorizzare l'accesso a un'API. L'API key è un token di sicurezza che consente di identificare e verificare l'utente o l'applicazione che effettua la richiesta, garantendo che solo gli utenti autorizzati possano accedere alle risorse protette.

Application logic

Parte di un'applicazione software che implementa la logica di business e le regole di funzionamento del sistema in una applicazione specifica. L'application logic è responsabile della validazione dei dati che arrivano in input dall'applicazione esterna, chiamati DTO, e li adatta verso tipi di business per poterli trasmettere alla business logic che ci compie le operazioni di dominio. I dati ricevuti in output dalla business logic vengono poi adattati verso tipi di DTO per poterli trasmettere all'applicazione esterna.

Applicazione web

Applicazione accessibile via web tramite un browser (Safari, Chrome, Firefox, Edge etc.) e può funzionare sia da mobile che da desktop. Si tratta di un'architettura tipicamente di tipo client-server, che offre determinati servizi all'utente.

Architettura

In ambito informatico, l'organizzazione di base di un sistema, espressa dai suoi componenti, dalle relazioni tra di loro e con l'ambiente, e dai principi che ne guidano il progetto e l'evoluzione. Essa comprende le strutture del sistema, necessarie per ragionare su di esso, che includono elementi software, le relazioni tra di essi e le loro proprietà.

Architettura logica

Esempio di organizzazione di un sistema software in strati o livelli, dove ogni livello ha una responsabilità specifica e comunica solo con i livelli adiacenti. Un esempio comune di architettura logica è l'architettura a strati (Layered Architecture), che suddivide il sistema in livelli come presentazione, logica di business e accesso ai dati. Un altro esempio è l'architettura esagonale (Hexagonal Architecture), che promuove la separazione delle responsabilità e l'isolamento del core del sistema dalle dipendenze esterne.

Architettura esagonale

Pattern architetturale che promuove la separazione delle responsabilità all'interno di un'applicazione, organizzando il codice in base a un'architettura a esagono. Questo approccio favorisce la modularità, la manutenibilità e la scalabilità del software, consentendo di isolare le componenti del sistema e di ridurre le dipendenze tra di esse.

Architettura di deployment

Configurazione dell'infrastruttura e delle risorse necessarie per distribuire e gestire un'applicazione software. L'architettura di deployment definisce come l'applicazione è distribuita su server, cloud o dispositivi, e come le risorse sono allocate e gestite per garantire prestazioni, scalabilità e sicurezza. Due esempi di architettura di deployment sono l'architettura monolitica e l'architettura a microservizi.

Architettura monolitica

Modello architetturale in cui un'applicazione software è progettata come un'unica unità monolitica, in cui tutte le funzionalità e i componenti sono raggruppati insieme e distribuiti come un'entità unica. Questo approccio è caratterizzato da una forte interdipendenza tra i moduli, rendendo il sistema più difficile da scalare e mantenere. Tuttavia, l'architettura monolitica è spesso più semplice da sviluppare e testare rispetto ad altre architetture più complesse.

Attore

Nel contesto dell'analisi dei requisiti e del design del software, il termine attore rappresenta chiunque interagisca con il sistema. Gli attori possono essere utenti, altri sistemi o qualsiasi altra entità che abbia un ruolo nelle interazioni con il sistema modellato.

AzzurroDigitale

AzzurroDigitale è una società italiana con sede a Padova. Si occupa di digitalizzazione di processi sia con prodotti proprietari che di terze parti, e ha l'obiettivo di accompagnare le aziende manifatturiere nella transizione 5.0.

B

Back-end

Termine che si riferisce alla parte di un'applicazione web o di un sistema software che gestisce le funzioni e le elaborazioni necessarie per rendere possibile l'utilizzo dell'applicazione stessa. Esso comprende tutti i componenti, i server e i sistemi di archiviazione dati che non sono accessibili direttamente dall'utente finale, ma che lavorano dietro le quinte per fornire funzionalità e dati al frontend. Normali attività gestite dal back-end includono la gestione dei database, l'elaborazione delle richieste dei client, la logica di business e la sicurezza dell'applicazione.

Backlog

Insieme di compiti/attività da completare per un certo obiettivo. All'interno del framework Scrum, ne esistono due tipi principali: il product backlog, che è la lista delle funzionalità da implementare, e lo sprint backlog, che contiene le attività da svolgere durante un particolare sprint. Un'attività interna al backlog porta valore ad un progetto perchè possiede:

- Stato, che segnala se l'attività è stata completata, in corso o non ancora iniziata.
- Priorità, che indica l'importanza dell'attività rispetto alle altre.
- Assegnatario, cioè una persona incaricata a svolgere l'attività. Questa assegnazione non è vincolante, infatti se un membro del team ha terminato la sua attività può prendersi a carico un'altra attività presente nel backlog anche se non era stata inizialmente assegnata a lui.
- Scadenza, cioè un termine temporale entro il quale l'attività deve essere svolta.

Baseline

Nel contesto dell'ingegneria del software, stato di avanzamento che rappresenta un insieme di punti di arrivo che ci si pone come obiettivo di raggiungere in una milestone, dimostrando che l'incremento delle modifiche condotte ha portato a un risultato.

Best Practices

Nello sviluppo software, metodologie che attraverso l'esperienza e la sperimentazione sono state identificate come modi efficaci e raccomandati di affrontare determinati problemi o compiti nel processo di sviluppo del software. Queste pratiche sono considerate migliori (best) perché hanno dimostrato di portare a risultati di alta qualità, facilitando la manutenzione del codice e promuovendo una migliore collaborazione nel team di sviluppo.

Bot

Un bot è un software progettato per automatizzare attività ripetitive o interagire con gli utenti. Funziona seguendo regole predefinite o usando algoritmi di intelligenza artificiale.

Branch

Letteralmente "ramo", indica un'entità che si sviluppa o si dirama da un punto principale. Nel contesto di un sistema di controllo delle versioni, un branch rappresenta una linea di sviluppo separata. Può essere utilizzato per sviluppare nuove funzionalità, risolvere bug o implementare modifiche senza influenzare direttamente il ramo principale del codice, noto come master o main.

Browser

Applicazione software progettata per consentire agli utenti di navigare in Internet, visualizzare pagine web e accedere a contenuti online.

Build

Build

Definito anche come costruzione, è il processo di compilazione di un progetto software, in cui il codice sorgente viene trasformato in un formato eseguibile. La build può includere una serie di attività come la compilazione del codice sorgente, la creazione di file di configurazione e la generazione di file di installazione o di pacchetti per la distribuzione del software. Normalmente viene automatizzato tramite l'utilizzo di strumenti affidabili e riproducibili, gestendo le dipendenze del progetto e automatizzando il processo di rilascio e il controllo di versione.

Burndown (diagramma di)

Burndown, diagramma di

Strumento grafico utilizzato nella gestione agile dei progetti per tracciare la quantità di lavoro rimanente nel tempo. Esso mostra la diminuzione progressiva ("burn down") delle attività o dei punti stima rimanenti nel corso del tempo, consentendo al team di progetto di valutare il proprio progresso e adattare la pianificazione in base alle esigenze. A differenza del diagramma di Gantt, il diagramma di Burndown si concentra sulla visualizzazione dell'avanzamento reale rispetto al piano temporale.

Business logic

Parte di un'applicazione software che implementa le regole di business e la logica di funzionamento del sistema. La business logic è responsabile dell'elaborazione dei dati, della gestione delle regole aziendali e delle operazioni di calcolo necessarie per il funzionamento dell'applicazione.

C

Camel Case

È una convenzione di scrittura usata nella programmazione per nominare variabili, funzioni e altri identificatori. Si distingue perché la prima parola inizia con una lettera minuscola, mentre le parole successive iniziano con una lettera maiuscola, senza spazi o separatori.

Capitolato

Documento privato tra chi commissiona il lavoro e il gruppo (ditta) che lo esegue, in cui viene esposto un problema che il proponente necessita di risolvere e specifica le norme e i vincoli da rispettare per lo sviluppo del prodotto software specifico.

Caso d'uso

Descrizione dettagliata di come un utente (attore) interagisce con l'applicazione per il compimento di un'attività specifica. È uno strumento utilizzato nel contesto dello sviluppo software per individuare i requisiti funzionali del prodotto e per fornire una visuale chiara delle interazioni che possono avvenire all'interno dell'applicazione.

ChatGPT

ChatGPT è un modello di intelligenza artificiale sviluppato da OpenAI, basato sulla famiglia di modelli GPT (Generative Pre-trained Transformer). È progettato per comprendere e generare testo in linguaggio naturale, rendendolo utile in numerosi scenari, come chatbot, assistenti virtuali, generazione di contenuti e risposte automatizzate. ChatGPT utilizza una vasta base di conoscenza pre-addestrata e può essere ulteriormente personalizzato per applicazioni specifiche, offrendo interazioni conversazionali fluide e contestualmente rilevanti.

Checklist

Lista dettagliata di elementi, attività o criteri specifici che devono essere controllati, esaminati o completati durante le diverse fasi del ciclo di vita del software. E' utilizzata come strumento di controllo e verifica.

Chroma

Chroma è un database open-source ottimizzato per la gestione di dati vettoriali, progettato per supportare applicazioni di intelligenza artificiale e machine learning. È utilizzato principalmente per il retrieval di informazioni basato su similarità, come la ricerca di embedding, e integra funzionalità avanzate per lavorare con modelli di linguaggio (LLM). Fornisce un'API semplice per memorizzare, indicizzare e interrogare dati multidimensionali, rendendolo adatto a scenari come motori di raccomandazione, sistemi di domande e risposte, o clustering.

Chunk

Nel contesto dell'elaborazione del linguaggio naturale, un chunk è una sequenza di parole o token che rappresenta un'unità semantica significativa. I chunk sono utilizzati per identificare e raggruppare parti di testo con significato simile, come nomi, frasi o entità, facilitando l'analisi e l'elaborazione del testo.

Ciclo di vita del software

Serie di fasi attraverso le quali un software passa dal suo concepimento iniziale fino al suo ritiro o dismissione. È un concetto chiave nell'ingegneria del software e fornisce una struttura organizzativa per il processo di sviluppo del software.

Codifica

Fase del processo di sviluppo software in cui gli sviluppatori traducono i requisiti e il design del sistema in linguaggio di programmazione, creando il codice sorgente. Durante questa fase, gli sviluppatori seguono gli standard di codifica e le buone prassi per assicurare la leggibilità, l'efficienza e la manutenibilità del codice.

Committente

Nell'ambito dell'ingegneria del software, il termine committente si riferisce alla persona, all'organizzazione o all'entità che richiede e finanzia lo sviluppo di un sistema software. Il committente è colui che ha un interesse diretto nel progetto e assume il ruolo di chi determina le aspettative per il prodotto software finale. Il committente svolge un ruolo chiave nel processo di sviluppo del software, in quanto fornisce la visione iniziale del progetto, identifica le esigenze degli utenti finali e stabilisce i criteri di successo.

Componente software

Unità logica e funzionale di un sistema software, progettata per svolgere specifiche funzioni o compiti. Le componenti software sono i mattoni fondamentali di un'applicazione, e possono essere riutilizzate, combinate e integrate per creare sistemi più complessi.

Confluence

Piattaforma di collaborazione e gestione della conoscenza, utilizzata per creare, condividere e collaborare su documenti, progetti e informazioni all'interno di un team o di un'organizzazione. Confluence offre funzionalità come la creazione di wiki aziendali, la gestione di progetti, la documentazione tecnica e la collaborazione in tempo reale.

Consuntivo

Bilancio dei risultati ottenuti a rendiconto di un certo periodo temporale di attività, in termini di tempo e risorse.

Container Docker

Un container Docker è un'unità standardizzata di software che contiene tutto il necessario per eseguire un'applicazione, inclusi il codice sorgente, le librerie, le dipendenze e le configurazioni. I container Docker sono leggeri, portabili e autosufficienti, e possono essere eseguiti in modo consistente su qualsiasi ambiente, garantendo che l'applicazione funzioni correttamente indipendentemente dall'host o dalla configurazione del sistema.

Contesto applicativo

Ambito o scenario in cui un'applicazione software è progettata per essere utilizzata. Il contesto applicativo definisce le condizioni, le esigenze e le caratteristiche specifiche dell'ambiente in cui l'applicazione deve operare, influenzando il design, le funzionalità e le prestazioni del software.

Controller

Componente di un'applicazione web che gestisce le richieste degli utenti, coordina il flusso di dati e controlla il comportamento degli altri componenti. Il controller è responsabile di interpretare le azioni dell'utente, di invocare i servizi necessari e di restituire una risposta appropriata all'interfaccia utente.

Controllo di versione

Strumento che consente di gestire e tracciare le modifiche apportate al codice sorgente o ad altri file di progetto.

Coverage

Metrica che misura la percentuale di codice eseguita durante i test. Include vari livelli, come la copertura delle istruzioni, dei rami e delle condizioni, per valutare l'efficacia dei test e individuare parti di codice non verificate.

Cron

Nel contesto informatico, cron è un demone (un processo che si esegue in background) presente in molti sistemi operativi Unix-like (come Linux e macOS) che permette di pianificare l'esecuzione di comandi a intervalli di tempo regolari o a date e orari specifici.

CSS

Acronimo di Cascading Style Sheets, linguaggio di stile utilizzato per definire l'aspetto e lo stile delle pagine web HTML. CSS permette di separare il contenuto della pagina dalla sua presentazione, consentendo un maggiore controllo sulla formattazione e lo stile degli elementi della pagina. Con questo possono essere definiti degli stili, i quali possono definire proprietà come il colore, il font, la dimensione, la posizione e l'animazione.

D

Data binding

Il data binding è un meccanismo che stabilisce una connessione tra i dati e l'interfaccia utente di un'applicazione. Consente di sincronizzare automaticamente i dati del modello con la vista, in modo che qualsiasi modifica ai dati venga riflessa immediatamente nell'interfaccia utente e viceversa. Il data binding semplifica lo sviluppo delle applicazioni, riducendo la complessità e il codice necessario per mantenere sincronizzati i dati e l'interfaccia utente.

Database relazionale

Un database relazionale è un tipo di database che organizza i dati in tabelle strutturate, con righe e colonne. Ogni tabella rappresenta un'entità e le relazioni tra le tabelle sono definite tramite chiavi primarie e chiavi esterne. Questo modello facilita la gestione e l'interrogazione dei dati utilizzando il linguaggio SQL.

Database vettoriale

Un database vettoriale è un tipo di sistema di archiviazione progettato per gestire e interrogare dati rappresentati come vettori, spesso in spazi ad alta dimensionalità. Questi vettori sono comunemente utilizzati per rappresentare oggetti, concetti o entità in un formato numerico, solitamente prodotto da algoritmi di machine learning, reti neurali o tecniche di embedding.

Demo

Una demo (abbreviazione di "dimostrazione") è una presentazione interattiva o registrata di un software o di una funzionalità specifica all'interno di un progetto. Viene usata per mostrare lo stato attuale dello sviluppo, illustrando come funzionano le caratteristiche implementate e come si comporta il sistema rispetto ai requisiti stabiliti. Le demo sono solitamente presentate a stakeholder, clienti, o team interni per ottenere feedback, verificare funzionalità e orientare lo sviluppo futuro. Le demo sono particolarmente utili nei processi iterativi e agili, dove è importante monitorare e comunicare costantemente i progressi del progetto.

Dependency Injection

La Dependency Injection è un design pattern di programmazione che permette di ridurre le dipendenze tra i componenti di un sistema. Invece di creare esplicitamente le dipendenze all'interno di un componente, queste vengono passate dall'esterno, rendendo il componente più flessibile, riutilizzabile e facilmente testabile. La Dependency Injection è spesso utilizzata per migliorare la modularità e la manutenibilità del codice.

Design pattern

Soluzione generale e riutilizzabile a un problema che si verifica comunemente nell'ambito dello sviluppo del software. I design pattern forniscono un approccio testato e comprovato per la progettazione e l'implementazione del software.

Diagramma di Gantt

Strumento di visualizzazione temporale utilizzato nella gestione dei progetti per rappresentare le attività pianificate nel tempo. È composto da una barra orizzontale che rappresenta l'arco temporale totale del progetto e da barre orizzontali più piccole che rappresentano le singole attività del progetto. Ogni barra è posizionata lungo l'asse temporale in base alle date di inizio e fine previste per l'attività.

Diagramma UML

Acronimo di Unified Modeling Language, un diagramma UML diagramma utilizzato per modellare, descrivere e visualizzare sistemi software e processi di sviluppo software. È uno standard industriale nel campo dell'ingegneria del software e fornisce una serie di diagrammi, ognuno dei quali si concentra su un aspetto specifico del sistema o del processo. Sono diagrammi UML ad esempio i diagrammi dei casi d'uso, i diagrammi delle classi e i diagrammi delle funzionalità.

Diagramma UML dei casi d'uso

Diagramma UML che rappresenta le interazioni tra utenti (attori) e il sistema, descrivendo come gli utenti utilizzano il sistema per raggiungere obiettivi specifici. Ogni caso d'uso rappresenta una funzione o un'attività significativa, utile per descrivere i requisiti funzionali. È spesso il primo passo nella progettazione di un sistema software e aiuta a identificare le funzioni principali e il modo in cui il sistema interagisce con gli utenti.

Diagramma UML delle classi

Diagramma UML che descrive la struttura statica di un sistema, mostrando le classi, i loro attributi, i metodi e le relazioni tra di esse (come ereditarietà, associazioni e aggregazioni). Questo diagramma è fondamentale per la programmazione orientata agli oggetti poiché fornisce una rappresentazione visiva della struttura del codice, aiutando a comprendere le interconnessioni tra le varie entità e a definire i componenti principali.

Diagramma UML di sequenza

Diagramma UML che descrive come gli oggetti in un sistema interagiscono tra loro e in quale sequenza. Mostra le chiamate di metodo, le risposte e le interazioni tra gli oggetti in un contesto specifico. È particolarmente utile per visualizzare il flusso di controllo e le interazioni tra i vari componenti di un sistema software.

Diario di Bordo

Nel contesto del progetto didattico, il “diario di bordo” è un'attività settimanale in cui ogni gruppo di progetto documenta e presenta pubblicamente il progresso del proprio lavoro. Questa attività include la descrizione delle attività svolte, delle difficoltà incontrate, dei dubbi e delle incertezze, e dei piani per il periodo successivo. L'obiettivo è promuovere una discussione aperta e riflessiva per migliorare la consapevolezza e la gestione del progetto collaborativo.

Discord

Piattaforma VoIP (Voice over IP: tecnologia che rende possibile effettuare una conversazione sfruttando una connessione internet), messaggistica istantanea e distribuzione digitale progettata per la comunicazione.

Docker

Docker è un software progettato per eseguire processi informatici in ambienti isolabili, minimali e facilmente distribuibili chiamati container, con l'obiettivo di semplificare i processi di deployment di applicazioni software.

Docker Compose

Strumento progettato per gestire il deploy di applicazioni tramite la definizione di container. Questo tool di Docker permette di raggruppare e avviare un insieme di container con un solo comando.

Docker Desktop

Docker Desktop è un'applicazione per Windows, Linux e macOS che fornisce un'esperienza di sviluppo Docker completa, inclusi Docker Engine, Docker CLI, Docker Compose, Kubernetes e altre funzionalità. È progettata per semplificare l'installazione e la configurazione di Docker sul desktop, consentendo agli sviluppatori di creare, testare e distribuire applicazioni in container direttamente sul proprio computer.

Dockerfile

Un Dockerfile è un file di testo che contiene una serie di istruzioni per creare un'immagine Docker. Queste istruzioni definiscono come l'immagine deve essere costruita, quali dipendenze devono essere installate, quali comandi devono essere eseguiti e come l'applicazione deve essere configurata. Il Dockerfile è utilizzato da Docker per automatizzare il processo di creazione di immagini e garantire che l'ambiente di esecuzione sia coerente e riproducibile.

Docstring

Una docstring è una stringa speciale inserita all'interno del codice sorgente di un programma per documentare una funzione, un modulo o una classe. In pratica, è una descrizione dettagliata e concisa dello scopo, dei parametri, del valore di ritorno e di eventuali eccezioni sollevate da un elemento di codice.

Domain business model

Il Domain business model è il nucleo dell'architettura esagonale, contenente la logica di business e le regole di dominio dell'applicazione.

Draw.io

Draw.io (ora chiamato diagrams.net) è uno strumento gratuito per la creazione di diagrammi, disponibile sia come applicazione web che come app desktop per vari sistemi operativi. Viene utilizzato ampiamente per progettare e documentare diagrammi di flusso, architetture software, diagrammi UML, organigrammi, mappe mentali, wireframe, e altri tipi di rappresentazioni visive.

DTO (Data Transfer Object)

Un DTO (Data Transfer Object) è un oggetto utilizzato per trasferire dati tra sottosistemi di un'applicazione. È progettato per essere semplice, leggero e contenere solo i dati necessari per la comunicazione tra i diversi componenti del sistema. I DTO sono spesso utilizzati per ridurre il traffico di rete e migliorare le prestazioni dell'applicazione. In particolare, nell'architettura esagonale, i DTO sono il tipo di dati gestito dall'application logic, la quale riceve DTO dall'esterno e li adatta per inviarli alla business logic, e riceve dati di dominio dalla business logic e li adatta per inviarli all'esterno.

E

Efficacia

Capacità di raggiungere gli obiettivi desiderati o di produrre gli effetti previsti.

Efficienza

Capacità di svolgere un compito, un'attività o un processo nel modo più economico e con il minimo spreco di risorse.

Embedding (modello di)

Un Modello di embedding è un modello di intelligenza artificiale progettato per trasformare dati, come parole, frasi o immagini, in rappresentazioni vettoriali numeriche in uno spazio multidimensionale. Questi vettori catturano caratteristiche semantiche o strutturali dei dati, rendendoli utili per confronti, analisi o classificazioni basate su similarità. I modelli di embedding sono comunemente utilizzati in applicazioni come la ricerca semantica, il clustering e il machine learning.

Entity

Una Entity è un oggetto di persistenza di una applicazione, che nell'architettura esagonale viene gestita da un repository. Essa rappresenta l'adattamento interno all'applicazione di un tipo di dato giunto da un servizio esterno, come ad esempio un database, ed ha il compito di fornire un layer intermedio tra tipi di dato di business e tipi di dato esterni.

ESLint

ESLint è uno strumento ampiamente utilizzato per l'analisi statica del codice JavaScript (e relativi framework come React, Vue, e Node.js), con l'obiettivo di individuare errori, applicare standard di codifica e migliorare la qualità complessiva del codice.

F

Facade

Il Facade è un design pattern strutturale che fornisce un'interfaccia unificata per un insieme di interfacce in un sistema. Nasconde la complessità delle interazioni tra sottosistemi e fornisce un punto di accesso semplificato per il client. Il Facade semplifica l'utilizzo del sistema e riduce la dipendenza del client dai sottosistemi.

FastAPI

FastAPI è un framework web Python moderno e performante, progettato per creare API RESTful in modo rapido e intuitivo. Offre una sintassi concisa e tipo-hintata, che riduce significativamente il numero di bug e aumenta la velocità di sviluppo. FastAPI è ideale per costruire microservizi e backend per applicazioni web, grazie alla sua eccellente scalabilità e integrazione con strumenti di test e documentazione.

Feature

Una feature è una specifica funzionalità o capacità di un sistema software che ne arricchisce il comportamento e le possibilità d'uso. Le feature sono aggiunte o miglioramenti che rispondono a esigenze degli utenti, migliorano l'esperienza d'uso, o risolvono specifici problemi. L'introduzione o il miglioramento di una feature segue un ciclo di sviluppo completo, dalla raccolta dei requisiti fino alla verifica e al testing.

Feedback

Il feedback è un'informazione, un'opinione o una valutazione che viene fornita in risposta a un'azione, a una prestazione o a un comportamento. Ha lo scopo di evidenziare punti di forza e aree di miglioramento, permettendo a chi lo riceve di comprendere meglio come il proprio operato viene percepito e come può migliorare o adattarsi. Il processo di feedback è un processo per cui il risultato dell'azione di un sistema si riflette sul sistema stesso per correggerne o modificarne il comportamento.

File .env

Un file di testo che contiene variabili di ambiente utilizzate per configurare un'applicazione. Consente di separare le impostazioni specifiche di un ambiente (es. sviluppo, produzione) dal codice sorgente, facilitando la gestione e la sicurezza delle informazioni sensibili come chiavi API, password e URL.

Flask

Flask è un leggero framework web Python per lo sviluppo di applicazioni web. Conosciuto per la sua flessibilità e semplicità d'uso, Flask offre una base solida su cui costruire applicazioni personalizzate. Permette agli sviluppatori di scegliere le librerie e gli strumenti che meglio si adattano al loro progetto, senza imporre una struttura rigida.

Fogli Google

Fogli Google è un'applicazione web di Google, parte della suite di produttività Google Workspace, che consente di creare, modificare e condividere fogli di calcolo online. È uno strumento particolarmente apprezzato per il suo accesso immediato da browser, le funzionalità collaborative in tempo reale, e l'integrazione con altri servizi Google. Così come ogni foglio di calcolo, include la possibilità di creare grafici e diagrammi basati su dati tabellari.

Framework

Infrastruttura software che fornisce un'architettura generale per lo sviluppo di applicazioni. Un framework è progettato per facilitare il processo di sviluppo fornendo strutture, librerie, linee guida e pattern comuni che possono essere seguiti dagli sviluppatori. L'obiettivo è semplificare il lavoro degli sviluppatori fornendo un ambiente predefinito in cui possono costruire le loro applicazioni.

Front-end

Termine riferito alla parte grafica e funzionale dell'interfaccia utente con cui l'utente interagisce direttamente. Esso comprende elementi grafici, un'interfaccia utente, la disposizione dei pulsanti, gli elementi di input e la validazione dei dati inseriti dall'utente. Gli strumenti più comuni per lo sviluppo del front-end includono HTML, CSS e JavaScript.

Funzionalità

Nella programmazione, indica una capacità specifica offerta da un sistema o un'applicazione di soddisfare un determinato requisito o necessità.

G

Gantt, diagramma di

Vedi [Diagramma di Gantt](#).

Git

Software per il controllo di versione distribuito utilizzabile tramite interfaccia a riga di comando.

Git Flow

Git Flow è un modello di branching di Git che definisce un insieme di regole e procedure per gestire il ciclo di vita di un progetto software. Offre una struttura ben definita per la collaborazione tra sviluppatori e la gestione dei rilasci, garantendo un flusso di lavoro efficiente e organizzato.

GitHub

Servizio di hosting per progetti software. Il sito è principalmente utilizzato da sviluppatori che caricano il codice sorgente di programmi in dei repository e lo rendono scaricabile e migliorabile da altri sviluppatori. Questi ultimi possono interagire con i proprietari dei repository tramite un sistema per inviare segnalazioni di bug o richieste di funzionalità (issue tracker), un sistema per copiare il software in una versione modificabile (fork), un sistema per proporre modifiche agli sviluppatori originali (pull request) e un sistema di discussione legato al codice del repository (commenti).

GitHub Pages

GitHub Pages è un servizio di hosting gratuito offerto da GitHub che permette agli utenti di creare e pubblicare facilmente siti web statici direttamente dai loro repository GitHub. È utilizzato comunemente per creare siti di documentazione, pagine personali o di progetto, blog e portali di portfolio. GitHub Pages è particolarmente apprezzato perché permette di ospitare un sito senza costi e con aggiornamenti automatici ogni volta che il repository viene modificato.

GitHub Projects

GitHub Projects è uno strumento di gestione dei progetti integrato in GitHub, ideato per aiutare sviluppatori e team a organizzare, pianificare e tracciare il lavoro sui progetti direttamente all'interno dell'ambiente GitHub. Si basa su un sistema flessibile di "project board" simile a Kanban, che offre un modo visuale per coordinare i task e monitorare l'avanzamento del lavoro. GitHub Projects è uno strumento ideale per team che lavorano su progetti di sviluppo software, in quanto permette di gestire l'intero processo di sviluppo all'interno di GitHub stesso. Grazie alla sua integrazione nativa con il codice, le issues e le pull requests, aiuta a mantenere sincronizzati i task e a ridurre il contesto di cambiamento per gli sviluppatori, migliorando il flusso di lavoro e facilitando la collaborazione su GitHub.

Glossario

Elenco organizzato di termini tecnici, acronimi e definizioni utilizzati nel contesto del progetto. Questo documento fornisce una chiara comprensione dei concetti e dei linguaggi specifici impiegati nel progetto, aiutando a ridurre ambiguità e fraintendimenti tra i membri del team e gli stakeholder.

Google Chrome

Browser web che consente di accedere a Internet e interagire con siti e applicazioni online.

Google Meet

Servizio di videoconferenza sviluppato da Google. È parte di Google Workspace e offre funzionalità di videochiamata, chat, condivisione dello schermo e registrazione delle riunioni. Google Meet è ampiamente utilizzato per incontri di lavoro, lezioni online, webinar e conferenze, grazie alla sua facilità d'uso, alla stabilità della connessione e alla possibilità di partecipare senza dover scaricare alcun software.

Gulpease, indice di

Indice di leggibilità di un testo specificamente in lingua italiana, che utilizza il numero delle parole, delle frasi e delle lettere per facilitare il calcolo automatico della leggibilità.

H

Hosting

Servizio che fornisce l'infrastruttura necessaria per rendere accessibili siti web, applicazioni o dati attraverso Internet.

HTML

Acronimo di HyperText Markup Language, linguaggio utilizzato per la creazione e l'impaginazione di documenti ipertestuali disponibili sul web. Consente di strutturare il contenuto delle pagine web utilizzando tag e attributi, permettendo la visualizzazione di testo, immagini, link e altri elementi multimediali nei browser web.

I

IA

Acronimo di intelligenza artificiale

IDE

Acronimo di Integrated Development Environment, è un ambiente di sviluppo, ovvero un software che, in fase di programmazione, supporta i programmatori nello sviluppo e debugging del codice sorgente di un programma, segnalando errori di sintassi del codice direttamente in fase di scrittura, oltre a fornire una serie di strumenti e funzionalità di supporto alla fase stessa di sviluppo e debugging.

Immagine Docker

Un'immagine Docker è un file leggero, stand-alone e eseguibile che comprende tutto il necessario per eseguire un'applicazione: codice sorgente, runtime, librerie, variabili d'ambiente e configurazioni. Le immagini Docker sono utilizzate per creare container Docker, che sono istanze eseguibili di un'immagine e permettono di eseguire applicazioni in ambienti isolati e portatili.

Implementazione

L'implementazione è il processo mediante il quale un progetto, un piano o un sistema viene realizzato e reso operativo, traducendo specifiche o requisiti in soluzioni concrete e funzionanti. In termini generali, implica la costruzione, l'integrazione e la configurazione degli elementi necessari per ottenere il risultato desiderato. Oltre a ciò, l'implementazione include la responsabilità di restituire un utensile usabile ai fini attesi, ossia fornire un prodotto che soddisfi le esigenze per cui è stato progettato, assicurandosi che sia funzionale, accessibile e adatto all'uso previsto dagli utenti finali.

Incrementale (modello di sviluppo)

Incrementale, modello di sviluppo

Approccio alla creazione di software che suddivide il progetto in piccoli segmenti o incrementi. Ogni incremento rappresenta una versione funzionante del software che include nuove funzionalità o miglioramenti rispetto alla versione precedente.

Intelligenza artificiale

Disciplina che studia come realizzare sistemi informatici in grado di simulare il pensiero umano. Sistemi basati su di essa hanno dunque l'abilità di mostrare capacità umane quali il ragionamento, l'apprendimento, la pianificazione e la creatività.

ISO/IEC 31000:2018

La norma ISO 31000 "Risk management - Principles and guidelines", in italiano UNI ISO 31000 Gestione del rischio - Principi e linee guida. È una guida che fornisce principi e linee guida generali per la gestione del rischio. Può essere utilizzata da qualsiasi organizzazione pubblica, privata o sociale, associazione, gruppo o individuo, e non è specifica per nessuna industria o settore. La ISO 31000 può essere applicata nel corso dell'intero ciclo di vita di un'organizzazione, ed essere adottata per molte attività come la definizione di strategie e decisioni, operazioni, processi, funzioni, progetti, prodotti, servizi e beni. Può inoltre essere applicata a qualsiasi tipo di rischio, sia per conseguenze di tipo positivo che negativo.

ISO/IEC 9126

La norma ISO/IEC 9126 è uno standard internazionale per la valutazione della qualità del software. Definisce un modello di qualità software che si basa su sei caratteristiche generali di qualità (funzionalità, affidabilità, usabilità, efficienza, manutenibilità, portabilità) e su 27 sotto-caratteristiche. Lo standard è stato sostituito dalla norma ISO/IEC 25010:2011, che definisce un modello di qualità software più ampio e aggiornato.

ISO/IEC/IEEE 12207

Lo standard ISO 12207 stabilisce un processo di ciclo di vita del software, compreso processi ed attività relative alle specifiche ed alla configurazione di un sistema. Ad ogni processo corrisponde un insieme di risultati (outcome): in totale ci sono 43 processi, 133 attività, 325 sottoattività e 236 risultati (la nuova ISO/IEC 12207:2008 definisce 43 sistemi e processi software). Lo standard ha come obiettivo principale quello di fornire una struttura comune che permetta a clienti, fornitori, sviluppatori, tecnici, manager di usare gli stessi termini e lo stesso linguaggio per definire gli stessi processi. La struttura dello standard è stata concepita per essere flessibile e modulare in modo che sia adattabile alle necessità di chiunque lo utilizzi.

Issue

Una issue su GitHub (e altre piattaforme di gestione del codice e dei progetti) è un elemento utilizzato per tracciare problemi, richieste di funzionalità, idee o miglioramenti relativi a un progetto. È uno strumento fondamentale per organizzare il lavoro collaborativo e garantire che tutti i membri del team siano aggiornati sui task e le priorità. Ogni issue rappresenta un singolo elemento che richiede attenzione o azione, e fornisce un luogo centralizzato per discuterlo, seguirlo e risolverlo.

Issue Tracking System (ITS)

Issue Tracking System (ITS)

Un Issue Tracking System (ITS) è un software o un sistema di gestione che consente di creare, monitorare e gestire issue (problemi, task, richieste o bug) all'interno di un progetto. Questo strumento è utilizzato principalmente per facilitare la collaborazione tra membri del team e per garantire che tutte le problematiche e le richieste siano gestite in modo organizzato, trasparente e tracciabile. Un ITS è spesso impiegato da team di sviluppo software, ma può essere utile anche in contesti di supporto clienti, gestione di progetto, e qualsiasi ambito dove occorra tracciare richieste di lavoro o problemi.

J

Jasmine

Jasmine è un framework di testing per JavaScript, progettato per essere semplice e intuitivo. È spesso utilizzato per lo sviluppo di applicazioni web, in particolare per il testing di codice front-end. Jasmine offre una sintassi chiara e leggibile, che facilita la scrittura e l'esecuzione di test automatizzati, e fornisce una serie di funzionalità per la gestione delle asserzioni, dei mock e dei test asincroni. Jasmine è compatibile con diversi runner di test e framework di sviluppo, come Karma e AngularJS.

JavaScript

Linguaggio di programmazione ad alto livello utilizzato per aggiungere interattività alle pagine web, grazie alla capacità di essere orientato agli eventi esterni ed interagire con essi in modo asincrono. Esso viene utilizzato sia nella programmazione lato client (front-end), che si occupa della parte grafica e dell'interfaccia utente, sia nella programmazione lato server (back-end), che gestisce la logica e la comunicazione del sistema. JavaScript è uno dei linguaggi fondamentali del web, insieme a HTML e CSS, ed è supportato da tutti i principali browser.

Jira

Jira è uno strumento di gestione dei progetti e di issue tracking sviluppato da Atlassian, ampiamente utilizzato per la pianificazione, il monitoraggio e il controllo di progetti, in particolare nell'ambito dello sviluppo software. Nato come strumento di gestione dei bug e dei problemi, Jira è diventato uno dei principali strumenti per il project management, soprattutto per le organizzazioni che adottano metodologie agili come Scrum e Kanban.

K

Kebab Case

È una convenzione di scrittura in cui le parole sono separate da un trattino (-) e sono tutte in minuscolo.

L

LangChain

LangChain è un framework Python e JavaScript open-source progettato per sviluppare applicazioni che sfruttano i Large Language Models (LLM). Offre una struttura modulare e flessibile per la creazione di catene di elaborazione (chains) che combinano modelli di linguaggio con altre fonti di dati e strumenti.

LaTeX

Linguaggio di marcatura per la preparazione di testi, basato sul programma di composizione tipografica TeX. LaTeX è ampiamente utilizzato per la creazione di documenti scientifici e tecnici grazie alla sua capacità di gestire formule matematiche complesse e alla sua alta qualità tipografica.

Large Language Model (LLM)

Il termine Large Language Model si riferisce a modelli di linguaggio avanzati e complessi che sono stati addestrati su enormi quantità di dati testuali. Questi modelli, basati su tecniche di intelligenza artificiale come il deep learning, sono in grado di comprendere e generare testo in linguaggio naturale in modo più sofisticato rispetto a modelli più piccoli (Small Language Model, SML).

Logging

Il processo di registrazione di eventi, errori e altre informazioni rilevanti all'interno di un'applicazione. Questi dati vengono solitamente salvati in un file di log, un database o inviati a un servizio esterno per analisi successive. Il logging è fondamentale per la risoluzione dei problemi, il monitoraggio delle prestazioni e la sicurezza delle applicazioni.

M

Manuale Utente

Uno dei documenti che è richiesto che vengano redatti per la Product Baseline, con lo scopo di fornire istruzioni dettagliate su come utilizzare il sistema sviluppato. Nell'ambito dell'ingegneria del software, il manuale utente è progettato per aiutare gli utenti finali a comprendere le funzionalità del software, eseguire operazioni specifiche e risolvere problemi comuni. Include descrizioni delle interfacce utente, spiegazioni delle funzionalità principali, guide passo-passo, e soluzioni ai problemi più frequenti. Il manuale utente è essenziale per garantire che gli utenti possano utilizzare il software in modo efficace e autonomo.

Manuale Sviluppatore

Un manuale sviluppatore è un documento tecnico che fornisce istruzioni dettagliate, linee guida e riferimenti necessari per progettare, sviluppare e mantenere software o hardware. Contiene le funzionalità, gli strumenti, le best practice e gli esempi di codice destinati specificamente ai programmatori che lavorano con una particolare tecnologia, linguaggio di programmazione o piattaforma.

Manutenibilità

Caratteristica di un software che ne facilita la manutenzione, ovvero la capacità di apportare modifiche, correzioni e miglioramenti al codice in modo efficiente, sicuro e affidabile.

Markup

Un sistema di codifica che utilizza tag o annotazioni per strutturare e formattare documenti digitali (es. HTML o XML).

Merge

Operazione fondamentale nei sistemi di controllo delle versioni. Essa è utilizzata per combinare le modifiche apportate in due branch separati in un singolo branch.

Metrica

Misura quantitativa utilizzata per valutare, quantificare e analizzare diversi aspetti del processo di sviluppo del software, del prodotto software stesso o della gestione del progetto. Le metriche forniscono dati numerici che consentono di valutare l'andamento del progetto, la qualità del software, l'efficacia dei processi e altri aspetti rilevanti.

Milestone

In ingegneria del software e nella gestione dei progetti, punto di riferimento o traguardo significativo che sancisce il termine di un periodo nel ciclo di vita di un progetto. Le milestone rappresentano generalmente eventi chiave, compimenti o obiettivi importanti che indicano il progresso del progetto. L'obiettivo che ci si pone durante una milestone è realizzare una baseline.

Mock

Un oggetto simulato che riproduce il comportamento di un oggetto reale in modo controllato. I mock sono spesso utilizzati nei test di unità per sostituire le dipendenze esterne e garantire che il codice venga testato in isolamento.

Model (MVVM)

Nel pattern architetturale MVVM (Model-View-ViewModel), il Model rappresenta i dati e la logica di business dell'applicazione. Il Model è responsabile della gestione dei dati, della logica di business e delle operazioni di accesso ai dati, senza dipendere dalla View o dal ViewModel. Il Model fornisce i dati necessari alla View attraverso il ViewModel, che li prepara e li esporta in un formato adatto alla visualizzazione.

Modello di embedding

Vedi [Embedding \(modello di\)](#).

Modello di question answering

Vedi [Question answering \(Modello di\)](#).

MVC

MVC (Model-View-Controller) è un pattern architetturale che suddivide un'applicazione in tre componenti principali. Il Model gestisce i dati e la logica dell'applicazione, mentre la View si occupa della presentazione e dell'interfaccia utente. Il Controller, infine, funge da intermediario tra il Model e la View, gestendo le interazioni dell'utente e aggiornando i dati o la visualizzazione di conseguenza.

Minimum Viable Product (MVP)

Versione ridotta di un prodotto, che incorpora solo le funzioni essenziali per soddisfare le esigenze di base. Viene utilizzato per rilasciare un prodotto come test e ricevere feedback dall'utenza per migliorare poi il prodotto finito con tutte le funzionalità.

MVVM

MVVM (Model-View-ViewModel) è un pattern architetturale che separa l'applicazione in tre componenti. Il Model si occupa dei dati e della logica, mentre la View gestisce la visualizzazione. Il ViewModel, a differenza del Controller di MVC, prepara i dati in un formato che la View può legare facilmente, spesso tramite data binding, e non gestisce la logica di navigazione. Questo approccio è particolarmente utile nelle applicazioni che supportano il data binding, come nelle app mobili.

N

NestJS

Framework Node.js per lo sviluppo di applicazioni lato server, basato su TypeScript. Utilizza un'architettura modulare e concetti come iniezione delle dipendenze e decoratori, ispirandosi a framework come Angular, per creare applicazioni scalabili e facilmente mantenibili.

Next.js

Un framework open-source full stack basato su React, progettato per lo sviluppo di applicazioni web moderne. Consente la gestione sia del frontend sia del backend, grazie al supporto integrato per API e funzioni serverless.

Node.js

Runtime JavaScript open-source e multiplatforma, progettato per l'esecuzione di codice JavaScript lato server. Basato sul motore V8 di Google Chrome, permette di creare applicazioni veloci, scalabili e basate su eventi, sfruttando un modello non bloccante per la gestione delle operazioni I/O.

Norme di Progetto

Insieme di linee guida, procedure e regole stabilite per regolare e standardizzare l'approccio, il processo e l'output del lavoro all'interno del progetto. Queste norme possono riguardare diversi aspetti del progetto, come la gestione del codice, la documentazione, la comunicazione e la gestione dei rischi. L'obiettivo delle norme di progetto è promuovere la coerenza, la qualità e l'efficienza nel processo di sviluppo del software, consentendo al team di lavorare in modo più efficace e collaborativo.

O

Onboarding

Il processo di integrazione e formazione di un nuovo dipendente, collaboratore o utente in un'organizzazione o piattaforma. L'onboarding ha l'obiettivo di fornire tutte le informazioni e le risorse necessarie per adattarsi al nuovo ambiente di lavoro, comprendere la cultura aziendale, acquisire competenze specifiche e diventare operativi nel più breve tempo possibile.

OpenAI

OpenAI è un'organizzazione di ricerca e sviluppo nel campo dell'intelligenza artificiale, fondata nel 2015. Si dedica alla creazione di AI avanzate e sicure, con l'obiettivo di garantire che i benefici dell'intelligenza artificiale siano condivisi da tutta l'umanità. OpenAI è nota per lo sviluppo di modelli come GPT (Generative Pre-trained Transformer), ChatGPT e DALL-E, utilizzati in applicazioni come chatbot, creazione di contenuti, e generazione di immagini. Combina ricerca all'avanguardia con implementazioni pratiche per promuovere l'accessibilità e la sostenibilità dell'AI.

P

Pascal Case

È una convenzione di scrittura utilizzata nella programmazione per nominare variabili, funzioni, classi e altri identificatori. In Pascal Case, ogni parola della stringa inizia con una lettera maiuscola e non ci sono spazi o separatori tra le parole.

Pattern architetturale

Soluzione generale e riutilizzabile a un problema che si verifica comunemente nell'ambito dello sviluppo del software. I pattern architetturali forniscono un approccio testato e comprovato per la progettazione e l'implementazione del software.

Persistent logic

La logica di persistenza è la parte di un'applicazione che si occupa di salvare e recuperare i dati da un database o da un altro meccanismo di memorizzazione permanente. Questa logica è responsabile della gestione delle operazioni di lettura e scrittura dei dati, dell'interazione con il database e della sincronizzazione dei dati tra l'applicazione e lo storage persistente. Essa, nell'architettura esagonale, è rappresentata da un repository, che riceve un input un oggetto Entity dall'adapter, che lo aveva convertito partendo da un tipo di business logic, e lo salva nella piattaforma di storage collegata. In modo speculare, quando viene richiesto, il repository preleva i dati dalla piattaforma di persistenza, li converte in oggetti Entity e li restituisce all'Adapter, che si occupa di convertirli in tipi di business logic e di fornirli all'applicazione.

Piano di Progetto

Documento formale che delinea in dettaglio la pianificazione, l'esecuzione, il monitoraggio e il controllo di tutte le attività coinvolte nella realizzazione di un progetto. Questo documento fornisce una roadmap chiara e organizzata, comprensiva di obiettivi, risorse, scadenze e strategie di gestione dei rischi. Essenziale per la gestione efficace di un progetto, il piano di progetto serve come guida per il team di lavoro e gli stakeholder, fornendo una struttura che facilita il coordinamento delle attività e l'assegnazione delle risorse.

Piano di Qualifica

Documento che stabilisce gli standard di qualità, i processi e le attività di testing che saranno implementati durante lo sviluppo di un progetto. Contiene una descrizione dettagliata delle strategie di testing, delle metriche di valutazione e dei criteri di accettazione del prodotto finale. L'obiettivo principale del Piano di Qualifica è garantire che il prodotto soddisfi gli standard di qualità prefissati e che il processo di sviluppo segua procedure coerenti ed efficaci.

Piattaforma

In ambito informatico, un ambiente software che supporta l'esecuzione di applicazioni o servizi.

Pinecone

Pinecone è una piattaforma cloud-native progettata per l'indicizzazione e il retrieval di dati vettoriali. Consente di costruire e gestire motori di ricerca vettoriale altamente scalabili, spesso utilizzati in applicazioni di intelligenza artificiale come motori di raccomandazione, ricerca semantica e assistenti virtuali. Pinecone offre un'infrastruttura ottimizzata per il retrieval basato su similarità, supportando scenari di alto volume e garantendo basse latenze, con un'interfaccia API intuitiva per integrarsi facilmente con applicazioni esistenti.

Polling REST

Il polling è una tecnica utilizzata nei servizi web RESTful per ottenere aggiornamenti da un server. In pratica, un client effettua periodicamente richieste HTTP al server per verificare se sono disponibili nuovi dati o eventi. Il polling può essere implementato in vari modi, come polling periodico, polling a intervalli regolari o polling basato su eventi. Sebbene il polling sia semplice da implementare, può comportare un consumo elevato di risorse e una latenza aggiuntiva, poiché richiede al client di inviare continuamente richieste al server per ottenere aggiornamenti.

Port

Interfaccia che definisce i punti di comunicazione tra la logica di business e il mondo esterno. Permette ai Service di interagire con sistemi di persistenza e altre componenti senza dipendere direttamente dalla loro implementazione.

Postgres (PostgreSQL)

PostgreSQL (o Postgres) è un sistema di gestione di database relazionali open-source, noto per la sua stabilità, flessibilità e conformità agli standard SQL. Supporta funzionalità avanzate come transazioni ACID, indicizzazione full-text, query JSONB per dati semi-strutturati e estensibilità tramite funzioni definite dall'utente. Postgres è ampiamente utilizzato in applicazioni che richiedono scalabilità, sicurezza e gestione efficiente di dati complessi, ed è compatibile con molte estensioni per database vettoriali e applicazioni di machine learning.

Preventivo

Preventivo

Stima dei costi e delle risorse necessarie per completare un determinato lavoro o progetto.

Processo

Processo

Insieme di attività correlate e coese che trasformano ingressi (bisogni) in uscite (prodotti) secondo regole date, consumando risorse nel farlo.

Prodotto software

Un prodotto software è un'applicazione, un sistema o un programma informatico risultante dal processo di sviluppo del software. In altre parole, è il risultato tangibile e funzionante di attività di progettazione, sviluppo, test e manutenzione svolte dal team di sviluppo.

Product Baseline (PB)

È la seconda revisione di avanzamento del progetto didattico. Comprende un prodotto software con design definitivo, chiamato *Minimum Viable Product (MVP)*.

Progettazione

Attività nel processo di sviluppo che mira a definire l'architettura di un prodotto in grado di soddisfare le esigenze di tutti gli stakeholder. Questa fase fornisce documentazione dettagliata sulla struttura del prodotto, specifiche tecniche e scelte tecnologiche adottate per la sua realizzazione.

Progettazione logica

Fase del processo di progettazione del software in cui si definiscono le specifiche tecniche e funzionali del sistema, senza considerare l'implementazione dettagliata. La progettazione logica si concentra sulla struttura interna del sistema, definendo le componenti, i moduli e le interfacce necessarie per realizzare i requisiti identificati nella progettazione concettuale.

Progettazione di dettaglio

Fase del processo di progettazione del software in cui si definiscono in modo dettagliato le specifiche tecniche e funzionali del sistema, considerando l'implementazione concreta delle componenti e dei moduli. La progettazione di dettaglio si concentra sui singoli aspetti del sistema, definendo le interfacce, le classi, i metodi e le relazioni tra le varie parti del software.

Proof of Concept (PoC)

Versione preliminare di un'applicazione o di una soluzione software che viene sviluppata per dimostrare la fattibilità tecnica di un'idea o di un concetto. Viene utilizzata per testare rapidamente l'efficacia di un approccio, identificare eventuali limitazioni delle tecnologie scelte e valutare se l'idea può essere realizzata in modo pratico.

Prompt

Si riferisce al testo o alla richiesta che appare su un'interfaccia, chiedendo all'utente di fornire un input.

Proponente

Nel contesto dell'ingegneria del software, colui che presenta un'idea, un progetto o una proposta e ne sostiene la realizzazione. Il gruppo *SWEg Labs* ha come proponente l'azienda *AzzurroDigitale*.

Pull request

Pull request

Nel sistema di versionamento integrato da GitHub, operazione che permette di proporre modifiche al codice sorgente di un progetto. Una pull request consente agli sviluppatori di richiedere l'integrazione delle loro modifiche nel branch principale del repository, facilitando la revisione e la collaborazione.

PyLinit

Pylint è uno dei più diffusi linters per il linguaggio Python, utilizzato per rilevare errori nel codice, garantire aderenza alle convenzioni di stile (come PEP 8) e migliorare la qualità e la manutenibilità del codice.

Pytest

Pytest è un framework di testing per il linguaggio di programmazione Python. È progettato per essere semplice da usare e permette di scrivere test in modo chiaro e conciso. Pytest supporta test di unità, test funzionali e test di integrazione, offrendo funzionalità avanzate come l'assertion introspection, la parametrizzazione dei test e l'integrazione con altri strumenti di testing e di sviluppo.

Python

Un linguaggio di programmazione open-source, interpretato, dinamico e multi-paradigma, ampiamente utilizzato per lo sviluppo di software, applicazioni web, analisi dati, intelligenza artificiale e automazione. Creato da Guido van Rossum nel 1991, si distingue per la sua sintassi semplice e leggibile, che favorisce la produttività e la manutenzione del codice. Supporta programmazione procedurale, orientata agli oggetti e funzionale. Dotato di una vasta libreria standard e di un ecosistema ricco di framework e pacchetti, è adatto sia a principianti sia a sviluppatori esperti per applicazioni front-end e back-end.

Python Crontab

Python Crontab è un modulo Python che semplifica la gestione dei crontab. Un crontab è un file di configurazione utilizzato nei sistemi operativi Unix-like (come Linux e macOS) per pianificare l'esecuzione di comandi a intervalli di tempo regolari. In pratica, un crontab è una tabella (tabella dei comandi) che indica al sistema operativo quali comandi eseguire e a quale ora.

Q

Qdrant

Qdrant è un motore open-source progettato per la gestione e la ricerca di dati vettoriali. Ottimizzato per il nearest neighbor search, Qdrant è utilizzato per applicazioni basate su similarità, come motori di ricerca semantica, raccomandazioni e clustering. Fornisce un'API RESTful, supporta query scalabili in tempo reale e integra funzionalità per filtrare e classificare dati multidimensionali. La sua architettura è progettata per essere facilmente scalabile e per garantire alte prestazioni con dataset di grandi dimensioni.

Qualità

Insieme delle caratteristiche di un'entità che ne determinano la capacità di soddisfare esigenze sia espresse che implicite. Si parla di qualità del prodotto software in termini di:

- **Qualità Intrinseca:** conformità ai requisiti, idoneità all'uso;
- **Qualità Relativa:** soddisfazione del cliente;
- **Qualità Quantitativa:** misurazione oggettiva, per confronto.

Question answering (Modello di)

Un sistema di intelligenza artificiale progettato per rispondere a domande poste in linguaggio naturale. Nell'ambito dell'ingegneria del software e dell'elaborazione del linguaggio naturale (NLP), un modello di question answering utilizza tecniche avanzate di machine learning e deep learning per comprendere il contesto della domanda e fornire risposte accurate e pertinenti. Questi modelli possono essere addestrati su grandi quantità di dati testuali e sono in grado di estrarre informazioni da documenti, articoli, database e altre fonti di conoscenza per rispondere alle domande degli utenti.

R

Refactoring

Il processo di riorganizzazione del codice sorgente di un programma senza modificarne il comportamento esterno. L'obiettivo è migliorare la leggibilità, la manutenibilità e la struttura del codice, senza alterarne la funzionalità.

Repository

In termini informatici, un luogo o un archivio dove vengono conservati e gestiti dati, documenti o, nel contesto del software, il codice sorgente di un progetto. Nell'ambito dei sistemi di controllo delle versioni come Git, un repository è una struttura dati che archivia anche la cronologia completa delle modifiche apportate al codice sorgente di un progetto.

Repository (Architettura Esagonale)

Nell'architettura esagonale, il repository è un componente che si occupa della persistenza dei dati, fornendo un'interfaccia per l'accesso ai dati e nascondendo i dettagli della loro implementazione. Il repository è responsabile di recuperare, salvare e aggiornare i dati da e verso una piattaforma di persistenza, come un database o un servizio esterno. Questo componente è essenziale per separare la logica di business dalla logica di persistenza, garantendo che le operazioni di accesso ai dati siano incapsulate e gestite in modo indipendente dal resto dell'applicazione. Il repository riceve oggetti Entity dall'adapter, li converte in un formato compatibile con la piattaforma di persistenza e li salva o li recupera secondo le richieste dell'applicazione. In caso di recupero di dati, il repository li fornisce all'adapter come Entity, e poi l'adapter li convertirà in tipi di dato di business. In questo modo, il repository fornisce un'interfaccia uniforme per l'accesso ai dati, indipendentemente dal tipo di storage utilizzato, semplificando la gestione e l'evoluzione della persistenza dei dati.

Requirement and Technology Retrospective (RTB)

La prima revisione di avanzamento del progetto didattico. Fissa i requisiti da soddisfare in accordo con il proponente, motiva le tecnologie, i framework e le librerie adottate dimostrandone adeguatezza e compatibilità tramite il Proof of Concept (PoC).

Requisiti software

I requisiti software sono le specifiche che definiscono le funzionalità, le prestazioni, i vincoli e le caratteristiche che un sistema software deve soddisfare per rispondere alle esigenze degli utenti e degli stakeholder.

Retrospettiva

Vedi [Sprint Retrospective](#).

S

Scalabilità

La scalabilità è la capacità di un sistema, di un'applicazione o di un'infrastruttura di crescere e adattarsi in modo flessibile e affidabile per gestire un aumento della domanda o del carico di lavoro. La scalabilità può essere orizzontale (aggiungendo più risorse o nodi) o verticale (aggiungendo più potenza di elaborazione o risorse a un singolo nodo) e può essere progettata per essere elastica, automatica o manuale in base alle esigenze del sistema.

Scenario alternativo

Uno scenario alternativo nei casi d'uso è una sequenza di azioni o eventi che deviano dal flusso principale a causa di condizioni o circostanze specifiche. Questi scenari descrivono come il sistema dovrebbe comportarsi quando si verificano situazioni diverse rispetto al percorso standard, garantendo che tutte le possibilità siano considerate nel processo di progettazione.

Scenario d'uso (scenario principale)

Uno scenario d'uso è una descrizione narrativa di come un utente interagisce con un sistema software per raggiungere un determinato obiettivo. Rappresenta una sequenza di azioni e reazioni tra l'utente e il sistema in un contesto specifico e fornisce una visione dettagliata del comportamento atteso del software in una situazione reale.

Scheduler

Uno scheduler è un componente software che gestisce l'esecuzione di processi, attività o operazioni in un sistema informatico. Può essere utilizzato per programmare l'avvio e la terminazione di processi, assegnare risorse, gestire la priorità delle attività e garantire che le operazioni vengano eseguite in modo efficiente e sincronizzato. Gli scheduler sono ampiamente utilizzati nei sistemi operativi, nei database, nei server web e in altre applicazioni per gestire il tempo e le risorse in modo ottimale.

Script

Uno script è un insieme di istruzioni scritte in un linguaggio di scripting (come Python, JavaScript, Bash, ecc.) progettato per automatizzare compiti specifici o per eseguire operazioni in modo diretto su un sistema, applicazione o ambiente web. A differenza di un programma complesso, uno script tende a essere più leggero e adatto per operazioni mirate e ripetibili, senza richiedere una fase di compilazione lunga o complessa.

Service

Un service è un componente software che fornisce funzionalità specifiche o servizi a un'applicazione o a un sistema. I service sono progettati per essere riutilizzabili, modulari e indipendenti dal contesto, consentendo di separare la logica di business, l'accesso ai dati e altre funzionalità in unità distinte e ben definite. I service sono ampiamente utilizzati nell'architettura orientata ai servizi (SOA), nei microservizi e in altri approcci di progettazione software per promuovere la modularità, la scalabilità e la manutenibilità del codice.

Similarità

La similarità nell'apprendimento automatico indica quanto due o più elementi (immagini, testi, numeri) sono simili tra loro. Questa misura, fondamentale per raggruppare dati simili, classificare nuovi dati e fare raccomandazioni, si basa spesso sul calcolo della similarità tra vettori che rappresentano numericamente gli elementi. Esistono diverse misure di similarità, ciascuna adatta a specifici tipi di dati e problemi.

Singleton

Il Singleton è un design pattern creazionale che garantisce che una classe abbia una sola istanza e fornisce un punto di accesso globale a tale istanza. Questo pattern è utile quando si desidera limitare il numero di istanze di una classe e fornire un modo per accedere a quella singola istanza da qualsiasi parte del programma. Il Singleton è ampiamente utilizzato per la gestione delle risorse condivise, la configurazione globale e la creazione di oggetti unici in un'applicazione.

Slack

Piattaforma di messaggistica istantanea e collaborazione progettata per team e organizzazioni. Permette di comunicare tramite canali tematici, inviare messaggi diretti, condividere file, e integrare altre applicazioni e strumenti di lavoro. Slack favorisce la comunicazione in tempo reale, migliorando la produttività e la gestione dei progetti, e viene utilizzato soprattutto in ambienti di lavoro agili e collaborativi.

Snake Case

Lo Snake Case è uno stile di scrittura di nomi in cui le parole sono separate da un trattino basso o underscore (-) e tutte le lettere sono generalmente minuscole (es.: nome_variabile_uno). Viene spesso contrapposto allo stile Camel Case, in cui invece le parole successive alla prima vengono concatenate con l'iniziale maiuscola (es.: nomeVariabileUno).

Snippet

Uno snippet è un piccolo blocco di codice riutilizzabile che svolge una funzione specifica o rappresenta un esempio di utilizzo di una determinata funzionalità. Gli snippet sono utilizzati per risparmiare tempo durante la scrittura del codice, poiché forniscono pezzi di codice già pronti che possono essere facilmente inseriti e adattati nel contesto del progetto.

Sottocaso d'uso

Un Sottocaso d'uso è una specificazione di un caso d'uso che descrive un flusso alternativo o dettagliato di azioni, mantenendo una relazione diretta con il caso d'uso principale. Viene utilizzato per suddividere comportamenti complessi in scenari più gestibili, evidenziando variazioni o estensioni del flusso principale, pur rimanendo parte integrante del caso d'uso generale.

Specifica tecnica

Si tratta di un documento formale con lo scopo di servire da linea guida per gli sviluppatori che andranno ad estendere o mantenere il prodotto: al suo interno si trovano tutte le informazioni riguardanti i linguaggi e le tecnologie utilizzate, l'architettura del sistema e le scelte progettuali effettuate per il prodotto.

Specifiche funzionali

Specifiche funzionali

Le specifiche funzionali sono una descrizione dettagliata delle funzionalità e dei comportamenti che un sistema software deve avere per soddisfare i requisiti degli utenti. Le specifiche funzionali definiscono cosa il software deve fare, come deve rispondere a determinate azioni e quali risultati deve produrre in risposta a determinati input.

Specifiche tecniche

Specifiche tecniche

Le specifiche tecniche sono una descrizione dettagliata delle caratteristiche tecniche e delle prestazioni che un sistema software deve avere per soddisfare i requisiti tecnici e funzionali. Le specifiche tecniche definiscono come il software deve essere progettato, implementato e testato per garantire che funzioni correttamente e risponda alle esigenze degli utenti.

Spring

Framework Java che fornisce un ecosistema modulare per sviluppare applicazioni lato server. Facilita la gestione delle dipendenze, configurazioni, sicurezza, accesso ai dati e integrazioni, promuovendo un'architettura scalabile e flessibile basata su principi come l'inversione del controllo (IoC) e la programmazione orientata agli aspetti (AOP).

Spring Boot

Framework Java che semplifica lo sviluppo di applicazioni Spring, ovvero applicazioni basate sul framework Spring per la gestione di componenti, dipendenze e funzionalità lato server come sicurezza, accesso ai dati e integrazione. Spring Boot offre configurazioni predefinite, un server embedded e una struttura modulare, permettendo di creare applicazioni pronte all'uso con meno configurazioni manuali.

Sonarqube for IDE

Estensione che integra le funzionalità di analisi della qualità del codice direttamente all'interno degli ambienti di sviluppo (IDE) più diffusi. Questa integrazione permette agli sviluppatori di rilevare problemi nel codice (come bug, vulnerabilità, code smells e problemi di mantenibilità) in tempo reale.

Sprint

Periodo di tempo prefissato entro il quale lavorare producendo dei risultati documentati. Gli sprint sono al centro delle metodologie Agile, atte a produrre risultati piccoli e in maniera costante.

Sprint Retrospective

Incontro finalizzato ad analizzare l'andamento dello sprint quando questo è terminato, per migliorare la performance futura del team di sviluppo. La riunione retrospettiva dello sprint è quindi propedeutica allo sprint successivo.

SQL

Acronimo di Structured Query Language, linguaggio di programmazione utilizzato per la gestione dei database relazionali. Con SQL è possibile creare, modificare e interrogare database, definire tabelle, inserire, modificare ed eliminare dati, e gestire le autorizzazioni di accesso. SQL è uno standard de facto per la gestione dei dati e viene utilizzato in una vasta gamma di applicazioni e sistemi informatici.

Stakeholder

Individui, gruppi o entità che hanno un interesse o un coinvolgimento in un progetto, un'organizzazione o un'iniziativa specifica. Essi possono influenzare o essere influenzati dalle attività e dalle decisioni associate al progetto o all'organizzazione. La gestione degli stakeholder è una parte fondamentale del processo di pianificazione ed esecuzione di progetti ed è essenziale per il successo complessivo di un'iniziativa.

Streamlit

Streamlit è un framework open-source in Python progettato per creare applicazioni web interattive e data-driven in modo rapido e intuitivo. È utilizzato principalmente per sviluppare dashboard e strumenti di visualizzazione dati, grazie a una sintassi semplice che permette di trasformare script Python in applicazioni web complete con poche righe di codice. Streamlit è particolarmente apprezzato nella comunità scientifica e tra i data scientist per la sua capacità di integrare facilmente grafici, widget interattivi e modelli di machine learning in un'interfaccia user-friendly.

Stub

Uno stub è un componente software che simula il comportamento di un modulo o di una funzionalità specifica in un sistema, al fine di testare altre parti del software o di verificare l'interazione tra i componenti. Gli stub vengono utilizzati per sostituire temporaneamente i moduli non ancora implementati o per simulare condizioni specifiche, consentendo di testare il software in modo isolato e controllato.

Supabase

Supabase è una piattaforma open-source che fornisce un backend completo per applicazioni web e mobili, simile a Firebase. Basato su PostgreSQL, offre un database relazionale, un sistema di autenticazione, un'API RESTful generata automaticamente, storage per file e supporto per database vettoriali. Questa funzionalità consente di gestire embedding e lavorare con dati multidimensionali, rendendolo adatto ad applicazioni di machine learning e intelligenza artificiale. Supabase è pensato per semplificare lo sviluppo di applicazioni, fornendo strumenti pronti all'uso per gestire dati, utenti e file, con la possibilità di eseguire query in tempo reale. È molto apprezzato per la sua flessibilità, l'integrazione con standard open-source e la facilità d'uso.

T

Telegram

Applicazione multiplatforma che permette una facile comunicazione con gruppi/canali, organizzando la comunicazione sulla base di strumenti e funzionalità automatiche offerte (bot), condividendo facilmente file e messaggi in un canale unico di comunicazione.

Test

In ingegneria del software, processo sistematico e controllato per valutare un sistema software o una sua componente allo scopo di individuare eventuali difetti, errori o comportamenti indesiderati.

Test di integrazione

Fase del processo di testing software in cui i singoli moduli o componenti di un'applicazione vengono combinati e testati come un gruppo. L'obiettivo è verificare che le diverse parti del sistema interagiscano correttamente e che l'intera applicazione funzioni come previsto. Questo tipo di test è cruciale per identificare e risolvere problemi di interoperabilità tra le componenti e garantire la coesione del sistema nel suo complesso.

Test di unità

Pratica dell'ingegneria del software che aiuta a garantire la qualità e l'affidabilità del codice attraverso la verifica delle singole unità. Tali unità si riferiscono alle parti più piccole e atomiche del software, come funzioni, metodi, classi o moduli. L'obiettivo principale del test di unità è isolare e testare ogni singola unità di codice in modo indipendente dagli altri componenti del sistema. Questo consente agli sviluppatori di individuare e risolvere eventuali difetti o bug in modo più efficiente e tempestivo.

TDD

Il Test-Driven Development (TDD) è una metodologia di sviluppo software in cui i test vengono scritti prima del codice di produzione. Il processo segue un ciclo iterativo di tre fasi: scrittura di un test fallimentare, sviluppo del codice minimo per superarlo e successiva refactorizzazione per migliorare la qualità del codice.

Token

Un token è una sequenza di caratteri che rappresenta un'unità di testo significativa per un modello di linguaggio (LLM). I token sono utilizzati per suddividere un messaggio di testo in parti più piccole e gestibili, come parole, sottoparole o caratteri, a seconda del modello. Ogni LLM ha un limite massimo di token che può elaborare in un singolo input, il che influisce sulla lunghezza massima del testo che può essere processato. La gestione dei token è cruciale per garantire che il testo rientri nei limiti del modello e per ottimizzare le prestazioni e l'accuratezza del modello stesso.

Tracciamento

Il tracciamento è il processo di monitoraggio e documentazione dell'evoluzione e delle modifiche di un sistema, dei suoi requisiti e dei suoi componenti, all'interno di un progetto. Questo processo è essenziale per mantenere una visione chiara dello stato attuale del sistema, delle sue modifiche, e delle relazioni tra diversi elementi (come requisiti, attività di design, e componenti di codice).

Trigger

Un trigger (dall'inglese "grilletto") è un meccanismo che innesca automaticamente un'azione o un evento in risposta a un particolare stimolo o condizione. Nell'ambito informatico, un trigger è una procedura predefinita che viene eseguita in modo automatico quando si verifica un evento specifico.

Txtai

Txtai è una piattaforma open-source per la ricerca semantica e l'indicizzazione di dati basati su embedding. Fornisce strumenti per creare motori di ricerca, sistemi di domande e risposte, classificazione di documenti e clustering, utilizzando modelli di machine learning. Txtai supporta vari formati di dati, integra API intuitive e può essere facilmente utilizzato per costruire applicazioni di AI scalabili.

TypeScript

TypeScript è un linguaggio di programmazione open-source sviluppato da Microsoft, basato su JavaScript ma con l'aggiunta di un sistema di tipi statici. Progettato per migliorare la scalabilità e la manutenibilità del codice, TypeScript permette di rilevare errori durante la scrittura del codice grazie alla tipizzazione, mantenendo la compatibilità con il runtime di JavaScript. È particolarmente utilizzato in applicazioni web e progetti di grandi dimensioni, offrendo una migliore esperienza di sviluppo e strumenti avanzati come il completamento automatico e il refactoring.

U

UML, diagramma

Vedi Diagramma UML.

Use Case

Unità logica che rappresenta un caso d'uso specifico del sistema. Definisce un'operazione di business e viene implementato da un Service per eseguire la logica di business associata.

V

Verifica

Processo che include un insieme di attività volte a garantire che il lavoro svolto durante lo sviluppo del software rispetti gli standard, i requisiti e le aspettative stabilite. La verifica è essenziale per garantire che il software sia di alta qualità, risponda alle esigenze degli utenti e riduca il rischio di errori e difetti. Si svolge a periodi regolari durante il corso del progetto.

Versionamento

Processo che realizza il cosiddetto “controllo di versione”, stabilendo la storia cronologica delle azioni fatte per una certa attività, tracciando i cambiamenti occorsi e permettendo di tornare a uno stadio precedente qualora necessario.

View (MVVM)

Nel pattern architetturale MVVM (Model-View-ViewModel), la View rappresenta l'interfaccia utente dell'applicazione. La View è responsabile della presentazione dei dati e della gestione delle interazioni dell'utente, senza dipendere direttamente dal Model o dal ViewModel. La View riceve i dati dal ViewModel e li visualizza in modo appropriato per l'utente, gestendo anche gli eventi e le azioni dell'utente.

ViewModel

Nel pattern architetturale MVVM (Model-View-ViewModel), il ViewModel funge da intermediario tra il Model e la View. Il ViewModel è responsabile di preparare i dati del Model in un formato adatto alla visualizzazione nella View, gestendo anche la logica di presentazione e le interazioni dell'utente. Il ViewModel non contiene la logica di business o l'accesso ai dati, ma si concentra sulla presentazione e sulla gestione delle interazioni dell'utente.

Visual Studio Code (VSC)

IDE libero e gratuito sviluppato da Microsoft per Windows, Linux e macOS. Permette sia di scrivere codice sorgente per un prodotto software sia di scrivere testo di documentazione.

W

Way of working

Il termine “Way of working” (modo di lavorare) si riferisce al modo in cui un individuo, un team o un’organizzazione svolge le proprie attività lavorative. Questo concetto può includere processi, metodologie, abitudini, strumenti e culture aziendali che influenzano la gestione del lavoro e la collaborazione. Un modo di lavorare efficace ed efficiente può migliorare la produttività, la qualità del lavoro e la soddisfazione dei membri del team, promuovendo un ambiente di lavoro positivo e collaborativo.

Il way of working deve essere:

- *Sistematico*, cioè non deve dipendere dalle singole persone, bensì deve essere adattabile anche a futuri membri del team.
- *Disciplinato*, cioè deve far seguire norme di comportamento che blocchino l’istinto e favoriscano il ragionamento.
- *Quantificabile*, cioè si deve poter verificare concretamente la sua implementazione.

Web server

Un web server è un’applicazione software che, in esecuzione su un computer dedicato o virtuale, riceve richieste HTTP da client (solitamente browser web) e restituisce le risorse richieste, come pagine HTML, file CSS, JavaScript, immagini e altri asset digitali. Opera come un endpoint di rete, ascoltando su una specifica porta TCP (tipicamente la 80 o la 443 per HTTPS) per le connessioni in ingresso.

WebSocket

WebSocket è una tecnologia che fornisce un canale di comunicazione bidirezionale e full-duplex su una singola connessione TCP. È progettato per essere implementato sia dai browser che dai server web, consentendo una comunicazione in tempo reale tra client e server. WebSocket è spesso utilizzato per applicazioni web che richiedono aggiornamenti in tempo reale o comunicazione bidirezionale, come chatbot, giochi online, applicazioni di collaborazione e dashboard di monitoraggio.

Whisper.ai

Sistema di riconoscimento automatico del parlato (ASR) sviluppato da OpenAI. È progettato per trascrivere il linguaggio parlato in testo con alta precisione.

X

...

Y

...

Z

...