

Manuale di Installazione

C04 – Esistere



Riferimento	C04_Esistere_MDI_V1.0
Versione	1.0
Data	15/01/2024
Destinatario	Prof.ssa Filomena Ferrucci, Prof.re Fabio Palomba
Presentato da	C04 Team Esistere: <ul style="list-style-type: none">• Antonio D'Auria (AA)• Luca Casillo (LC)• Maria Giovanna Della Pietra (MGP)• Ogham If Dell'Erba (OE)• Raffaele Forte (RF)• Rosa Carotenuto (RC)• Valentino Dragone (VD)
Approvato da	Alessandra Parziale, Saverio Napolitano



Esistere

Per poter abbracciare ogni istante.

Revision History

Data	Versione	Descrizione	Autori
15/01/2024	1.0.0	Creazione Manuale di Installazione	AA, LC, MGP, OE, RC, VD



Esistere

Per poter abbracciare ogni istante.

Team Members

Nome	Ruolo nel progetto	Acronimo	Informazione di controllo
Alessandra Parziale	Project Manager	AP	a.parziale8@studenti.unisa.it
Saverio Napolitano	Project Manager	SN	s.napolitano44@studenti.unisa.it
Antonio D'Auria	Team Member	AA	a.dauria123@studenti.unisa.it
Luca Casillo	Team Member	LC	l.casillo16@studenti.unisa.it
Maria Giovanna Della Pietra	Team Member	MGP	m.dellapietra10@studenti.unisa.it
Rosa Carotenuto	Team Member	RC	r.carotenuto16@studenti.unisa.it
Ogham If Dell'Erba	Team Member	OE	o.dellerba@studenti.unisa.it
Valentino Dragone	Team Member	VD	v.dragone5@studenti.unisa.it
Raffaele Forte	Team Member	RF	r.forte12@studenti.unisa.it



Esistere

Per poter abbracciare ogni istante.

Sommario

REVISION HISTORY	1
TEAM MEMBERS	2
SOMMARIO	3
1. INTRODUZIONE	4
1.1 SCOPO DEL SISTEMA	4
1.2 SCOPO DEL DOCUMENTO	4
2. PREREQUISITI PER L'INSTALLAZIONE	4
2.1 APPLICAZIONE WEB	5
2.1 DATABASE	5
3. INSTALLAZIONE APPLICAZIONE WEB	5
4. INSTALLAZIONE DEL DATABASE	6
5. TEST DEL PROGETTO	7
<i>Guida ai Test di Unità con Jest (Back-End)</i>	7
<i>Guida ai Test di Sistema con Nightwatch (Front-End)</i>	7
6. GLOSSARIO	8



1. Introduzione

1.1 Scopo del sistema

Il sistema che si vuole realizzare ha l'obiettivo di creare una piattaforma che consenta un collegamento user-friendly tra medico e caregiver di pazienti affetti da Alzheimer.

Tramite tale piattaforma i medici potranno seguire i pazienti in tempo reale e in maniera più semplice ed efficace, con l'aggiunta di poter assegnare, in maniera esclusiva e personalizzata, attività e compiti da svolgere.

In maniera particolare il sistema si concentra sui seguenti aspetti:

- Facilitare il lavoro dei medici, mettendogli a disposizione una gestione accessibile e centralizzata delle informazioni dei propri pazienti, fornendo monitoraggio in tempo reale.
- Aiutare i familiari fornendogli strumenti per l'interfacciamento con il medico e il paziente.
- Fornire al paziente un supporto più concreto e umano tramite funzionalità che possano aiutarlo nella sua routine e durante la terapia.

1.2 Scopo del documento

Questo documento è stato creato con l'intento di fornire istruzioni dettagliate e una guida passo passo per l'installazione e la configurazione del progetto. Il manuale si rivolge principalmente agli sviluppatori software, agli amministratori di sistema e a tutti coloro che necessitano di implementare e gestire l'applicazione nel loro ambiente di sviluppo, test o produzione.

2. Prerequisiti per l'installazione

I prerequisiti necessari per l'installazione di Esistere sono:

- **Node.js e npm:**
 - Node.js [20.10.0] o superiore.
 - npm (Node Package Manager) [10.2.3] o superiore
 - Verificare l'installazione eseguendo `node -v` e `npm -v` nel terminale
- **Browser web:**
 - Mozilla Firefox
 - Google Chrome
 - Microsoft Edge
 - Opera
 - Safari
- **Spazio su Disco:**
 - Almeno 2GB di spazio libero



Esistere

Per poter abbracciare ogni istante.

2.1 Applicazione Web

Esistere è un'applicazione Web in React con TypeScript e raggruppa diversi tipi di file:

Lato back-end:

- Adapter: design pattern adapter per convertire i dati nel formato adatto (.ts)
- Dao: data access object per l'accesso al database (.ts)
- Entity: dto del sistema (.ts)
- Routes: gestione delle richieste (.ts)
- Services: strato di servizio tra le routes e i dao (.ts)
- Test: test di unità (.ts)

Lato front-end:

- Components: componenti React (.tsx)
- Control: control per reperire i dati dal Web Server (.ts)
- Css: fogli di stile (.css)
- Interfaces: interfacce (.ts)
- Test: test di sistema (.ts)

2.1 Database

Il Database usato per l'applicazione è PostgreSQL, un avanzato sistema di gestione di database relazionali a oggetti. Scelto per la sua affidabilità, robustezza e compatibilità con gli standard SQL, PostgreSQL offre una soluzione ottimale per la gestione efficace e sicura dei dati dell'applicazione.

3. Installazione applicazione web

Dopo aver reperito i progetti (<https://github.com/Esistere/>), il primo passo per l'installazione, per entrambi, è quello di installare le dipendenze necessarie. All'interno del root di ciascun progetto scrivere su terminale il comando: `npm install`

Fatto ciò, è possibile procedere alle configurazioni specifiche.

Lato back-end:

Per l'installazione del lato back-end è necessario un server per effettuare l'hosting con Express (Web framework Node.js scelto). Si può quindi procedere a effettuare il port-forwarding sulla porta desiderata, nel caso di Esistere è stata scelta la porta 3001 (configurabile dal file `server.ts`). L'applicazione necessita inoltre della configurazione delle variabili di ambiente e della configurazione dei certificati per il SSL.

- Configurazione delle variabili di ambiente e del database:
 - Nel root del progetto creare un file `.env` che manterrà le variabili necessarie al funzionamento del progetto senza esporle nel codice sorgente.



Esistere

Per poter abbracciare ogni istante.

- Per il Database è necessario definire le seguenti variabili:
 - POSTGRES_USER: nome dell'utente Postgres
 - POSTGRES_HOST: host sul quale è in esecuzione il database
 - POSTGRES_DATABASE: nome del database
 - POSTGRES_PASSWORD: password del database
 - POSTGRES_PORT: porta del database (5432, default per Postgres)
- Per l'autenticazione è necessario definire la seguente variabile:
 - SECRET_KEY: stringa utilizzata per la crittografia del token JWT, utilizzato per l'autenticazione
- Configurazione SSL per HTTPS:
 - Il server utilizza HTTPS, ci si assicuri di possedere la chiave e il certificato. Questi due file devono trovarsi nella stessa cartella del file principale (server.ts) e hanno estensione. pem. Nel caso si vogliano spostare i file altrove, all'interno di server.ts è possibile definire i percorsi

Lato front-end:

L'installazione del lato front-end non necessita di requisiti particolari se non l'installazione delle dipendenze precedentemente menzionate. Si noti tuttavia che nella cartella app è presente un file di configurazione (config.ts) che fa riferimento all'indirizzo del Web Server, modificare tali riferimenti e adattarli al proprio Web Server.

4. Installazione del database

Per supportare l'applicazione è necessario installare il database relazionale PostgreSQL, Questa sezione fornisce una guida dettagliata per l'installazione e la configurazione di PostgreSQL su diverse piattaforme:

Per Windows:

1. **Download:** Andare sul sito ufficiale di PostgreSQL ([postgresql.org](https://www.postgresql.org)) e scaricare l'ultima versione dell'installer per Windows.
2. **Installazione:** Eseguire l'installer scaricato. Seguire la procedura guidata, selezionare i componenti da installare e specificare la cartella di installazione.
3. **Configurazione Iniziale:** Durante l'installazione, impostare una password per l'utente postgres (un account superuser predefinito).
4. **Completamento dell'Installazione:** Al termine dell'installazione, verificare che il servizio PostgreSQL sia attivo.

Per MacOS:

1. **Download:** Utilizzare Homebrew (un gestore di pacchetti per MacOS) eseguendo `brew install postgresql` nel terminale.
2. **Avvio del Servizio:** Dopo l'installazione, avviare PostgreSQL con `brew services start postgresql`.
3. **Verifica dell'Installazione:** Verificare che PostgreSQL sia stato installato correttamente eseguendo `postgres -v` nel terminale.

Per Linux (Debian/Ubuntu):



Esistere

Per poter abbracciare ogni istante.

1. **Installazione:** Aprire il terminale ed eseguire `sudo apt-get update` seguito da `sudo apt-get install postgresql postgresql-contrib`.
2. **Avvio del Servizio:** Assicurarsi che il servizio PostgreSQL sia avviato con `sudo service postgresql start`.
3. **Creazione di un Nuovo Ruolo:** Opzionalmente, creare un nuovo ruolo utente digitando `sudo -u postgres createuser --interactive`.

Una volta terminata l'installazione con successo, all'interno del progetto back-end, nella cartella db è presente il file di DDL del database dal quale è possibile reperire l'intera struttura del database.

5. Test del progetto

Il progetto presenta i test di unità nel lato back-end e i test di sistema nel lato front-end. Di seguito sono indicati i passaggi per configurare ed effettuare entrambi i tipi di test:

Guida ai Test di Unità con Jest (Back-End)

Jest: Un framework di test JavaScript focalizzato sulla semplicità.

1. **Installazione:** Già installato poiché presente nelle dipendenze
2. **Configurazione:** La configurazione è presente nel file `jest.config.ts` (presente nel root del progetto)
3. **Scrittura dei Test:** Crea eventualmente altri file di test con nomi che terminano in `test.ts` e inserirli all'interno della cartella `test`. Usa `describe` e `it` per organizzare i test ed `expect` per le asserzioni.
4. **Esecuzione dei Test:** Esegui i test con il comando `npm test`. Jest rileva automaticamente i file di test ed esegue i relativi casi.

Guida ai Test di Sistema con Nightwatch (Front-End)

Nightwatch: Uno strumento di testing end-to-end per applicazioni web.

1. **Installazione:** Già installato poiché presente nelle dipendenze
2. **Configurazione:** La configurazione è presente nel file `nightwatch.conf.ts` (presente nel root del progetto)
3. **Scrittura dei Test:** Scrivi eventualmente altri test di sistema creando file `.ts` nella cartella dei test. Usa comandi Nightwatch come `browser.url()`, `browser.waitForElementVisible()`, e `browser.assert` per interagire con la tua applicazione web.
4. **Esecuzione dei Test:** Avvia i test eseguendo `nightwatch` dalla riga di comando. Nightwatch aprirà il browser, eseguirà i test, e fornirà un report dei risultati.



6. Glossario

Sigla/Termine	Definizione
Esistere	Nome dell'applicativo che si andrà a realizzare.
Piattaforma	Base software o hardware su cui sono sviluppate o eseguite applicazioni.
Browser	Rappresentazione dell'intera categoria dei browser come motori di ricerca.
Web Server	Un software di sistema che ospita e gestisce i file di un sito web. Quando un browser web richiede una pagina da un server web attraverso il protocollo HTTP o HTTPS, il server web accetta la richiesta, trova il contenuto richiesto (se disponibile), e lo invia al browser.
HTTP (HyperText Transfer Protocol)	Il protocollo standard utilizzato per la trasmissione di documenti ipertestuali sul web, ovvero la base della comunicazione dati su Internet. HTTP definisce come i messaggi vengono formattati e trasmessi, e come i server web e i browser dovrebbero rispondere a vari comandi.
HTTPS (HyperText Transfer Protocol Secure)	Una versione di HTTP che utilizza la crittografia per aumentare la sicurezza. HTTPS crittografa i dati trasmessi, proteggendo così le informazioni da intercettazioni o alterazioni durante il loro viaggio attraverso la rete. È comunemente utilizzato per transazioni sicure come il banking online e gli acquisti online.
Node.js	Un ambiente di runtime JavaScript utilizzato per eseguire codice JavaScript sul lato server.
npm (Node Package Manager)	Uno strumento per la gestione di pacchetti in ambienti Node.js, facilitando la condivisione e la riutilizzo di codice.
PostgreSQL	Un sistema avanzato di gestione di database relazionali a oggetti, noto per la sua affidabilità e compatibilità con gli standard SQL.
React	Un framework JavaScript per costruire interfacce utente.
Typescript	Un linguaggio di programmazione JavaScript aggiungendo tipizzazione statica.
SSL (Secure Sockets Layer)	Una tecnologia standard di sicurezza per stabilire un collegamento crittografato tra un server web e un browser.
JWT (JSON Web Token)	Un metodo standard per la creazione di token di accesso che permettono la comunicazione sicura tra client e server.
Jest	Un framework di test per JavaScript, utilizzato prevalentemente per test di unità nel back-end. Jest è noto per la sua semplicità, velocità e integrabilità con progetti basati su React e Node.js.



Esistere

Per poter abbracciare ogni istante.

Nightwatch	Uno strumento di test end-to-end per applicazioni web basate su browser. Viene utilizzato per simulare interazioni dell'utente con l'interfaccia front-end, garantendo che i componenti della pagina web funzionino come previsto in un ambiente reale.
Express	Un framework web per Node.js, utilizzato per costruire applicazioni web e API. È noto per la sua flessibilità e semplicità, offrendo un robusto set di funzionalità per applicazioni web e mobile. Express semplifica il routing, la gestione delle richieste e delle risposte, e integra middleware per estendere le sue funzionalità.