

Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA “TULLIO LEVI-CIVITA”

CORSO DI LAUREA IN INFORMATICA



**MoviORDER, modulo agenti per gestione  
clienti**

*Tesi di Laurea Triennale*

*Relatore*

Prof. Vardanega Tullio

*Laureando*

Oseliero Antonio

Matricola 1226325

---

ANNO ACCADEMICO 2023-2024



# Sommario

Il presente documento descrive il lavoro svolto dallo studente Oseliero Antonio durante il suo *stage* presso l'azienda VisioneImpresa Software House.

Lo scopo del tirocinio è stato studiare il codice dell'applicazione mobile MoviORDER e sviluppare un modulo di autenticazione di agenti aziendali in un'applicazione pensata per clienti terzi. Più precisamente era richiesto che, dopo l'autenticazione, l'agente possa scegliere un cliente da una lista e operare all'interno dell'applicazione come il cliente selezionato senza la necessità di autenticarsi come tale.

Raggiungere questo obiettivo ha richiesto lo studio del codice e dell'architettura dell'applicazione: ***front-end***, ***back-end*** e **base dati sottostante**, e un certo insieme di tecnologie e strumenti tra i quali **React Native** e **ASP.NET Core**, **Visual Studio**, e **Server Management Studio**. Il progetto ha incluso la realizzazione delle API e l'interfaccia grafica del modulo richiesto, insieme a una batteria di test automatici per la *Business Logic* di alcune di tali API.

Ho segnalato tutte le parole non italiane in *corsivo* all'interno del documento.

Utilizzo il carattere **monospaziato** per i nomi di tabelle e colonne del *database*, classi, funzioni, *component*, *view* o altre parti del codice.

Evidenzio le parole del glossario con una G a pedice, in corsivo e in blu (ad esempio *CSR<sub>G</sub>*).

Uso il **grassetto** per enfatizzare la parola chiave di un punto elenco per migliorare la leggibilità e rendere immediatamente identificabili i concetti chiave.

Ecco come appare un elenco puntato:

---

"Elenco numeri primi:

- 1
- 2
- 3

..."

Ho diviso il documento in 4 macro sezioni:

**Capitolo 1 - VisioneImpresa:** Descrivo l'azienda dove ho svolto il tirocinio, riportando brevemente clienti, prodotti, organizzazione aziendale, strumenti e tecnologie utilizzate, e la propensione all'innovazione.

**Capitolo 2 - Descrizione del progetto:** Presento il progetto assegnatomi dall'azienda, specificando obiettivi, vincoli tecnologici e temporali. Approfondisco inoltre il rapporto dell'azienda con gli *stage* in generale e le motivazioni dietro la scelta di questo specifico progetto.

**Capitolo 3 - *Stage*:** Racconto la mia esperienza di *stage*.

**Capitolo 4 - Retrospettiva:** Offro un giudizio obiettivo sul raggiungimento degli obiettivi di *stage*, personali e aziendali. Faccio inoltre un resoconto delle conoscenze acquisite con questa esperienza e una valutazione personale del percorso di studi universitario.

# Indice

<b>1</b>	<b>VisioneImpresa</b>	<b>1</b>
1.1	L'azienda . . . . .	1
1.2	Clienti e servizi . . . . .	2
1.3	Organizzazione aziendale . . . . .	5
1.3.1	Aree di competenza . . . . .	5
1.3.2	Metodologie di sviluppo <i>software</i> . . . . .	7
1.4	Tecnologie . . . . .	9
1.4.1	Elenco delle tecnologie utilizzate . . . . .	9
1.4.2	Integrazione delle tecnologie con i processi aziendali . . . .	12
1.5	Propensione all'innovazione . . . . .	15
<b>2</b>	<b>Descrizione del progetto</b>	<b>16</b>
2.1	Strategia aziendale e rapporto con gli <i>stage</i> . . . . .	16
2.2	Descrizione progetto . . . . .	17
2.3	Scelta dell'attività di <i>stage</i> . . . . .	19
2.4	Vincoli . . . . .	20
2.4.1	Vincoli tecnologici . . . . .	20
2.4.2	Vincoli temporali . . . . .	23
2.5	Obiettivi . . . . .	24
2.5.1	Obiettivi aziendali . . . . .	24
2.5.2	Obiettivi personali . . . . .	24
	<b>Glossario</b>	<b>i</b>
	<b>Sitografia</b>	<b>iii</b>

## Elenco delle figure

1.1	Obiettivi delle società benefit. . . . .	2
1.2	Organizzazione di uno <i>sprint</i> con il <i>framework</i> Scrum . . . . .	8
1.3	Integrazione metodologie e tecnologie . . . . .	12
2.1	Alcune <i>view</i> di MoviORDER . . . . .	17
2.2	Tecnologie utilizzate per lo sviluppo di MoviORDER . . . . .	20
2.3	Pianificazione attività di <i>stage</i> . . . . .	23

## Elenco delle tabelle

2.1	Tabella Obiettivi . . . . .	24
2.2	Tabella Obiettivi Personali . . . . .	25



# Capitolo 1

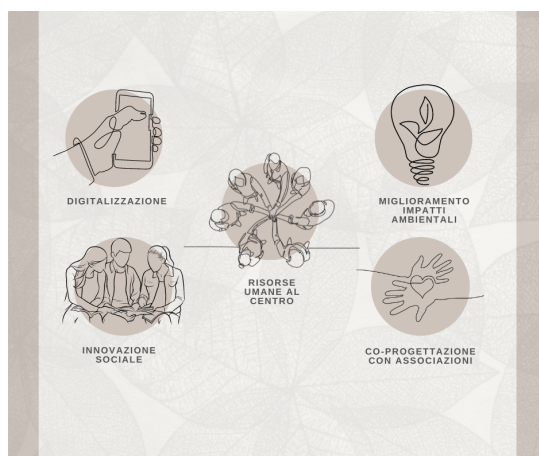
## VisioneImpresa

### 1.1 L'azienda

VisioneImpresa è un'azienda con quarant'anni di esperienza nell'offrire a piccole e medie imprese soluzioni informatiche per la gestione e l'automazione dei processi aziendali. I suoi prodotti di punta sono infatti sistemi *ERP<sub>G</sub>* (*Enterprise Resource Planning*) ovvero sistemi che permettono di coordinare il flusso di dati tra i processi di un'azienda, fornendo un'unica fonte di informazioni e semplificandone le operazioni.

VisioneImpresa è situata a Pernumia (Padova) e opera in prevalenza nel Nord Italia, dal 2016 è entrata a far parte del gruppo Officegroup, azienda che riunisce diverse *software house* specializzate nella progettazione e sviluppo di sistemi gestionali evoluti. Dal 2023 inoltre è diventata una società *benefit*, ovvero è un'azienda che opera con l'obiettivo di generare un impatto positivo sulla società e sull'ambiente, oltre al profitto finanziario.





**Figura 1.1:** Obiettivi delle società benefit.

fonte: <https://www.vsh.it/azienda/societa-benefit/>

La figura 1.1 mostra le iniziative promosse da questo tipo di società, dalla dematerializzazione e digitalizzazione alla promozione di politiche a sostegno della conciliazione vita-lavoro.

Altri obiettivi delle società *benefit* possono essere: investire nelle energie rinnovabili e la sostenibilità, l'investimento in tecnologie ad alta efficienza energetica, rispetto della parità di genere, formazione professionale del lavoratore, progetti con scuole ed università, co-progettazione con associazioni e istituzioni del territorio con il duplice obiettivo di stimolare la partecipazione dei dipendenti a “buone cause” della comunità e valorizzare il lavoro di associazioni no-profit del territorio, generando così valore sociale.

## 1.2 Clienti e servizi

VisioneImpresa ha come clienti piccole e medie imprese situate in prevalenza in Veneto e in generale nel Nord Italia, possiamo trovare però anche clienti dal Centro Italia e dalla Sardegna.

Il gestionale che propone può adattarsi a qualsiasi tipo di azienda indipendentemente dal settore in cui operi (anche se come vedremo vengono venduti dei gestionali ad hoc per i settori: petrolifero, ittico, assistenza post-vendita, ortofrutticolo, antincendi e antinfortunistica, trasporti).

Una volta implementato il gestionale all'interno dell'azienda del cliente viene

offerta una formazione all'utilizzo del *software* per i dipendenti, che parteciperanno a delle riunioni tenute da un consulente tecnico che ne illustrerà le funzionalità e insegnerà come sfruttarle al meglio.

VisioneImpresa propone due linee di prodotti principali: Vision e MoviDAT. La prima, Vision, è la linea di gestionali dell'azienda, con VisionENTERPRISE, che è il loro *ERP<sub>G</sub>* di punta, e poi una serie di soluzioni verticali per venire incontro alle specifiche esigenze delle varie aziende con cui VisioneImpresa opera. Ognuna delle soluzioni verticali offerte dall'azienda è una variazione di VisionENTERPRISE, che viene arricchita con funzionalità specifiche per adattarsi a specifici settori. In particolare quindi nella linea Vision abbiamo:

- **VisionENTERPRISE**, *ERP<sub>G</sub>* di punta dell'azienda e dedicato ad imprese che non hanno necessità di funzionalità specifiche.
- **VisionENERGY**, gestionale con specifiche funzionalità pensate per le aziende che lavorano nel settore petrolifero, come la possibilità di gestire la vendita di carburante, manutenzione valvole, ecc.;
- **VisionBLUE**, gestionale con specifiche funzionalità pensate per le aziende che lavorano nel settore ittico, come la possibilità di gestire lotti, prodotti e imballaggi;
- **VisionASSISTANCE**, gestionale con specifiche funzionalità pensate per le aziende specializzate nell'assistenza post-vendita, come la possibilità di gestire richieste di assistenza, contratti e assegnare gli ordini di intervento ai singoli tecnici;
- **VisionFRESH**, gestionale con specifiche funzionalità pensate per le aziende che lavorano nel settore ortofrutticolo, come la possibilità di gestire movimentazione merce, inserimento pesate, interfacciamento con bilance elettroniche, ecc.;
- **VisionANTINCENDI**, gestionale con specifiche funzionalità pensate per le aziende che lavorano nel settore antincendi e antinfortunistica, come la possibilità di gestire chiamate ed interventi straordinari, buoni di manutenzione e geolocalizzare gli interventi;

- **VisionTRASPORTI**, gestionale con specifiche funzionalità pensate per le aziende che lavorano nel settore trasporti, come la possibilità di gestire listini, anagrafiche, dotazioni, manutenzione, pianificazione viaggi, ecc.

Nella linea di prodotti MoviDAT invece troviamo una gamma di applicazioni sviluppate per i principali sistemi operativi per dispositivi *mobile*: Android e iOS. Queste applicazioni sono state sviluppate per integrarsi direttamente con i gestionali della linea Vision e permettono di semplificare il lavoro di dipendenti che operano in mobilità e non hanno a disposizione un *computer* con cui lavorare durante le trasferte (ed anche se ce lo avessero il suo utilizzo risulterebbe scomodo).

In questa linea dunque troviamo:

- **MoviDOC** è un *web app<sub>G</sub>* (ovvero un *app* a cui è possibile accedere direttamente da *browser* senza necessità di installarla sul dispositivo) che consente la gestione e condivisione dei documenti;
- **Handy** è un *app* per palmare che integrata a VisionENTERPRISE supporta la movimentazione della merce del magazzino o del punto vendita;
- **MoviSELL** è un *app* sviluppata per *tablet* iOS dedicata agli agenti aziendali, permette di: visualizzare i clienti su una mappa, avere visibilità dello stato contabile e inserire ordini clienti direttamente nel ciclo attivo dell'azienda.
- **MoviREP** è un *app* sviluppata per *tablet* iOS per la gestione digitalizzata dei rapportini da parte di operatori addetti alla manutenzione o all'assistenza post vendita.
- **MoviALERT** è una *web app<sub>G</sub>* che permette di inviare *mail* di notifica automatiche all'avvenire di specifici eventi nel gestionale;
- **MoviCHECK** è un *web app<sub>G</sub>*, scaricabile anche su dispositivi Android e iOS per consultare i dati di business in mobilità;
- **MoviEXPENSE** è un *app* per Android e iOS, per la registrazione automatica delle note spese;

- **MoviCHECKIN** è una *web app* per la registrazione dei visitatori in azienda;
- **MoviORDER** applicazione per *smartphone e tablet* iOS e Android che l'azienda può fornire ai propri clienti per l'invio di ordini e richieste di approvvigionamento.

Nel caso in cui un'azienda richieda funzionalità specifiche per uno dei *software* sopra elencati, VisioneImpresa offre la possibilità di creare una versione modificata dei propri prodotti. Per evitare di avere troppe variazioni dello stesso prodotto il codice delle personalizzazioni (così vengono chiamate le funzioni in più richieste dal cliente) vengono inserite direttamente nel codice del *software* principale, e "attivate" da specifici parametri controllati all'avvio del sistema. Nel caso di MoviORDER, che ho avuto la possibilità di esaminare per questo progetto, a seconda del valore del campo *Company* ottenuto a seguito dell'autenticazione del cliente venivano apportate alcune variazioni grafiche (loghi, tema). Questo si può ottenere grazie ad un'attenta progettazione e appropriate scelte architetture.

## 1.3 Organizzazione aziendale

### 1.3.1 Aree di competenza

VisioneImpresa è strutturata in tre aree di competenza, ognuna con ruoli e responsabilità specifiche.

Reparto Assistenza: qui operano i consulenti tecnici gestionali, il cui compito è assistere l'azienda nell'implementazione dei nuovi gestionali e nella gestione del cambiamento assicurandosi che il personale aziendale sia formato sull'uso delle nuove tecnologie. Ogni consulente è responsabile di uno o più *software* di cui hanno un'ampia conoscenza operativa. Inoltre, forniscono assistenza ai clienti, aiutandoli nella risoluzione dei problemi e, se necessario, segnalando le problematiche al reparto sviluppo che aprirà quindi un *ticket* all'interno della piattaforma Jira (vedi capitolo 1.4).

Area Amministrazione Commerciale e *Marketing*: In quest'area si trovano diverse competenze, tra cui:

- **Responsabile *marketing***: ha il compito di realizzare strategie per promuovere l'azienda e i suoi prodotti ai potenziali clienti;
- **Risorse umane**: ha il compito di amministrare stipendi, pensioni e *benefit*, nonché di assicurarsi il rispetto da parte dell'azienda delle normative sul lavoro;
- **Contabilità**: ha il compito di gestire e registrare le transazioni finanziarie, garantendo che tutte le attività economiche siano documentate in modo accurato e trasparente;
- **Segreteria generale**: ha il compito di gestire e indirizzare le chiamate in entrata, gestire la posta elettronica e la corrispondenza, pianificare eventi aziendali;
- **Segreteria commerciale**: ha il compito di mantenere le comunicazioni con i clienti fornendo informazioni su prodotti e servizi e preparando offerte commerciali, contratti di vendita e documenti correlati;
- **Amministrazione ciclo attivo**: ha il compito di garantire una gestione efficiente delle vendite e della riscossione dei pagamenti;
- **Commerciale rete diretta**: ha il compito di occuparsi della vendita dei prodotti direttamente ai clienti finali;
- **commerciale rete indiretta**: ha il compito di gestire le vendite attraverso intermediari come distributori, rivenditori, agenti o partner commerciali;
- **Responsabile d'impatto**: si occupa della valutazione, pianificazione e promozione delle misure di responsabilità sociale d'impresa (*CSR<sub>G</sub>*), ovvero di tutte le iniziative attuate dall'azienda in ambito sociale e di transizione ecologica.

- **Amministratore** è responsabile di dirigere e gestire l'azienda nel suo complesso, assicurando che tutti i dipartimenti e le attività lavorino insieme per raggiungere gli obiettivi strategici e operativi.
- **Product manager** ha il compito di assicurare che i prodotti siano sviluppati in linea con le esigenze del mercato, lanciati con successo e gestiti efficacemente durante il loro ciclo di vita.

Reparto Sviluppo *Software*: In quest'area troviamo:

- **Sviluppatori**: possono essere *front-end*, specializzati nello sviluppo di interfacce e della gestione dell' interazione uomo-macchina, *back-end* specializzati nello sviluppo della logica del *software* e nella manipolazione dei dati, o *full-stack*, in grado di operare sia come sviluppatore *front-end* che *back-end*.
- **Project manager**: ha il compito di assicurarsi che vengano rispettati obiettivi, tempi, costi e vengano soddisfatti i parametri di qualità;
- **Direttore dello sviluppo**: ha il compito di prendere le decisioni implementative e scegliere l'architettura del *software*, si occupa inoltre di dirigere il team e di pianificare e assegnare i lavori da svolgere;
- **Analista**: si occupa di interagire con i clienti per delineare i requisiti del progetto *software* e documentarli in un documento di analisi.

### 1.3.2 Metodologie di sviluppo *software*

Ho potuto notare, durante la mia esperienza di tirocinio, che gli sviluppatori utilizzano metodologie Agile per la gestione dei loro progetti.

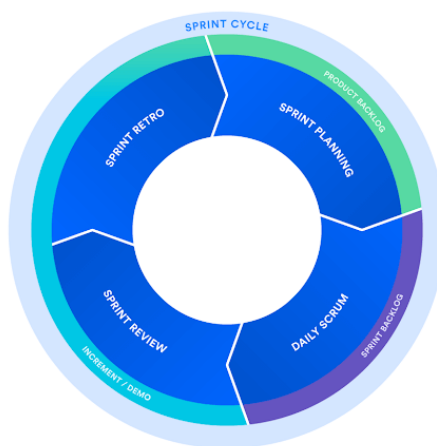
Le metodologie Agile sono un'approccio alla gestione dei progetti che prevede la suddivisione del progetto in fasi e del lavoro in cicli brevi, al termine dei quali verranno introdotti cambiamenti che avvicinano il progetto sempre di più al soddisfacimento di tutti i requisiti. Questo approccio è particolarmente adattabile agli imprevisti, permettendo di reagire velocemente e riducendo al minimo i danni, come lo slittamento della data di completamento e conseguentemente

l'aumento di denaro da destinare al progetto. Il manifesto Agile riporta i punti principali punti di questa filosofia <sup>1</sup>:

- Gli **individui e le interazioni** più che i processi e gli strumenti;
- Il **software funzionante** più che la documentazione esaustiva;
- La **collaborazione col cliente** più che la negoziazione del contratto;
- **Rispondere al cambiamento** più che seguire un piano.

Sebbene il manifesto Agile, pubblicato nel 2001, abbia posto le basi di questo approccio, le pratiche Agile si sono notevolmente evolute nel corso degli anni. Attualmente, esistono diverse interpretazioni e implementazioni delle metodologie Agile, adattate alle specifiche esigenze delle organizzazioni e dei *team* di sviluppo.

In particolare, VisioneImpresa adotta il *framework* Agile Scrum, che definisce una serie di principi, pratiche e cerimonie per riuscire ad assimilare nel proprio metodo di lavoro la metodologia Agile. Scrum richiede di suddividere il lavoro in *sprint* dalla durata variabile di una fino a quattro settimane. VisioneImpresa pianifica *sprint* di una settimana in modo da rispondere tempestivamente a gli imprevisti ed effettuare una pianificazione più efficace.



**Figura 1.2:** Organizzazione di uno *sprint* con il *framework* Scrum.

fonte: <https://www.atlassian.com/it/agile/scrum>

---

<sup>1</sup>Manifesto Agile per lo sviluppo *software*, fonte riportata nella sitografia [2.5.2](#).

La figura 1.2 mostra come ogni *sprint* è strutturato in una serie di incontri che avvengono solitamente in video chiamata usando 3CX (vedi capitolo 1.4).

Si comincia il lunedì, all'inizio dello *sprint*, quando programmatori, direttore dello sviluppo e *project manager* partecipano ad un *meeting* chiamato *sprint planning* dove si pianifica il lavoro da svolgere per lo *sprint* in corso. Quindi ogni giorno si tiene un breve *meeting* prima della pausa pranzo chiamato *daily scrum* dove si discute dello stato dei lavori ed eventuali problemi emersi. Il venerdì si tiene l'ultimo *meeting* dello *sprint* chiamato *sprint review* dove si discute dello stato dei lavori rispetto alle aspettative e discutendo dei problemi emersi durante lo *sprint* si cercano modi per migliorare.

VisioneImpresa organizza inoltre un ulteriore *meeting* a cadenza mensile dove non solo le persone interessate al progetto, ma tutti i dipendenti dell'azienda si riuniscono per discutere dello stato dei lavori di ogni settore: evoluzione dei prodotti, vendite, *feedback* dei clienti, aggiornare il reparto *marketing* e commerciale sulle nuove funzionalità dei *software* ecc.. Questo incontro ha lo scopo di dare a tutti i dipendenti dell'azienda una visione d'insieme evitando il cosiddetto "effetto sottomarino", ovvero quando una persona o un gruppo si focalizzano soltanto in uno specifico ambito, favorendo l'isolamento rispetto al resto dell'azienda, che ha invece bisogno di lavorare coordinando i vari settori.

## 1.4 Tecnologie

### 1.4.1 Elenco delle tecnologie utilizzate

L'azienda utilizza diversi strumenti sia per lo sviluppo, che per lo svolgimento dei normali processi aziendali.

- **Portatili:** ad ogni impiegato viene messo a disposizione un portatile con Windows 10 o 11, il sistema operativo di Microsoft o all'occorrenza un Mac con macOS, il portatile sviluppato da Apple con il suo sistema operativo proprietario;



- **Dispositivi *mobile*:** all'interno dell'azienda troviamo molti dispositivi *mobile* con diversi sistemi operativi e dimensioni dello schermo, usati per il *testing* delle applicazioni Android e iOS;
- **Microsoft Office 365:** servizio in abbonamento di Microsoft che include diversi *software* come Word e PowerPoint;
- **Zimbra:** sistema di posta elettronica utilizzato dall'azienda, durante il mio *stage* è stato cambiato in favore di un'integrazione di Zimbra con Outlook;
- **Bitbucket:** strumento per la gestione della versione Git basato sul *web*, che consente di creare *repository* pubbliche o private per caricare il proprio codice e gestirlo in modo collaborativo con il proprio *team*;
- **Jira:** *suite* di *software* proprietari per il tracciamento delle segnalazioni sviluppato da Atlassian, che consente il *bug tracking* e la gestione dei progetti;
- **Confluence:** strumento che permette ai *team* di condividere e organizzare documenti e contenuti in un ambiente centralizzato e strutturato;
- **3CX:** centralino telefonico PBX (*Private Branch Exchange*), ovvero una rete telefonica privata utilizzata all'interno di un'azienda o organizzazione. Gli utenti del sistema telefonico PBX possono comunicare internamente ed esternamente, tramite il classico telefono fisso o chat da *smartphone*. Questo sistema permette anche di effettuare video chiamate e di scambiarsi messaggi all'interno di *chat*;
- **Visual Studio:** Si tratta di un ambiente di sviluppo integrato completo (*IDE<sub>G</sub>*) che è possibile usare per scrivere, modificare, eseguire il *debug* e compilare codice. Visual Studio include compilatori, strumenti di completamento del codice, controllo del codice sorgente, estensioni e molte altre funzionalità per migliorare ogni fase del processo di sviluppo *software*;

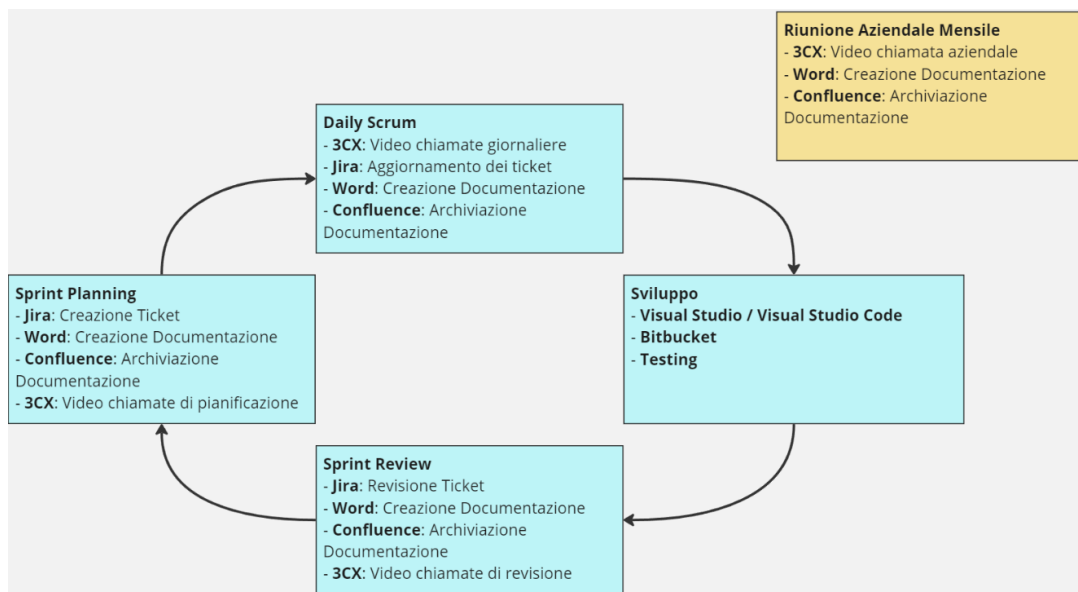
- **Visual Studio Code:** *editor* di codice sorgente particolarmente leggero ed estensibile grazie ad una gamma di estensioni che è possibile integrargli. È inoltre *open source* e compatibile con una vasta gamma di sistemi operativi;
- **SQL Server Management Studio (SSMS):** è un ambiente integrato per la configurazione, la gestione e l'amministrazione di tutti i componenti, le istanze e i *database* all'interno di Microsoft SQL Server. SSMS include sia *editor* di *script* che strumenti grafici che lavorano con oggetti e funzionalità del *server*.
- **Swagger:** un insieme di strumenti e tecnologie per la progettazione, la costruzione e la documentazione delle *API<sub>G</sub>* RESTful, ovvero delle interfacce che permettono a i vari *software* di comunicare tra loro.

L'azienda utilizza una vasta gamma di linguaggi di programmazione, *framework* e librerie per diversi motivi. Alcuni di questi includono l'acquisizione e l'adattamento di codice sorgente da altre aziende, il fatto che il codice sia stato scritto molti anni fa con tecnologie ormai obsolete, e la necessità di utilizzare linguaggi specifici per soddisfare esigenze particolari. Tuttavia, l'azienda si impegna a uniformare quanto più possibile i linguaggi e a ridurre il numero di tecnologie in uso, al fine di semplificare e rendere più efficiente la gestione delle risorse tecnologiche. Oltre a quelle che ho usato per il mio progetto (che sono discusse più approfonditamente nel capitolo 2.4.1) queste sono alcune delle tecnologie che l'azienda usa per lo sviluppo dei suoi prodotti.

- **Angular:** *framework* per lo sviluppo di *web app<sub>G</sub>* basato su TypeScript, sviluppato e mantenuto da Google;
- **Librerie Dev Express:** Dev Express è un'azienda incentrata sulla creazione di librerie di componenti grafici di cui le più famose sono Blazor e MAUI. Queste librerie sono supportate per lo sviluppo in React, Angular e Vue;

- **FoxPro**: un sistema di gestione di *database* e un linguaggio di programmazione procedurale orientato agli oggetti. Originariamente sviluppato da Fox Software e successivamente acquisito da Microsoft.

### 1.4.2 Integrazione delle tecnologie con i processi aziendali



**Figura 1.3:** Integrazione metodologie e tecnologie.

La figura 1.3 mostra come l'applicazione del *framework* Scrum richieda l'impiego di diverse tecnologie:

- **Documentazione**: a seguito dei *meeting* aziendali viene prodotto un documento che può avere diverse finalità: trascrivere gli argomenti dell'incontro, le difficoltà incontrate e le soluzioni da applicare per correggerle o semplici appunti. Gli sviluppatori usano quindi Word per redarre la documentazione data la sua estrema semplicità e la velocità con la quale si possono produrre tali documenti;
- **Archiviazione di documenti**: eccezion fatta per gli appunti personali, la documentazione richiede di essere condivisa e organizzata. Nonostante in VisioneImpresa esista una serie di cartelle di rete accessibili tramite il *file manager* di Windows, questa soluzione risulta inadeguata allo scopo

in quanto povera di funzionalità e personalizzazioni. Confluence permette non solo di organizzare la documentazione in una serie di *directory on-line* in modo da rendere i documenti sempre consultabili, ma permette anche di definire una serie di privilegi per consentire o meno l'accesso alla documentazione.

- **Comunicazione:** 3CX agisce come canale di comunicazione per la maggior parte delle comunicazioni interne. È il luogo dove si svolgono i *meeting*, fondamentali per mantenere il *team* coordinato, ma permette anche di effettuare chiamate e scrivere in *chat* confinando la maggior parte delle comunicazioni in un unico luogo. Ovviamente anche le *mail* sono un valido strumento di comunicazione, ed infatti viene usato Zimbra per comunicare con persone esterne all'azienda o per alcune *mail* interne (comunicazioni di servizio, *reminder*, ecc.). Quindi 3CX viene usato maggiormente per i *meeting* o per un tipo di comunicazione veloce ed informale, mentre il servizio di posta elettronica viene usato principalmente per comunicazione con persone esterne all'azienda. Ovviamente i clienti che hanno necessità di assistenza possono chiamare l'assistenza tramite il numero di telefono che viene loro fornito per mettersi in contatto con uno dei tecnici del reparto assistenza;
- **Sviluppo:** Per quanto riguarda lo sviluppo gli strumenti utilizzati e le tecnologie impiegate variano a seconda del progetto a cui lo sviluppatore sta lavorando o da preferenze personali (come nel caso del sistema operativo utilizzato nel proprio *computer*). Anche per quanto riguarda l'*editor* viene lasciata libertà di scelta, personalmente per lo sviluppo di *API<sub>G</sub>* in .NET (le tecnologie che ho impiegato per lo sviluppo del progetto sono discusse in un capitolo a parte [2.4.1](#)) ho preferito utilizzare Visual Studio perché offre molti strumenti di supporto e *debug* integrati. Per quanto riguarda lo sviluppo del *front-end* ho preferito utilizzare Visual Studio Code perché più leggero, personalizzabile.
- **Testing:** anche qui gli strumenti utilizzati dipendono dal progetto che si sta sviluppando. Nel mio caso ho utilizzato uno *smartphone* e *tablet*

Android per testare l'applicazione nel suo insieme e la piattaforma Swagger per testare manualmente le *API* (le tecnologie che ho impiegato per lo sviluppo del progetto sono discusse in un capitolo a parte 2.4.1, mentre per la descrizione accurata di come ho effettuato il testing del codice consulta il capitolo TODO 3.6). Riporto per completezza che per quanto riguarda il *testing* di applicazioni Android è possibile emulare un dispositivo con gli strumenti offerti dallo strumento di sviluppo Android Studio e installare l'applicazione su questo *device* virtuale. Il problema di questo strumento è che richiede *computer* con prestazioni molto alte per funzionare in maniera fluida altrimenti rischia di paralizzare l'elaboratore. Non sono sicuro che sia utilizzato dagli sviluppatori di VisioneImpresa quindi ho evitato di includerlo nelle tecnologie utilizzate;

- **Collaborazione e gestione del codice:** Bitbucket viene utilizzato come piattaforma per depositare il codice sorgente e gestire lo sviluppo collaborativo. Qui gli sviluppatori caricano il loro codice suddiviso in *repository* per ogni progetto. Ogni *repository* vede diversi *branch* attivi: *main* che contiene l'ultima versione rilasciata al pubblico del *software*, *develop* ovvero il *branch* di lavoro dove nascono e confluiscono tutti i *feature branch* prima di effettuare il rilascio in *main*, i *feature branch* che viene creato dal programmatore per sviluppare una specifica funzione del programma che sarà, una volta terminata e testata, aggiunta in *develop*;
- **Gestione dei progetti:** Jira è una piattaforma particolarmente utile per pianificare i vari compiti da svolgere nello *sprint* e assegnarli ai vari componenti del *team* di sviluppatori. Questi compiti sono chiamati *ticket* e possono essere di diverso tipo: *epic* che rappresentano grosse porzioni di lavoro e sono quindi usate come raccolte di *ticket*, *task* il singolo compito che deve essere completato e *bug* che rappresenta una problematica da risolvere. I *bug* possono essere avere diverse origini: gli sviluppatori stessi nel caso in cui si accorgano di un difetto di programmazione o da i clienti che telefonando all'assistenza riportano il problema, quindi il tecnico riporterà la problematica al *project manager* che creerà la *task*. Jira offre

inoltre molti altri strumenti per la gestione di metodologie Agile come la possibilità di creare diagrammi di Gantt, che permettono di avere una rappresentazione visiva delle attività programmate nel tempo, aiutando il *team* a comprendere meglio la sequenza delle attività, o la definizione di un *backlog*, ovvero una lista con le *task* rimaste incompiute durante gli *sprint* precedenti;

## 1.5 Propensione all'innovazione

VisioneImpresa non dispone di un ufficio specificamente dedicato alla ricerca e sviluppo, ma questo non significa che non vengano effettuati aggiornamenti costanti delle tecnologie e degli strumenti utilizzati. Ad esempio, l'azienda ha in programma di migrare i propri sistemi *ERP<sub>G</sub>*, attualmente scritti in FoxPro (un linguaggio il cui supporto da parte di Microsoft è terminato nel 2015), verso tecnologie più moderne. Questo progetto, data la grandezza e complessità dei *software* coinvolti, richiederà anni per essere completato.

Inoltre, l'attività di *stage* rappresenta un'opportunità per l'azienda di innovare. Durante il mio tirocinio, ho osservato un apprezzamento particolare per l'indipendenza degli stagisti nel cercare e implementare soluzioni o tecnologie originali. Per ulteriori dettagli sul rapporto dell'azienda con gli *stage*, consulta il capitolo [2.1](#).

# Capitolo 2

## Descrizione del progetto

### 2.1 Strategia aziendale e rapporto con gli *stage*

Come ho descritto nel capitolo 1.5, gli *stage* rappresentano per VisioneImpresa un'opportunità di innovazione e crescita.

Gli stagisti, prossimi alla conclusione del corso triennale di laurea in informatica, possiedono una buona conoscenza di quali sono le nuove tecnologie e hanno la curiosità di studiarle e implementarle.

L'azienda decide quindi di sfruttare l'occasione offerta dai tirocini per innovare e rinnovare i propri prodotti.

I progetti assegnati agli stagisti devono essere formativi, richiedendo uno studio approfondito del codice e delle tecnologie. Questi compiti non devono essere banali e devono mirare a stimolare la crescita e l'apprendimento dello studente. Inoltre, devono basarsi su necessità reali, identificate mediante colloqui con i clienti, e sono pertanto di interesse per VisioneImpresa nell'ottica di una futura implementazione dei progetti nei prodotti reali.

Al termine del lavoro, lo studente presenta il suo operato agli sviluppatori e al *project manager* del prodotto in questione. Questo *meeting* è cruciale per valutare la qualità del lavoro svolto e comprendere la logica di sviluppo adottata. Permette inoltre agli sviluppatori e al *project manager* di esaminare il progetto in funzione e valutare i benefici di una sua futura implementazione.

Successivamente, si richiede allo stagista di consegnare il codice sorgente e la

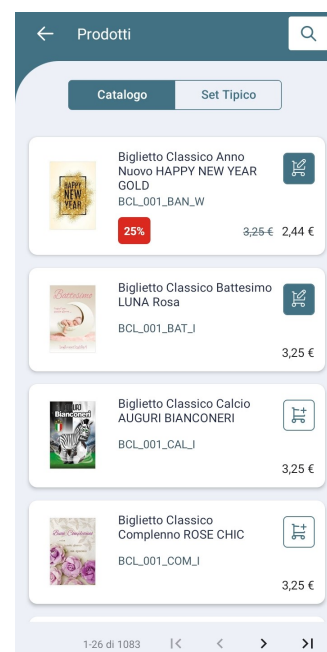
documentazione tecnica, che serviranno da guida per l'eventuale implementazione futura.

Gli *stage* per VisioneImpresa sono inoltre un modo per mettersi in contatto con studenti che si sono distinti in azienda e che hanno intenzione di interrompere il loro percorso universitario dopo la laurea, in modo da poter proporre loro un colloquio dove poterne valutare l'assunzione.

## 2.2 Descrizione progetto



(a) *Homepage* della versione *tablet* di MovORDER



(b) Pagina prodotti della versione *smartphone* di MovORDER

**Figura 2.1:** Alcune *view* di MovORDER

MovORDER è un'applicazione fornita dalle aziende ai loro clienti per gestire gli ordini ed effettuare il *restock* della merce. L'app permette inoltre di: visualizzare lo storico degli ordini precedenti, accedere a un *set* tipico (ovvero un ordine predefinito con i prodotti solitamente acquistati), visualizzare il carrello per il riepilogo dell'ordine e uno storico dei documenti generati dopo la conferma dell'ordine.

La sua *homepage* è visibile in figura 2.1a, mentre il catalogo prodotti è mostrato



in figura 2.1b.

Il progetto assegnatomi da VisioneImpresa richiede di sviluppare un modulo per MoviORDER chiamato "Modulo Agenti", ovvero l'insieme di interfacce, funzioni, *API<sub>G</sub>*, ecc. che permettono l'autenticazione di un nuovo tipo di utente, gli agenti aziendali, i quali, selezionando da una lista uno dei clienti a loro assegnati, possono operare nell'*app* come il cliente selezionato.

La necessità di implementare questo modulo nasce da alcuni bisogni segnalati dai suoi clienti a VisioneImpresa e che mi sono stati descritti durante il primo incontro in azienda con il *tutor* aziendale. I principali motivi sono:

- **MoviSELL, l'*app* pensata per gli agenti aziendali, è disponibile solo per *tablet* iOS.** Questo può costituire un problema per alcune aziende che, per dotare i propri agenti dell'*app*, sono obbligate ad acquistare questi *tablet* per i propri agenti. Per aziende con agenti plurimandatari, cioè che rappresentano più aziende contemporaneamente, questo requisito si rivela essere particolarmente oneroso da soddisfare, dato che si richiede di fornire i *tablet* non ai propri dipendenti, ma a professionisti esterni;
- **Non tutti gli agenti si trovano a loro agio ad usare il *tablet*,** trovandolo ingombrante e scomodo, soprattutto per chi lavora molto in mobilità. Pertanto, avere un'alternativa per smartphone risulta preferibile.
- MoviSELL è un'*app* ricca di funzionalità, ma può risultare di difficile utilizzo per chi non ha dimestichezza con gli strumenti digitali. **MoviORDER risulta molto più semplice e intuitiva, rimuovendo la barriera tecnologica per alcuni agenti** e permettendo loro di svolgere il loro lavoro;
- Alcuni clienti preferiscono contattare direttamente gli agenti per effettuare i loro ordini, invece di usare MoviORDER. **Il modulo agenti semplifica l'operazione di creazione dell'ordine per gli agenti**, evitando loro di dover appuntare la merce da ordinare e poi effettuare l'ordine dal *computer* una volta rientrati in ufficio.

### 2.3 Scelta dell'attività di *stage*

L'incontro con VisioneImpresa è avvenuto durante l'evento StageIT 2024, organizzato dall'Università di Padova e promosso da Confindustria Veneto Est.

Questo evento offre alle aziende l'opportunità di incontrare gli studenti e condurre brevi colloqui, illustrando le caratteristiche dell'azienda e i progetti offerti per lo *stage*.

In questa occasione, ho avuto l'opportunità di esplorare diverse realtà operanti in settori distinti, dallo sviluppo *web* a progetti in ambito *cyber security*. Non mi sono limitato a cercare progetti allineati alle mie conoscenze pregresse, ma ho esplorato diverse opzioni.

Durante l'incontro con VisioneImpresa, ho approfondito la conoscenza dell'azienda e mi sono state fornite ulteriori informazioni riguardo i loro progetti di *stage*, descritti in un elenco diffuso prima dell'evento per tutte le aziende coinvolte.

Successivamente ho selezionato i progetti più interessanti e fissato ulteriori colloqui con le aziende per discutere in maniera più approfondita delle proposte. I colloqui si sono focalizzati principalmente su: il progetto in dettaglio, le tecnologie utilizzate, l'azienda e una discussione ad alto livello sulle possibili implementazioni del progetto.

Durante l'incontro con VisioneImpresa ho inoltre potuto fare un *tour* dell'azienda, che mi ha permesso di conoscere i dipendenti e osservare il loro ambiente di lavoro e le interazioni con i clienti.

La scelta di lavorare al progetto "Modulo agenti" è stata motivata da diversi fattori: innanzitutto, mi ha consentito di lavorare ad un'applicazione Android, ambito di grande interesse personale, utilizzando tecnologie moderne e ricercate come React Native. Questa tecnologia non mi è del tutto estranea, grazie all'esperienza pregressa con React, sebbene non includa le funzionalità specifiche per applicazioni *mobile*. Il progetto ha previsto anche l'utilizzo di ASP.NET Core per lo sviluppo di *API*.

In secondo luogo, l'opportunità offerta da VisioneImpresa mi ha permesso di lavorare sia come sviluppatore *front-end*, creando le interfacce per l'*app*, che

come sviluppatore *back-end*, creando e modificando le *API<sub>G</sub>* dell'applicazione.

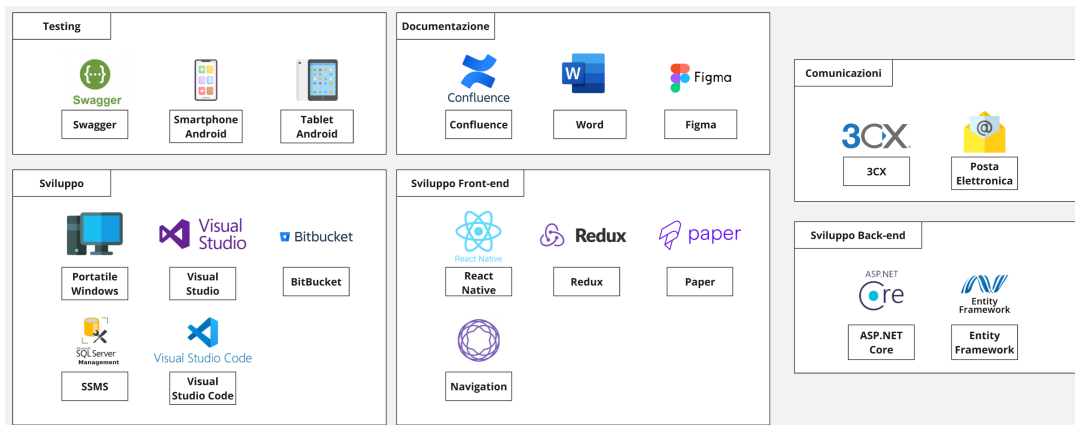
## 2.4 Vincoli

### 2.4.1 Vincoli tecnologici

VisioneImpresa non impone vincoli specifici sulle tecnologie da utilizzare. Tuttavia, essendo lo scopo del progetto lo sviluppo di un modulo per un'applicazione esistente, emergono vincoli tecnologici impliciti. È necessario utilizzare gli stessi *framework* già impiegati per lo sviluppo, per poter integrare il modulo nell'*app* ed eventualmente riportarlo nel *software* commercializzato.

Non vi sono divieti circa l'introduzione di nuove tecnologie. Come riporto nel capitolo 2.1, uno degli obiettivi aziendali per il progetto di *stage* è valutare il beneficio di nuove librerie, *framework* e tecnologie per favorire innovazione e crescita.

Anche per gli strumenti da utilizzare non ho ricevuto vincoli espliciti, eccetto l'uso di BitBucket come piattaforma per conservare il codice.



**Figura 2.2:** Tecnologie utilizzate per lo sviluppo di MovieORDER

Le tecnologie che ho utilizzato per sviluppare il modulo agenti, come mostra la figura 2.2, sono:

- **Visual Studio:** per lo sviluppo delle *API<sub>G</sub>*. Offre numerosi strumenti per il *debug* del codice e facilita la gestione dei pacchetti necessari. È

particolarmente efficace per lo sviluppo in C#, grazie alle funzioni di auto completamento e all'aggiornamento automatico degli *import*;

- **Visual Studio Code:** per lo sviluppo del *front-end*, grazie alle numerose estensioni disponibili per lo sviluppo in React (come Prettier, che assicura una formattazione uniforme del codice);
- **Computer con Windows 10:** preferito al Mac per la mia esperienza pregressa con Windows, evitando così l'apprendimento di un nuovo sistema operativo parallelamente alle nuove tecnologie necessarie;
- **Smartphone e tablet Android:** per il *testing* dell'applicazione;
- **BitBucket:** per conservare il codice all'interno di un *branch* del progetto e usufruire delle funzionalità di Git;
- **Confluence:** per conservare la documentazione creata;
- **Word:** per creare la documentazione tecnica, poiché consente di creare rapidamente e agevolmente un documento facilmente consultabile e modificabile;
- **Figma:** per creare il manuale utente. Permette una gestione dello stile più precisa e personalizzabile rispetto a Word, ma richiede più tempo per la creazione di un documento. Il risultato è un impatto grafico migliore, meno rilevante per un documento tecnico, ma apprezzabile per un documento destinato all'utente finale;
- **Servizio di posta elettronica:** per ricevere gli annunci aziendali;
- **SQL Server Management Studio (SSMS):** per operare sul *database* usato per lo sviluppo dell'applicazione;
- **Swagger:** per testare manualmente le *API* e valutarne il corretto funzionamento.

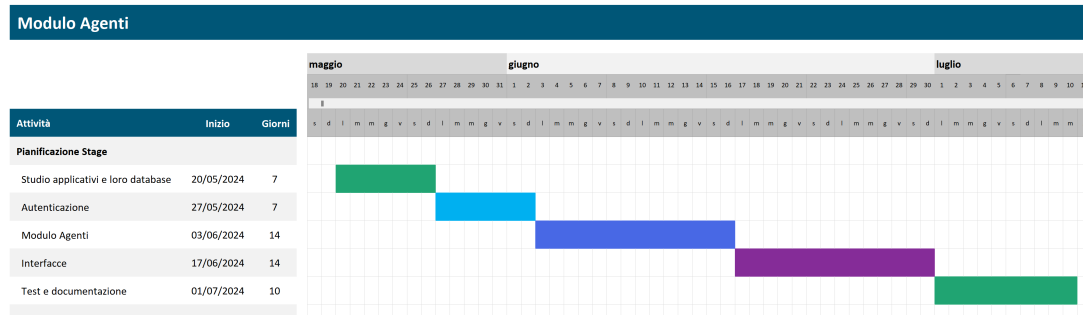
Per completezza, menziono anche 3CX tra le tecnologie che mi sono state fornite. Tuttavia, non ho mai avuto necessità di utilizzarlo, avendo avuto la possibilità

di confrontarmi personalmente con gli sviluppatori e il *tutor* aziendale. Inoltre, durante il mio *stage*, il *meeting* mensile generale è stato annullato, impedendomi di partecipare alla video chiamata.

I *framework* e le librerie utilizzate sono:

- **React Native:** *framework* che consente lo sviluppo di applicazioni Android e iOS utilizzando il *framework* React. Include funzionalità specifiche per dispositivi *mobile*, distinguendosi da React, principalmente utilizzato per lo sviluppo di siti *web*.  
Permette di scrivere e mantenere un unico codice funzionante per entrambi i sistemi operativi, eliminando la necessità di sviluppatori specializzati separatamente nello sviluppo Android e iOS;
- **React Native Paper:** libreria di componenti per le interfacce;
- **React Native Navigation:** libreria per la gestione della navigazione tra le *view*;
- **React Native Redux:** libreria per la gestione dello stato dell'applicazione in React Native;
- **ASP.NET Core:** *framework open source* moderno per lo sviluppo di applicazioni connesse a *internet*, utilizzato in questo caso per lo sviluppo delle *API*<sub>G</sub> necessarie;
- **Entity Framework Core:** *ORM*<sub>G</sub> (*Object-Relational Mapping*) per .NET, un *mapper* che semplifica l'accesso e la gestione dei dati nel *database*, permettendo di lavorare con oggetti .NET invece di *query* SQL.

### 2.4.2 Vincoli temporali



**Figura 2.3:** Pianificazione attività di *stage*

La figura 2.3 illustra la pianificazione delle mie attività di *stage*, come riportata nel piano di lavoro concordato con il *tutor* aziendale e approvato dal relatore. Lo stage si divide in cinque fasi principali:

1. **Studio degli applicativi e del *database*** (una settimana): si concentra sull'apprendimento delle tecnologie utilizzate e sull'analisi del codice sorgente;
2. **Modifica della *API<sub>G</sub>* di autenticazione** (una settimana): l'obiettivo è modificare la *API<sub>G</sub>* di *login* per il nuovo tipo di utente agente;
3. **Sviluppo del modulo agenti** (due settimane): prevede la creazione delle *API<sub>G</sub>* necessarie e la modifica del *front-end*;
4. **Modifica delle interfacce** (due settimane): prevede la creazione dei componenti, definizione dello stile della UI (*User Interface*) e l'ottimizzazione per *tablet*;
5. **Documentazione e *testing*** (dieci giorni): prevede la creazione della documentazione tecnica , operativa e il *testing* dell'applicazione.

## 2.5 Obiettivi

### 2.5.1 Obiettivi aziendali

La tabella 2.1 illustra gli obiettivi del progetto di *stage* richiesti dall'azienda. Questi mirano a produrre un prodotto usabile e funzionale, costituendo una solida base di studio per l'implementazione reale del modulo. Gli obiettivi sono categorizzati in obbligatori, contrassegnati dalla lettera M (*mandatory*), e opzionali, contrassegnati dalla lettera O (*optional*).

Tipo	Obiettivo
M	Modifica del sistema di autenticazione
M	Aggiornamento delle interfacce per adeguamento al nuovo modulo
M	Sviluppo del modulo agenti
M	Stesura documentazione tecnica e operativa
M	<i>Testing</i> delle nuove funzionalità
O	Ottimizzazione per <i>tablet</i>

**Tabella 2.1:** Tabella Obiettivi

### 2.5.2 Obiettivi personali

Per quanto riguarda gli obiettivi personali che ho stabilito per questo *stage*, sono elencati dalla tabella 2.2, nello stesso modo descritto per la tabella precedente.

Tipo	Obiettivo
M	Raggiungere una buona conoscenza delle principali tecnologie utilizzate (React Native e .NET)
Continua nella prossima pagina...	

**Tabella 2.2 – Continuo della tabella**

<b>Tipo</b>	<b>Obiettivo</b>
M	Analizzare e comprendere l'architettura <i>software</i> di MoviORDER
M	Apprendere ed implementare le metodologie di sviluppo aziendali
M	Migliorare nel <i>problem solving</i> trovando soluzioni efficienti ai problemi incontrati durante lo sviluppo
M	Collaborare con il <i>team</i> di VisioneImpresa per la realizzazione del progetto

**Tabella 2.2:** Tabella Obiettivi Personali



# Glossario

**API** *Application Programming Interface* (interfaccia di programmazione delle applicazioni) sono un insieme di definizioni e protocolli con i quali vengono realizzati e integrati *software applicativi*. Le API stabiliscono il contenuto e la forma dei dati necessari per la chiamata e quelli restituiti in risposta. [fonte riportata in sitografia]. [11](#), [13](#), [14](#), [18–23](#)

**CSR** Per Responsabilità Sociale delle Imprese (e delle organizzazioni) o secondo l'acronimo inglese CSR, *Corporate Social Responsibility*, si intende l'integrazione su base volontaria, da parte delle imprese, delle preoccupazioni sociali e ambientali nelle loro operazioni interessate. [fonte riportata in sitografia]. [iii](#), [6](#)

**ERP** *Enterprise Resource Planning*, è un tipo di sistema *software* che aiuta le organizzazioni ad automatizzare e gestire i processi aziendali principali per ottenere le prestazioni ottimali. Il *software* ERP coordina il flusso di dati tra i processi di un'azienda, fornendo un'unica fonte di informazioni e semplificando le operazioni nell'azienda. È in grado di collegare le attività finanziarie, della catena di approvvigionamento, delle operazioni, del commercio, dei *report*, della produzione e delle risorse umane di un'azienda in una sola piattaforma. [fonte della definizione inglese riportata in sitografia]. [1](#), [3](#), [15](#)

**IDE** *Integrated Development Environment* o ambiente di sviluppo integrato, è un *software* progettato per la realizzazione di applicazioni che aggrega strumenti di sviluppo comuni in un'unica interfaccia utente grafica. In genere è costituito da: *editor* del codice sorgente, strumenti che consentono

di automatizzare la *build* locale, un *debugger* e strumenti per l'esecuzione di test automatici.

[*fonte riportata in sitografia*]. [10](#)

**ORM** *Object-Relational Mapping*, è uno strumento che facilita l'interazione tra il codice orientato agli oggetti e i *database* relazionali. Esso permette agli sviluppatori di manipolare i dati del *database* usando oggetti e metodi del linguaggio di programmazione, anziché scrivere *query* SQL dirette. Questo approccio aumenta la produttività, migliora la manutenibilità del codice e riduce il rischio di errori legati alla gestione diretta del database . [22](#)

**Web App** L'applicazione *web*, o abbreviato *web app*, nell'ambito dell'informatica e della programmazione, si riferisce alle applicazioni accessibili e fruibili attraverso il *web*, quindi accessibili dall'utente tramite un *browser web* con una connessione attiva . [4](#), [5](#), [11](#)

# Sitografia

*Definizione API.* URL: <https://www.redhat.com/it/topics/api/what-is-a-rest-api> (visitato il 04/08/2024).

*Definizione CSR.* URL: <https://www.lavoro.gov.it/temi-e-priorita/terzo-settore-e-responsabilita-sociale-imprese/focus-on/responsabilita-sociale-imprese-e-organizzazioni/pagine/default> (visitato il 11/08/2024).

*Definizione ERP.* URL: <https://www.microsoft.com/en-us/dynamics-365/topics/erp/what-is-erp> (visitato il 11/08/2024).

*Definizione IDE.* URL: <https://www.redhat.com/it/topics/middleware/what-is-ide> (visitato il 11/08/2024).

*Funzionalità di 3CX.* URL: <https://www.3cx.it/> (visitato il 28/07/2024).

*Funzionalità di SSMS.* URL: <https://learn.microsoft.com/it-it/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16> (visitato il 28/07/2024).

*Manifesto Agile.* URL: <https://www.atlassian.com/it/agile/manifesto> (visitato il 27/07/2024).

*Scrum.* URL: <https://www.atlassian.com/it/agile/scrum> (visitato il 27/07/2024).

*Sito di VisioneImpresa.* URL: <https://www.vsh.it/> (visitato il 27/07/2024).

*Swagger.* URL: <https://www.geekandjob.com/wiki/swagger> (visitato il 04/08/2024).

*Tipologia di ticket Jira.* URL: <https://www.atlassian.com/it/software/jira/guides/issues/overview#what-are-issue-types> (visitato il 27/07/2024).