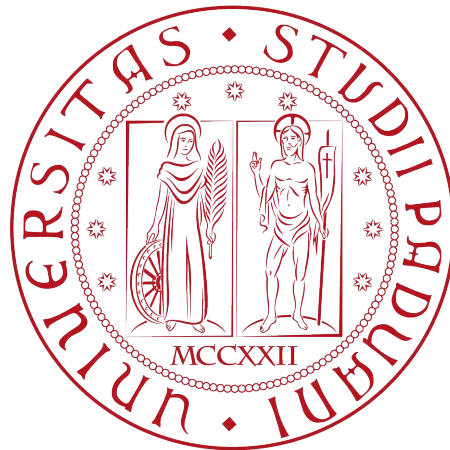


Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA “TULLIO LEVI-CIVITA”

CORSO DI LAUREA IN INFORMATICA



Un calcolatore di consumi per forni realizzato con tecnologia React

Tesi di laurea triennale

Relatore

Prof. Tullio Vardanega

Laureando

Michele Cazzaro

ANNO ACCADEMICO 2022-2023

Sommario

Scopo e obiettivi del documento

Questo documento dettaglia in toto l'esperienza di tirocinio che ho svolto, a partire dal contesto aziendale in cui mi sono immerso per 300 ore per realizzare il mio progetto di *stage*.

Il presente testo nasce per rispondere all'esigenza di certificare e documentare la mia esperienza di tirocinio, e per fornire un quadro completo del lavoro svolto, dei risultati ottenuti, e delle competenze acquisite.

Struttura del documento

Questa relazione è suddivisa in 4 capitoli:

- **L'azienda:** descrizione dell'azienda, del contesto in cui opera, e dei processi aziendali.
- **Progetto di stage: la proposta:** descrizione del progetto, degli obiettivi, e dei risultati ottenuti.
- **Progetto di stage: lo svolgimento:** descrizione delle tecnologie utilizzate durante lo *stage*, e delle competenze acquisite.
- **Valutazione retrospettiva:** valutazione retrospettiva del tirocinio, con una panoramica dei punti di forza e di debolezza.

Convenzioni tipografiche

All'interno di questo documento sono state adottate alcune convenzioni per migliorare l'esperienza di lettura:

- I termini in lingua straniera, diversa dall'italiano, sono evidenziati utilizzando il *corsivo*.
- I termini tecnici, gli acronimi e altre parole che necessitano di definizione o disambiguazione, avranno una sezione dedicata all'interno del **glossario**
- La bibliografia, in coda alla relazione, conterrà eventuali riferimenti a documenti esterni utilizzati per la stesura, o utili per fornire un quadro completo al lettore.

- Per i riferimenti a sezioni o capitoli di questo documento, utilizzo il simbolo § seguito dal numero del capitolo o della sezione. Ad esempio, il riferimento al primo capitolo sarà §1.

Ringraziamenti

Ringrazio molto i miei genitori, per la fiducia e le sponsorizzazioni elargite durante questi anni.

Desidero ringraziare il Prof. Tullio Vardanega, il mio relatore, per il suo prezioso supporto, le sue precise indicazioni, e la sua professionalità, durante la stesura di questa tesi. Il suo contributo è stato fondamentale per il mio lavoro.

Ringrazio poi gli amici tutti, sia quelli conosciuti durante il percorso, che quelli cresciuti con me, per essere stati di supporto reciproco anche nei momenti più difficili.

Padova, Settembre 2023

Michele Cazzaro

Indice

1	L'azienda	1
1.1	Presentazione	1
1.2	Processi aziendali	2
1.2.1	Organizzazione aziendale	3
1.2.2	Processi di sviluppo <i>software</i>	3
1.2.3	Digital Experience <i>team</i>	4
1.3	Tecnologie e applicazioni utilizzate	4
1.3.1	<i>Stack</i> tecnologico	4
1.3.2	Applicativi utilizzati	6
1.4	Propensione all'innovazione	7
2	Progetto di <i>stage</i>: la proposta	9
2.1	Visione aziendale degli <i>stage</i>	9
2.2	Introduzione al progetto	10
2.3	Importanza del progetto per l'azienda	10
2.3.1	Politiche <i>green</i> e responsabilità ambientale	10
2.3.2	Differenziazione dei prodotti di fascia alta	10
2.3.3	Miglioramento dell'immagine aziendale	11
2.4	Obiettivi del progetto	11
2.5	Vincoli	13
2.6	Scelta del progetto	14
2.6.1	Motivazioni	14
2.6.2	Obiettivi personali	15
3	Progetto di <i>stage</i>: lo svolgimento	16
3.1	Studio delle tecnologie	16
3.1.1	Obiettivi del periodo preliminare di studio	16
3.1.2	Metodologie utilizzate per l'apprendimento	18
3.2	Analisi dei requisiti: contesto ed esecuzione	19
3.2.1	Analisi dei requisiti in Unox	19
3.2.2	Identificazione e raccolta dei requisiti aziendali	19
3.2.3	Analisi dei requisiti	20
3.3	Progettazione	25
3.3.1	Scelte progettuali fondamentali	25
3.3.2	Diagrammi e documentazione della progettazione	28
3.4	Codifica	30
3.4.1	Sviluppo dei componenti	30
3.4.2	Creazione dei componenti più complessi	31

<i>INDICE</i>	vi
3.4.3 Sperimentazione di gerarchie di componenti	31
3.4.4 Esempi	32
3.4.5 Salvataggio di un file JSON	36
3.4.6 Strumenti di sviluppo e metodologie di <i>coding standard</i>	37
3.4.7 Creazione dello script per la generazione del <i>dataset</i>	37
3.5 Risultati raggiunti	38
3.5.1 Descrizione del tirocinio	38
3.5.2 Prodotto realizzato	39
4 Valutazione retrospettiva	42
4.1 Confronto tra gli obiettivi iniziali e i risultati ottenuti	42
4.2 Soddisfacimento degli obiettivi personali	43
4.3 Maturazione professionale	43
4.4 Conoscenze acquisite	44
4.5 Confronto con il corso di studi	45
4.6 Confronto con le aspettative	46
Acronimi e abbreviazioni	47
Glossario	48
Bibliografia	50

Elenco delle figure

1.1	Tutti i teams di Unox S.p.A. - unox.com	3
1.2	Processo di sviluppo <i>software</i> di Unox S.p.A.	5
1.3	<i>Stack</i> tecnologico di Unox S.p.A. - Credito immagini forni: unox.com	6
2.1	Flusso del wizard per il calcolatore di consumi energetici.	12
3.1	Use case individuati	23
3.2	Requisiti individuati	24
3.3	<code>RadioSelector</code> e <code>RadioGroup</code> in azione	27
3.4	<code>FormInput</code> e <code>InputGroup</code> in azione; vedi fig 3.8 per un frammento di codice che ne esplica il funzionamento	28
3.5	Diagramma dei componenti principali sviluppati	29
3.6	Diagramma dei componenti per la visualizzazione dei dati	30
3.7	Frammento di codice per la creazione di un <code>DonutChart</code>	32
3.8	Frammento di codice che implementa <code>InputGroup</code>	34
3.9	Creazione di un oggetto <code>Oven</code> mediante l'uso di un <code>factory method</code>	35
3.10	Salvataggio di un file JSON dopo l'estrapolazione dei dati relativi ai forni	36
3.11	Landing screen	39
3.12	Selezione dei parametri di utilizzo	40
3.13	Selezione dei parametri di costo	41
3.14	Alcuni componenti del riepilogo dati	41
4.1	Percentuale di requisiti raggiunti: 95%	42

Elenco delle tabelle

1.1	Tabella per il confronto tra tipologie di innovazione	8
-----	---	---

2.1	Suddivisione delle ore di lavoro per le attività di progetto.	14
3.1	Tabella di alcuni dei requisiti individuati	24
4.1	Tabella degli obiettivi raggiunti. Vedi obiettivi	42

Capitolo 1

L'azienda

Le informazioni di seguito riportate riguardo a Unox S.p.A. sono frutto della mia personale esperienza di tirocinio e di ricerca. Un appunto a mio avviso rilevante è il seguente: Unox S.p.A. è stata la prima azienda di grandi dimensioni presso la quale sono stato impiegato. Ritengo questo abbia influenzato la mia esperienza in quanto non ero familiare con le dinamiche interne di un ufficio, tantomeno con le dinamiche di un *team* di sviluppo. Ho quindi esplorato queste dinamiche durante il tirocinio.

1.1 Presentazione

Unox S.p.A. è un'azienda attiva dal 1990 nel settore della produzione di elettrodomestici professionali per la ristorazione e la panificazione. È quindi un'azienda di prodotto: il focus principale non è la produzione di *software*, ma la vendita e il supporto clienti per i propri forni. L'azienda opera a livello [Business to Business \(B2B\)](#), operando con clientela diversificata: dalle piccole panetterie alle grandi catene di ristorazione o centri cotture.

Inizialmente l'azienda sfruttò un vuoto di mercato, producendo forni destinati principalmente a fornitori di *croissant* congelati in Europa meridionale. La destinazione principale dei prodotti era la cessione in comodato d'uso a piccoli commercianti da parte di grandi fornitori.¹ La visione aziendale si allargò poi alla produzione di prodotti di qualità più elevata, destinati alla vendita diretta agli utilizzatori finali. Questo permise all'azienda di espandersi in Europa settentrionale e pose le basi per l'espansione in altri continenti.

Unox S.p.A. mantiene ad oggi diverse fasce di prodotti, che testimoniano la storia dell'azienda e la sua evoluzione nel tempo. Negli ultimi anni l'azienda ha iniziato a dare sempre più peso all'evoluzione digitale dei propri prodotti, con l'obiettivo di fornire ai clienti un'esperienza sempre più completa e soddisfacente. Tutti i modelli di punta dell'azienda, dispongono ora di un pannello touch-screen, possibilità di controllo vocale, controllo remoto, aggregazione e gestione di dati per grandi aziende ed altri accessori all'avanguardia.

Per rispondere alla necessità di evoluzione digitale, l'azienda ha iniziato a sviluppare internamente *software* per fornire questi servizi, ampliando il proprio organico con

¹Massimo Frasson. *Corporate Governance System - Teorie organizzative applicate alla realtà delle PMI familiari: Il caso Unox S.p.A.*. 2017.

sviluppatori *software*. Ad oggi dispone di 2 diversi *team* di sviluppo *software* e un *team* di [Research and development \(RD\)](#)

Nonostante sia diventata una multinazionale, l'azienda è rimasta a conduzione familiare. Inoltre, la produzione è rimasta in Italia, poco lontano dalla sede in cui è nata Unox. Lo sviluppo verso l'estero c'è stato invece in senso commerciale, dove ora l'azienda è presente con sedi commerciali.

L'azienda adotta un buon grado di integrazione verticale, producendo internamente o con l'ausilio di aziende controllate la maggior parte dei componenti dei propri prodotti. Proprio questo ha permesso a Unox di superare le *supply chain disruptions* avvenute durante la pandemia di Covid-19 senza grossi problemi produttivi, registrando anzi utili record.

1.2 Processi aziendali

Essendo Unox un'azienda di prodotto, la maggior parte dei processi aziendali da essa istanziati non riguardano la produzione di *software*. Nonostante ritengo degno di nota citare i processi di cui sono venuto a conoscenza:

- **Produzione (sviluppo, design):** Riguarda lo sviluppo e la progettazione dei prodotti dell'azienda. Comprende lo sviluppo di prodotti o servizi digitali, il miglioramento di quelli esistenti, nonché la definizione del loro design e delle specifiche tecniche.
- **Manifattura (produzione, assemblaggio):** Si occupa dell'effettiva produzione fisica dei prodotti. Include la fabbricazione, l'assemblaggio e la creazione di prodotti finiti o componenti chiave necessari per la produzione.
- **Vendita (marketing, vendita, supporto clienti):** Copre tutto ciò che è legato alla commercializzazione e alla vendita dei prodotti dell'azienda. Ciò include attività di marketing per promuovere i prodotti, il processo di vendita effettiva e il supporto post-vendita ai clienti.
- **Amministrazione (contabilità, risorse umane):** L'aspetto amministrativo si concentra sul corretto funzionamento interno dell'azienda. Questo processo comprende la gestione finanziaria e contabile dell'azienda, nonché la gestione delle risorse umane, tra cui l'assunzione, la formazione e la gestione del personale.
- **Ricerca e sviluppo (sviluppo *software*, ricerca):** È responsabile dell'innovazione e dello sviluppo di nuove tecnologie o prodotti. Include la ricerca, lo sviluppo *software* e l'implementazione di nuove idee per migliorare i prodotti esistenti o creare nuove soluzioni.
- **Logistica (trasporto, magazzino):** La logistica si concentra sulla gestione efficiente del flusso di materiali e prodotti all'interno dell'azienda. Ciò include la pianificazione del trasporto, la gestione dei magazzini e la distribuzione dei prodotti ai clienti.
- **Qualità (controllo qualità, sicurezza):** Mira a garantire che i prodotti rispettino gli standard di qualità e sicurezza. Include il controllo qualità durante la produzione, nonché l'implementazione di normative e procedure per garantire la sicurezza dei prodotti e dei processi aziendali.

Io sono rimasto totalmente estraneo ai processi manifatturieri e logistici, ma ho avuto modo di entrare in contatto con praticamente tutti gli altri. Infatti all'interno dell'azienda non vi è una compartimentazione ermetica dei processi, ma una forte interazione tra i diversi *team*. Questo non mi è stato molto utile per comprendere a pieno i processi aziendali esterni allo sviluppo *software*, ma mi ha permesso di comprendere meglio il contesto in cui opera l'azienda e di apprezzare la sua organizzazione.

1.2.1 Organizzazione aziendale

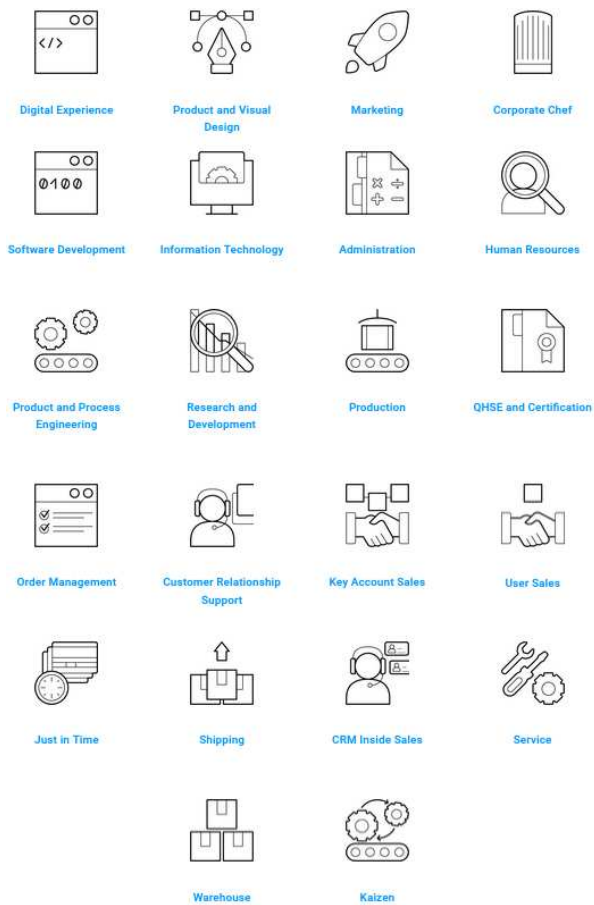


Figura 1.1: Tutti i teams di Unox S.p.A. - unox.com

1.2.2 Processi di sviluppo *software*

In azienda sono presenti diversi *team* che si occupano di sviluppare *software*. Questi sono il Digital Experience *team*, *software* Development *team*, IT *team* e il *team* di RD. Ognuno di questi *team* sviluppa *software* in modo pressochè indipendente, e ognuno di essi ha un proprio processo di sviluppo *software*. Queste differenze sono dovute ai

diversi fini dei prodotti sviluppati dai *team*. RD sviluppa *software* utile a rilevazioni o esperimenti, oltre che per eventuale analisi di dati prodotti. Il *team* IT si occupa di sviluppare e mantenere *software* interno all'azienda. Il *team software Development* sviluppa generalmente algoritmi o *software* più a basso livello per il funzionamento dei forni. Il Digital Experience *team* infine, si occupa dello sviluppo e della manutenzione di *software* a contatto con gli utenti finali: siti web e web apps, applicazioni mobile e interfaccia dei forni.

1.2.3 Digital Experience *team*

All'interno del Digital Experience *team*, quello in cui mi trovavo io durante il tirocinio, l'azienda adotta un approccio ben strutturato per la gestione dei progetti e delle attività. Questo processo si basa su una pianificazione semestrale, che consente al management di discutere e stabilire le attività principali per i successivi sei mesi. Questa pratica garantisce un'ottima visibilità sul carico di lavoro delle risorse, consentendo di bilanciare lo sviluppo di progetti strategici con la gestione delle attività quotidiane e delle richieste impreviste.

Ogni progetto attivo all'interno dell'azienda è dotato di almeno un *owner*, una figura chiave responsabile della gestione della roadmap del progetto. Questa persona si assicura che le attività siano svolte in linea con le scadenze e mantenga un costante allineamento con gli stakeholder, garantendo che le scelte progettuali, i tempi e le priorità siano chiari e coerenti. Il *team* di progetto, a sua volta, gode di una notevole autonomia nella gestione delle attività, sempre in accordo con l'*owner*.

Un elemento importante nella gestione dei progetti è rappresentato dalle riunioni ricorrenti denominate **Takt time (TT)**, un termine che deriva dalla filosofia lean production. Queste riunioni sono analoghe alla sprint review nel contesto [Agile](#) e si concentrano sulla condivisione dello stato di avanzamento del progetto, consentendo di discutere le scelte tecniche e le priorità. Ad esempio, nel caso del sito web Unox, l'*owner* tecnico, Matteo Bressan, collabora con gli "clienti" interni, come il reparto marketing, per definire le priorità relative ai compiti da assegnare al *team* del sito.

1.3 Tecnologie e applicazioni utilizzate

All'interno dell'azienda, vengono utilizzate diverse tecnologie per gestire vari aspetti della produzione *software*. In questa sezione, esploreremo le tecnologie di supporto ai processi di sviluppo *software* con cui ho acquisito esperienza diretta. Analizzeremo, in sequenza, i linguaggi di programmazione impiegati dall'azienda per sviluppare i propri prodotti, i *framework* utilizzati per strutturare tali implementazioni e gli strumenti dedicati alla comunicazione e al coordinamento delle attività.

1.3.1 *Stack* tecnologico

Linguaggi di programmazione

È interessante notare che l'azienda fa uso di diversi linguaggi di programmazione, ciascuno applicato in base al contesto specifico:

- **TypeScript**: viene utilizzato per lo sviluppo di applicazioni legate agli ultimi modelli di forni, Digital.ID, all'infrastruttura web e alle applicazioni mobili. La sua forte tipizzazione e le moderne caratteristiche lo rendono una scelta eccellente per la creazione di componenti *software* robusti e manutenibili.

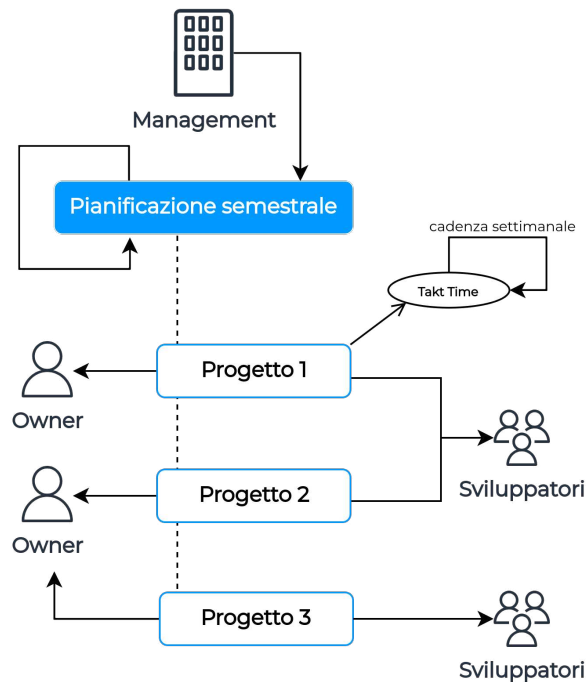


Figura 1.2: Processo di sviluppo *software* di Unox S.p.A.

- **C++**: per lo sviluppo dell'interfaccia utente dei forni Mind.Maps, C++ è il linguaggio scelto. Questo linguaggio a livello più basso fornisce il controllo necessario sugli elementi hardware, garantendo prestazioni ottimizzate per questi forni specifici.

Framework in uso

Unox dispone di uno *stack* tecnologico ben definito: vengono adottati vari *framework* per lo sviluppo *software*.

In particolare, per lo sviluppo di interfacce per i forni in TypeScript e per lo sviluppo web, è stato scelto il *framework* **React**. Per lo sviluppo di applicazioni mobile, invece, viene utilizzato **React Native**. In tutti gli ambienti di sviluppo che utilizzano TypeScript, viene utilizzato **Node.js** come runtime per l'esecuzione del codice e come package manager.

- **React**: viene impiegato per la realizzazione delle interfacce dei forni e per lo sviluppo web. La sua architettura basata su componenti e il virtual DOM lo rendono una scelta efficiente per la creazione di interfacce utente reattive e interattive.
- **React Native**: per lo sviluppo di applicazioni mobili, Unox utilizza React Native. Consente al *team* di scrivere codice una sola volta e distribuirlo su piattaforme iOS e Android, semplificando gli sforzi di sviluppo.

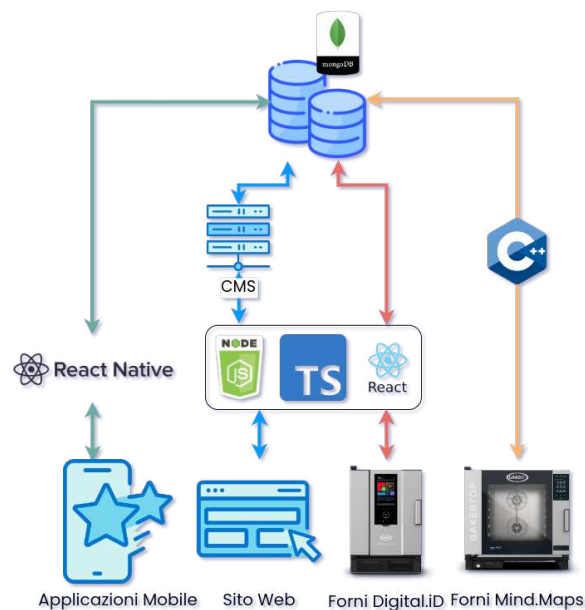


Figura 1.3: Stack tecnologico di Unox S.p.A. - Credito immagini fornite: unox.com

- **Node.js:** viene utilizzato per l'esecuzione del codice TypeScript e come gestore di pacchetti. Facilita le operazioni lato server, consentendo una gestione efficiente di compiti come il *rendering* del server e lo sviluppo di API.

MongoDB

MongoDB è impiegato per memorizzare e gestire i dati relativi ai prodotti digitali dell'azienda. La sua natura NoSQL e la flessibilità lo rendono adatto a gestire le diverse esigenze di dati delle moderne applicazioni. Durante la mia esperienza non ho avuto modo di accedere direttamente al [Database \(DB\)](#).

1.3.2 Applicativi utilizzati

Di seguito, verranno elencati gli applicativi utilizzati all'interno dell'azienda per la gestione delle attività di sviluppo *software*.

Git

Git viene utilizzato come sistema di controllo delle versioni per il tracciamento delle modifiche al codice sorgente.

Microsoft Teams

Microsoft Teams è adottato come strumento di comunicazione e collaborazione in tempo reale all'interno dell'azienda, facilitando le discussioni, le videochiamate e la condivisione di documenti. Viene utilizzato anche per la calendarizzazione di eventi e meetings. Utile sia durante il lavoro *on-premise*, sia durante lo *smart working*.

Microsoft Office

Microsoft Office è lo strumento aziendale e ufficiale per la creazione, la modifica e la condivisione di documenti, fogli di calcolo e presentazioni. Tutti i dipendenti dell'azienda hanno una licenza per l'utilizzo di Microsoft Office 365, che permette di utilizzare le applicazioni sia *online*, sia *offline*. Ancora una volta è possibile l'accesso agli strumenti aziendali a distanza in caso di *smart working*.

Microsoft Azure

Microsoft Azure è una piattaforma *cloud* utilizzata per l'*hosting* di applicazioni, servizi e dati aziendali. Viene utilizzato da Unox per l'*hosting* di alcuni dei servizi principali, tra cui il sito web. Sempre dalla suite di Azure, viene utilizzato Azure DevOps per la gestione delle attività di sviluppo *software*, tra cui la gestione dei *repository* Git, la gestione delle *build* e la gestione delle attività.

AWS

Amazon Web Services (AWS) è un altro servizio *cloud* utilizzato per le risorse di calcolo, archiviazione e servizi di rete. Alcuni dei servizi secondari sono hostati su AWS.

1.4 Propensione all'innovazione

La propensione all'innovazione tecnologica rappresenta a mio avviso una qualità importante per la sopravvivenza a lungo termine di un'azienda. Parlando di Unox nello specifico e volendo analizzare la propensione di questa organizzazione all'innovazione, partirei osservando la sua storia, brevemente raccontata in §1.1

Inizialmente l'azienda produceva poco più che forni domestici modificati in modo artigianale. Nel corso degli anni l'azienda ha continuato a modificare la propria offerta per rispondere al mercato e proponendo prodotti sempre più avanzati tecnologicamente. Nell'ultimo periodo, l'azienda sta esplorando un nuovo segmento di mercato con i propri prodotti *flagship*: i forni digitali. Questi prodotti sono dotati di un pannello touch-screen, possibilità di controllo vocale, controllo remoto, aggregazione e gestione di dati per grandi aziende ed altri accessori all'avanguardia.

Penso questo sia un buon esempio di propensione all'innovazione, in quanto l'azienda ha saputo evolversi nel tempo, adattandosi al mercato e cercando di creare una nuova nicchia.

Questo è quanto può testimoniare la storia, ma io posso aggiungere qualcosa. In azienda è presente un reparto di **RD** che vede assegnate *task* di esplorazione tecnologica e sperimentale. Un progetto di cui ho avuto modo di venire a conoscenza è la misurazione dell'influenza della posizione o velocità delle ventole interne sull'uniformità della cottura. Altri progetti che, secondo me, testimoniano la propensione all'innovazione sono ad esempio: produzione del primo forno a microonde con sistema di lavaggio automatico, studio approfondito delle cotture combinando convezione e microonde o lo sviluppo di un manteneritore di vivande a caldo.

In ambito informatico, è in corso di sviluppo un accessorio che, utilizzando un sistema di visione artificiale, permette di riconoscere il cibo inserito nel forno e di impostare automaticamente i parametri di cottura.

Penso che questi progetti possano testimoniare la propensione all'innovazione di Unox, ma non sono certo che sia sufficiente o del tutto oggettivo parlare solo per esperienza

personale. Per questo motivo ho deciso di analizzare la propensione all'innovazione dell'azienda utilizzando il *framework* esposto dai ricercatori Tidd, Bessant e Pavitt². Secondo questo *framework* l'azienda in esame rientra nella categoria di **Innovazione incrementale**, ovvero un'azienda che si concentra su miglioramenti incrementali di prodotti e processi esistenti. Ciò non significa che vengono ignorati i miglioramenti radicali, ma che questi non sono l'obiettivo principale dell'azienda.

Tabella 1.1: Tabella per il confronto tra tipologie di innovazione

Innovazione	Sintomo	Implicazioni
Incrementale	Ottimizzazione e il miglioramento graduale delle offerte esistenti	L'azienda può mantenere la stabilità e la competitività a breve termine. Gli investimenti sono generalmente meno rischiosi. Tuttavia, può essere vulnerabile a cambiamenti repentini o a concorrenti che introdurranno innovazioni più radicali.
Radicale	L'azienda sta sviluppando nuovi concetti, prodotti o tecnologie rivoluzionari.	Questo tipo di innovazione può aprire nuovi mercati e offrire un vantaggio competitivo significativo. Richiede risorse considerevoli e comporta un alto grado di rischio. Il successo non è garantito, e il tempo di sviluppo può essere lungo.
Architetturale	L'azienda sta cercando di ridefinire o riconfigurare sistemi o tecnologie esistenti in modi innovativi.	Questo tipo di innovazione può migliorare l'efficienza, ridurre i costi o aprire nuove opportunità di mercato senza dover reinventare completamente la ruota. Richiede una comprensione approfondita degli attuali sistemi e delle tecnologie, nonché la capacità di implementare con successo le modifiche architetturali.

²Joe Tidd, John Bessant e Keith Pavitt. *Managing Innovation: Integrating Technological, Market and Organizational Change*. Wiley, 2001.

Capitolo 2

Progetto di *stage*: la proposta

2.1 Visione aziendale degli *stage*

Unox sembra dare molta importanza ai nuovi talenti. Tenendo a mente che è una azienda che ha subito, e auspicabilmente per loro continuerà a subire, una forte espansione, è importante un afflusso di nuove risorse umane. L'azienda potrebbe rivolgersi a figure con esperienza per acquisire nuovi talenti, ma non credo sia quello che sta succedendo. All'interno dell'azienda l'età media si aggira attorno ai 30 anni, testimoniando una buona presenza di giovani.

Unox porta avanti diverse iniziative per avvicinarsi a giovani: in primis la convenzione con l'Università di Padova, che permette di ospitare tirocinanti e studenti in *stage*. Parlando con i miei colleghi, ho scoperto che molti di loro sono stati degli alumni dell'ateneo padovano, e che buona parte di loro è venuta in contatto con l'azienda attraverso la possibilità di un tirocinio, curricolare o meno. Credo sia appurato che Unox da valore ai tirocini come strumento di reclutamento.

Non penso che la visione aziendale degli *stage* si fermi qui, in quanto ho avuto modo di provare sulla mia pelle e posso testimoniare che l'azienda si impegna a far sì che lo *stage* sia un'esperienza formativa e di crescita per lo stagista, oltre che un'opportunità per l'azienda di reclutare nuove risorse. Ero infatti in contatto con l'organizzazione da ben prima dell'inizio del mio tirocinio, e ho avuto modo di valutare e scegliere, anche con il consiglio del tutor aziendale, i progetti a me proposti. Durante il tirocinio, avevo a disposizione tutto il tempo necessario all'auto-apprendimento, e in caso di difficoltà venivo assistito dai miei colleghi. L'azienda spende quindi risorse per far sì che il tirocinio sia un'esperienza positiva per lo stagista, e non solo per l'azienda. Per quanto riguarda i progetti assegnati ai nuovi talenti si può dire siano proporzionati alla durata dello *stage* e all'esperienza che l'assegnatario ha maturato. Per esempio, il progetto che mi è stato assegnato è stato pensato per poter essere svolto in autonomia da un laureando del mio indirizzo. Un collega laureando magistrale, ha ricevuto un progetto più corposo e proporzionato alla sua esperienza.

Unox non utilizza solo i tirocini per entrare o restare in contatto con i giovani del settore: altre iniziative quali la possibilità di avere uno *student job* o quella di lavorare come dipendente part-time durante l'anno accademico, sono attività che a mio avviso testimoniano la volontà di dare valore agli studenti e ai giovani in generale.

2.2 Introduzione al progetto

A causa di ritardi sia da parte mia che dall'azienda ospitante, purtroppo, non è stato possibile portare a termine il progetto di *stage* iniziale inizialmente discusso con il tutor aziendale. Questo progetto era incentrato sullo sviluppo di un sistema di *testing end-to-end* automatico specifico per i forni Unox, un compito ambizioso ma cruciale per garantire la qualità dei prodotti. Tuttavia, comprendendo l'importanza di adattarsi alle circostanze, ho collaborato con il mio tutor aziendale per trovare una soluzione alternativa che potesse ancora apportare valore alla mia esperienza.

Durante uno scambio di messaggi con Matteo, il mio tutor aziendale, lui mi ha presentato un progetto alternativo, che si è rivelato essere un'ottima opportunità per me, in quanto ho particolarmente a cuore le tematiche ambientali. La nuova direzione che abbiamo intrapreso è stata sviluppare un'applicazione web per il calcolo dei consumi energetici dei forni Unox. Questa decisione è stata motivata dalla crescente preoccupazione per l'ambiente e dalla necessità di fornire ai clienti uno strumento che consentisse loro di valutare l'impatto ambientale dei loro prodotti. Inoltre, l'azienda ha compreso che offrire un calcolatore di consumi energetici avrebbe reso i loro prodotti di fascia alta ancora più attraenti per i clienti, aprendo così la possibilità di aumentare il fatturato.

Le specifiche per questa nuova applicazione erano chiare fin dall'inizio dei colloqui con il tutor aziendale. Doveva trattarsi di un calcolatore di consumi energetici accessibile direttamente ai clienti Unox tramite una semplice pagina web. Questo avrebbe permesso ai clienti di inserire i parametri specifici dell'utilizzo previsto del forno e ottenere una stima precisa dei consumi energetici. L'obiettivo era di rendere questa funzionalità facilmente accessibile da qualsiasi dispositivo, in modo che i clienti potessero utilizzarla in qualsiasi momento.

2.3 Importanza del progetto per l'azienda

Sin dall'inizio, è emerso chiaramente che il progetto da me intrapreso non rappresentasse una priorità vitale per Unox. Tuttavia, ciò non significa che il progetto fosse trascurabile o privo di significato per l'azienda. Al contrario, questo progetto ha portato con sé una serie di benefici strategici e operativi che hanno contribuito positivamente all'immagine e al posizionamento di Unox nel mercato.

2.3.1 Politiche *green* e responsabilità ambientale

Uno dei principali motivi per cui questo progetto è stato considerato importante è la crescente attenzione all'ambiente e alla sostenibilità nell'attuale panorama aziendale. L'azienda Unox, come molte altre, è orientata a diventare sempre più sostenibile e a ridurre l'impatto ambientale dei suoi prodotti. L'introduzione di un calcolatore di consumi energetici è stata una dimostrazione tangibile dell'impegno di Unox nei confronti delle politiche *green* e della sua responsabilità ambientale. Questo strumento ha permesso ai clienti di valutare l'impatto ambientale dei prodotti Unox, evidenziando il desiderio dell'azienda di operare in modo trasparente e responsabile.

2.3.2 Differenziazione dei prodotti di fascia alta

Un altro aspetto cruciale è legato alla strategia di mercato di Unox. L'azienda è conosciuta per i suoi prodotti di fascia alta e mira a mantenere questa posizione nel

mercato. Il calcolatore di consumi energetici è stato un elemento di differenziazione chiave per i prodotti Unox di fascia alta. Consentendo ai clienti di valutare l'efficienza energetica dei forni, Unox ha reso i suoi prodotti più appetibili per coloro che cercano soluzioni di alta qualità e sostenibili. Questo ha aperto la possibilità per l'azienda di aumentare il fatturato e conquistare una fetta di mercato sempre più consapevole delle tematiche ambientali.

2.3.3 Miglioramento dell'immagine aziendale

Un aspetto cruciale da considerare è il potenziale miglioramento dell'immagine aziendale che il progetto può apportare. Nonostante il calcolatore di consumi energetici non sia ancora stato rilasciato, l'azienda Unox dimostra la sua dedizione alla sostenibilità e alla responsabilità ambientale attraverso la sua intenzione di sviluppare e introdurre questo strumento. La semplice idea di offrire ai clienti uno strumento per valutare l'efficienza energetica dei prodotti Unox contribuisce a creare un'immagine di Unox come un'azienda all'avanguardia nell'adozione di tecnologie sostenibili. Questa iniziativa può attirare l'attenzione positiva da parte di clienti, partner commerciali e *opinion leader* nel settore. È importante sottolineare che il progetto non rappresenta solo un passo verso la sostenibilità ambientale, ma anche un segno di impegno nei confronti dei clienti. Anche se il calcolatore non è ancora operativo, il fatto che l'azienda stia lavorando a questo strumento dimostra la sua volontà di fornire ai clienti strumenti informativi per prendere decisioni più consapevoli riguardo ai propri acquisti.

In sintesi, sebbene il progetto potesse non rappresentare una priorità immediata per Unox, ha apportato una serie di benefici significativi, tra cui il supporto alle politiche ambientali, la differenziazione dei prodotti di fascia alta e il miglioramento dell'immagine aziendale. Questi risultati strategici e operativi hanno dimostrato che il progetto aveva un'importanza reale e misurabile per l'azienda.

2.4 Obiettivi del progetto

Il progetto del calcolatore di consumi energetici per i forni Unox è stato delineato con diversi obiettivi chiave che definiscono la sua portata e il suo scopo. Questi obiettivi sono stati stabiliti per garantire che il progetto fosse ben definito e avesse un chiaro percorso di sviluppo.

Obiettivi obbligatori

- **Sviluppo di elementi altamente riutilizzabili per il *design system*:** Uno degli obiettivi principali è stato creare componenti e elementi di design altamente riutilizzabili. Questi elementi dovrebbero essere integrati nel *design system* dell'azienda, consentendo una coerenza visiva e funzionale tra diverse applicazioni e servizi offerti da Unox.
- **Astrazione del modello matematico:** Un altro obiettivo cruciale è stato l'astrazione del modello matematico utilizzato per calcolare i consumi energetici. L'obiettivo era creare un modello matematico che potesse essere facilmente compreso, adattato e ampliato per futuri sviluppi. Questa astrazione avrebbe consentito di gestire in modo più efficiente le complessità dei calcoli energetici.
- **Implementazione di grafici:** Un obiettivo essenziale è stato l'implementazione di grafici per visualizzare in modo intuitivo i dati relativi ai consumi energetici.

Questi grafici sono stati considerati uno strumento fondamentale per rendere più accessibili e comprensibili le informazioni ai clienti.

- **Creazione di una pagina web:** Il progetto mirava a sviluppare una pagina web accessibile direttamente ai clienti e ai commerciali Unox. Questa pagina web avrebbe costituito il vero e proprio calcolatore di consumi, guidando gli utenti attraverso uno *wizard*. Durante le prime riunioni in azienda, sono emersi la struttura e il flusso del *wizard* per il calcolatore di consumi. Questo avrebbe condotto gli utenti attraverso una serie di passaggi:



Figura 2.1: Flusso del *wizard* per il calcolatore di consumi energetici.

L'obiettivo era rendere l'interfaccia intuitiva e di facile utilizzo, in modo che i clienti potessero ottenere le informazioni desiderate senza difficoltà ed in autonomia.

Obiettivi desiderabili

In aggiunta agli obiettivi obbligatori precedentemente delineati, sono stati identificati obiettivi desiderabili che, se raggiunti, avrebbero apportato ulteriori benefici alla completezza del progetto, alla mia esperienza e all'azienda.

- **Aderenza al 100% alle specifiche grafiche:** Un obiettivo desiderabile era raggiungere una totale aderenza alle specifiche grafiche stabilite dall'azienda. Questo avrebbe garantito una coerenza visiva completa tra l'applicazione del calcolatore di consumi energetici e il resto dei prodotti e servizi offerti da Unox, contribuendo a rafforzare l'identità visiva dell'azienda.
- **Performance ottimizzate:** Un altro obiettivo desiderabile era ottimizzare le prestazioni dell'applicazione in modo da non compromettere la velocità e la reattività. Questo avrebbe contribuito a garantire una buona esperienza utente, evitando ritardi nell'elaborazione dei calcoli e nella visualizzazione dei risultati. Inoltre, l'ottimizzazione delle prestazioni avrebbe favorito una valutazione positiva da parte di strumenti come Google PageSpeed Insights, riflettendo un impegno per fornire un'applicazione di alta qualità. Questa operazione sarebbe stata svolta a prescindere: il sito aziendale non può permettersi di avere un punteggio basso su Google PageSpeed Insights. Nonostante ciò essendo il [Search Engine](#)

Optimization un argomento piuttosto complesso, non è stato inserito come obiettivo obbligatorio.

Questi obiettivi desiderabili sono stati identificati per migliorare ulteriormente il progetto e per contribuire agli obiettivi strategici di Unox.

Più in generale, questi obiettivi sono stati definiti per garantire che il progetto fosse ben strutturato e allineato alle esigenze aziendali. L'obiettivo principale era sviluppare un'applicazione che offrisse un servizio utile e di alta qualità ai clienti Unox, contribuendo allo stesso tempo al miglioramento dell'immagine aziendale e alla promozione di pratiche sostenibili.

2.5 Vincoli

Nel corso dello sviluppo del progetto relativo al calcolatore di consumi energetici per i forni Unox, è emersa una serie di vincoli che hanno delineato il contesto in cui il lavoro è stato condotto. È fondamentale considerare questi vincoli poiché hanno influenzato le scelte progettuali e l'approccio alla realizzazione dell'applicazione. I principali vincoli possono essere suddivisi in diverse categorie, tra cui vincoli aziendali, tecnologici, temporali e di design.

Vincoli tecnologici

Uno dei vincoli principali era il rispetto delle tecnologie adottate dall'azienda Unox. L'applicazione doveva essere sviluppata utilizzando le tecnologie e gli strumenti già in uso all'interno dell'azienda per garantire l'integrazione e la coerenza con l'ecosistema tecnologico esistente. Un certo grado di libertà mi è stato fornito dove possibile, ma nella maggior parte dei casi ho dovuto attenermi alle tecnologie e agli strumenti già utilizzati da Unox.

Vincoli temporali

Un vincolo temporale cruciale era rappresentato dalla data di conclusione del periodo di *stage*, fissata per il 5 settembre. Ciò ha imposto una scadenza rigida per il completamento del progetto.

Inoltre, gli orari di lavoro erano limitati dall'orario di apertura dell'azienda. L'ingresso era consentito dalle 7:30 alle 9:00 e l'uscita dalle 16:30 alle 18:00. Questi orari dovevano essere rigorosamente rispettati.

Per garantire il progresso del progetto, è stata anche stabilita una suddivisione temporale dettagliata nel piano di lavoro.

Vincoli di design

I vincoli di design includevano il rispetto del *design system* aziendale e delle specifiche grafiche fornite dall'azienda. L'applicazione doveva essere allineata al *design system* esistente e rispettare le palette di colori e le linee guida visive stabilite dall'azienda. Inoltre, era essenziale rispettare le specifiche grafiche fornite per garantire la coerenza visiva tra l'applicazione del calcolatore di consumi energetici e altri prodotti e servizi offerti da Unox.

Durata in ore	Descrizione dell'attività
40	Inserimento in azienda
40	Studio dell'architettura e delle tecnologie rilevanti
80	Sviluppo dell'architettura <i>software</i> della pagina web
80	Sviluppo di grafici ed altri componenti complessi
80	Test e ottimizzazione delle prestazioni pagina web
Totale ore	320

Tabella 2.1: Suddivisione delle ore di lavoro per le attività di progetto.

2.6 Scelta del progetto

La scelta di affrontare il progetto relativo al calcolatore di consumi energetici per i forni Unox è stata motivata da una combinazione di interessi personali, opportunità di apprendimento e rilevanza aziendale. Questa sezione del rapporto di *stage* espone le ragioni che hanno guidato la mia decisione.

2.6.1 Motivazioni

Interesse personale

Fin dall'inizio del mio percorso di studi in informatica, ho avuto un forte interesse per le tematiche ambientali e per l'applicazione della tecnologia per affrontare le sfide legate alla sostenibilità. La possibilità di lavorare su un progetto che unisse le mie competenze informatiche alla promozione della sostenibilità ambientale è stata un'opportunità entusiasmante. La creazione di uno strumento che aiutasse i clienti Unox a valutare e ridurre il consumo energetico dei loro forni ha rappresentato una sfida stimolante e significativa per me. Ritengo interessante lo sviluppo di applicazioni web: non avendo mai lavorato con un *framework* come React, ed essendo interessato allo sviluppo web di applicazioni, ho colto l'occasione per imparare qualcosa di nuovo. Inoltre, il progetto mi ha permesso di acquisire conoscenze nel campo della sostenibilità energetica e dei calcoli energetici, argomenti che ho affrontato con grande interesse.

Opportunità di apprendimento

Il progetto offriva un'opportunità di apprendimento unica in diversi aspetti. In primo luogo, mi ha consentito di approfondire le mie competenze nello sviluppo web, in particolare nella creazione di interfacce utente intuitive e nella gestione di dati non aggregati. In secondo luogo, ho avuto l'opportunità di acquisire conoscenze nel campo dei calcoli energetici e una infarinatura in sostenibilità energetica, argomenti che ho affrontato con grande interesse. Inoltre, lavorare all'interno dell'ambiente aziendale di Unox mi ha permesso di sperimentare le dinamiche del mondo del lavoro, interagire con professionisti e affinare le mie competenze di comunicazione e *problem solving*.

Esperienza pregressa

Per quanto l'azienda fosse disponibile ad aiutarmi ad apprendere, ritengo che aver avuto un minimo di esperienza pregressa mi abbia donato un grado di serenità che non sarei stato in grado di avere se avessi lavorato in un ambito a me totalmente sconosciuto. Avendo già avuto l'opportunità di lavorare con TypeScript e Node.js per

dei progetti personali, mi sentivo più sicuro di me, ed è quindi stato più facile scegliere questo progetto.

Rilevanza aziendale

La scelta del progetto è stata influenzata anche dalla rilevanza dell'azienda Unox. Non conoscevo questa azienda prima dei contatti per il tirocinio, ma sono rimasto colpito dal fatto che un'azienda di tali dimensioni esistesse a pochi chilometri da casa mia. L'idea di lavorare in una azienda locale, ma allo stesso tempo con un buon grado di internazionalizzazione e modernità era allettante. La possibilità di contribuire con le mie competenze a un'azienda del mio territorio ha reso questo progetto ancora più significativo per me.

2.6.2 Obiettivi personali

Prima dell'inizio della mia esperienza di tirocinio, ho stabilito alcuni obiettivi personali che avrei voluto raggiungere. Questi obiettivi sono stati definiti per garantire che il mio tirocinio fosse un'esperienza di apprendimento significativa e per massimizzare il valore che avrei potuto trarre dal mio lavoro. I principali obiettivi personali erano:

Consolidamento delle competenze di sviluppo web Avendo già una minima esperienza nello sviluppo web, ho identificato il consolidamento di queste competenze come un obiettivo fondamentale. Volevo approfondire e rafforzare le mie conoscenze, affinando le abilità esistenti e scoprendo nuovi metodi e best practice nel campo dello sviluppo web.

Uso di una libreria a me nuova L'utilizzo di una libreria sconosciuta rappresentava un'opportunità di apprendimento significativa. Ho voluto sfidarmi ad acquisire competenze in una nuova libreria o framework, ampliando così il mio bagaglio tecnologico e sfruttando l'occasione per imparare qualcosa di nuovo.

Esperienza del ruolo di sviluppatore La mia curiosità mi ha spinto a voler comprendere appieno cosa significhi lavorare come sviluppatore. Ho fissato come obiettivo assorbire al massimo l'esperienza pratica, imparando dai compiti assegnati e dall'interazione con il team, allo scopo di avere una visione più concreta di ciò che potrebbe essere il mio futuro lavoro.

Capitolo 3

Progetto di *stage*: lo svolgimento

3.1 Studio delle tecnologie

La prima parte del mio tirocinio in Unox, è stata completamente dedicata ad un periodo di inserimento in azienda e di studio delle tecnologie da utilizzare. Non sapevo che cosa mi avrebbe aspettato né come comportarmi, ma riflettendo prima dell'inizio dello *stage*, dialogando con il mio referente e rileggendo i documenti a mia disposizione sono riuscito a farmi un'idea migliore riguardo al periodo di studio e inserimento in azienda. Il fine di questo periodo era farmi avere le necessarie conoscenze pregresse per poter svolgere senza intralcio il progetto proposto. Poiché ciò fosse possibile, dovevo conoscere la struttura e le dinamiche interne al mio ufficio, protocolli di comunicazione, convenzioni e usi. Oltre a questo necessitavo di conoscenze tecniche: dovevo sapere quali e in che misura fossero in linguaggi di codifica utilizzati, oltre che familiarizzarmi con essi.

3.1.1 Obiettivi del periodo preliminare di studio

Più chiaramente, gli obiettivi di questo periodo erano:

Conoscere il luogo di lavoro Questo obiettivo includeva l'ottenimento di conoscenze sul *layout* dell'ufficio, sui protocolli aziendali, sugli strumenti di comunicazione, sulle persone e sui rispettivi ruoli. Inoltre, era importante comprendere l'utilizzo di applicazioni specifiche, come Microsoft Teams, per facilitare la comunicazione interna.

Comprendere le tecnologie aziendali Un altro aspetto fondamentale era l'apprendimento delle tecnologie utilizzate in azienda, compresi i linguaggi di programmazione, i *framework* e le librerie. Ho dedicato tempo a studiare la documentazione e ad eseguire esercizi pratici per padroneggiare queste tecnologie.

Familiarizzare con il *Design System* L'azienda aveva definito uno standard visivo e funzionale noto come "*Design System*" che doveva essere seguito nello sviluppo delle applicazioni. Era cruciale comprenderne le linee guida, tra cui specifiche grafiche, palette di colori e componenti di design.

Conoscere il luogo di lavoro

Durante il periodo iniziale del mio tirocinio presso Unox, mi sono posto diversi obiettivi chiave, finalizzati a garantire una solida integrazione nell'ambiente aziendale e a prepararmi per il progetto imminente. In particolare:

- Comprendere l'ambiente di lavoro: acquisire una conoscenza completa dell'ambiente fisico e operativo dell'ufficio, compresi i luoghi fisici come uffici, sale riunioni, aree comuni e servizi come la mensa e i bagni. Inoltre, familiarizzarsi con i protocolli aziendali e le procedure operative, inclusi i mezzi di comunicazione interna ed esterna.
- Riconoscere le persone e i ruoli: identificare e stabilire relazioni con i colleghi, riconoscendo i loro ruoli e le responsabilità all'interno dell'azienda. Questo include comprendere la struttura gerarchica e le catene di comando.
- Apprendere gli strumenti aziendali: familiarizzarsi con gli strumenti *software* utilizzati nell'azienda, compresi sistemi di gestione di progetti, applicazioni di comunicazione e altri *software* aziendali. Questo include l'uso di piattaforme come Microsoft Teams per la comunicazione interna.
- Navigare la logistica quotidiana: comprendere e aderire alle procedure quotidiane, come il timbrare il cartellino, gestire le presenze e la postazione di lavoro. Questo include la familiarità con le procedure per la pausa pranzo, il ristoro e altre esigenze quotidiane.

Comprendere le tecnologie aziendali

Nel corso del mio tirocinio presso Unox, ho dedicato una parte significativa del tempo all'approfondimento delle tecnologie aziendali. Questo periodo è stato fondamentale per acquisire una solida comprensione delle tecnologie utilizzate nell'ambito del progetto di calcolatore dei consumi energetici per i forni Unox. Gli obiettivi principali di questi momenti erano:

- Conoscenza dei principali progetti aziendali: per poter studiare con maggiore efficacia le tecnologie in uso, necessitavo di avere un quadro generale un po' più definito. Proprio a questo scopo, assieme al tutor aziendale, ho individuato la necessità di conoscere i principali progetti di sviluppo presenti e passati. Questo mi avrebbe permesso di comprendere meglio le tecnologie in uso e di avere un quadro più chiaro di come queste venissero utilizzate.
- Conoscenza delle tecnologie in uso: più approfonditamente, una volta conosciuti i progetti ho provato ad individuare i singoli strumenti utilizzati, provando ad identificare librerie e *framework* esterni con i quali non avevo confidenza, o linguaggi che avrei dovuto eventualmente imparare.
- Studio delle tecnologie in adozione: una volta che le conoscenze necessarie in ambito di tecnologie di sviluppo erano più chiare ho potuto procedere con lo studio delle suddette. Avere una comprensione approfondita delle tecnologie principali, e una conoscenza quanto meno *surface-level* di quelle secondarie, mi avrebbe permesso di affrontare il progetto con maggiore sicurezza.

Allineamento al *Design System*

Per assicurarsi che l'applicazione da sviluppare fosse coerente con il *Design System* aziendale, esisteva la necessità che comprendessi che cosa è e quali sono i principi di utilizzo alla base di un *Design System*. Solo comprendendo appieno le linee guida grafiche, la palette dei colori e le componenti di design esistenti questo sarebbe stato possibile.

Il *Design System* aziendale rappresenta un elemento chiave per mantenere un aspetto coerente tra le applicazioni aziendali. Ho studiato attentamente le specifiche grafiche, la palette dei colori e le componenti di design esistenti all'interno del *Design System* per garantire che l'applicazione che stavo per sviluppare rispettasse appieno le linee guida aziendali.

3.1.2 Metodologie utilizzate per l'apprendimento

Durante il mio percorso di *stage* presso Unox, ho riconosciuto l'importanza di acquisire rapidamente le competenze necessarie per affrontare con successo il progetto assegnato. La sezione §3.1.1 ha già delineato gli obiettivi principali di questa fase, ora concentreremo l'attenzione sulle metodologie adottate per raggiungere tali obiettivi.

Approccio allo studio delle tecnologie

Un approccio metodico è stato cruciale per l'apprendimento delle tecnologie aziendali. La mia strategia si è basata su diverse fasi: **in primo luogo**, ho acquisito una visione generale dell'ambiente di lavoro, comprendendo l'organizzazione dell'ufficio, i protocolli aziendali e le procedure operative. Questo è avvenuto attraverso interazioni sociali, utilizzando un buon grado di curiosità e onestà, mettendosi in gioco senza paura di sbagliare. Tutto ciò mi ha permesso di sentirmi a mio agio nell'ambiente e di comunicare efficacemente con i colleghi. **In secondo luogo**, ho individuato grazie all'aiuto del tutor aziendale i progetti chiave e ho compreso come le tecnologie fossero state utilizzate in passato e fossero applicate attualmente. Questo è avvenuto grazie ad uno scambio di informazioni informale, consultando le pagine dei progetti e cercando di risolvere i dubbi che sorgevano da parte mia. Questa fase mi ha fornito un contesto più ampio per comprendere il ruolo delle tecnologie all'interno dell'azienda. Successivamente, ho studiato le tecnologie specifiche, tra cui linguaggi di programmazione, *framework* e librerie. Questo studio è stato guidato dalla documentazione ufficiale e dalla pratica attiva attraverso l'esecuzione di esercizi mirati.

Utilizzo di materiale formativo gratuito

Per approfondire le mie conoscenze tecniche, ho fatto ampio uso di risorse di disponibili gratuitamente *online*. Questa modalità di apprendimento mi ha consentito di accedere a corsi e tutorial riguardanti React e Typescript. In particolare, per apprendere React ho utilizzato una serie di video-corsi gratuiti, così come la documentazione ufficiale. Per quanto riguarda Typescript invece avendo già un po di esperienza con il linguaggio, ho letto la documentazione ufficiale quando ne sentivo la necessità, per colmare delle lacune.

Collaborazione con colleghi

Una risorsa inestimabile durante il mio periodo di inserimento in azienda è stata la collaborazione con i colleghi più esperti nell'azienda. Per affrontare alcuni temi, come

ad esempio l'utilizzo o la creazione di nuove funzionalità per il [Content Management System](#) aziendale o il collegamento al [DB MongoDB](#) ho interagito con loro. Ho avuto modo di venire affiancato in particolare al mio referente aziendale, ma tutti i colleghi si sono messi a disposizione per aiutarmi durante o in seguito al mio periodo di studio. Effettuando alcuni esercizi di prova ho ricevuto dei feedback di cui ho cercato di far tesoro per il resto del periodo di tirocinio. Questa collaborazione ha accelerato notevolmente il mio apprendimento e mi ha fornito un'ottica pratica sulle migliori pratiche all'interno dell'azienda.

Applicazione pratica delle conoscenze

Oltre allo studio teorico, ho applicato attivamente le competenze acquisite. L'apprendimento attraverso la pratica è stato un passo cruciale per consolidare le conoscenze e padroneggiare le tecnologie. Ho affrontato progetti o esercizi specifici che coinvolgevano le tecnologie oggetto di studio sottoponendomi poi ad una revisione da parte dei colleghi. Ad esempio, per quanto riguarda React, ho creato un progetto [Hello World](#) per testare le conoscenze acquisite, per poi provare qualche piccolo progetto suggerito dai corsi che stavo seguendo.

3.2 Analisi dei requisiti: contesto ed esecuzione

Durante il primo periodo del mio progetto di *stage*, ho svolto la funzione di analista al fine di individuare i requisiti del progetto che avrei affrontato.

3.2.1 Analisi dei requisiti in Unox

Dialogando con i colleghi ho scoperto che la redazione di un documento di analisi dei requisiti esaustivo non è una operazione abitualmente svolta all'interno dell'azienda. O perlomeno, non è usanza comune effettuare un'operazione simile per progetti così piccoli, ai quali vengono assegnate risorse limitate. Non è chiaro per me se progetti più grandi e indipendenti ricevano un trattamento simile, ma penso sarebbe impossibile e alquanto improbabile. Osservando la scala di progetti in fase di sviluppo in azienda infatti trovo difficile che le analisi dei requisiti non siano mai state effettuate. Tuttavia, grazie anche al progetto di Ingegneria del *software* svolto durante il mio percorso di studi, l'ho ritenuto un passaggio utile anche per un progetto da portare avanti autonomamente. Durante e dopo il processo di redazione mi sono sentito in grado di donare una visione più chiara del progetto, una comprensione migliore dei requisiti e ho ottenuto una maggiore sicurezza nello sviluppo.

Non essendo questa una pratica comune, o per lo meno non che ne sia a conoscenza, per la redazione del documento di Analisi dei requisiti della web-app, mi sono mosso in modo indipendente, sia per quanto riguarda la raccolta che la formalizzazione dei requisiti. Il documento prodotto potrebbe non essere conforme agli standard dell'industria, ma è stato certamente di vitale importanza per la comprensione del progetto e per completarne con successo lo sviluppo.

3.2.2 Identificazione e raccolta dei requisiti aziendali

Nel contesto del mio tirocinio presso Unox, ho adottato un approccio pragmatico per individuare e raccogliere i requisiti aziendali necessari per il progetto di calcolatore dei consumi energetici. Benché l'azienda non abbia una pratica consolidata di redazione

formale dei requisiti, ho ritenuto essenziale svolgere questa attività per garantire una comprensione chiara del progetto e dei suoi obiettivi.

Metodologia di identificazione dei requisiti

Ho iniziato il processo di identificazione dei requisiti attraverso incontri approfonditi con i colleghi del mio *team* e con altri stakeholder chiave. Questi incontri, sia in formato di gruppo che one-on-one, sono stati preziosi per chiarire i dettagli del progetto e definire i requisiti necessari. Nonostante l'assenza di un *framework* strutturato, ho adottato un approccio interattivo, ponendo domande chiave come:

- Quali sono le funzionalità e le caratteristiche principali richieste per il progetto?
- Ci sono vincoli tecnici, finanziari o di risorse che devono essere presi in considerazione?
- Chi sono gli utenti finali dell'applicativo e quali sono le loro specifiche esigenze?

Coinvolgimento degli *stakeholders*

Durante questo processo, ho identificato il manager del mio *team* e il mio referente aziendale come i principali stakeholder del progetto. Sono stati fondamentali nell'apportare una visione chiara degli obiettivi e delle aspettative aziendali. Inoltre, sono stati organizzati incontri specifici con membri del *team* di Design e del *team* di RD per esporre le loro esigenze e ottenere un feedback costruttivo. Durante questi meeting gli *stakeholders* hanno esposto a me e agli altri coinvolti nel progetto quelle che erano le aspettative dell'applicativo una volta terminato. Ognuna delle parti ha poi contribuito alla discussione esponendo diverse tipologie di problemi che avrebbero potuto sorgere durante lo sviluppo. Questo ha permesso di individuare e risolvere alcuni problemi prima ancora che si presentassero, oltre che a rendere un po' più specifici i requisiti dell'applicativo.

Documentazione dei requisiti

Per catturare e organizzare le informazioni raccolte durante gli incontri, ho adottato un approccio basato su appunti. Durante le riunioni, ho preso appunti rapidi su carta o utilizzando strumenti elettronici per assicurarmi di non perdere alcun dettaglio. Successivamente, ho elaborato questi appunti per creare un documento di analisi dei requisiti. Questo documento ha fornito una struttura chiara per organizzare i requisiti e comprendere meglio il contesto del progetto.

3.2.3 Analisi dei requisiti

Obiettivi del prodotto

L'obiettivo del progetto *software* (sw) desiderato è la creazione di una "fascia" includibile tramite l'uso del CMS aziendale all'interno di pagine del sito web. Ciò implica lo sviluppo di un componente React complesso, e componenti più semplici; alcuni dei quali saranno inclusi nel *Design System* aziendale.

Funzionalità del prodotto

Le funzionalità esposte dal prodotto all'utente finale dovranno essere:

- La possibilità di selezionare il forno per il quale effettuare il calcolo dei consumi
- La possibilità di selezionare il tipo di utilizzo che si effettua del prodotto
- La possibilità di visualizzare un riepilogo dei dati inseriti, nonché le stime di consumo effettuate sulla base dei dati ricevuti.

Caratteristiche degli utenti

La *user base* attesa è un *subset* degli utenti del sito unox.com. Pertanto sarà un'utenza con interesse per il mondo della cucina di alto livello: pasticceria, gastronomia, cucine ad alti volumi. In particolare, ci si aspetta che gli utenti che desiderano calcolare i consumi dei prodotti prestino una particolare attenzione all'efficienza energetica e/o al possibile risparmio economico.

Vincoli generali

L'utente, per usufruire del servizio, necessita di un browser web e di una connessione ad internet.

Il prodotto finale sarà quindi un componente *React* per guidare gli utenti nel processo di selezione del proprio prodotto e nell'inserimento dei dati relativi al consumo. Pensando a come questo sarebbe stato effettivamente utilizzato, ho individuato i seguenti attori:

Utente generico L'utente generico è un qualsiasi utilizzatore del sito web di Unox che accede al servizio.

Cliente Il Cliente è un utente generico. Questa è una distinzione puramente di *business* e non garantisce accesso a nessuna funzionalità diversa dall'utente generico, allo stato attuale della specifica.

Commerciale Il Commerciale è un addetto alle vendite di Unox. Anche questo attore esiste solo per differenziare a livello *business* e non ha nessun impatto a livello di sistema.

Fornitore di informazioni sull'energia Un servizio esterno in grado di fornire prezzi di gas e corrente elettrica. È un attore secondario.

Durante le riunioni ho individuato i seguenti *use case* per la *web-app* da sviluppare:

UC1

- **Attore primario:** utente generico;
- **Scopo e descrizione:** l'utente seleziona il prodotto UNOX del quale desidera effettuare le stime di consumo;
- **Scenario principale:** l'utente accede alla pagina per utilizzarla;
- **Precondizione:** il sistema è raggiungibile e funzionante e l'utente desidera usufruire del calcolatore;
- **Postcondizione:** l'utente ha selezionato un prodotto e ha ora la possibilità di inserirne il modo d'uso.

UC1.1

- **Attore primario:** utente generico;
- **Scopo e descrizione:** l'utente seleziona da una lista localizzata in una pagina dedicata alle politiche green;
- **Scenario principale:** l'utente accede alla pagina per utilizzare il calcolatore i consumi;
- **Precondizione:** il sistema è raggiungibile e funzionante e l'utente desidera usufruire del calcolatore;
- **Postcondizione:** l'utente ha selezionato un prodotto e ha ora la possibilità di inserirne il modo d'uso.

UC1.2

- **Attore primario:** utente generico;
- **Scopo e descrizione:** l'utente seleziona dalla pagina del prodotto l'opzione per stimarne i consumi;
- **Scenario principale:** l'utente accede alla pagina per utilizzare il calcolatore i consumi;
- **Precondizione:** il sistema è raggiungibile e funzionante e l'utente desidera usufruire del calcolatore;
- **Postcondizione:** l'utente ha selezionato un prodotto e ha ora la possibilità di inserirne il modo d'uso.

UC2

- **Attore primario:** utente generico;
- **Scopo e descrizione:** l'utente inserisce nello specifico il modo in cui utilizza/-pianifica di utilizzare un prodotto;
- **Scenario principale:** l'utente prosegue nel wizard di calcolo consumi;
- **Precondizione:** il sistema è raggiungibile e funzionante, un prodotto è già stato selezionato;
- **Postcondizione:** il sistema conosce l'utilizzo ipotetico del prodotto di cui calcolare i consumi, l'utente ha la possibilità di specificare il tipo e il costo dell'energia.

UC3

- **Attore primario:** utente generico;
- **Scopo e descrizione:** l'utente specifica il tipo e il costo dell'energia per poter ricevere una stima di consumi più precisa;

- **Scenario principale:** l'utente che desidera una stima dei consumi prosegue nel wizard di calcolo consumi;
- **Precondizione:** l'utente ha completato tutti i passi precedenti del wizard ed il sistema è raggiungibile e funzionante;
- **Postcondizione:** l'utente ottiene accesso alla visualizzazione dei consumi stimati.

UC4

- **Attore primario:** utente generico;
- **Scopo e descrizione:** l'utente ottiene una panoramica dei consumi e delle emissioni, oltre che un confronto con la controparte elettrica se ha selezionato un forno a gas;
- **Scenario principale:** l'utente completa il wizard di inserimento dati per il calcolo consumi e ottiene una panoramica;
- **Precondizione:** l'utente ha completato tutti i passi precedenti del wizard, portandolo a termine. Il sistema è raggiungibile e funzionante;
- **Postcondizione:** l'utente ottiene le stime richieste.

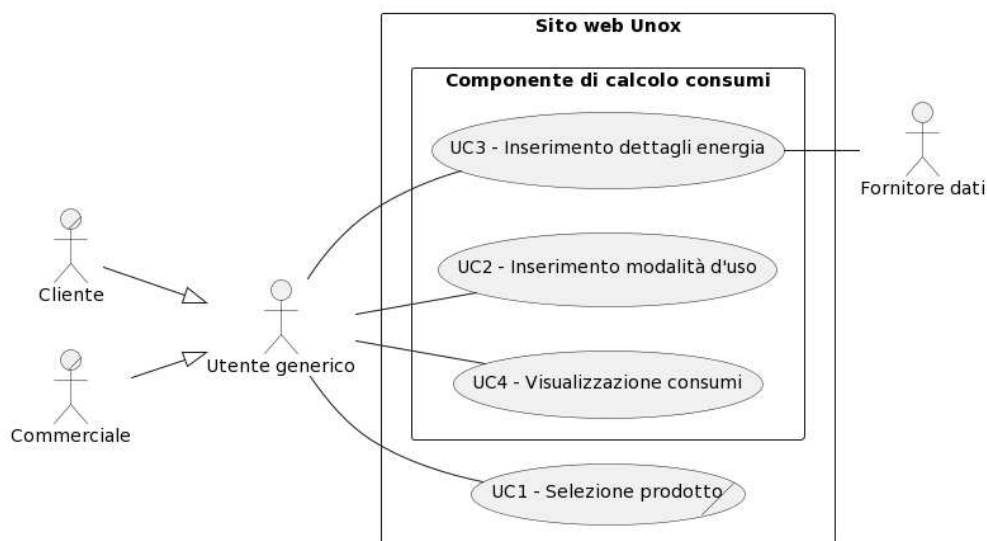


Figura 3.1: Use case individuati

Non ritenevo che approfondire ulteriormente gli use case avrebbe apportato sufficiente valore aggiunto, e pertanto mi sono fermato al primo livello.

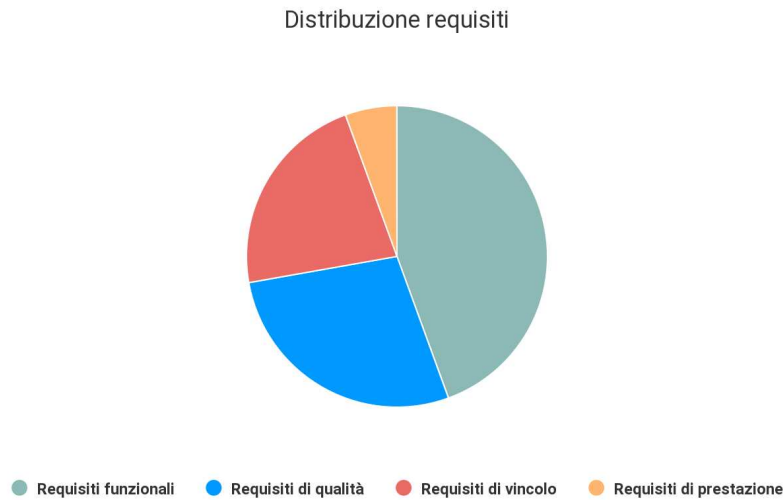


Figura 3.2: Requisiti individuati

Requisiti

Esempi di requisiti individuati

Identificatore	Categoria	Descrizione	Importanza
RF6	Funzionale	Il completamento del wizard permette la visualizzazione dei dati sui consumi	Obbligatorio
RP1	Prestazionale	Il prodotto non deve abbassare le statistiche di <i>rating</i> SEO del sito UNOX	Obbligatorio
RQ2	Di qualità	Tutti i componenti sviluppati seguono le convenzioni di design e colori UNOX	Obbligatorio
RV5	Di vincolo	Il componente deve rispettare le specifiche aziendali per l'inclusione tramite il CMS	Obbligatorio

Tabella 3.1: Tabella di alcuni dei requisiti individuati

È degno di nota che non mi era inizialmente chiaro cosa significasse [RV5](#). In seguito all'inizio della codifica, ho visto assieme al mio tutor aziendale alcuni esempi da cui prendere spunto, in progetti già terminati o ad uno stadio di completamento maggiore. Ho così avuto chiaro cosa significasse "adatto all'inclusione nel [CMS](#) aziendale".

3.3 Progettazione

Il processo di progettazione di questa web-app non ha occupato una grande quantità di tempo. Trattandosi infatti di un solo componente React, con limitata interazione con l'esterno, ho deciso di concentrare gli sforzi sul mantenimento di un ottimo grado di modularità, trovando superfluo l'utilizzo di design pattern particolarmente complessi. Durante questa fase ho avuto libertà totale, pur avendo sempre a disposizione i colleghi per un confronto. Ho effettuato le scelte che ritenevo più adatte alla situazione in cui mi trovavo, per poi esporle al mio tutor aziendale e ad altri colleghi per avere opinioni sul lavoro svolto. Questo mi ha permesso di ricevere *feedback* utili a migliorare quanto progettato.

3.3.1 Scelte progettuali fondamentali

Uso di componenti funzionali

L'uso di componenti funzionali piuttosto che componenti sotto forma di classe è dettata da

1. esigenze aziendali,
2. standard della tecnologia adottata,
3. compatibilità maggiore con il resto della codebase.

Questa scelta non è stata ovvia per me, in quanto abituato a sviluppare con paradigma ad oggetti fin dall'inizio della mia esperienza.

In azienda la totalità dei componenti React sono realizzati con il paradigma di programmazione funzionale e per assicurare la buona interoperabilità tra il mio lavoro ed il resto di quanto esisteva o sarebbe stato sviluppato in Unox, ho deciso di seguire questa convenzione. Inoltre è sicuramente stato utile per me lavorare con un paradigma diverso dalla programmazione ad oggetti in quanto sono fermamente convinto che lavorare fuori dalla propria *comfort zone* rappresenti una grande opportunità di crescita personale e professionale. Oltre a questo, nonostante React sia nato con componenti rappresentati da classi TypeScript, lo standard attuale del *framework* è ora quello funzionale.

Modularizzazione

Durante la fase di progettazione ho diretto la maggior parte degli sforzi in questa direzione. Innanzitutto ho ritenuto che produrre dei componenti altamente modulari favorisse il riutilizzo e semplificasse per me il processo di codifica. Produrre componenti modulari in questo caso ha significato individuare le funzionalità principali dell'applicazione web e suddividere le responsabilità di queste funzionalità in componenti specifici.

Ad esempio, ho individuato la responsabilità di immagazzinare lo stato e gestire l'avanzamento del Wizard di calcolo consumi nel componente `component.tsx`. Conseguentemente ho creato un componente per ogni passo del processo, in grado di gestire la grafica di alto livello e suddividere ulteriormente le funzionalità. Di conseguenza molti elementi, anche semplici, della pagina che richiedevano un minimo di personalizzazione grafica o funzionale, sono diventati dei componenti *ad-hoc* per consentire una maggiore leggibilità, manutenibilità e riutilizzo. Anche durante il mio tirocinio, ho avuto modo di rendermi conto dell'importanza di questa scelta, quando ho avuto necessità di modificare il comportamento o l'estetica di qualche elemento.

Classe Oven

Oltre ai componenti React ho creato una classe `Oven` per gestire tutte le informazioni legate ad un forno ed effettuare a richiesta calcolo dei consumi per i vari parametri. Questa classe dipende da `ckgData`, un *dataset* di dati sui consumi dei forni aziendali. Ho scelto di creare una classe per gestire le informazioni relative ai forni in quanto mi sembrava una scelta naturale, in quanto i forni sono il fulcro dell'applicativo. Inoltre, ho ritenuto che sarebbe stato utile avere un'entità che potesse gestire i dati dei forni in modo indipendente dalla grafica, in modo da poterla riutilizzare in futuro per altri scopi.

Utilizzo del *Factory Method*

Ho deciso intenzionalmente di non utilizzare *Design Pattern* complessi o che non ero in grado di padroneggiare. Nonostante ciò, progettando la classe `Oven` mi sono reso conto della corposità dei parametri e della possibile variabilità della provenienza di questi ultimi. Pertanto si è resa evidente la necessità di utilizzare un *Factory Method* per la creazione delle istanze di `Oven`.

Uso della struttura standard aziendale per componenti React

Per rendere fruibile attraverso il CMS aziendale il componente del Calcolatore di consumi, è stato necessario seguire le indicazioni del mio tutor. Il mio progetto non riguardava in minima parte il CMS interno e pertanto non ho avuto una formazione a riguardo. Ciononostante è stato necessario adottare la convenzione aziendale per lo sviluppo di componenti. Oltre all'inserimento di alcune *entry* nel DB per aggiungere la pagina al sito web in locale e per rendere visibile la "fascia" nel CMS, ho dovuto seguire una struttura standard per i *file* che compongono il componente. In particolare, ho dovuto creare un *file* `index.tsx` per esportare il componente, un *file* `completeProps.ts` per definire le proprietà del componente e un *file* `component.tsx` per definire il componente vero e proprio. Questa struttura è stata imposta come vincolo in fase di sviluppo, in quanto il CMS si aspetta di trovare questi *file* per poter integrare il componente.

index.tsx Questo *file* esporta il componente `Calculator` in modo da poterlo importare in altri componenti. Inoltre, esporta un'interfaccia `CalculatorProps` che definisce le proprietà del componente.

completeProps.ts Questo *file* definisce le proprietà del componente `Calculator` e le rende opzionali. Questo è necessario per poter utilizzare il componente in altri componenti React, in quanto il CMS non è in grado di fornire tutte le proprietà necessarie. `completeProps` si occupa pertanto di aggregare e filtrare le proprietà

Astrazione massima

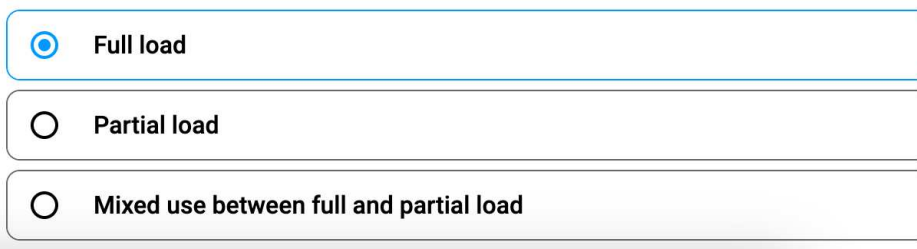
Mantenere un buon livello di astrazione è un elemento costante per uno sviluppo robusto. Per alcuni dei componenti sviluppati però, è stato un elemento cruciale. Ho avuto l'opportunità di sviluppare dei componenti che sono poi stati inseriti nel *Design System*: in questi casi ho deciso di provare a massimizzare l'astrazione e la separazione dei componenti grafici dal loro futuro contenuto. In questo modo i prodotti sarebbero stati di più facile manutenzione e fruibilità anche in ottica di un utilizzo più vasto rispetto al resto del mio lavoro. I componenti inseriti nel *design system* sono:

FormInput Rappresenta un singolo campo di input di un *form*. In base alla parametrizzazione può presentarsi in diverse varianti, con possibilità di aggiungervi icone, placeholder, testo in prefisso o suffisso o un bottone

RadioSelector Rappresenta un singolo campo di un radio *button* rivisto nel comparto estetico. Deve essere utilizzato assieme a **RadioGroup** per funzionare correttamente.

InputGroup Rappresenta un raggruppatore di **FormInput**, consentendo una creazione più veloce e coerente di gruppi di campi di testo.

RadioGroup Si occupa di costruire idoneamente i **RadioSelector** e ne gestisce la logica di selezione.



The image shows a vertical list of three radio button options. The first option, 'Full load', has a blue radio button and is selected. The second option, 'Partial load', has a white radio button and is not selected. The third option, 'Mixed use between full and partial load', also has a white radio button and is not selected. The options are contained within a light gray rounded rectangle with a thin blue border.

Figura 3.3: RadioSelector e RadioGroup in azione

The image shows a vertical stack of four input groups. Each group is a rounded rectangle with a thin border. The title of each group is in blue text at the top left. Below the title is a radio button icon, a label, and an information icon (a lowercase 'i' in a circle) on the right.

- Grilled**: Radio button icon, label "lb 3", information icon.
- Roasted**: Radio button icon, label "lb 4", information icon.
- Steamed**: Radio button icon, label "lb 3", information icon.
- Fried**: Radio button icon, label "lb 4", information icon.

Figura 3.4: FormInput e InputGroup in azione; vedi fig 3.8 per un frammento di codice che ne esplica il funzionamento

3.3.2 Diagrammi e documentazione della progettazione

Questa sottosezione offre dei diagrammi e documentazione tecnica che aiutano a comprendere meglio l'architettura e il funzionamento dell'applicazione:

Diagramma dei componenti

Viene fornito un diagramma dei componenti che illustra la struttura e le relazioni tra i componenti principali del sistema. Questo diagramma facilita la comprensione della gerarchia e delle interazioni.

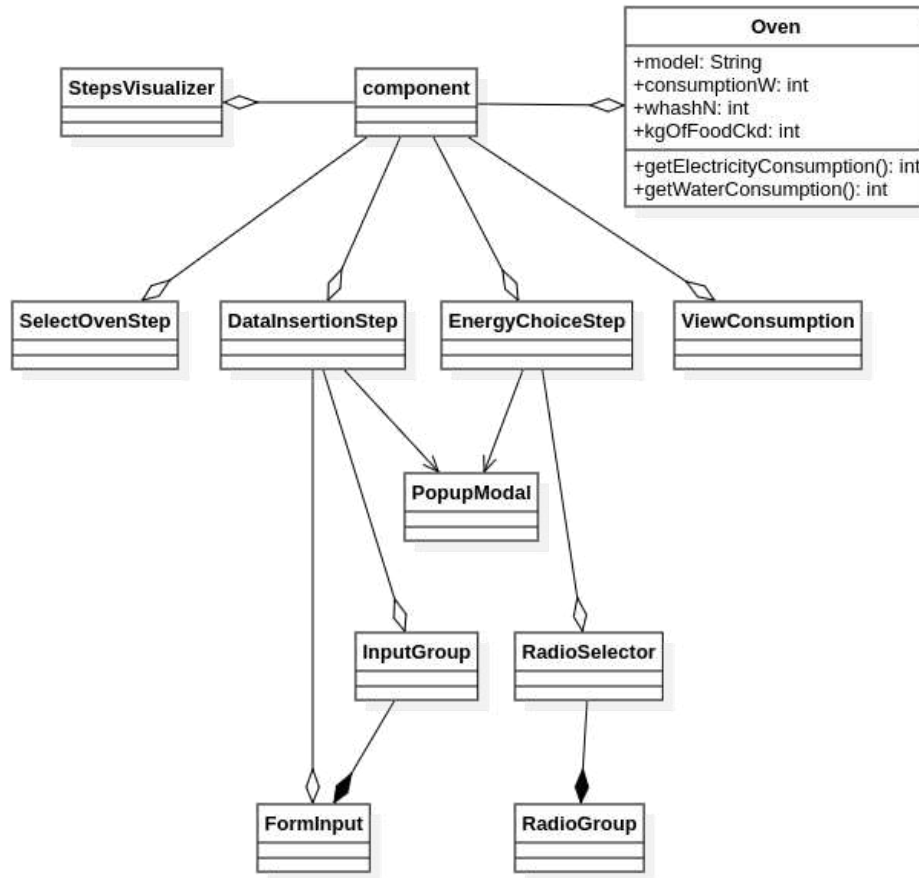


Figura 3.5: Diagramma dei componenti principali sviluppati

Come si può vedere dalla figura, **component** è l'elemento centrale dell'applicativo. La sua funzione è quella di conservare lo stato in modo da poter montare e smontare gli altri componenti al momento giusto e con le giuste informazioni. Questo componente si occupa di ottenere dal [CMS](#) i dati sui forni, filtrarli e costruire un **Oven** corrispondente alla scelta dell'utente. **component** si occupa anche di gestire la navigazione tra i vari step del wizard, in modo da poter passare le informazioni necessarie ai componenti figli. I componenti "Step" si occupano di gestire la grafica e la logica di ogni singolo step del wizard. Ad esempio, **SelectOvenStep** si occupa di fornire un'interfaccia grafica per riepilogare il forno scelto e fornire un'alternativa all'utente. Gli altri step si occupano rispettivamente di raccogliere i dati di utilizzo del forno, di raccogliere i dati relativi al costo dell'energia e di visualizzare i dati calcolati. I componenti dedicati alla raccolta dati, si occupano anche di aggiornare lo stato in tempo reale utilizzando delle funzioni di *callback*. Queste funzioni vengono passate come proprietà ai componenti figli e vengono chiamate quando l'utente modifica i dati inseriti. Questo permette di aggiornare lo stato in tempo reale e di poter fornire un'esperienza utente più fluida. Alcuni componenti minori sono stati riutilizzati in vari step. Questo è il caso di [InputGroup](#), [FormInput](#), [RadioSelector](#), [RadioGroup](#).



Figura 3.6: Diagramma dei componenti per la visualizzazione dei dati

A causa della maggiore complessità ho ritenuto che il componente che si occupa della visualizzazione, `ViewConsumption`, meritasse una suddivisione più capillare in sottocomponenti. Questo mi ha semplificato la separazione tra il comportamento e la rappresentazione della pagina. A partire dalla specifica grafica fornitami ho individuato la necessità di rappresentare molteplici elementi diversi tra loro sotto forma di card con estetica comune. `ConsumptionCard` assolve esattamente questo compito. Inoltre, ho ritenuto che la rappresentazione grafica dei dati di consumo e di emissioni meritasse un componente dedicato. `DonutChart` si occupa di generare grafici vettoriali a ciambella, utilizzando SVG (Scalable Vector Graphics). Questo componente è stato sviluppato da me e successivamente inserito nel *design system* aziendale. Ognuna delle card dalle quali è composto `ViewConsumption` riempie una `ConsumptionCard` con diversi elementi per riepilogare i consumi forniti dal componente padre.

3.4 Codifica

3.4.1 Codifica dei componenti

Per affrontare l'implementazione dei componenti precedenti descritti, ho intrapreso una strategia concentrata sullo sviluppo dei componenti più riutilizzati all'inizio. Il primo componente ad essere completato è stato infatti `FormInput`. Questo primo periodo è stato cruciale per adattarmi allo stile di programmazione richiesto dalla libreria React. Durante questa fase, mi assicuravo di star sviluppando quello che volevo utilizzando una pagina posticcia. Su questa pagina venivano montati i componenti man mano che li sviluppavo, in modo da poterli testare e verificare il corretto funzionamento.

Utilizzo di un diario di lavoro

Ho notato che alcuni dei miei colleghi tenevano un diario di lavoro. Ho deciso presto durante il mio tirocinio di imitarli e sono convinto questo abbia apportato dei benefici al processo di codifica. Innanzitutto, utilizzavo il diario per fare brainstorming e per creare dei piccoli diagrammi che potessero facilitarmi la comprensione di alcune *task*. Inoltre ho utilizzato in diverse occasioni il diario per tenere traccia del lavoro svolto o da svolgere, non essendomi affidato a nessun *Issue Tracking System*.

Suddivisione del lavoro in *task*

Poco dopo lo sviluppo dei primi componenti mi sono reso conto che avrei avuto bisogno di suddividere il carico in compiti più piccoli. Fare questo mi ha permesso di avere un'idea più chiara di quanto ancora avrei dovuto fare e di quanto tempo mi sarebbe servito. Inoltre, l'affrontare problemi più piccoli mi permetteva di ridurre lo stress e di avere un senso di progressione più tangibile.

3.4.2 Creazione dei componenti più complessi

Una volta sviluppati i componenti più atomici, ho iniziato a lavorare sui componenti più complessi, costruendo il flusso di navigazione tra i vari step del wizard. In questa fase ho avuto modo di apprezzare la modularità dei componenti sviluppati in precedenza, in quanto mi hanno permesso di concentrarmi sulla logica di navigazione e di non dovermi preoccupare di come i componenti fossero effettivamente rappresentati graficamente. Inoltre in questa fase ho compreso per la prima volta l'utilità di lavorare con una specifica precisa, fornitami in questo caso su Figma.

3.4.3 Sperimentazione di gerarchie di componenti

Essendo stato abituato a sviluppare con paradigmi ad oggetti, il mio pensiero iniziale durante la progettazione era quello di realizzare componenti sotto forma di classi. In questo modo avrei potuto costruire delle gerarchie di componenti via via più specializzati. Come abbiamo visto dalla fase di progettazione, questa non è stata la strada che ho seguito. Nonostante ciò, ho deciso comunque di sperimentare la creazione di una piccola gerarchia di componenti. Ho riscontrato diverse difficoltà, innanzitutto perché non conoscevo la sintassi di React per creare componenti-classe, ed in seguito per la compatibilità di diversi componenti tra loro. Ho deciso quindi di abbandonare questa strada e di utilizzare componenti funzionali, che mi hanno permesso di ottenere lo stesso risultato in modo più semplice e compatibile con il resto del codice.

3.4.4 Esempi

DonutChart

```

return (
  <svg width={width} height={height} viewBox={`0 0 ${width} ${height}`}>
    <g transform={`translate(${center},${center})`}>
      /* Draw the donut slices */
      {angles.map((angle, index) => {
        const startAngle = angles.slice(0, index).reduce((sum, value) => sum + value, 0)
        const endAngle = startAngle + angle

        // Convert angles to radians
        const startAngleRad = (startAngle - 90) * (Math.PI / 180)
        const endAngleRad = (endAngle - 90) * (Math.PI / 180)

        // Calculate the coordinates for the arc paths
        const innerX1 = Math.cos(startAngleRad) * hole
        const innerY1 = Math.sin(startAngleRad) * hole
        const innerX2 = Math.cos(endAngleRad) * hole
        const innerY2 = Math.sin(endAngleRad) * hole
        const outerX1 = Math.cos(startAngleRad) * radius
        const outerY1 = Math.sin(startAngleRad) * radius
        const outerX2 = Math.cos(endAngleRad) * radius
        const outerY2 = Math.sin(endAngleRad) * radius

        // Determine if the angle is larger than 180 degrees (for largeArc flag)
        const largeArcFlag = angle > 180 ? 1 : 0

        // Create the SVG path for the donut slice
        const pathData = `M ${innerX1},${innerY1}
          A ${hole},${hole} 0 ${largeArcFlag},1 ${innerX2},${innerY2}
          L ${outerX2},${outerY2}
          A ${radius},${radius} 0 ${largeArcFlag},0 ${outerX1},${outerY1}
          Z`

        return (
          <path key={index} d={pathData} fill={data[index].color}>
            <title`${data[index].label}: ${data[index].value}`</title>
          </path>
        )
      })}
    </g>
  </svg>
)

```

Figura 3.7: Frammento di codice per la creazione di un DonutChart

Questo frammento di codice JSX è utilizzato per disegnare un grafico a ciambella tramite SVG (Scalable Vector Graphics). Esso sfrutta le seguenti funzionalità:

- `<svg>`: Definisce un elemento SVG con proprietà di larghezza e altezza passate come parametri alla funzione in cui è inserito il codice.
- `<g>` (gruppo): Raggruppa gli elementi grafici all'interno del SVG, trasladoli al centro del disegno tramite l'attributo `transform`.

- Iterazione sull'*array* degli angoli: Disegna i segmenti del grafico a ciambella utilizzando l'*array* `angles`, che verosimilmente contiene gli angoli dei vari segmenti da rappresentare.
- Calcoli degli angoli e delle coordinate: Effettua calcoli trigonometrici per determinare le coordinate degli archi di ogni segmento del grafico.
- `<path>` (percorso): Per ciascun segmento, crea un percorso SVG utilizzando l'elemento `<path>` e il valore `d`, che specifica il percorso da disegnare.
- `<title>`: Inserisce un elemento `<title>` all'interno di ciascun `<path>` per mostrare etichette e valori quando il mouse vi si sovrappone (utile per l'accessibilità e per informazioni aggiuntive).

In sintesi, questo frammento di codice utilizza SVG e JSX per rappresentare un grafico a ciambella basato su un set di dati forniti, determinando la forma di ogni fetta del grafico tramite l'utilizzo dell'angolo e calcolando le coordinate necessarie per disegnare il percorso SVG corrispondente.

InputGroup

```
1 import Box from "design-system/src/components/Box";
2 import FormInput, { InputProps } from "./FormInput";
3
4 interface InputGroupProps {
5   topicLabel?: string;
6   inputs: InputProps[];
7 }
8
9 const InputGroup = (props: InputGroupProps) => {
10   const { topicLabel = "", inputs } = props;
11
12   return (
13     <Box>
14       <span>{topicLabel}</span>
15       <Box>
16         {inputs
17           .filter((input) => input && input !== null)
18           .map((input, index) => (
19             <FormInput key={index} {...input} />
20           ))}
21       </Box>
22     </Box>
23   );
24 };
25 export default InputGroup;
26
```

Figura 3.8: Frammento di codice che implementa InputGroup

Il precedente frammento di codice implementa un componente React denominato `InputGroup`, sviluppato utilizzando TypeScript. Questo componente è progettato per accettare proprietà specifiche e renderizzare un gruppo di campi di input all'interno di un *layout* definito.

Il componente `InputGroup` è strutturato in diverse parti:

- **Importazioni:** Include l'importazione del modulo `Box` da un percorso specifico all'interno del *design system* e importa `FormInput` insieme al tipo `InputProps` da un *file* locale chiamato `FormInput`.
- **Interfaccia `InputGroupProps`:** Definisce le proprietà accettate da questo componente. Queste proprietà includono `topicLabel`, un'etichetta opzionale che rappresenta il titolo dell'argomento associato al gruppo di input, e `inputs`,

un *array* di oggetti `InputProps` che contengono le specifiche degli input da visualizzare.

- **Componente `InputGroup`:** È una funzione componente che riceve le proprietà definite nell'interfaccia `InputGroupProps`. Estrae le proprietà `topicLabel` e `inputs` dalle *props* utilizzando la destructuring *assignment* di JavaScript/TypeScript.
- **Render:** Il componente restituisce un componente `Box` come contenitore principale. Include un elemento `span` per visualizzare il testo dell'etichetta tematica (`topicLabel`). Inoltre, contiene un altro componente `Box` che mappa gli elementi nell'*array* `inputs`, filtrando eventuali elementi nulli o indefiniti. Ogni elemento dell'*array* viene passato al componente `FormInput` utilizzando le *props* spread (`{...input}`). Questo popola il `FormInput` con le proprietà definite nell'oggetto `input`.

In sintesi, il componente `InputGroup` si configura come un contenitore per visualizzare un gruppo di campi di input (`FormInput`) sotto un'etichetta tematica (`topicLabel`). L'output dei campi di input dipende dalle proprietà passate al componente al momento dell'utilizzo.

Oven

```
const [oven, setOven] = useState<Oven>(  
  Oven.fromOvenCode(products, formattedTraysData, codes, userCountry),  
)
```

Figura 3.9: Creazione di un oggetto `Oven` mediante l'uso di un *factory method*

In questo frammento di codice, possiamo vedere la costruzione di un oggetto `Oven` mediante l'utilizzo di un *factory method*. L'oggetto viene costruito filtrando dai `products` un forno con consumi equivalenti a quelli del forno con il codice `ovenCode`. Questo avviene perché non sono presenti misure sperimentali per tutti i forni, ma solo per alcuni di essi. `codes` contiene i codici dei forni con i quali l'oggetto in costruzione verrà comparato. Questo passaggio avviene in quanto i forni da comparare sono parte dell'oggetto `forno`, e sono perciò costruiti ricorsivamente utilizzando lo stesso *factory method*. Il parametro `codes` è facoltativo. Il paese dell'utente (`userCountry`) e `formattedTraysData` sono parametri utili per la localizzazione geografica delle pagine prodotte.

3.4.5 Salvataggio di un *file* JSON

```
writeFile(summaryWB as Workbook, 'temp.xlsx')
// sleep(3000);
await excelOpenClose('temp.xlsx')

const columnValues = await readFrom('temp.xlsx', range)
columnValues.push(load)
columnValues.push(QUICK_WASH_INFO[model.modelN.substring(5, 9)])

OUTPUT_TO_CSV && (await writeToCSV(columnToCsv(columnValues)))
// console.log("columnToCsv: ", columnToCsv(columnValues));

OUTPUT_TO_JSON && (await writeToJSON(columnValues))

currentProgress.panIndex++
await saveProgress()
```

Figura 3.10: Salvataggio di un *file* JSON dopo l'estrapolazione dei dati relativi ai forni

Nel contesto dell'estrapolazione di dati dal modello matematico fornito dal reparto, era essenziale fornire i dati in un formato utilizzabile da altre applicazioni. Il frammento di codice qui sopra mostra come, dopo aver popolato il *file* Excel con le informazioni necessarie, il *file* venga salvato, aggiornato e tradotto in JSON o CSV a seconda delle impostazioni. Più in particolare:

- `excelOpenClose` è una funzione che si occupa di riconoscere il sistema operativo corrente e di aprire il *file* Excel con l'applicazione predefinita. Questo è necessario in quanto il *file* Excel deve essere aperto per poter essere aggiornato. Il *file* viene poi chiuso dopo un *timeout* fisso. È stato difficile trovare il *timeout* minore possibile per garantire che il *file* venisse aggiornato correttamente, e allo stesso tempo minimizzare l'attesa.
- Viene letto il *file* Excel nel `range` specificato. Il range è ottenuto da una funzione che deduce in quali righe e colonne si troveranno i dati in output.
- Alcuni dati, non dedotti dal *file* Excel vengono aggiunti a quelli appena letti. È il caso d'uso del metodo `push` in questo frammento di codice.
- I metodi da me definiti `writeToCSV` e `writeToJSON`, si occupano di trasformare in formato CSV o JSON i dati passati per parametro.
- `currentProgress` è una variabile che si occupa di tenere traccia del progresso dell'estrapolazione dei dati. Questo è necessario per aggiornare l'utente sullo stato di avanzamento del processo e permettere un recupero del progresso in caso di errori durante l'esecuzione. Infatti, il cambio di *focus* durante l'esecuzione del programma può causare l'interruzione del processo di estrazione dei dati.
- `saveProgress()` è una funzione che si occupa di salvare il progresso dell'estrapolazione dei dati in un *file* JSON. Questo è necessario per permettere all'utente di riprendere il processo di estrazione dei dati in caso di interruzione.

3.4.6 Strumenti di sviluppo e metodologie di *coding standard*

Durante lo sviluppo gli strumenti utilizzati sono stati pochi ma di importanza cardinale:

Git Version control system utilizzato per tenere traccia delle modifiche e per il versioning del codice. Ho utilizzato *Conventional commits* per seguire uno standard di commit message.

Visual Studio Code L'IDE designato per lo sviluppo di componenti React. Ho utilizzato le estensioni *ESLint* e *Prettier* per garantire la correttezza del codice e per formattarlo in un modo standardizzato. Alcune regole specifiche per ESLint sono utilizzate da tutto il *team* di sviluppo di Unox per prevenire problematiche già conosciute.

3.4.7 Creazione dello *script* per la generazione del *dataset*

Inizialmente questa fase ha subito ritardi a causa della mancanza in azienda del collega affidatario dei dati riguardanti i consumi dei forni. Purtroppo durante la sua assenza non ho avuto modo di farmi un'idea di che cosa mi aspettasse. Ero solamente a conoscenza che per la prosecuzione del resto del progetto e del *testing* avrei necessitato di dati reali sui consumi dei forni. Questi dati si sono presentati sotto forma di *file* Excel. Non un semplice foglio di calcolo sotto forma tabellare, bensì un *file* Excel con molteplici fogli, ognuno assegnato a rilevazioni sperimentali diverse e con modelli di astrazione per sopperire alle misurazioni mancanti. I fogli contenevano a volte riferimenti circolari tra loro e l'unico modo affidabile di estrarre informazioni sui forni era quello di utilizzare il foglio appositamente dedicato a funzione di interfaccia. La realizzazione di questo *script* ha richiesto più studio e più impegno di quanto inizialmente previsto.

Comprensione del problema

Il mio obiettivo era costruire un *dataset* semplice, sotto forma ad esempio di JSON o CSV, contenente i dati di un subset dei forni presenti all'interno del *file* Excel. Per ottenere questo risultato è stato evidente fin da subito che compilare manualmente il foglio di interfaccia non sarebbe stata una soluzione fattibile.

Soluzioni proposte

Per questo motivo ho deciso di automatizzare il processo costruendo uno *script* ausiliario: si sarebbe occupato di rilevare il *file* Excel e la zona adatta all'inserimento dati, popolarne i campi con dati riguardanti forno e cotture ed infine aggiornare il foglio di calcolo e l'estrazione dei dati. Per effettuare queste operazioni ho valutato diverse opzioni tra cui l'utilizzo della versione web di Excel in abbinata a Puppeteer. Un'altra idea era di utilizzare RobotJS¹ per simulare quello che avrei fatto manualmente, ma la soluzione era molto delicata e avrebbe probabilmente richiesto un processo di calibrazione lungo e tedioso. Invece, per lo sviluppo di questo *script* ho scelto di utilizzare una libreria che permette di modificare i *file* .xlsx, **XLSX**², in abbinata con **RobotJS**³, un tool in grado di simulare un input HID.

¹RobotJS. URL: <https://robotjs.io/>.

²XLSX. URL: <https://www.npmjs.com/package/xlsx>.

³RobotJS.

In questo scenario, XLSX si sarebbe occupata di modificare il *file* Excel nel quale inserire i parametri per ottenere i dati sui consumi. Per l'aggiornamento del foglio, il *file* veniva aperto con Excel tramite l'esecuzione di un comando `exec`, in seguito salvato e chiuso con l'utilizzo di RobotJS⁴, simulando l'uso di scorciatoie da tastiera. Infine, XLSX⁵ si sarebbe occupata di estrarre i dati dal *file* Excel e di salvarli in un *file* JSON.

Realizzazione e *debugging*

Durante la realizzazione dello *script*, ho inizialmente verificato la fattibilità di quanto proposto, realizzandone una versione piuttosto rudimentale. Ho via via raffinato il codice per renderlo più robusto, nell'ottica di dover ripetere questa operazione in futuro per un eventuale aggiornamento del *dataset*. Ho parallelamente prodotto una semplice guida dettagliante l'utilizzo dello *script*. Per arrivare a produrre con successo ed efficienza temporale il *dataset* contenente i consumi dei forni ho dovuto fallire diverse volte. L'efficienza temporale è particolarmente importante a mio avviso: per ogni forno, per ogni tipologia di cottura e per ogni tipologia di utilizzo, lo *script* sarebbe rimasto in attesa del salvataggio dei files. Pertanto ho perso parecchio tempo per provare a ridurre al minimo i *timeout* manualmente inseriti. Inoltre, ho dovuto documentarmi su come poter rendere il programma, che utilizza anche scorciatoie da tastiera, *platform independent*.

Dopo il lavoro effettuato, lo *script* era in grado di generare il *dataset* contenente i consumi dei forni, circa 3000 righe, in 45 minuti. 45 preziosi minuti in cui il computer non poteva essere utilizzato in quanto per l'aggiornamento del foglio di calcolo era necessaria la ripetuta apertura e chiusura di Excel. Devo dire che non mi sento pienamente soddisfatto del risultato ottenuto, in quanto è un tempo lungo per il quale non poter utilizzare un computer, ma allo stesso tempo sono appagato dall'essere riuscito a risolvere un problema adattando il *dataset* esistente nonostante non fosse pensato per questo scopo.

3.5 Risultati raggiunti

3.5.1 Descrizione del tirocinio

Ho svolto il mio lavoro prevalentemente in autonomia, soprattutto nel periodo centrale del mio tirocinio. Se nei primi ed ultimi momenti avvenivano infatti molteplici interazioni spontanee al giorno, nel periodo centrale mi limitavo a portare avanti lo sviluppo dell'applicazione web, pezzo per pezzo. In questo periodo le interazioni con i colleghi e con il mio responsabile sono avvenute, ma in misura più limitata. Avvenivano ad esempio per sottoporre a revisione qualcosa di incerto da me prodotto, o alle riunioni a cadenza regolare.

Lo sviluppo dei primi componenti, più indipendenti e semplici, mi ha fornito il tempo necessario ad abituarli al nuovo *stack* tecnologico. Iniziare a costruire componenti più complessi è stato fin da subito soddisfacente, avendo il riscontro di colleghi e designer dalla mia parte.

⁴RobotJS.

⁵XLSX.

Sviluppo della classe Oven

Quando ho espresso la mia volontà di creare la classe *utility Oven*, ho ottenuto reazioni contrastanti. Alcuni colleghi erano molto entusiasti, ancor di più quando ho descritto la volontà di utilizzare un *factory method*. Altri avrebbero risolto il problema utilizzando la programmazione funzionale, ma hanno appoggiato allo stesso modo la mia idea. Ho preso questa strada in quanto più nella mia comfort zone.

3.5.2 Prodotto realizzato

Al termine del tirocinio, il frutto del mio lavoro si presentava come un prodotto utilizzabile in ambiente di sviluppo e *testing*, ma non integrato in ambiente di produzione. Il prodotto presentava tutte le caratteristiche grafiche richieste, e tutte le funzionalità core, evidenziate dai requisiti raccolti. La pagina web su cui ho lavorato non era ancora stata integrata nel sito aziendale, e pertanto era accessibile solamente conoscendone l'url esatto. Una volta entrati all'interno della pagina, si veniva ricevuti da un landing screen che riepiloga la scelta del forno. Non essendo ancora collegata al resto del sito, era possibile non aver selezionato nessun forno all'accesso alla pagina ed era necessaria una soluzione per continuare lo sviluppo ed il *testing* della pagina. A questo fine, se la pagina non riceveva un vero e proprio forno selezionato come parametro, simulava la scelta di un forno specifico.

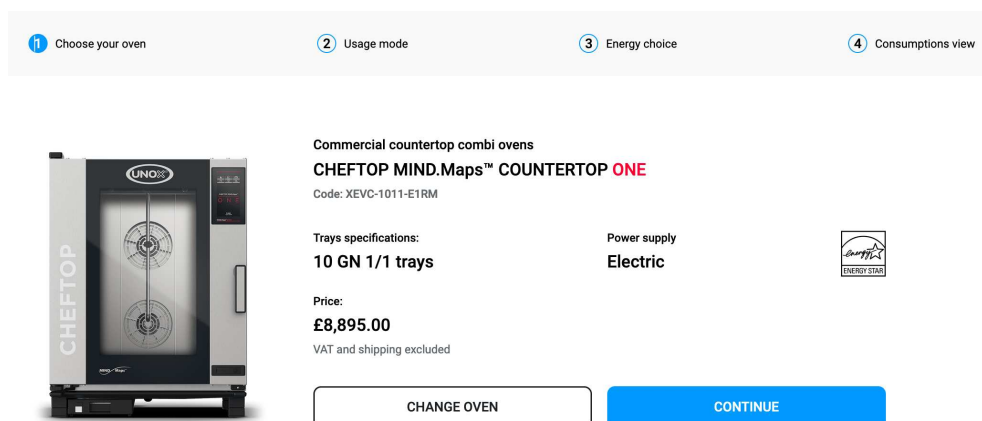


Figura 3.11: Landing screen

A questo punto all'utente veniva presentata la possibilità di proseguire con il calcolo dei consumi, oppure quella di tornare alla selezione del forno. Non essendo ancora parte integrante del sito web, la funzionalità di selezione del forno non era ancora stata implementata.

The screenshot shows a multi-step form with four progress indicators at the top: 1. Choose your oven (checked), 2. Usage mode (active), 3. Energy choice, and 4. Consumptions view.

The form contains three main sections:

- How many days per week do you use the oven?**: A text input field labeled "Days / week" with a calendar icon and the value "1".
- How many kilos of food/hours of cooking do you cook each day with the following modes?**: Four stacked input fields, each with a cooking mode icon and a value:
 - Grilled**: lb 3
 - Roasted**: lb 4
 - Steamed**: lb 3
 - Fried**: lb 4
- How do you use the oven in relation to its maximum capacity?**: A radio button group with three options:
 - Full load
 - Partial load
 - Mixed use between full and partial load

Figura 3.12: Selezione dei parametri di utilizzo

In caso di scelta di prosecuzione con il processo di inserimento dati, l'utente viene presentato di fronte a 2 diversi *form* per la raccolta di informazioni riguardanti l'utilizzo e il costo operativo del forno. Questi *form* sono stati realizzati utilizzando i componenti precedentemente descritti. In particolare, `InputGroup` è stato utilizzato per raggruppare i campi di input in modo coerente. `RadioGroup` e `RadioSelector` sono stati utilizzati per la selezione di un'opzione tra più alternative. `FormInput` è stato utilizzato per la raccolta di dati testuali. `FormInput` è stato utilizzato anche per la raccolta di dati numerici, in quanto non era disponibile un componente specifico per questo scopo.

Which type of energy do you use?

100% certified renewable

National power grid

Select your country

United Kingdom

Insert electricity price

Use national mean

Electricity price

£/kWh 5

CONTINUE

Figura 3.13: Selezione dei parametri di costo

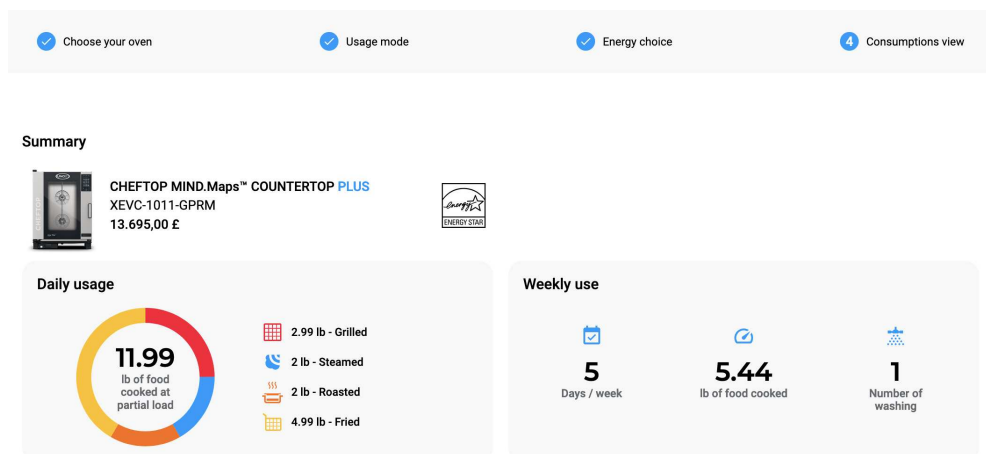


Figura 3.14: Alcuni componenti del riepilogo dati

Infine, l'utente visualizza i dati inseriti e una serie di consumi ipotetici calcolati.

Capitolo 4

Valutazione retrospettiva

4.1 Confronto tra gli obiettivi iniziali e i risultati ottenuti

Tutti i requisiti sono stati soddisfatti, ad esclusione di quelli riguardanti il *testing* dell'applicazione.



Figura 4.1: Percentuale di requisiti raggiunti: 95%

Tabella 4.1: Tabella degli obiettivi raggiunti. Vedi [obiettivi](#).

Obiettivo	Raggiunto/no
Sviluppo di elementi altamente riutilizzabili per il <i>design system</i>	Raggiunto
Astrazione del modello matematico	Raggiunto
Implementazione di grafici	Raggiunto
Creazione di una pagina web	Raggiunto

Mancato *testing*

Rispetto ai requisiti fissati, alcuni non sono stati soddisfatti. In particolare tutti i requisiti di *testing* dell'applicazione. Questo è avvenuto perché è stato deciso a codifica iniziata dai miei responsabili che l'applicazione avrebbe dovuto supportare tutte le linee di punta dell'azienda. Il problema presentatosi è stato che non esistevano dati a sufficienza per completare il *testing*, e nemmeno lo sviluppo. Questa modifica richiedeva infatti sforzi supplementari non solo da parte mia, ma anche da parte del *team* di design che avrebbe dovuto produrre il nuovo flusso di navigazione completo per i forni da supportare. Purtroppo non c'è stato tempo a sufficienza perché questo potesse accadere, e pertanto il *testing* è stato rimandato a quando i dati sarebbero stati disponibili. Alla consegna del mio lavoro, ho stilato una lista di problematiche note o questioni aperte cosicché chi prendesse in mano il mio lavoro potesse avere un'idea più chiara di cosa fosse stato fatto e di cosa mancasse.

4.2 Soddisfacimento degli obiettivi personali

Posso dire di aver soddisfatto tutti gli obiettivi che mi ero posto prima dell'inizio del tirocinio. Questi sono visionabili nella sezione §2.6.2.

Infatti durante il mio periodo di *stage* ho avuto modo di consolidare le mie competenze di sviluppo web, imparando a lavorare con tecnologie più moderne e all'avanguardia rispetto a quanto affrontato durante il mio corso di studi. Questo è testimoniato dal mio uso più intensivo di quanto mai fatto prima di TypeScript. Ho inoltre imparato ad utilizzare il framework React, e a lavorare con la sintassi JSX.

Ho anche avuto modo di lavorare con più di una nuova libreria: ho utilizzato RobotJS¹, xlsx² e utilizzato Node.js in maniera più consapevole e approfondita di quanto non avessi mai fatto.

Per il soddisfacimento del terzo obiettivo, l'esperienza della posizione da programmatore, non posso presentare una metrica davvero oggettiva, ma posso portare la mia testimonianza. Questo documento testimonia a mio avviso quello che ho avuto modo di vivere nel periodo di tirocinio. Sono soddisfatto di aver messo in pratica in modo professionale quanto appreso negli ultimi anni, e altrettanto soddisfatto dal feedback ricevuto all'interno dell'azienda dai colleghi.

4.3 Maturazione professionale

Questa esperienza in Unox, è stata per me sotto molti aspetti una prima volta. Ritengo che le prime esperienze siano sempre in qualche modo costruttivo, lasciando più conoscenza di quanta ve ne fosse prima.

Ritengo che essere riuscito a gestire la vita professionale e personale, anche se per un esiguo periodo di 2 mesi, con un impiego a tempo pieno possa essere un sinonimo di maturazione personale. Durante il periodo del mio tirocinio i miei ritmi sono radicalmente cambiati rispetto a quando stavo frequentando l'Università. Sono stato costretto a gestire il mio tempo in maniera più efficiente, cercando di sfruttare al meglio le ore della giornata. Inoltre, ho dovuto imparare a gestire il mio tempo libero, cercando di sfruttarlo al meglio per rilassarmi e ricaricare le energie per il giorno successivo.

Per quanto stia diventando uno standard, con sempre più posti di lavoro appartenenti al settore terziario, per me lavorare all'interno di un ufficio è stata un'esperienza nuova. Ho dovuto abituarci ad uno spazio chiuso e limitato, e ad una stretta convivenza con delle persone che non conoscevo. Ritengo di essere riuscito a comprendere le dinamiche di un posto di lavoro in ufficio costituisca un vantaggio per i prossimi impieghi.

Fino ad ora, non ho mai avuto l'opportunità di lavorare all'interno di una grande impresa. Ritengo che questa esperienza mi abbia permesso di comprendere meglio le dinamiche di un'azienda di grandi dimensioni, e di capire come funziona la comunicazione tra i vari reparti. Inoltre, ho avuto modo di comprendere come vengono gestiti i progetti, e come vengono assegnati i compiti ad alcuni dei *team*.

Anche se non in contemporanea sullo stesso progetto, ho avuto modo di lavorare con altri sviluppatori. Ho riutilizzato codice o librerie prodotti dai miei colleghi, e predisposto il mio lavoro ad un riutilizzo futuro da parte loro. In più, ho tratto vantaggio dall'esperienza di colleghi che permangono in Unox da anni.

¹RobotJS.

²XLSX.

Anche se nel 2020 ho avuto modo di partecipare alle lezioni da remoto, non posso dire di aver mai fornito una prestazione di lavoro da remoto. L'azienda che mi ha ospitato mi ha fornito la possibilità di lavorare da casa, e di partecipare alle riunioni da remoto. Non sapevo se mi sarei trovato bene, essendo particolarmente sensibile ai cambi di contesto in cui mi trovo a lavorare o studiare. Sono rimasto piacevolmente sorpreso nello scoprire che lavorare da casa mi ha permesso di essere più produttivo, e di avere più tempo libero per me stesso. Inoltre, ho avuto modo di comprendere come funziona la comunicazione all'interno di un'azienda che permette il lavoro da remoto, e come vengono gestiti i progetti.

Ritengo che tutti questi possano essere dei fattori che hanno contribuito all'inizio di una maturazione professionale, che mi permetterà di affrontare con più sicurezza i prossimi impieghi.

4.4 Conoscenze acquisite

Durante il periodo di *stage*, ho acquisito competenze notevoli nell'utilizzo di tecnologie frontend come React e TypeScript. Ho anche familiarizzato con la sintassi JSX che non conoscevo prima. Queste competenze mi hanno permesso di contribuire in modo significativo allo sviluppo di moderne applicazioni web. La comprensione e l'esperienza pratica acquisite in queste tecnologie rappresentano un importante pilastro del mio percorso formativo. Ho inoltre avuto modo di assorbire informazioni riguardo alla struttura di un sito web complesso, dotato di [CMS](#) e basato su Node.js.

La collaborazione e la comunicazione efficace sono fondamentali in qualsiasi contesto lavorativo. Durante il mio *stage*, ho appreso l'uso di strumenti di collaborazione come Microsoft Teams per la comunicazione e la pianificazione di riunioni. Questo percorso ha migliorato le mie competenze nella gestione del lavoro di squadra e nella comunicazione aziendale. Ho dovuto rispettare protocolli di comunicazione, ed in un paio di occasioni organizzare delle riunioni per ottenere le informazioni necessarie per portare a termine il mio lavoro.

La gestione del codice sorgente è una componente cruciale nello sviluppo *software* collaborativo. Ho imparato ad utilizzare Git più in profondità di quanto potessi prima, comprendendo le best practice e gli strumenti avanzati. Questa competenza ha influenzato positivamente la mia capacità di contribuire in modo efficace a progetti complessi e collaborativi.

La modalità di lavoro da remoto è sempre più diffusa e rappresenta una parte significativa dell'ambiente lavorativo contemporaneo. Durante il mio tirocinio, ho avuto l'opportunità di lavorare da remoto e partecipare alle riunioni a distanza. Questa esperienza ha fornito un'interessante prospettiva sulle dinamiche del lavoro a distanza e sull'importanza della comunicazione in tale contesto. Ho fatto la piacevole scoperta che, al contrario di quanto pensassi, apprezzo lavorare da remoto in quanto mi consente di essere più produttivo e di avere più tempo libero per me stesso.

Per la prima configurazione della mia postazione, ho ricevuto delle nozioni di ergonomia utili a mantenere una postura corretta. Ora comprendo meglio gli effetti a medio e lungo termine di uno scorretto posizionamento e sono in grado di apprezzare strumenti più ergonomici per la comodità fornita.

Durante il mio tirocinio, ho avuto l'opportunità di immergermi nel funzionamento di un'azienda di grandi dimensioni. Questa esperienza mi ha permesso di comprendere le dinamiche tra i vari reparti e la gestione dei progetti. La conoscenza del contesto aziendale è essenziale per operare in modo efficace in un ambiente di lavoro complesso.

Nel corso del mio *stage*, ho acquisito conoscenze specifiche relative al funzionamento dei forni a convezione, forni a microonde e forni combinati. Ho inoltre approfondito per lo sviluppo della pagina di selezione dell'energia dati relativi alle forniture energetiche e alle loro unità di misura. Questa conoscenza settoriale è di particolare rilevanza per affrontare progetti futuri legati al settore, permettendomi di contribuire in modo più competente ed informato.

4.5 Confronto con il corso di studi

Durante la mia esperienza lavorativa in azienda, ho avuto l'opportunità di confrontare le tecnologie front-end utilizzate con quelle affrontate nel corso di studi. Ho sviluppato competenze significative nell'uso di *framework* moderni come React e TypeScript, aprendomi alla sintassi JSX. Queste competenze rappresentano un progresso importante nella mia formazione, preparandomi per progetti di sviluppo web avanzati.

La gestione del codice sorgente è stata un'altra area di differenza significativa. In azienda, ho imparato l'importanza di una gestione rigorosa attraverso l'uso standard e obbligatorio di *repository* Git. Questo contrasta con il mio corso di studi, dove ho avuto modo di utilizzare Git ed impararne il funzionamento solo in modo autonomo, non avendo frequentato il corso facoltativo che trattava questo argomento. Questo sottolinea la necessità di una maggiore enfasi su quest'aspetto fondamentale. Durante il mio tirocinio, ho notato una differenza significativa nella gestione dei progetti rispetto a quanto insegnato nel mio corso di studi. Nel mio progetto aziendale, la pianificazione e la progettazione non erano processi prioritari, e l'uso di [ITS](#) non era suggerito. La coordinazione tra i *team* non era sempre garantita, e spesso mancavano dati dettagliati sui requisiti. In questa situazione specifica, l'analisi dei requisiti non era ritenuta necessaria in quanto la priorità era avere un prodotto *software* funzionante piuttosto che un prodotto estremamente robusto. Questo contrasta con quanto appreso nel corso di studi, dove la pianificazione, la progettazione e l'analisi dei requisiti erano considerati aspetti fondamentali nel processo di sviluppo *software*, con un'enfasi maggiore sull'importanza di processi ben definiti.

Inoltre, ho imparato a gestire esigenze particolari, riconoscendo che in alcune situazioni è necessario adattarsi e non sempre seguire rigidamente le *best practices*. Ad esempio, ritengo che accorciare i processi di Analisi e Progettazione sia possibile solo grazie alla presenza di stretta collaborazione interna tra team. In altri casi, la priorità era avere un prodotto [sw](#) funzionante piuttosto che robusto, favorendo magari una raffinazione in un secondo momento. Devo dire che ho ritenuto molto importante tenere a mente questo punto, in quanto questo mi ha permesso di mitigare i rischi che ne derivavano. Al contrario dell'ambiente accademico, dove si tende a perseguire sempre il massimo secondo un punta di vista scientifico, in azienda non sempre è così, e pertanto esiste il rischio di acquisire cattive abitudini.

Infine, la gestione degli imprevisti è stata una lezione importante. All'interno dell'ambiente universitario, durante lo svolgimento di progetti può capitare di imbattersi in imprevisti, ma raramente dell'entità di quelli che è possibile incontrare durante un tirocinio. Mi rendo conto che è importante avere una buona capacità di adattamento, e di saper gestire le situazioni di stress. Allo stesso modo mi rendo conto della difficoltà di replicare imprevisti così grandi all'interno di un ambiente accademico.

4.6 Confronto con le aspettative

Ho cercato di non formare aspettative troppo forti per non incappare in delusioni. Mi sono reso conto che molto spesso arrivare con una mente priva di aspettative significa godersi di più le esperienze e fornisce una mentalità più aperta e flessibile. Nonostante ciò sarei un'automata se non avessi fatto nessun tipo di speculazione su quanto stesse per succedermi, ed in questa sezione affronterò le aspettative che avevo prima di iniziare il mio tirocinio, e come queste si sono rivelate essere diverse dalla realtà.

Inizialmente, mi aspettavo che l'azienda fosse di dimensioni più contenute. Tuttavia, ho scoperto che l'azienda è effettivamente più grande di quanto avevo immaginato. Questo ha portato a un ambiente di lavoro con una struttura organizzativa più complessa e una maggiore diversità di *team* e progetti. Questa scoperta ha influenzato la mia percezione del contesto lavorativo.

Pensavo che l'ambiente lavorativo sarebbe stato molto più impersonale e orientato solo alla produttività. Sorprendentemente, ho trovato che i rapporti umani sono un elemento importante in azienda. C'è spazio per la collaborazione e per costruire relazioni significative con i colleghi. Questo ha reso l'ambiente di lavoro più accogliente e appagante, influenzando il mio approccio alle relazioni interpersonali sul posto di lavoro.

Inoltre, temevo che la routine lavorativa sarebbe stata monotona e noiosa. Tuttavia, ho scoperto che la routine lavorativa è gestibile e che ci sono sfide interessanti da affrontare. Questo ha reso il lavoro meno noioso di quanto temessi. La mia percezione della routine lavorativa è cambiata notevolmente.

Ulteriormente, non avevo previsto che la risoluzione di problemi potesse generare stress. In alcune occasioni, ho sperimentato stress a causa della risoluzione dei problemi. Tuttavia, ho anche imparato a gestirlo e a vedere queste situazioni come opportunità di crescita. Questo aspetto ha influito sulla mia capacità di affrontare le sfide con una mentalità più resiliente.

Le mie aspettative iniziali per il bilancio tra vita e lavoro non erano tra le più rosee. Tuttavia, ho notato che con un po' di abitudine e forza di volontà è assolutamente possibile riuscire a mantenere un buon equilibrio tra interessi personali e vita lavorativa. A volte mi sono trovato comunque in difficoltà, ma devo dire che è accaduto meno di quanto mi aspettassi.

Acronimi e abbreviazioni

B2B *Business to Business.* 1, 48

CMS *Content Management System.* 19, 20, 24–26, 29, 44, 48

DB *Database.* 6, 19, 26

ITS *Issue Tracking System.* 31, 45, 48

RD *Research and developement.* 2, 3, 7, 20, 48

SEO *Search Engine Optimization.* 12, 24, 48

sw *Software.* 20, 45

TT *Takt time.* 4, 49

Glossario

Agile Il termine *Agile* si riferisce a un gruppo di metodi di sviluppo del *software* basati su processi iterativi e incrementali, dove i requisiti e le soluzioni si evolvono attraverso la collaborazione tra *team* auto-organizzati e interfunzionali. Questi metodi o approcci sono stati definiti nel 2001 nel *Manifesto per lo sviluppo agile del software* (ing. *Manifesto for Agile software Development*³). 4

Business to Business con il termine *Business to Business* (ing. azienda ad azienda) si indica un'attività commerciale tra aziende, ovvero un'attività di scambio di beni o servizi tra due o più imprese. 47

Content Management System con il termine *Content Management System* (ing. sistema di gestione dei contenuti) si indica un programma *software* che consente la creazione e la modifica di contenuti digitali utilizzando un'interfaccia grafica. I sistemi di gestione dei contenuti sono spesso utilizzati per gestire siti Web, intranet, portali e applicazioni di gestione dei contenuti aziendali. 47

Hello World Nel mondo dell'informatica, *Hello World* è un semplice programma che, quando eseguito, stampa il messaggio "Hello, World!". È spesso usato per illustrare la sintassi di base di un linguaggio di programmazione per un programma funzionante.. 19

Issue Tracking System con il termine *Issue Tracking System* (ing. sistema di tracciamento delle problematiche) si indica un *software* che aiuta a gestire e mantenere una lista di problemi, come bug, errori e miglioramenti, riscontrati durante lo sviluppo di un progetto *software*. 47

Research and Development con il termine *Research and developement* (ing. ricerca e sviluppo) si indica un'attività di ricerca e sviluppo di nuovi prodotti o servizi, o di miglioramento di quelli esistenti. 47

Search Engine Optimization con il termine *Search Engine Optimization* (ing. ottimizzazione per i motori di ricerca) si indica l'insieme delle attività volte a migliorare la visibilità di un sito Web nei risultati organici dei diversi motori di ricerca. 47

³*Agile Manifesto*. URL: <http://agilemanifesto.org/>.

Takt time *Takt time* è un termine adottato nel contesto del *team* di sviluppo *software*, con origini nel settore manifatturiero e strettamente correlato al concetto di Lean Production. Rappresenta l'intervallo di tempo ideale tra le riunioni ricorrenti o altre attività pianificate, al fine di garantire che il *team* mantenga un ritmo sostenibile di lavoro e comunicazione. Questa metrica, ispirata dalla produzione industriale snella (Lean Production), aiuta il *team* a sincronizzarsi, a identificare tempestivamente i problemi e a mantenere un flusso costante di lavoro. [47](#)

Wizard Nell'ambito dell'informatica, un wizard è un'interfaccia utente guidata progettata per semplificare e automatizzare processi complessi o ripetitivi. Questo strumento fornisce una serie di passaggi o schermate interattive che guidano l'utente attraverso una procedura specifica, come l'installazione di un *software*, la configurazione di un'applicazione o la creazione di un nuovo documento. Ogni schermata del wizard presenta istruzioni chiare e opzioni predefinite per facilitare le scelte dell'utente.. [vii](#), [12](#)

Bibliografia

Riferimenti bibliografici

Frasson, Massimo. *Corporate Governance System - Teorie organizzative applicate alla realtà delle PMI familiari: Il caso Unox S.p.A.* 2017 (cit. a p. 1).

Tidd, Joe, John Bessant e Keith Pavitt. *Managing Innovation: Integrating Technological, Market and Organizational Change.* Wiley, 2001 (cit. a p. 8).