



UNIVERSITÀ DEGLI STUDI DI PADOVA

---

FACOLTÀ DI ...

*Corso di Laurea in ...*

**THESIS TITLE**

**(SUBTITLE)**

*Laureando*

**nome cognome**

*Relatore*

**Prof. nome cognome**

*Co-relatore*

**nome cognome**

---

ANNO ACCADEMICO 20NN/20NN



A ...

Quote  
*Author*



# Indice

<b>1</b>	<b>L'azienda</b>	<b>1</b>
1.1	Presentazione . . . . .	1
1.2	Thron . . . . .	2
1.3	Processi . . . . .	4
1.3.1	Sistema di ticketing . . . . .	5
1.4	Tecnologie . . . . .	6
1.5	Clienti . . . . .	6
<b>2</b>	<b>La proposta di stage</b>	<b>9</b>
2.1	Scopo dello stage . . . . .	9
2.1.1	Introduzione . . . . .	9
2.2	Obiettivi aziendali . . . . .	10
2.2.1	Obiettivi obbligatori . . . . .	10
2.2.2	Obiettivi opzionali . . . . .	10
2.3	Vincoli temporali . . . . .	11
2.4	Aspettative . . . . .	11
2.5	Obiettivi personali . . . . .	11
<b>3</b>	<b>Lo stage</b>	<b>13</b>
3.1	Piano di lavoro . . . . .	13
3.2	Analisi . . . . .	13
3.2.1	Analisi fase 1 . . . . .	14
3.2.2	Analisi fase 2 . . . . .	18
3.2.3	Analisi fase 3 . . . . .	19
3.2.4	Obiettivi . . . . .	19
3.2.5	Tecnologie utilizzate . . . . .	22
3.3	Progettazione e sviluppo . . . . .	23
<b>4</b>	<b>Chapter title</b>	<b>25</b>
4.1	Section title . . . . .	25
4.1.1	Sub-section title . . . . .	25

## INDICE

---

## Sommario

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras mattis tincidunt ligula. Duis ante neque, convallis vel vulputate vel, dignissim vel enim. Proin et iaculis libero. Aliquam erat volutpat. Cras ac purus non ante ultricies scelerisque. Donec lobortis lorem imperdiet leo consequat nec iaculis velit adipiscing. Curabitur nec gravida neque. Nunc vel dui vitae ante dapibus sagittis ac non libero. Suspendisse gravida commodo arcu bibendum luctus. Nam placerat pharetra massa, aliquam rutrum arcu fermentum nec. In non ultrices ante. Pellentesque pretium, felis ac mattis condimentum, dui massa ultricies nisl, hendrerit malesuada magna risus eget dolor. Pellentesque lobortis eleifend nibh, sed gravida sem fringilla eget. Proin pretium, arcu in ornare pellentesque, elit ante faucibus sem, at convallis eros ante ut velit. Donec ornare erat non diam tristique vitae congue nulla commodo. Proin fermentum fringilla mattis. Pellentesque ut dolor hendrerit tellus tincidunt egestas at sit amet velit.





# L'azienda

## Presentazione

Thron è un'azienda nata dalla New Vision, una società per azioni italiana fondata nel 2000 da Nicola Meneghello con lo scopo di convincere le aziende ad usare internet come principale mezzo di comunicazione. L'azienda sin da quando è nata ha sempre avuto un grande interesse per applicazioni web, che non hanno bisogno di installazioni ma sono fruibili direttamente da browser come servizio, questo le rende accessibili da qualunque dispositivo facilitando non poco il lavoro per avere un'applicazione multi-piattaforma.

Inizialmente l'azienda sviluppava un software in flash per videoconferenze multiple, successivamente abbandonato questo progetto si sono focalizzati su un prodotto chiamato 4ME il quale è stato integrato nella piattaforma Thron col passare del tempo.

La struttura dell'azienda è gerarchica con 4 livelli: ogni livello è comandato da quello superiore che gestisce l'operato di quelli inferiori. Nel gradino più in alto si trova il consiglio di amministrazione mentre nel livello inferiore si trovano diversi settori in cui si divide l'azienda, come:

1. Direzione Tecnica
2. Direzione Marketing
3. Direzione Commerciale
4. Direzione Amministrativa.

Ognuno di questi settori è diviso in team o micro team che costituiscono il terzo e quarto livello. La Direzione Tecnica si divide in 4 team:

1. Engineering
2. Ricerca
3. Formazione

## 1. L'azienda

---

### 4. Product Specialist

Il team di Engineering che si occupa dello sviluppo del prodotto è divisa in diversi micro team:

1. Core
2. User Experience
3. DevOps
4. Intelligence
5. Marketplace
6. ItOps
7. Design

Nel mio periodo di stage sono stato inserito nel team di Intelligence, un team recente che si occupa di raccogliere i dati ed analizzarli per poter dare al cliente una visione degli interessi degli utenti così da poter migliorare le strategie di business e di marketing.

## Thron

Thron azienda sviluppa un prodotto omonimo. Questo prodotto è pensato per raggruppare in un unico posto tutti gli strumenti per gestire contenuti digitali di ogni tipo.

L'idea che ha portato alla realizzazione di questo prodotto è che i contenuti sono il vero valore, quindi l'organizzazione deve essere slegata dalla piattaforma di distribuzione. Le piattaforme di distribuzione sono in continua evoluzione, ma il contenuto stesso ed il suo valore rimane e rimarrà sempre lo stesso. Pensando al futuro, quando verranno inventate nuove piattaforme (in particolar modo oggetti smart o wearable nel prossimo futuro) Thron sarà pronta a riceverle ottimizzando il contenuto per la fruizione sulla nuova piattaforma con poco sforzo.

La piattaforma supporta molti contenuti come:

- Documenti come pdf, word, excel, HTML, file di testo
- Immagini di ogni tipo comprese immagini vettoriali e bitmap
- Video

- Video e audio on-demand
- Video e audio live

e molti altri.

Tutti questi contenuti sono raggruppati in un'unica piattaforma che analizza il contenuto e lo arricchisce di informazioni automaticamente con tag<sub>G</sub> e metadati<sub>G</sub>. Questo lavoro è svolto da un motore semantico e analitico che aggiunge queste informazioni sia appena caricato il contenuto sia durante il ciclo di vita di esso. Naturalmente è possibile aggiungere, modificare o eliminare queste informazioni manualmente. Queste informazioni oltre ad arricchire i contenuti di informazioni utili aiutano anche la loro organizzazione e gestione, soprattutto quando la mole di dati inizia ad essere notevole, poichè si possono ordinare o raggruppare contenuti per tag. In TODO Figura 1 vediamo una pagina di un contenuto su piattaforma Thron.

TODO aggiungere figura contenuto con tag... Inoltre per ogni contenuto è possibile scegliere dei permessi, come scegliere che sia visualizzabile solo da alcuni utenti per renderlo privato o visibile solo da un team di interesse, si può dare il permesso di modifica del contenuto o si può garantire il permesso di condividere il contenuto in modo da delegare ad altri la condivisione all'esterno della piattaforma del contenuto.

Per ogni contenuto viene creato uno storico, in questo modo vengono tracciate le azioni fatte su di esso.

Al momento sono tracciate azioni come:

1. Creazione del contenuto (chi lo ha caricato in piattaforma)
2. Modifica dei permessi e quali
3. Visualizzazioni del contenuto, utile per effettuare statistiche di chi li usa, vengono salvate anche le informazioni sul dispositivo utilizzato per la visualizzazione
4. Condivisioni del contenuto
5. Sostituzioni del contenuto, è possibile infatti caricare una nuova versione del file, la vecchia in questo caso viene automaticamente salvata nel caso si volesse tornare ad una versione precedente.

Queste informazioni sono filtrabili per diversi parametri:

1. Data: impostando un arco di tempo predefinito o inserendo data di inizio e di fine

## 1. L'azienda

---

2. Utente o applicazione: vengono visualizzare solo le modifiche fatte da una certa persona o da una specifica applicazione
3. Tipo: per filtrare attraverso i tag o metadati di ogni contenuto
4. Azione: per filtrare per azione eseguita, ad esempio per vedere solo le visualizzazioni.

Un esempio si può vedere in Figura 2. TODO aggiungere immagine history contenuto. Ogni file viene convertito in diversi formati e dimensioni in modo da ottimizzare la sua fruizione in base al canale di distribuzione finale. Ad esempio un'immagine, una volta caricata, viene copiata e ridimensionata con le varie risoluzioni che si usano maggiormanete nella piattaforma.

A seconda della banda disponibile il sistema controlla che la qualità dell'erogazione dei contenuti sia ottimale ove questo sia possibile. Ad esempio, durante la riproduzione di un video, il sistema controlla la velocità con la quale il client riceve il video e automaticamente aumenta o riduce la qualità. Naturalmente anche la qualità può essere impostata manualmente.

Il prodotto viene infine fornito a siti o servizi esterni da dei connettori, in modo da interfacciarsi e integrarsi senza modifiche alla piattaforma. Per esempio è possibile installare dei connettori per i social network più famosi o per CMS<sub>G</sub> come WordPress all'interno di Thron. Questo permette di gestire contemporaneamente un contenuto ovunque esso sia usato. Ad esempio se viene caricata una nuova versione del file questo viene automaticamente modificato ovunque sia stato condiviso. Queste opzioni sono gestite nella sezione Shareboard in Thron come vediamo in Figura 3. TODO aggiungere figura

## Processi

Thron non usa un modello di sviluppo standard per il suo prodotto, ma il tutto è molto paragonabile ad un modello agile.

Il progetto è ormai avviato e ben consolidato perciò la sua evoluzione procede attuando piccoli passi incrementali. Questi vengono fatti interagendo in maniera molto forte con i clienti che vengono spesso interpellati per suggerire anche modifiche o proporre miglioramenti.

Periodicamente vengono fatti degli incontri con clienti ai quali vengono mostrati le evoluzioni, attraverso dei prototipi, per avere dei riscontri sulla qualità e sull'efficacia della soluzione raggiunta.

Dopo questi incontri segue un incontro interno per valutare l'andamento, discutendo sul grado di soddisfazione del cliente e sulle eventuali proposte mosse da quest'ultimo. Sempre in questi incontri interni vengono presentate anche nuove idee per sviluppi futuri.

Tutte queste nuove idee, miglioramenti, modifiche ed errori vengono scremati dal Project Management<sub>G</sub> che le gestisce attraverso un sistema di Ticketing<sub>G</sub> aziendale.

Riassunto in un diagramma il flusso delle attività è come in Figura 4. TODO figura

## Sistema di ticketing

Thron utilizza diversi tipi di ticket per differenziare diverse procedure documentate internamente.

La classificazione dei ticket è:

1. Ticket obiettivo: indica l'obiettivo da raggiungere al termine dell'attività, comprende casi d'uso tipici, i suoi requisiti ed i vincoli
2. Ticket funzionalità: indica la funzionalità di un prodotto che si vuole ottenere con relativa motivazione
3. Ticket attività: definisce un'attività da svolgere con relativa motivazione
4. Ticket fase: definisce il passo evolutivo della funzionalità su cui si sta lavorando
5. Ticket sprint: definisce l'unione di più fasi di lavorazione
6. Ticket lavorazione: descrive il passo evolutivo dell'attività in lavorazione
7. Ticket richiesta: descrive una richiesta da parte di un cliente o di un ticket sprint
8. Ticket difetto: descrive un errore riscontrato sul prodotto.

Questi ticket possono essere assegnati direttamente dal Project Management nel caso richiedano specifiche conoscenze, altrimenti sono presi in carico da persone che hanno finito le proprie mansioni attuali e rientra nella loro sfera di competenza.

## Tecnologie

Occupandosi l'azienda di molti settori dell'informatica, ed essendo divisa in diversi team, non possiede una suite di strumenti comune per lo sviluppo, perciò ogni team decide a se gli strumenti che ritengono più opportuni per gli obiettivi e le attività che gli vengono assegnati.

In comune ci sono gli strumenti di versionamento, comunicazioni e documentazione.

Per il versionamento l'azienda ha un dominio proprio di gitlab con svariate repositories per ogni progetto.

Per la comunicazione l'azienda utilizza Microsoft Outlook per la posta elettronica, questo perchè è un software che funziona su entrambi i sistemi operativi utilizzati per lo sviluppo (Windows e OS X) e poichè consente una facile gestione di eventi e riunioni su calendario. Ogni membro dell'azienda ha un proprio indirizzo email con la forma *nome.cognome@thron.com*. Per comunicazioni più brevi si utilizza un sistema di messaggistica istantanea interno alla piattaforma Thron come possiamo vedere in Figura 5 TODO figura.

Per la documentazione viene utilizzata la suite Office, in particolare Word è utilizzato per la stesura di documenti provvisori e informali, che vengono convertiti in pdf quando il documento viene approvato. Si utilizza anche Power Point per creare presentazioni sia per riunioni interne, per spiegare più efficacemente, sia negli incontri con i clienti.

L'ambiente di sviluppo più comune in azienda è Windows, visto il grande supporto per i software utilizzati e per il vasto uso anche da parte dei clienti, è presente anche OS X in particolare nei team di Design e User Experience. Viene poi utilizzato Ubuntu per i servizi server per la maggiore stabilità e semplicità in questo ambito. Sono poi presenti e accessibili a tutti delle macchine preconfigurate con sistemi operativi e versioni di browser mirate per testare il comportamento del prodotto su ambienti meno comuni.

## Clienti

L'azienda mira ad una vasta ed eterogenea clientela, per questo motivo ha creato un prodotto ampio e facilmente configurabile. In questo modo è il prodotto che si adatta alle esigenze dei diversi clienti. Questo è un punto molto forte, infatti inizialmente il prodotto viene venduto solo con il modulo base che offre le funzionalità standard utili a tutti, poi i vari clienti possono attraverso la sezione Marketplace comprare o aggiungere moduli gratuiti che offrono diverse funzionalità rendendo molto flessibile la piattaforma e

con sempre e soltanto le funzionalità volute. Vediamo in Figura 6 la sezione Marketplace TODO figura marketplace.

L'azienda, nel caso queste soluzioni non siano sufficienti, si propone, qual'ora possibile, di creare e confezionare soluzioni personalizzate e mirate per il caso d'uso specifico del cliente.

Le molteplici e diverse richieste da parte dei clienti contribuiscono l'evoluzione della piattaforma che deve innovarsi continuamente per soddisfarli. Ognuna di queste evoluzioni conferisce a Thron una completezza sempre maggiore. In linea generale il client tipo di Thron cerca un modo semplice di gestire una mole di contenuti più o meno grande, di distribuirli su vasta scala sempre mantenendole il controllo.

Per avere una erogazione il più possibile efficiente l'azienda ha creato una  $CDN_G$  che possiamo vedere in Figura 7 TODO figura cdn.





# La proposta di stage

## Scopo dello stage

### Introduzione

Come già detto in precedenza Thron è una piattaforma che mira a facilitare tutta la gestione dei contenuti. Un punto molto importante per Thron è il poter dare informazioni sui contenuti ai propri clienti per poter migliorare le tecniche di business e di marketing.

Questo non è fatto soltanto aggiungendo tags e metadati ai contenuti ma anche tracciando molteplici informazioni riguardanti la fruizione e le visite sui contenuti stessi sui vari canali di distribuzione con l'uso di una libreria javascript proprietaria.

Le informazioni raccolte vengono poi mostrate sotto forma di grafici di semplice lettura al cliente in apposite sezioni come si vede in Figura x.

TODO figura statistiche contenuti Tra i vari moduli e widget  $G$  che Thron offre ai suoi clienti troviamo il Predictive Content Recommendation (PCR). Questo è un motore raccomandativo in grado di suggerire, in tempo reale, contenuti in base al profilo degli interessi della singola persona e allo storico delle sue precedenti navigazioni. Grazie alla capacità di Thron di capire gli interessi dell'utente, l'azione del PCR è diversa da quella dei tradizionali sistemi raccomandativi che basano i loro suggerimenti su "cosa hanno fatto o visto utenti simili". Thron invece permette una comunicazione personalizzata in linea con gli specifici interessi del singolo utente. L'aspetto rivoluzionario del PCR è che, a partire dalla sua capacità di aggregare contenuti da qualsiasi canale, permette di personalizzare la comunicazione su qualsiasi touch point (dal sito web all'app mobile, dall'e-commerce ai social network), aumentando in modo esponenziale il coinvolgimento dell'utente. Il PCR permette di filtrare inoltre contenuti "già visti o fruiti" dalla raccomandazione, supporta il multi-lingua ed estende la sua azione su qualsiasi canale digitale.

## Obiettivi aziendali

Il PCR è già disponibile ai clienti ed è composto da due parti: una console di gestione della raccomandazione e una serie di widget "ready to use" non direttamente personalizzabili (al netto di modifiche CSS).

Lo step evolutivo che si vuole ottenere è:

- **Console di monitoring delle prestazioni del PCR:** si tratta di integrare nella console di management presente una sezione che consente al cliente di verificare le prestazioni di erogazione della struttura raccomandativa; da una prima analisi emergono i seguenti sviluppi plausibili:
  - Diagramma che indica con granularità oraria il numero di richieste per la singola PCR
  - Lista o diagramma che indica lo sfioramento del numero di richieste rispetto al modello di business attivato
  - Visualizzazione dello storico delle prestazioni del PCR negli ultimi 30 giorni, con intervallo selezionabile
- **Inserimento di un meccanismo di personalizzazione per i widget:** è necessario implementare una consola tramite la quale il cliente scelga il tipo di widget da configurare e ne indichi le personalizzazioni attraverso un'interfaccia usabile anche da utenti non tecnici. I widget così configurati potranno essere inseriti all'interno dei progetti dei clienti tramite copia e incolla.

## Obiettivi obbligatori

- Creare dei diagrammi che indichino, con granularità oraria, il numero di richieste per la singola PCR
- Creare una lista o un diagramma che indica lo sfioramento del numero di richieste rispetto al modello di business attivato
- Creare una visualizzazione dello storico delle prestazioni del PCR negli ultimi 30 giorni, con intervallo selezionabile

## Obiettivi opzionali

- Aggiungere un meccanismo di personalizzazione per i widget PCR

Gli obiettivi sono stati poi elaborati e definiti totalmente durante la fase di analisi a cui ho partecipato di cui si può leggere in [TODO link capitolo](#)

## Vincoli temporali

Per lo stage, l'Università di Padova ha imposto dei vincoli per la durata, che deve essere compresa tra le 300 e le 320 ore. Per questo motivo, prima dell'inizio dello stage, è stato redatto insieme all'azienda un piano di lavoro che comprende 320 ore, contando anche una chiusura aziendale estiva.

## Aspettative

L'azienda aveva un forte interesse per questo progetto, questo perché il progetto è frutto di diverse richieste da parte dei clienti.

Nello specifico i clienti hanno avanzato richieste per avere informazioni riguardanti quanto lo strumento dei raccomandazione sia effettivamente efficace, per cosa questo significhi leggere il paragrafo [TODO LINK](#) in cui si parla delle attività di analisi.

Inoltre questo progetto aprirebbe altre evoluzioni specifiche per il motore di raccomandazione che verranno discusse in seguito.

## Obiettivi personali

Il mio obiettivo più grande era affrontare uno stage che mi permettesse sia di aprire sbocchi lavorativi all'interno dell'azienda sia di arricchire il più possibile il mio bagaglio personale, affrontando sfide mai viste da me prima e che fossero richieste vere e proprie di un'azienda.

A StageIT Thron mi ha fatto subito una buona impressione, proponendomi questo progetto in maniera chiara e facendomi capire che mi avrebbero fatto lavorare in un team coinvolgendomi in tutte le attività legate al progetto così da darmi l'opportunità di ottenere una buona visione del progetto e di una buona formazione con gli altri membri.

In più mi è subito interessato poiché il progetto avrebbe avuto risultati immediati e visibili poiché in caso di esito positivo l'azienda lo avrebbe utilizzato come passo evolutivo mettendo la nuova versione in produzione per i clienti. Questo sommato alle tecnologie da me mai utilizzate o viste brevemente nel corso di studi mi ha spinto a scegliere questo progetto.

## 2. *La proposta di stage*

---

# Lo stage

## Piano di lavoro

Con il tutor aziendale era stato redatto un piano di lavoro di 320 ore suddiviso in questo modo: **Prima settimana**

- Introduzione all'azienda e alle tecnologie di sviluppo

### **Seconda settimana**

- Fine studio tecnologie e partecipazione alla fase di analisi

### **Terza settimana**

- Continua fase di analisi

### **Quarta settimana**

- Studio e progettazione dell'interfaccia di trasferimento dei dati

### **Quinta, sesta, settima ed ottava settimana**

- Implementazione della console di monitoring e del meccanismo di personalizzazione dei widget

## Analisi

La fase di analisi è stata molto importante e più lunga del previsto.

Uno dei motivi che mi hanno spinto a scegliere questo progetto di stage è dato dal fatto che copriva diversi ambiti, non focalizzandosi soltanto ad esempio in uno sviluppo backend o frontend, questo ha significato anche il dover interagire con diversi team aziendali effettuando molti brainstorming con componenti di vari team per collegare il tutto.

Infatti per effettuare la dashboard e presentare i dati in maniera elegante ed

intuitiva ho collaborato col team di design e UX per lo sviluppo delle grafiche e l'integrazione in  $Bacheca_G$ . Per la raccolta dei dati relativi al tracciamento dei dati, la raccolta e modellazione di quest'ultimi ho lavorato col team di Intelligence mentre per la raccolta e modellazione dei dati relativi alla quantità di raccomandazioni effettuate dalle varie applicazioni ho lavorato col team Core.

Ho potuto suddividere quindi tutte le attività in tre fasi sebbene poi interoperino tra di loro:

- **fase 1:** collaborazione col team di intelligente per raccolta e modellazione dati utili alla creazione dei grafici qualitativi $_G$ , ovvero i grafici per soddisfare le richieste dei clienti relative alla bontà delle raccomandazioni;
- **fase 2:** collaborazione col team di core per la raccolta e modellazione dati utili alla creazione dei grafici quantitativi $_G$ , ovvero i grafici per soddisfare le richieste dei clienti relative alla quantità di raccomandazioni fatte ogni ora e la comunicazione in caso di sfioramento di richieste;
- **fase 3:** collaborazione col team di design e di UX per lo sviluppo dei grafici e l'integrazione di quest'ultimi all'interno del software  $bacheca_G$ .

## Analisi fase 1

Per prima cosa si è scelto di analizzare come raccogliere i dati necessari per poter capire e fornire al cliente delle metriche e valutazioni di efficienza dello strumento  $PCR_G$ . Infatti i clienti tra le richieste volevano sapere quanto efficace il  $PCR_G$  fosse, questo però è un concetto non basilare e non definito dai clienti stessi perciò molto tempo dell'analisi è stato impiegato a pensare come valutare questo.

Ho collaborato con tutto il team di Intelligence e con uno data scientist $_G$  per cercare di definire questo concetto. Sono state vagliate diverse proposte e fatte diverse ricerche in internet, alla fine si è scelto l'approccio che conciliasse una buona precisione sull'efficacia dello strumento dovendo però rientrare in tempi non molto lunghi per i vincoli di stage.

La definizione risultata è la seguente:

per ogni contenuto devo essere tracciate almeno tre azioni:

1. Contenuto proposto
2. Contenuto proposto e visto dall'utente
3. Contenuto proposto, visto e l'utente ha espresso un interesse esplicito

D'ora in poi useremo terminologie specifiche per questi eventi, in particolare

1. Load: contenuto proposto
2. Impression: contenuto proposto e visto dall'utente
3. Interact o click: contenuto proposto, visto e l'utente ha espresso un interesse esplicito

Tramite questi eventi l'azienda è in grado di fornire al cliente varie informazioni tra cui:

- I contenuti più proposti
- I contenuti più visti
- I contenuti più cliccati

Mentre è facile capire che i contenuti più proposti siano i contenuti caricati più volte dai widget di raccomandazione (ovvero contenuti caricati nella pagina web), è importante capire la differenza tra visto e cliccato. Per contenuto visto (impression) si intende un contenuto proposto dal PCR e che è entrato nella viewport<sub>G</sub> dell'utente finale (ma che non necessariamente l'utente ci ha interagito). Per contenuto cliccato si intende un contenuto visto dall'utente in più su cui quest'ultimo ha fatto un click per aprirlo.

Infatti i widget di raccomandazione contengono una lista di contenuti con una thumbnail<sub>G</sub> e una breve descrizione, per aprire il contenuto vero e proprio l'utente può cliccarci sopra. Figura x TODO img per

Durante l'analisi si è capito che la sola azione di caricare il contenuto dal widget (evento load) non portava necessariamente informazioni molto utili poiché se un contenuto non è neanche entrato nella viewport dell'utente si può non contare proprio così abbiamo deciso di tracciare questo evento per avere più informazioni magari in futuro ma per ora ci concentreremo più sugli altri due eventi.

Abbiamo infatti, dopo brainstorming e ricerche in internet, trovato una metrica che desse un'indicazione soddisfacente dell'efficacia di un contenuto, questa metrica si chiama **click through rate**. In pratica è un valore calcolato dal rapporto del numero di interazioni con il contenuto per il numero di viste (clicks diviso impressions). Un valore alto infatti sta ad indicare che il contenuto è stato aperto molte volte, se non la totalità, rispetto alle volte in cui è stato visto. Questo fatto indicherebbe che il contenuto raccomandato è interessante per quasi tutti gli utenti a cui è stato proposto indicando un ottimo lavoro da parte del motore di raccomandazione. Valori invece molto

bassi di questo indice significherebbero che il contenuto è stato visto molte volte ma aperto poche, quindi provocando scarso interesse da parte degli utenti finali.

Si è deciso quindi che questo fosse uno dei valori, per ogni contenuto, da presentare al cliente, però eliminando i casi particolari come gli elementi con un numero di visite al di sotto di una certa soglia. Questo perché uno scarso numero di visite e pochi click avrebbe comunque un click through rate più alto di contenuti magari più significativi. Per esempio un contenuto con 3 click e 4 impression avrebbe un click through rate di 0.75, indice molto alto, mentre un contenuto con 1000 impression e 500 click avrebbe un click through rate di 0.5, il che porterebbe a pensare che il contenuto sia molto meno interessante ma è molto probabile che in realtà sia il contrario perché con numeri molto bassi è facile avere un rapporto alto e non veritiero.

Tutte queste considerazioni valgono non solo per contenuti caricati dal widget di raccomandazioni ma per qualunque contenuto THRON messo in pagina. Questo è stato motivo di ulteriore stimolo poiché erano funzionalità aggiunte non solo per uno specifico progetto ma che saranno riutilizzate in futuro in altri progetti dell'azienda.

#### **Tracciamento eventi - libreria dei tracciamenti**

Per raccogliere le informazioni necessarie Thron usa una libreria javascript che espone dei metodi per tracciare eventi. Questa libreria deve essere inclusa nelle pagine web o applicazioni e aggiunta in pagina la logica per tracciare queste informazioni. Un esempio banale: se si volesse acquisire l'informazione di quante volte viene aperta la scheda di un contenuto in un sito di e-commerce basterebbe lanciare un evento load al caricamento della pagina relativa dell'oggetto da vendere sul contenuto di tipo immagine relativo all'oggetto.

La libreria dei tracciamenti Thron supportava diversi eventi da loro già tracciati, tuttavia non disponeva di metodi per tracciare impressions e clicks perciò era necessario aggiungere queste funzionalità. Mi è stato proposto di effettuare io stesso queste aggiunte e ho accettato poiché andava ad aumentare ancor di più l'arricchimento personale che cercavo.

Inoltre si è scelto di aggiungere alla libreria la logica per tracciare questi eventi in automatico, cioè senza aggiunte di codice lato client in pagina, per facilitare ulteriormente i tracciamenti non solo per il widget<sub>C</sub> raccomandativo ma anche per qualunque altra pagina web che necessiti di tracciare eventi come click, load, impression o altri eventi che verranno aggiunti in futuro.

Questo ha provocato una modifica al mio piano di lavoro immettendo qualche giorno di lavoro (stimati cinque) per studiare la libreria dei tracciamenti,



le tecnologie utilizzate, progettare come apportare le modifiche e svilupparle prima dello sviluppo del resto del progetto.

#### Raccolta ed elaborazione dati tracciamenti

La libreria dei tracciamenti di THRON utilizza `snowplowG`, una libreria atta appunto a tracciare eventi anche personalizzati. Grazie a questa tecnologia tutti i dati relativi ai tracciamenti restano proprietari dell'azienda al contrario di altri servizi di tracciamenti come Google Analytics. Il vantaggio di `snowplow` è sicuramente il fatto di essere proprietari di tutti i dati tracciati e la totale personalizzazione che si può fare, sia negli eventi da tracciare sia nelle fasi successive di elaborazione di questi dati. Il lato negativo è che bisogna scriversi tutto il codice necessario per gli eventi personalizzati e scrivere il codice dei software che ricevono i tracciamenti, che gli elaborano e che li mettono a disposizione di eventuali client.

I dati tracciati dalla libreria `javascript` presente in tutte le pagine necessarie vengono raccolti in un collector che può essere visto come un calderone in cui arrivano e vengono raggruppati tutti gli eventi tracciati. Questi dati vengono letti da un enricher, realizzato con `Apache SparkG`, che si occupa di arricchirli con altre informazioni, ad esempio viene effettuato il `mergeG` dei contatti, questo va a rendere possibili filtraggi molto mirati e analisi di dati per strategie business impossibili altrimenti.

I dati a questo punto vengono elaborati ulteriormente da diversi enricher (altre istanze `Spark`) ed ognuno aggiunge informazioni utili per diversi scopi. Al momento ne hanno tre:

- Recommendation: questo contiene tutta la logica per la raccomandazione dei contenuti e l'aggiunta di importanti informazioni
- Behaviour: questo serve per analizzare il comportamento degli utenti
- Reports: questo serve per arricchire con moltissimi dati gli eventi per rendere possibile la fruizione di informazioni utili per business strategy

Tutte queste istanze salvano poi i dati in diversi `Elasticsearch` che possono essere interrogati da web services appositi dall'esterno.

Andando ad aggiungere nuovi eventi è necessario modificare il primo enricher comune per aggiungere il support a questi nuovi eventi. Per i tre enricher specifici invece è necessario effettuare queste modifiche:

- Recommendation: filtrare i nuovi eventi, per ora gli algoritmi di raccomandazione non utilizzeranno impressions e interacts. Questa è una

modifica che sarà effettuata nel prossimo futuro poiché non banale da applicare. Infatti il recommendation utilizza algoritmi di machine learning per essere più preciso ed efficiente

- Behaviour: anche in questo caso filtrare i nuovi eventi, come per il recommendation sono modifiche da applicare con molta cautela in algoritmi delicati e lo stage non prevede abbastanza tempo
- Reports: qui al contrario bisogna fare in modo che i nuovi eventi vengano letti, elaborati ed arricchiti con tutte le informazioni necessarie. Inoltre i dati che vengono salvati su Elasticsearch è necessario indicizzarli raggruppandoli per tipo di evento per maggiore ordine

I web services per ricavare questi di dati verranno sviluppati da altri membri del team di Intelligence essendo presenti già servizi molto simili sono modifiche che applicheranno molto velocemente.

## **Analisi fase 2**

Successivamente ho partecipato a delle riunioni col team Core per capire come rendere disponibili ai clienti i dati relativi alla quantità di raccomandazioni fatte dalle loro applicazioni. Questo è molto importante poiché i clienti al momento dell'acquisto dello strumento di raccomandazioni possono scegliere diversi Business Model che prevedono una quantità di raccomandazioni massima prefissata. Superata questa soglia lo strumento non raccomanda più secondo algoritmi intelligenti ma propone solo contenuti simili.

THRON possiede un Elasticsearch<sub>G</sub> in cui vengono immagazzinati tutti gli accessi a tutte le loro API. Tuttavia questi dati vengono salvati per un massimo di sessanta giorni poiché la mole di dati è non indifferente (si parla di almeno quattro gigabyte di log di accessi al giorno contando solo gli accessi alle API, senza contare quindi le fruizioni delle loro cdn<sub>G</sub>).

I dati da fornire ai clienti devono avere una retention<sub>G</sub> di almeno due anni, per questo motivo si è deciso di creare un qualche processo che ad intervalli regolari andasse ad interrogare l'Elasticsearch di monitoring, ricavesse tutte le informazioni riguardanti gli accessi alle raccomandazioni e le andasse a salvare in qualche altro database.

I grafici potranno avere una granularità oraria, giornaliera o mensile dovranno essere quindi salvati i dati raggruppati per queste granularità per avere un output immediato.

Per rendere poi questi dati accessibili all'esterno della rete aziendale è necessario anche creare dei webservices da aggiungere a quelli già esistenti in THRON. Questo lavoro mi era di molto facilitato grazie al fatto che ci fossero

già molti altri webservices, infatti potevo riutilizzare molti componenti già esistenti come il meccanismo di autenticazione e meccanismi di accesso ai database.

Essendo la quantità di dati elevata di è deciso, qual'ora il tempo lo permettesse, di aggiungere anche un servizio che andasse a ripulire i dati più vecchi di due anni con granularità maggiore di quella mensile.

## Analisi fase 3

Infine sono state fatte le analisi per discutere come presentare questi dati ai clienti.

col team di design si sono scelti i tipi dei grafici da presentare per ottenere una maggior usabilità possibile e per rendere i dati di facile comprensione, cosa non banale avendo moltissime informazioni da esporre.

Col team UX si è discusso invece di come integrare questi grafici nel loro software in cui sono già presenti molti grafici per la presentazione di informazioni di Intelligence. TODO aggiungere img grafici intelligente

## Obiettivi

Dopo tutta la fase di analisi sono emersi i seguenti obiettivi

### Fase 1

#### Obbligatorî:

- Dovrà essere aggiunta la funzionalità di tracciare impression nella libreria dei tracciamenti
- Dovrà essere aggiunta la funzionalità di tracciare click nella libreria dei tracciamenti
- Dovrà essere aggiunto un meccanismo per tracciare eventi come click, load e impression in automatico.
- Le modifiche da apportare alla libreria dei tracciamenti dovranno mantenere una retro compatibilità totale
- Tutte le modifiche devono essere supportate nelle seguenti versioni dei browsers:
  - Internet Explorer 9 e successivi

### 3. *Lo stage*

---

- Firefox 3.5 e successivi
  - Chrome tutte le versioni
  - Safari OSX 5 e successivi
  - Android 4.1 e successivi: Chrome tutte le versioni
  - IOS 4.x e successivi: Safari 4 e successivi, Chrome tutte le versioni
- Dovrà essere modificato il collector per salvare i nuovi tipi di evento
  - Dovrà essere modificato l'enricher per aggiungere il supporto ai nuovi eventi
  - L'enricher dovrà salvare tutti gli eventi in indici diversi raggruppati per tipo di evento

#### **Desiderabili:**

- Il meccanismo contenente la logica di automazione degli eventi dovrà essere aperto ad estensioni per futuri eventi

### **Fase 2**

#### **Obbligatori:**

- Creare un servizio che interroghi Elasticsearch di monitoring, ricavi i dati degli accessi alle raccomandazioni e li salvi in un database
- Questo servizio deve salvare gli accessi con granularità oraria
- Questo servizio deve salvare gli accessi con granularità giornaliera
- Questo servizio deve salvare gli accessi con granularità mensile
- Sviluppare dei web services interrogabili dall'esterno che permettano di interrogare il database
- Questi web services devono prevedere politiche di accesso e un sistema di permessi uguali agli altri web services THRON

#### **Desiderabili:**

- Sviluppare un servizio che ripulisca i dati, con granularità oraria e giornaliera, più vecchi di due anni

### Fase 3

#### Obbligatorî:

- Realizzare un grafico che indichi il numero di raccomandazioni fatte da un applicazione raccomandativa master (grafico quantitativo)
- Questo grafico deve permettere diverse granularità, in particolare oraria, giornaliera e mensile
- Questo grafico deve poter dare la possibilità di scegliere una data di inizio e di fine per filtrare i dati
- Questo grafico deve poter mostrare, con possibilità di filtraggio, il numero di raccomandazioni delle singole applicazioni raccomandative slave
- Per ogni applicazione slave devono essere realizzati altri grafici per analizzare la bontà delle raccomandazioni (grafici qualitativi)
- Un grafico che mostri i cinquanta contenuti raccomandati visti più volte (ovvero con più impression) con la relativa quantità di click sul contenuto
- Un grafico che mostri gli argomenti relativi ai contenuti più visti
- Un grafico che mostri il profilo dei contatti associato ai contenuti più visti
- Al click su un contenuto del primo grafico apra un popup che contenga
  - Una descrizione del contenuto
  - Il numero di impression
  - Il numero di click
  - I tag del contenuto
  - Le tipologie degli utenti che hanno interagito col contenuto
  - Gli argomenti più trattati da queste tipologie di utenti
- Dare la possibilità di filtrare per periodo qualunque
- Dare la possibilità di cercare specifici contenuti (nel caso non siano tra i primi cinquanta)
- Dare la possibilità di ordinare per più visti o più cliccati

#### Opzionali:

- Sviluppare una sezione in cui vengono dati consigli sulle raccomandazioni, ad esempio consigliare quali contenuti si consiglia di togliere dalle possibili raccomandazioni

## Tecnologie utilizzate

### Javascript

Javascript è un linguaggio di scripting orientato agli oggetti e agli eventi. Principalmente è utilizzato per la programmazione lato Client negli ambienti Web per aggiungere funzionalità, effetti dinamici o interattività alle pagine o applicazioni web.

Da qualche tempo è però utilizzato in molti altri ambiti come la programmazione di applicazioni mobile tramite frameworks come ionic o appcelerator. Essendo Thron una piattaforma web Javascript è un linguaggio pressoché obbligatorio da imparare per lavorare in azienda.

Nel mio stage l'ho utilizzato largamente per apportare le modifiche alla libreria dei tracciamenti visto che quest'ultima è interamente una libreria realizzata in javascript. In più l'ho utilizzato anche per aggiungere il tracciamento automatico nei widget di raccomandazione, essendo anche quest'ultimi realizzati in javascript.

Per comodità e necessità ho utilizzato diverse librerie come **Lodash**, **CommonJS**, **JQuery** TODO aggiungere lib.

TODO fatto? Inoltre ho utilizzato javascript anche per l'esposizione dei dati, ovvero per la creazione della console in cui sono esposti dei grafici come spiegato nei capitoli a seguire. Per questi l'azienda mi ha dato piena facoltà di scegliere e decidere cosa usare così ho studiato e provato le tre librerie che mi sembravano più adatte: **FusionCharts**, **KendoUI** e **d3js**.

FusionCharts è una libreria a pagamento nata nel 2003 e offre una vasta gamma di diagrammi, mappe, widgets e dashboards già fatti con cui arricchire i propri siti o applicazioni web. Dopo averla provata è risultata essere molto facile da imparare e immediata all'uso, quasi non necessita di conoscenze di programmazione. Molto belli e veloci i risultati che si possono ottenere tuttavia meno personalizzabili delle altre librerie provate proprio per la sua grande semplicità. Un altro punto a favore è la retro compatibilità molto estesa e resa automatica dalla libreria.

D3js, che sta per data driven documents, è un'altra libreria nata nel 2011 per la visualizzazione dinamica e interattiva di dati organizzati, quindi offre funzionalità a basso livello per creare grafici. Si può arrivare a dei risultati

sbalorditivi e davvero di forte impatto, tuttavia a me non servivano diagrammi particolarmente complessi e in più necessita di molto più lavoro per ottenere buoni risultati. In questo caso la retro compatibilità non è sempre garantita per le versioni dei browser non recenti.

KendoUI è un framework che offre una grandissima quantità di funzionalità built-in ma io mi sono concentrato nelle API per creare grafici. Anche in questo caso molto semplice da usare, risultati immediati e molto belli, in più offre più personalizzazione e una buona retro compatibilità.

La scelta è ricaduta su FusionCharts poichè molti grafici già realizzati da THRON sono creati usando questa libreria, quindi risultava essere tutto più omogeneo utilizzando la stessa, inoltre essendo un pò stretti con i tempi l'uso di FusionCharts ho pensato potesse aiutare a farmi rientrare nei tempi.

## Progettazione e sviluppo

### 3. *Lo stage*

---



# Chapter title

## Section title

Sed varius rhoncus libero a consequat. Cras facilisis magna eget tellus laoreet sit amet mattis nulla posuere. Nullam magna est, porta a feugiat quis, mollis vel urna. Nunc ante dolor, pretium eget laoreet in, tincidunt vitae ipsum. Nulla et diam risus. Ut auctor auctor vestibulum. Vestibulum vitae turpis sit amet lacus pulvinar dictum laoreet vitae enim. Aliquam erat volutpat. Aliquam ultricies posuere sem, ac mollis nunc interdum in. Vivamus tempor felis a tellus volutpat ut elementum lorem congue. Proin purus tortor, ultricies vitae viverra non, feugiat ac felis. Fusce condimentum dignissim volutpat. Proin quis augue ac tortor mollis congue at et magna. Nullam a velit est, nec ultrices justo (vedi Figura 4.1).

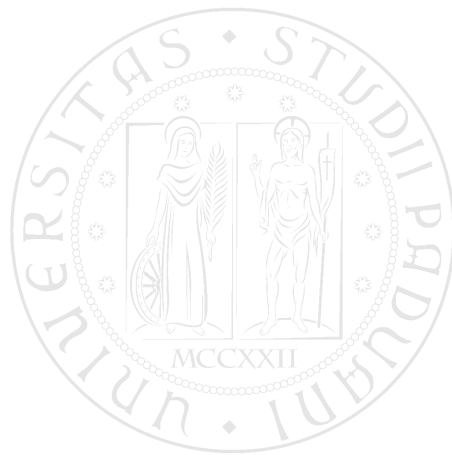


Figura 4.1: Image caption

## Sub-section title

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas (vedi Listing 4.1). Suspendisse arcu magna, faucibus ut tincidunt non, ultrices ut turpis. Nullam tristique vehicula massa, id commodo orci sollicitudin vel. Donec nibh ante, ultrices non facilisis sed, mattis id ligula. Sed sed orci sit amet nulla egestas gravida. Suspendisse laoreet, massa vel sagittis gravida, lectus ligula feugiat risus, a aliquam dolor eros ac orci. Nulla egestas tortor quis nunc scelerisque sed tincidunt massa scelerisque. Pellentesque vulputate pharetra lectus, vitae ultricies nisi luctus eu. Nam congue dui eu quam euismod vitae fermentum sem vehicula. Etiam ac leo id nisi placerat posuere. Curabitur mattis augue eget dolor tempus accumsan consequat diam imperdiet. Sed tristique orci id lacus vulputate rhoncus. Morbi tincidunt ante sed turpis luctus tincidunt et sit amet augue. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nunc viverra urna non libero sodales euismod et eleifend sapien. Donec aliquet risus non massa dignissim sollicitudin. Integer a ligula eros. Morbi et lacinia augue [?].



```
<p>
Pellentesque ac tortor eget eros iaculis euismod
vitae vitae augue.
</p>
<!-- comment -->
```

Listing 4.1: caption text