



UNIVERSITÀ DI PISA

Laurea Magistrale in Informatica Umanistica

Visual Analytics

A.A. 2021/2022

## Urban Economic Pulse

Exploring Economic Models and Trends of City Life

**Giacomo Cerretini**

g.cerretini2@studenti.unipi.it

543999

### Sommario

In questa relazione, vengono delineate le diverse fasi di progettazione e sviluppo del progetto d'esame relativo al corso di Visual Analytics, focalizzandosi in particolare sulle sfide proposte dalla VAST Challenge 2022 e ponendo un'enfasi predominante sull'aspetto economico.

GitHub repository: [https://github.com/giacomocerre/VAST\\_2022/tree/master](https://github.com/giacomocerre/VAST_2022/tree/master)

# Indice

<b>1</b>	<b>Introduzione</b>	<b>2</b>
1.1	VAST Challenge 2022 . . . . .	2
1.2	Il Dataset . . . . .	2
<b>2</b>	<b>Pre-Processing</b>	<b>4</b>
2.1	Costruzione e Caratterizzazione del Dataset . . . . .	4
2.1.1	Creazione di <i>buildings.json</i> . . . . .	5
2.1.2	Creazione di <i>places.json</i> . . . . .	5
2.1.3	Creazione di <i>calendar_wesfr.json</i> . . . . .	6
2.1.4	Creazione di <i>calendar_path_expenses.json</i> . . . . .	7
2.1.5	Creazione di <i>calendar.json</i> . . . . .	8
2.1.6	Creazione di <i>participants.json</i> . . . . .	9
2.1.7	Creazione di <i>global_financials.json</i> . . . . .	10
<b>3</b>	<b>Struttura del Progetto</b>	<b>11</b>
3.1	Client: Struttura e Progettazione della Web App . . . . .	11
3.1.1	Struttura della Dashboard . . . . .	12
3.1.2	Utilizzo delle API . . . . .	13
3.2	Server: Progettazione Strutturale e API . . . . .	14
<b>4</b>	<b>Design Process</b>	<b>16</b>
4.1	Scelta dei Colori . . . . .	16
<b>5</b>	<b>Usare e Interagire con le Visualizzazioni</b>	<b>18</b>

# 1 Introduzione

## 1.1 VAST Challenge 2022

Riguardo alla proposta avanzata per l'evento VAST Challenge 2022<sup>1</sup>, l'obiettivo principale del presente progetto è sviluppare una soluzione di Visual Analytics volta a delineare la realizzazione di un'applicazione web in relazione a una delle sfide proposte dall'evento.

Le sfide proposte da tale evento sono le seguenti:

### 1. Sfida 1: Demografia e Relazioni

Questa sfida richiede la comprensione approfondita dei dati demografici della città. Utilizzando i dati provenienti dai social network e altre fonti informative sulla città, l'analisi si concentrerà sulla preparazione di una scheda informativa di una pagina relativa ai dati demografici della città, dei suoi quartieri e della sua base commerciale.

### 2. Sfida 2: Patterns of Life

La seconda sfida si concentra sui modelli di vita quotidiana nell'intera città. L'obiettivo è descrivere le routine quotidiane di individui rappresentativi, caratterizzare i modelli di viaggio per identificare potenziali colli di bottiglia o pericoli, ed esaminare come questi modelli variano nel tempo e nelle stagioni.

### 3. Sfida 3: L'economia

La terza sfida riguarda la valutazione della salute finanziaria della città nel tempo. Sarà analizzato l'andamento delle imprese, le variazioni nel mercato del lavoro, e le dinamiche degli standard di vita nel corso del tempo.

Ogni singola sfida sopra citata trova sostegno per la sua realizzazione nella vasta base di dati messa a disposizione dai promotori dell'evento<sup>2</sup>.

Attraverso un'analisi approfondita e un adeguato pre-processing di questi dati (Capitolo 2), è stato possibile dedicarsi alla progettazione e alla concretizzazione del progetto finale, focalizzandosi in modo specifico sulla terza sfida.

## 1.2 Il Dataset

Il dataset proposto si compone di diverse tipologie di file CSV divisi in cartelle. All'interno di esso, è possibile identificare una varietà di categorie, tra le quali:

### 1. /Activity Logs

1.1 **ParticipantStatusLogs<n>.csv:** Contiene informazioni sullo stato di ciascuno dei circa 1.000 partecipanti nel corso del periodo di raccolta dati di 15 mesi, registrato in incrementi di 5 minuti. Questi dati (18Gb) sono suddivisi in 72 file.

### 2. /Attributes

2.1 **Apartments.csv:** Contiene informazioni sugli appartamenti residenziali nella città.

2.2 **Buildings.csv:** Contiene informazioni sugli edifici della città coinvolti nello studio, compresi quelli commerciali, residenziali e scolastici.

2.3 **Employers.csv:** Contiene informazioni sulle aziende e le attività commerciali all'interno dei limiti della città che impiegano partecipanti allo studio o che hanno aperture di posti di lavoro disponibili.

---

<sup>1</sup>VAST Challenge - <https://vast-challenge.github.io/2022/about.html>

<sup>2</sup>I dati possono essere scaricati in formato compresso tramite il sito e sono formattati in file di tipo CSV (Comma-Separated Values).

- 2.4 **Jobs.csv:** Contiene informazioni sulle opportunità di lavoro disponibili nella città all'inizio dello studio. Alcuni di questi lavori sono occupati dai partecipanti all'inizio dello studio, mentre altri sono posizioni aperte.
  - 2.5 **Participants.csv:** Contiene informazioni sui residenti di Engagement, OH che hanno accettato di partecipare a questo studio.
  - 2.6 **Pubs.csv:** Contiene informazioni sui pub all'interno dei limiti della città.
  - 2.7 **Restaurants.csv:** Contiene informazioni sui ristoranti all'interno dei limiti della città.
  - 2.8 **Schools.csv:** Contiene informazioni sulle cinque scuole della città.
3. **/Journals**
- 3.1 **CheckinJournal.csv:** Contiene informazioni sui check-in dei partecipanti presso varie località. Fornisce un riassunto delle principali informazioni sugli eventi di posizione contenute nei Log dei partecipanti.
  - 3.2 **FinancialJournal.csv:** Contiene informazioni sulle transazioni finanziarie.
  - 3.3 **SocialNetwork.csv:** Contiene informazioni sulle relazioni sociali in evoluzione dei partecipanti.
  - 3.4 **TravelJournal.csv:** Contiene informazioni sulla motivazione degli spostamenti dei partecipanti nella città. Fornisce un riassunto e ulteriore contesto riguardo alle informazioni sugli eventi di posizione e alle transazioni finanziarie contenute nei Log dei partecipanti.

Nella sezione seguente di questo documento, sarà esposta un'analisi riguardante la natura e il metodo di utilizzo dei dati impiegati, ponendo particolare attenzione sulla fase di pre-processing e sui parametri adottati in questo contesto al fine di rispondere adeguatamente ai requisiti correlati alla sfida selezionata.

## 2 Pre-Processing

Il progetto ha avuto inizio con una fase preliminare di analisi dei dati, mirata a comprendere approfonditamente le potenzialità e le peculiarità degli stessi.

Concretamente, si è proceduto alla ricerca delle informazioni pertinenti all'interno dei diversi dataset disponibili, al fine di estrarre dati rilevanti e identificare connessioni significative. L'obiettivo principale di questa fase è stato quello di massimizzare l'utilizzo efficace dei dati al fine di guidare la successiva fase di sviluppo del progetto.

Nell'implementazione operativa del progetto, si è fatto ricorso a script Python allo scopo di effettuare un'elaborazione dei dati in maniera altamente configurabile. La scelta di utilizzare il formato JSON<sup>3</sup> anziché CSV è stata motivata dalla natura finale del nostro progetto che vede la sua realizzazione in una Web App basata su Vue.js<sup>4</sup>.

L'utilizzo di JSON si sposa in maniera più coesa con le esigenze della nostra applicazione web, in quanto JSON offre una struttura gerarchica e la possibilità di rappresentare dati complessi in maniera più articolata rispetto a CSV. La flessibilità di JSON nel gestire dati annidati e strutture più complesse risulta particolarmente vantaggiosa per il nostro contesto, in cui la presentazione e la manipolazione di informazioni complesse costituiscono requisiti chiave.

Inoltre, l'interoperabilità naturale di JSON con JavaScript, linguaggio principale per lo sviluppo del frontend con Vue.js, ha influito positivamente sulla scelta, facilitando l'integrazione fluida dei dati elaborati nella nostra applicazione web. Tale decisione è stata guidata da considerazioni pratiche volte a garantire una gestione agevole e ottimizzata dei dati nel contesto specifico del nostro progetto.

Nel capitolo precedente (1), è stata descritta l'ampia natura dei dati caratterizzati da diversità e molteplicità. Al fine di ottimizzare la gestione di tali informazioni, è stata adottata la strategia di ridurre la quantità totale di file JSON, privilegiando la qualità espositiva e la rappresentatività del contesto d'uso. Tale selezione mira principalmente a mettere in relazione i dati di natura economica, affinché possano essere presentati in maniera più esaustiva e significativa.

### 2.1 Costruzione e Caratterizzazione del Dataset

Come precedentemente delineato, la fase di pre-processing ha coinvolto l'impiego del linguaggio di programmazione Python per la creazione di un insieme di file JSON che costituiscono il dataset finale utilizzato così composto:

#### 1. Dati Città

1.1 `building.json` – descrive i dati relativi alla costruzione della mappa della città

1.2 `place.json` – descrive le informazioni di ogni singolo place della città

#### 2. Dati Temporali

3.1 `calendar_wesfr.json` – descrive gli andamenti temporali di ogni utente per "wage", "education", "shelter", "food", "recreation"

3.2 `calendar_path_expenses.json` – descrive i percorsi tra i vari place fatti da ogni singolo utente giorno per giorno

3.3 `calendar.json` – struttura dati di un calendario che prevede le aggregazioni globali, annuali, mensili e giornaliere dei dati di "wage", "education", "shelter", "food", "recreation", "saving index"

---

<sup>3</sup>JSON - JavaScript Object Notation è un formato per lo scambio dati basato sul linguaggio di programmazione JavaScript. È utilizzato in programmazione web come alternativa al formato XML. Originariamente descritto nella RFC 4627, è diventato uno standard ECMA nel 2013 come ECMA-404

<sup>4</sup>Vue.js è un framework JavaScript open source in configurazione Model-view-viewmodel per la creazione di interfacce utente e single-page applications.

### 3. Dati Demografici

2.1 `participants.json` – descrive i dati relativi ad ogni singolo utente

2.2 `global_financials.json` – descrive le informazioni economiche globali per giorno, settimana, mese e anno

Il processo di generazione di ciascun file si articola attraverso una sequenza di fasi strategiche. Inizialmente, avviene la fase di **acquisizione dati (Data Acquisition)**, durante la quale vengono estratti dati da uno o più file CSV, e in alcuni casi, anche da file JSON precedentemente generati.

Successivamente, si prosegue con una fase di **esplorazione dei dati (Data Exploration)**, attentamente progettata per comprendere la natura intrinseca dei dati e ottimizzare le operazioni nelle fasi successive.

Dopodiché, i dati acquisiti attraversano una **fase di elaborazione (Data Processing)** estesa, dove vengono sottoposti a una serie di operazioni e trasformazioni mirate. Questa fase è concepita per garantire la coerenza, completezza e validità dei dati, tramite l'applicazione di logiche specifiche. Le operazioni eseguite possono includere filtraggio, normalizzazione e arricchimento dei dati, a seconda delle necessità del contesto.

Infine, il processo si conclude con la **fase di export o salvataggio (Data Transformation)** del nuovo file JSON. In questa fase, il risultato dell'elaborazione viene attentamente strutturato e formattato secondo gli standard definiti, prima di essere salvato come un nuovo file JSON. Questa fase non solo comprende la trasformazione dei dati, ma anche la loro preparazione per un utilizzo ottimale in contesti successivi.

#### 2.1.1 Creazione di *buildings.json*

##### Fase di Acquisizione Dati

Per generare il file `buildings.json`, è stato impiegato il file `Building.csv` proveniente dalla base di dati iniziale.

##### Fase di Esplorazione Dati e di Data Processing

Durante questa fase, non è stata condotta un'elaborazione complessa dei dati; si è, piuttosto, effettuata una mera trasformazione da CSV a JSON, preservando le informazioni nella loro forma originale.

##### Fase di Data Transformation

La struttura JSON finale prevede un array di oggetti, dove ogni oggetto identifica un singolo edificio ed è strutturato così:

- `id` (Number) - Identificatore di un edificio
- `location` (String) - Poligono che Identifica l'impronta geografica dell'edificio.
- `buildingType` (String) - Può essere "Commercial" o "Residential" o "School"
- `maxOccupancy` (Number) - Numero massimo di persone ammesse in quel edificio.
- `units` (Array<Number>) - Lista di `PlaceId` che fanno parte dell'edificio

#### 2.1.2 Creazione di *places.json*

##### Fase di Acquisizione Dati

Per la generazione del file `places.json`, si è proceduto all'impiego di tutti i file associati ai singoli luoghi ("place"). Nello specifico, i file utilizzati per tale operazione sono `Pubs.csv`, `Schools.csv`, `Restaurants.csv`, `Apartments.csv`, `Employers.csv` i quali costituiscono una parte dei dati originari.

## Fase di Esplorazione Dati

In questa fase, è emerso che tutti i file del dataset originale presentavano sostanzialmente le stesse informazioni. Di conseguenza, è stato deciso di procedere con una consolidazione unificata, selezionando esclusivamente i dati rilevanti ai nostri obiettivi.

## Fase di Data Processing

Nel corso della complessa fase di elaborazione dati, si è attuata una generalizzazione delle informazioni per la maggior parte dei file, al fine di garantire uniformità e coerenza. Tuttavia, va evidenziato che si è apportata un'eccezione per il file `Employer.csv`, il quale presenta, nella sua struttura finale, dettagli specifici sulla lista dei posti di lavoro relativi a ciascun luogo.

Va inoltre sottolineato che per i file `Pubs.csv` e `Restaurant.csv`, l'attenzione è stata focalizzata esclusivamente sul campo *maxoccupancy*, che specifica il numero massimo di persone che possono essere ospitate nei rispettivi locali. Diversamente, per il file `Apartments.csv`, si è aggiunto un ulteriore campo, denominato *numberOfRooms*, per fornire informazioni dettagliate sul numero di stanze presenti in ciascun appartamento. Tale approccio mira a garantire una rappresentazione accurata e completa delle caratteristiche specifiche associate a ciascun tipo di struttura.

## Fase di Data Transformation

La struttura JSON finale prevede un array di oggetti, dove ogni oggetto identifica un singolo luogo ed è strutturato così:

- `id` (Number) - Identificatore di un luogo
- `location` (String) - Point che Identifica l'impronta geografica di quel luogo.
- `type` (String) - Può essere "Apartment" o "Pubs" o "School" o "Restaurant" o "Employer"
- `cost` (Object): (solo per Apartment, Pubs, Restaurant, School)
  - `type` (String) - Tipo di costo da sostenere
  - `amount` (Number) - Valore del costo da sostenere
- `jobs` (Array<Object>): (solo per Employer)
  - `id` (Number) - Identificatore del lavoro
  - `hourlyRate` (Number) - Stipendio orario
  - `startTime` (Number) - Orario inizio lavoro
  - `endTime` (Number) - Orario fine lavoro
  - `daysToWork` (Array<String>) - Lista dei giorni lavorativi della settimana
  - `educationRequirement` (String) - Livello di istruzione per quel ruolo lavorativo
- `maxOccupancy` (Number) - Numero massimo di persone ammesse in quel luogo (solo per Apartment, Pubs, Restaurant, School).
- `numberOfRooms` (Number) - Numero delle stanze (solo per Apartment)

### 2.1.3 Creazione di *calendar\_wesfr.json*

## Fase di Acquisizione Dati

Per ottenere una rappresentazione dettagliata degli aspetti connessi alla durata complessiva del periodo di registrazione dei dati in relazione a salari e spese per ciascun partecipante allo studio, si è deciso di utilizzare il file `"FinancialJournal.csv"`. Questo file offre una sintesi concisa dell'evoluzione temporale di ciascun singolo partecipante, riguardo a tutte le transazioni economiche registrate durante l'intero periodo dello studio.

## Fase di Esplorazione Dati e di Data Processing

Al fine di ottimizzare l'analisi dei dati contenuti nel file `FinancialJournal.csv`, il quale presenta registrazioni a intervalli di cinque minuti, si è deciso di ristrutturare il formato attraverso un processo di raggruppamento giornaliero. Questa scelta è motivata da diverse considerazioni. In primo luogo, il raggruppamento giornaliero mira a migliorare l'efficienza computazionale, semplificando le future analisi e ottimizzando le prestazioni computazionali. Inoltre, tale approccio riduce il rumore nei dati, focalizzandosi sugli andamenti di lungo periodo e migliorando la stabilità delle analisi finanziarie. La coerenza temporale viene garantita, consentendo una visione più chiara delle tendenze nel tempo. Infine, il passaggio a dati giornalieri favorisce una più agevole interpretazione degli andamenti, contribuendo a una comunicazione più efficace delle informazioni finanziarie. Il processo di raggruppamento si basa su metodi aggregativi, gestione attenta dei dati mancanti e verifica sistematica della coerenza per garantire risultati robusti e affidabili.

## Fase di Data Transformation

Ciascun oggetto all'interno di questo Array rappresenta un singolo utente coinvolto nello studio, riportando tutti i valori relativi alle diverse spese sostenute nelle categorie di alimentazione, svago, istruzione e alloggio, nonché i dati salariali, tutti dettagliatamente suddivisi su base giornaliera. Ad ogni partecipante è associata un'istanza specifica dei dati strutturata in questo modo:

- `participantId` (Number) - Id di un partecipante
- `wage` (Array<Object>) - Lista di tutti i gli stipendi
  - `timestamp` (String) - Data di un giorno specifico
  - `amount` (Number) - Valore dello stipendio giornaliero
- `food` (Array<Object>) - Poligono che Identifica l'impronta geografica dell'edificio.
  - `timestamp` (String) - Data di un giorno specifico
  - `amount` (Number) - Valore delle spese della categoria "food" giornaliere
- `recreation` (Array<Object>) - Può essere "Commercial" o "Residential" o "School"
  - `timestamp` (String) - Data di un giorno specifico
  - `amount` (Number) - Valore delle spese della categoria "recreation" giornaliere
- `education` (Array<Object>) - Numero massimo di persone ammesse in quel edificio.
  - `timestamp` (String) - Data di un giorno specifico
  - `amount` (Number) - Valore Number delle spese della categoria "education" giornaliere
- `shelter` (Array<Object>) - Lista di `PlaceId` che fanno parte dell'edificio
  - `timestamp` (String) - Data di un giorno specifico
  - `amount` (Number) - Valore delle spese della categoria "shelter" giornaliere

### 2.1.4 Creazione di `calendar_path_expenses.json`

## Fase di Acquisizione Dati

In questa analisi, si sono considerate le informazioni contenute nel file originale `TravelJournal.csv`. Tale documento presenta una compilazione dettagliata delle registrazioni effettuate ogni 5 minuti, esponendo la posizione geografica dell'utente in quei precisi istanti. Ogni registrazione include l'orario di partenza e arrivo, l'ubicazione di partenza e destinazione, le motivazioni dietro lo spostamento, la durata della permanenza in un determinato luogo e le transazioni finanziarie associate, indicando sia le spese che i guadagni.



## Fase di Esplorazione Dati e di Data Processing

Sulla base dell'analisi del file `TravelJournal.csv`, è stato deciso di aggregare i dati per ottenere una visione più esaustiva e gestibile del comportamento degli utenti nel contesto degli spostamenti e delle relative transazioni finanziarie. La scelta di effettuare l'aggregazione a livello utente, anziché mantenere una granularità sulla base delle singole registrazioni, è stata guidata dalla prospettiva di semplificare l'analisi, migliorare la leggibilità e favorire una comprensione più approfondita dei pattern comportamentali.

In questo processo di aggregazione, ogni utente è stato trattato come un'entità unica, consolidando gli spostamenti e le transazioni associate nell'intervallo di cinque minuti, bensì in uno giornaliero. Questo approccio giornaliero consente una rappresentazione più sintetica e significativa delle attività degli utenti, permettendo di individuare con maggiore chiarezza tendenze, abitudini e correlazioni nei loro movimenti e nelle relative spese o guadagni.

## Fase di Data Transformation

In questa fase abbiamo scelto di generare un file JSON che rappresenti le informazioni sotto forma di un array di oggetti. Ogni oggetto identifica le spese e i movimenti di ciascun utente coinvolto nello studio. La struttura di ogni singola istanza dell'array è definita nel seguente modo:

- `participantId` (Number) - Identificatore di un singolo partecipante
- `movements` (Array<Object>) - Lista di tutti i movimenti del singolo partecipante giorno per giorno
  - `timestamp` (String) - Data di un singolo giorno
  - `path` (String) - Sequenza di `placeId` che descrive i luoghi visitati in quel giorno
  - `total_expenses` (Number) - Totale delle spese sostenute in quel giorno
  - `expenses_list` (Array<Number>) - Lista delle spese o dei guadagni sostenuti nei luoghi visitati.

### 2.1.5 Creazione di `calendar.json`

## Fase di Acquisizione Dati

In questa specifica circostanza, la generazione del file `calendar.json` è stata condotta mediante una derivazione dal file `json_calendar_wesfr.json`, precedentemente creato. Tale scelta è stata motivata dalla convenienza di utilizzare le aggregazioni preesistenti per agevolare il processo di creazione del nuovo file `calendar.json`, evitando così l'impiego diretto dei dati originali proposti.

## Fase di Esplorazione Dati e di Data Processing

Il file `"calendar_wesfr.json"` costituisce la rappresentazione informativa dell'andamento giornaliero di salari e spese. Al fine di ottimizzare l'utilità di queste informazioni, è stata proposta la creazione di un nuovo file caratterizzato da una struttura più adatta alla configurazione di un calendario completo. Questo nuovo file avrebbe il compito di sintetizzare le informazioni relative alle medie giornaliere delle entrate, delle uscite e degli indici di risparmio per ciascun giorno, mese e anno, fornendo anche un aggregato globale. Tale struttura mira a semplificare e rendere più efficiente la visualizzazione dei dati, consentendo di ottenere in modo chiaro e immediato una panoramica delle tendenze finanziarie giornaliere, mensili e annuali. Questo approccio fornisce una visione più sintetica e gestibile delle informazioni finanziarie, migliorando l'accessibilità e facilitando l'analisi dei dati.

## Fase di Data Transformation

La struttura del file JSON risulta complessa in quanto identifica un singolo oggetto che formalizza una struttura a "calendario". La chiave principale è denominata `"global"`, la quale rappresenta i valori medi globali per spese, risparmi e salari. All'interno della struttura `"global"`, è presente la chiave `"years"`,

una lista di oggetti dove ciascun elemento rappresenta un anno e contiene aggregazioni relative alle spese, ai risparmi e ai salari di quell'anno. Ogni oggetto all'interno di "years" contiene la chiave "months", che è una lista di oggetti rappresentanti i singoli mesi. All'interno di ciascun mese, troviamo la chiave "weeks", contenente una lista di oggetti che rappresentano le singole settimane. Infine, all'interno di ogni settimana, è presente la chiave "days" che esprime l'ultima aggregazione giorno per giorno delle spese, risparmi e salari.

- `global` (Object) - Identificatori di "wage", "education", "shelter", "food", "recreation", "saving"
  - `years` (Array<Object>) - Identificatori di "wage", "education", "shelter", "food", "recreation", "saving" per ogni anno
    - \* `months` (Array<Object>) - Identificatori di "wage", "education", "shelter", "food", "recreation", "saving" per ogni mese
      - `weeks` (Array<Object>) - Identificatori di "wage", "education", "shelter", "food", "recreation", "saving" per ogni settimana
      - `days` (Array<Object>) - Identificatori di "wage", "education", "shelter", "food", "recreation", "saving" per ogni giorno

### 2.1.6 Creazione di *participants.json*

#### Fase di Acquisizione Dati

Il presente file è il risultato di una fusione tra uno dei file CSV appartenenti al dataset originale denominato `Participant.csv` e il file JSON pregenerato denominato `calendar_wesfr.json`.

#### Fase di Esplorazione Dati e di Data Processing

La decisione di consolidare le informazioni relative ai partecipanti in un unico file è motivata dalla necessità di presentare in maniera integrata tutti i dati associati a ciascun utente coinvolto nello studio. In particolare, le informazioni demografiche sono state estratte in modo esclusivo dal file `Participants.csv`, mentre i dati relativi alle spese e ai guadagni medi per giorno, settimana, mese e anno derivano dalle aggregazioni di tali valori contenuti nel file `calendar_wesfr.json`. Questa unificazione mira a fornire una rappresentazione completa e coesa delle informazioni pertinenti, facilitando così l'analisi e l'interpretazione dei risultati complessivi dello studio condotto.

#### Fase di Data Transformation

La struttura finale rappresenta i dati relativi ad ogni singolo partecipante. Ogni entità infatti esprime le informazioni relative all'età, al livello di istruzione, alla presenza di figli, i luoghi di lavoro, la stabilità economica, gli appartamenti abitati, il reddito giornaliero, mensile, settimanale e annuo, le spese quotidiane, settimanali, mensili e annuali, l'indice di risparmio. Ogni oggetto della lista relativo ad un singolo partecipante ha questa struttura:

- `id` (Number) - Identificatore di un singolo partecipante
- `age` (Number) - Età
- `educationLevel` (String) - Livello di educazione
- `haveKids` (Boolean) - Presenza o meno di figli
- `workplaces` (Array<Number>) - Lista di tutti i `placeId` dei lavori svolti
- `apartments` (Array<Number>) - Lista di tutti i `placeId` degli appartamenti abitati
- `stability` (Object) - Livelli di stabilità economica
  - `stable` (Number) - Percentuale di stabilità economica
  - `unstable` (Number) - Percentuale di instabilità economica

- `average[Day/Week/Month/Year]Wage` (Number) - Stipendio medio giornaliero, settimanale, mensile e annuo
- `average[Day/Week/Month/Year]Food` (Number) - Spese medie giornaliere, settimanali, mensili e annue in cibo
- `average[Day/Week/Month/Year]Education` (Number) - Spese medie giornaliere, settimanali, mensili e annue in educazione
- `average[Day/Week/Month/Year]Shelter` (Number) - Spese medie giornaliere, settimanali, mensili e annue in affitto
- `average[Day/Week/Month/Year]Recreation` (Number) - Spese medie giornaliere, settimanali, mensili e annue in attività ricreative

### 2.1.7 Creazione di `global_financials.json`

#### Fase di Acquisizione Dati

La generazione del file `global_financials.json` è derivata dall'elaborazione del file `calendar_wesfr.json` anziché direttamente dai dati originali. Questa scelta metodologica è fondata su ragioni analoghe a quelle che hanno guidato la creazione del file `calendar.json`.

#### Fase di Esplorazione Dati e di Data Processing

In questa fase di elaborazione, si è operato con l'intento di standardizzare l'analisi, sfruttando le aggregazioni già presenti nel file `calendar_wesfr.json`. L'obiettivo era definire una nuova struttura in grado di rappresentare globalmente i valori medi di guadagno e spesa, nonché i rispettivi trend di crescita o decrescita.

Con l'utilizzo delle aggregazioni giornaliere del file JSON `calendar_wesfr.json`, è stato generato un nuovo documento suddiviso in quattro categorie (giorno, settimana, mese, anno). Ciascuna di queste categorie esprime i valori giornalieri, settimanali, mensili e annuali relativi a "wage," "food," "recreation," "education," e "shelter." Il documento fornisce inoltre i valori medi e analizza i trend di crescita o decrescita dei guadagni e delle spese per ciascun intervallo temporale, riferendosi al totale dei partecipanti al processo.

#### Fase di Data Transformation

- `[day/week/month/year]` (Number) - Giorno, Settimana, Mese o Anno
  - `list` (Array<Object>) - Ogni oggetto è un giorno o settimana o mese o anno e ha i valori di "wage", "food", "recreation", "shelter", "education"
  - `[wage/food/shelter/education/recreation]Average` (Number) - Spese o guadagni medi giornalieri, settimanali, mensili e annuali per le categorie "wage" o "food" o "shelter" o "education" o "recreation"
  - `[wage/food/shelter/education/recreation]Trend` (Number) - Percentuale di crescita o decrescita delle spese o guadagni medi giornalieri, settimanali, mensili e annuali per le categorie "wage" o "food" o "shelter" o "education" o "recreation"

### 3 Struttura del Progetto

La progettazione di questa web app ha coinvolto scelte tecniche mirate per garantire un'esperienza utente ottimizzata e un funzionamento efficiente del sistema. Per la parte client, è stato adottato *Vue.js*, un framework JavaScript progressivo ampiamente utilizzato per la creazione di interfacce utente dinamiche e reattive. La scelta di *Vue.js* è motivata dalla sua struttura modulare, che facilita lo sviluppo di componenti riutilizzabili e la gestione dello stato dell'applicazione in modo coerente.

Per quanto riguarda la parte server (backend), abbiamo optato per *Node.js*<sup>5</sup> e *Express.js*<sup>6</sup>. *Node.js* offre un ambiente di esecuzione JavaScript server-side ad alte prestazioni, consentendo una gestione asincrona efficace delle richieste. *Express.js*, un framework leggero basato su *Node.js*, semplifica la creazione di *API*<sup>7</sup>, fornendo un'organizzazione chiara delle rotte e agevolando la gestione delle richieste HTTP.

I dati sono serviti localmente nel server e memorizzati all'interno della cartella `/data`. Questa scelta è stata effettuata per migliorare le prestazioni dell'applicazione, riducendo i tempi di accesso ai dati rispetto a un'architettura in cui i dati sarebbero stati recuperati da un database remoto. Inoltre, il mantenimento dei dati localmente ha semplificato il processo di sviluppo di questo specifico progetto.

Il server inoltre espone delle *API* che consentono al client di manipolare le informazioni nel dataset. Questa architettura permette una separazione chiara tra il frontend e il backend, consentendo loro di evolvere indipendentemente. L'utilizzo di *API* facilita l'interazione tra le due parti, consentendo al client di richiedere dati in modo efficiente e scalabile. In definitiva, la scelta di *Vue.js* per il frontend e *Node.js* ed *Express.js* per il backend è stata guidata dalla necessità di un'architettura flessibile, reattiva e facilmente gestibile per garantire un'applicazione web performante e manutenibile.

Nelle prossime sezioni vengo chiariti alcuni punti focali sulla progettazione sia lato client che lato server.

#### 3.1 Client: Struttura e Progettazione della Web App

Questo capitolo fornisce una dettagliata panoramica delle decisioni di progettazione a livello del client, concentrando l'attenzione sulle scelte architetturali, l'organizzazione dei componenti *Vue.js* e le strategie adottate per ottimizzare l'esperienza utente.

La decisione finale, risultato di un'approfondita valutazione dei dati disponibili e delle sfide progettuali, ha orientato la scelta verso la creazione di una **"dashboard a tre livelli"**. Questo approccio consente una chiara suddivisione delle informazioni, agevolando la navigazione del cliente attraverso ampie quantità di dati e facilitandone la comprensione. La dashboard offre una visione istantanea e intuitiva delle informazioni cruciali mediante una rappresentazione grafica dei dati, consentendo al cliente di identificare rapidamente tendenze, anomalie o relazioni chiave.

Il progetto di sviluppo della dashboard prevede una suddivisione chiara e organizzata dei componenti all'interno del client. Le tre view principali sono collocate nella directory `/views`, contenente tutti i file necessari per la visualizzazione delle pagine. La struttura aggiuntiva del client contribuisce alla modularità del sistema: la cartella `/components` è suddivisa in `/atoms`, dedicata ai componenti più piccoli e atomici, e `/widgets`, destinata ai componenti grafici più complessi del dashboard. Questa organizzazione agevola la gestione e la manutenzione dei diversi elementi del progetto.

---

<sup>5</sup>Node.js è un runtime system open source multiplatforma orientato agli eventi per l'esecuzione di codice JavaScript, costruito sul motore JavaScript V8 di Google Chrome

<sup>6</sup>Express.js è un framework per applicazioni web per Node.js, open source sotto Licenza MIT.

<sup>7</sup>API - Application Programming Interface (interfaccia di programmazione delle applicazioni), indica un insieme di definizioni e protocolli per la creazione e l'integrazione di applicazioni software.

Inoltre, la presenza della cartella `/api` nel client gestisce tutte le richieste di dati alle API disponibili. Questa struttura consente una gestione centralizzata delle chiamate alle API, garantendo un'implementazione efficace e facilitando l'aggiornamento delle informazioni visualizzate nella dashboard.

### 3.1.1 Struttura della Dashboard

#### View "Data Overview"

La sezione iniziale della dashboard, denominata **"Data Overview"**, è il punto di ingresso principale per gli utenti, fornendo una visione panoramica dei dati. In questa scheda sono riportati i dati generali come:

1. **Informazioni Demografiche di Base:** dove si esplicitano dati come: il numero totale dei partecipanti, l'età media, il livello di istruzione medio e la stabilità media globale
2. **Indicatori Chiave di Stabilità Economica:** utili per analizzare la distribuzione degli stipendi in relazione all'età, offrendo anche una valutazione della stabilità economica associata a ciascun gruppo di età. Questa correlazione offre un quadro completo della situazione economica in base alle diverse fasce d'età.
3. **Distribuzione della Ricchezza:** che esplora la distribuzione della ricchezza tra i partecipanti.

#### View "Participants Economic Profile"

La seconda scheda della dashboard, denominata **"Participants Economic Profile"**, si concentra in modo più dettagliato su ciascun partecipante del dataset. Questa sezione mira a presentare come le variabili relative a guadagni e spese variano nel tempo per ciascun individuo. Sono evidenziati i luoghi in cui i partecipanti spendono di più o di meno, oltre a evidenziare i percorsi più frequenti nel corso del tempo.

In questa sezione si propone quindi un'analisi dettagliata degli andamenti finanziari di ciascun partecipante, evidenziando con precisione le fluttuazioni nei guadagni e nelle spese. Le principali caratteristiche analizzate sono:

1. **Fluttuazioni di Guadagno e Spesa:** Vengono esplorate e delineate le variazioni significative nei livelli di guadagno e spesa di ogni partecipante, offrendo una chiara comprensione delle dinamiche finanziarie individuali.
2. **Visualizzazione di Tendenze Temporal:** La rappresentazione grafica delle tendenze temporali consente agli utenti di identificare facilmente i modelli e le variazioni nel tempo, facilitando un'analisi accurata dei cambiamenti finanziari nel corso di periodi specifici.
3. **Analisi Approfondita dei Comportamenti Finanziari:** La sezione fornisce un'analisi dettagliata dei comportamenti finanziari individuali, mettendo in luce le abitudini di spesa e le preferenze nel corso del tempo.
4. **Informazioni Dettagliate:** Vengono fornite informazioni dettagliate sulle preferenze di spesa dei partecipanti, offrendo una visione chiara e completa dei loro orientamenti finanziari.
5. **Monitoraggio degli Spostamenti Abituati:** La sezione include dettagli sugli spostamenti abituali nel tempo, permettendo agli utenti di comprendere come le abitudini di spesa e gli spostamenti possano influenzare i loro andamenti finanziari.

#### View "Businesses Economic Profile"

La terza sezione, denominata **"Businesses Economic Profile"**, approfondisce le dinamiche finanziarie delle attività commerciali, con un focus mirato su Pubs e Restaurants, e fornisce un'analisi dettagliata

sui datori di lavoro.

Per quanto concerne le *attività commerciali* si va ad effettuare in modo specifico un'analisi che si concentra su:

1. **Monitorare le performance economiche nel tempo**
2. **Identificare, dove presenti le fluttuazioni nei guadagni e nell'affluenza**

Mentre per le analisi rivolte ai *datori di lavoro* la dashboard predispone una visualizzazione dei dati che mira a:

1. **Esplorare le variazioni negli stipendi offerti**
2. **Analizzare i cambiamenti nei turnover aziendali**

### 3.1.2 Utilizzo delle API

L'utilizzo delle API lato client (Fig. 1) è stato progettato seguendo un approccio modulare e strutturato, con un'attenzione particolare all'efficienza e alla facilità di manutenzione. Ogni modulo corrisponde a un file specifico che gestisce le chiamate API relative a una categoria di dati specifica. Le scelte logiche di progettazione riflettono la coerenza nell'utilizzo delle API strutturate e seguono un modello uniforme per garantire coesione e facilitare la comprensione del codice.

Per ogni modulo, ad esempio *Building.js*, *Businesses.js*, *Demography.js*, *Economy.js*, e *Participants.js*, è stato mantenuto uno schema coerente. Per performare le chiamate API è stata utilizzata la libreria *axios*<sup>8</sup>.

Inoltre è stato pensato di creare un file *config.js* che permettesse di generalizzare, tramite l'export di specifiche costanti, il recupero della URL per API così da incrementare la flessibilità del sistema, consentendo una rapida modifica dell'URL in un unico punto senza dover modificare ogni singolo modulo. Inoltre, l'uso di costanti come *apiName* per identificare il tipo di dati semplifica la manutenzione e la scalabilità, permettendo l'aggiunta di nuovi moduli senza dover modificare il codice esistente.

Le funzioni definite all'interno di ciascun modulo seguono uno schema coerente di denominazione, con nomi come *fetchMap*, *fetchPlace*, *fetchPeriods*, *fetchHistory*, *fetchParticipants*, ecc. Questa uniformità rende il codice più leggibile e agevola la comprensione delle operazioni che vengono eseguite per ogni tipo di chiamata API.

Inoltre, le funzioni sono progettate per gestire parametri in modo opzionale, consentendo una maggiore flessibilità nell'utilizzo delle API senza dover scrivere molteplici versioni della stessa funzione.

```
1 // DemographyData.js
2 import axios from "axios";
3 import { API_URL } from "../config/config.js";
4
5 const apiName = 'buildings';
6
7 export const fetchMap = async () => {
8   const response = await axios.get(`${API_URL}/${apiName}/map`);
9   return response.data;
10 }
11
12 export const fetchPlace = async (id) => {
13   const response = await axios.get(`${API_URL}/${apiName}/place/${id}`);
14   return response.data;
15 }
```

Figura 1: Esempio del file *Demography.js* per il fetching dei dati tramite API

---

<sup>8</sup>Axios è una libreria JavaScript che viene utilizzata per effettuare richieste HTTP sia negli ambienti browser che Node.js. Axios è comunemente usata nello sviluppo web per interagire con le API web

## 3.2 Server: Progettazione Strutturale e API

Nel progettare il server per l'esposizione delle API (Fig. 2) del nostro progetto, abbiamo fatto delle scelte logiche mirate a garantire una struttura modulare. Abbiamo utilizzato il framework *Express.js* per la gestione delle richieste HTTP, poiché offre una sintassi chiara e una vasta gamma di funzionalità per lo sviluppo di applicazioni web. La modularità si basa invece sulla creazione di diverse funzionalità in moduli separati all'interno della cartella `"/api"`. Inoltre, abbiamo adottato il *middleware CORS*<sup>9</sup> per abilitare la comunicazione tra il server e domini esterni, consentendo così una più ampia accessibilità alle nostre API.

Inoltre è stato utilizzato il file di configurazione *dotenv* che ci consente di gestire facilmente le variabili d'ambiente, migliorando la portabilità dell'applicazione. La definizione delle API è stata strutturata in modo chiaro e coerente, assegnando a ciascuna entità del progetto un percorso distintivo sotto l'URL principale `"http://localhost:<PORT>/api"`<sup>10</sup>. Questi percorsi sono:

### 1. /buildings

Espone delle API relative alla gestione dei dati degli edifici della città come:

- (a) /map: restituisce i dati per la costruzione della mappa
- (b) /place/:id : restituisce i dati relativi ad un singolo luogo
- (c) /locations/:type/:id?: restituisce i dati relativi a una o tutte le attività di un certo tipo (employer, apartment, school, restaurant, pub, all)

### 2. /businesses

Espone delle API relative alla gestione dei dati delle attività commerciali e dei datori di lavoro.

- (a) /period/:placeType/:monthOnly?/:placeId?: restituisce i dati relativi a tutto o una parte del periodo di attività di una o tutte le attività commerciali (pub o restaurant) o i datori di lavoro (employer)
- (b) /history/:placeId/:month/:participantId?: restituisce l'andamento storico dei guadagni o dei salari emessi in un dato periodo da un attività o un datore di lavoro di tutti i partecipanti o di un partecipante specifico.

### 3. /demography

Espone delle API relative alla gestione degli aspetti demografici dei partecipanti.

- (a) /participants: restituisce il numero totale dei partecipanti
- (b) /participant/:id: restituisce tutte le informazioni di un dato partecipante
- (c) /average/:type: restituisce i valori medi globali di "age" o "education"
- (d) /kids: restituisce i dati percentuali globali dei partecipanti che hanno e non hanno figli

### 4. /economy

Espone delle API relativi a vari aspetti economici globali e particolari della città e dei singoli partecipanti.

---

<sup>9</sup>Il middleware CORS (Cross Origin Resource Sharing) è un componente delle applicazioni web che gestisce le richieste da domini diversi, consentendo il superamento della Same-Origin Policy del browser attraverso l'aggiunta di header CORS alle risposte HTTP. Essenziale per consentire la comunicazione cross-origin in modo controllato e sicuro.

<sup>10</sup>La porta <PORT> della URL è modificabile in fase di progettazione e sviluppo tramite il file *dotenv*.

- (a) /global/:type: restituisce i valori globali relativi alle categorie di indice di risparmio (saving) e di stabilità/instabilità economica (stable, unstable).
- (b) /trends/:filter/:type?: restituisce il valore del trend positivo o negativo, calcolato giornalmente, settimanalmente, mensilmente o annualmente, in base agli stipendi o alle spese dei partecipanti.
- (c) /lorenz/:filter: restituisce i punti della curva di Lorenz per un periodo specifico (giorno, settimana, mese o anno) in base ai salari dei partecipanti
- (d) /distribution/:filter/:age?: restituisce i dati statistici della distribuzione del reddito e della stabilità economica, fornendo informazioni sia per singoli valori di età che per intervalli di età.

## 5. /participants

Espone delle API relative alla gestione dei dati sulle evoluzioni temporali dei partecipanti durante il periodo coperto dal dataset.

- (a) /calendar/info/:type/:date?/:participantId?: restituisce i valori relativi a un calendario globale o a una data specifica, quale giorno, mese o anno. Se viene specificato anche un participantId, restituisce il range temporale associato a quel participantId solamente
- (b) /calendar/:type: restituisce i valori selezionati dal calendario, che possono includere salario, alimenti, tempo libero, istruzione, alloggio e risparmio
- (c) /path/:type/:date/:participantId?: restituisce il percorso più frequente in un intervallo temporale globale o specifico di un partecipante o di tutti i partecipanti.
- (d) /distribution/:type/:date/:sort: restituisce i valori salariali, di spesa, o l'indice di risparmio su base giornaliera, mensile, annuale o complessiva, ordinati per ID del partecipante o in ordine decrescente di valore.

```

41  /*****
42  *****/
43  * @api {get} /buildings/places/:type/:id? Get Places
44  * @apiName GetPlaces
45  *
46  * @param type can be {employer, commercial, school, residential}
47  * @param id = number
48  *****/
49  /*****
50  router.get("/place/:id", async (req, res) => {
51    const { id } = req.params;
52    try {
53      const data = await places;
54      let response = data.find(item => item.id.toString() === id);
55      res.json(response);
56    } catch (err) {
57      console.error("Error:", err);
58      res.status(500).json({ error: "Server Error" });
59    }
60  });

```

Figura 2: Esempio di un API lato server

La fase di inizializzazione del server include anche la funzione di caricamento dati tramite il modulo "loader.js", garantendo che il server venga avviato solo dopo il completo caricamento delle informazioni necessarie.



## 4 Design Process

La fase di progettazione dell'interfaccia si è configurata come un passaggio successivo alla progettazione strutturale del client e del server. Questa parte del processo si articola in due principali aree:

1. la selezione delle scale di colori
2. lo sviluppo e l'interazione con i componenti grafici dedicati all'interfacciamento con i dati

Nella prima parte la scelta delle scale di colori riveste un ruolo fondamentale nel determinare l'estetica visiva del progetto, influenzando l'esperienza utente complessiva. La palette cromatica selezionata non solo contribuisce a creare un'identità visiva distintiva, ma può anche comunicare efficacemente informazioni e gerarchie. Nella seconda componente, lo studio, la realizzazione e l'interazione con i componenti grafici sono centrali per la creazione di un'interfaccia utente efficace, ponendo l'attenzione sull'organizzazione visiva degli elementi e la presentazione chiara dei dati.

Per semplificare e ottimizzare la gestione della progettazione grafica lato client, è stato adottato l'utilizzo di **Tailwind CSS**. Questo framework CSS utility-first offre un approccio pragmatico e altamente personalizzabile per la creazione di interfacce utente. La sua metodologia basata su classi predefinite permette una progettazione più modulare, facilitando la costruzione e la gestione dei componenti grafici.

### 4.1 Scelta dei Colori

La selezione dei colori per l'intera dashboard è stata guidata da un principio fondamentale di ricerca di semplicità e chiarezza visiva. A tale scopo, sono stati adottati colori neutri per gli sfondi, utilizzando tonalità come il bianco (`#FFFFFF`) e il grigio chiaro (`#F6F6F6`) (Fig. 3). Questa scelta mira a garantire una leggibilità ottimale e a evitare un impatto visivo eccessivamente pesante.



Figura 3: Scala di grigi utilizzata

Inoltre, è stato condotto uno studio approfondito sulla distinzione visiva tra elementi interattivi, quali bottoni, filtri e switch, e i colori associati ai dati. A tal fine, è stato scelto un colore blu (`#3B82FF`) (Fig. 4) per gli elementi interattivi che apportano modifiche ai dati visualizzati sulla dashboard. Per quanto riguarda i dati quantitativi continui che seguono un andamento temporale, sono state implementate due scale di colori differenziate: una scala rossa (Fig. 5) per evidenziare un andamento negativo e una scala verde (Fig. 6) per indicare un andamento positivo. Questa strategia di colore mira a enfatizzare visivamente le variazioni significative nei dati, contribuendo così a una comprensione più rapida ed efficace delle informazioni presentate. Mentre per i dati nominali e quelli puntuali sono stati utilizzati sempre dei colori neutri che fanno parte della scala di grigi descritta in figura 3.

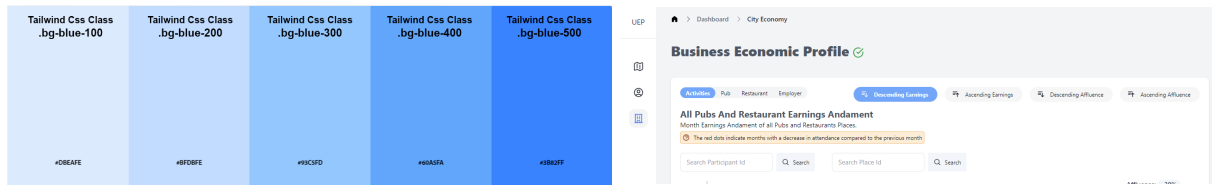


Figura 4: Scala di blue ed elementi attivi della Dashboard



Figura 5: Scala di rossi ed visualizzazione del range

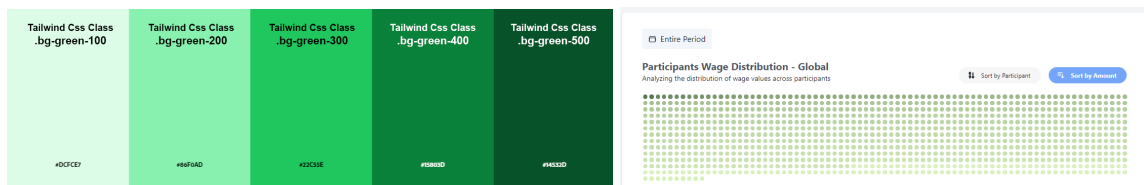


Figura 6: Scala di verdi ed visualizzazione del range

In conclusione, si è optato per associare un colore (Fig. 7) distintivo a ciascuna entità nominale che rappresenta un oggetto relativo alle categorie destinate agli edifici che sono:

1. *"Commercial"*: che identifica tutti gli edifici dove risiedono attività commerciali o datori di lavoro
2. *"School"*: che identifica gli edifici che contengono scuole
3. *"Residential"*: che identifica gli edifici che contengono appartamenti

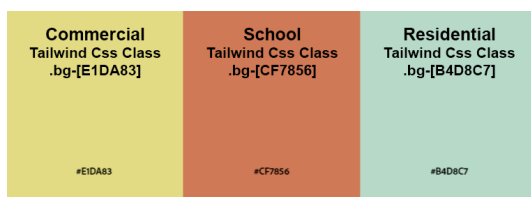


Figura 7: Colori scelti per le variabili nominali School, Commercial, Residential

## 5 Usare e Interagire con le Visualizzazioni

Nel contesto delle analisi dei dati, le visualizzazioni sono progettate seguendo un approccio metodologico dettagliato. All'interno della sezione **"Data Overview"** (Capitolo 3.1.1), vengono presentate in modo esauriente le informazioni chiave del dataset a livello aggregato, fornendo una panoramica chiara ed esaustiva degli aspetti economici e demografici dei dati in esame. Questa visualizzazione offre la possibilità di applicare diversi filtri per esplorare in modo approfondito i dati, consentendo l'aggregazione su base giornaliera, settimanale, mensile o annuale. I filtri, come illustrato nella figura 8, servono per modificare i valori medi globali dei salari e delle spese, mostrati all'interno dei due widget **"Wage Average and Trend"** e **"Expenses Average and Trends"**. Contestualmente, vengono forniti anche i trend temporali associati, consentendo così una valutazione dinamica della crescita o decrescita nel periodo di aggregazione selezionato delle spese e dei salari. Inoltre, i filtri temporali influenzano direttamente sia la forma della curva di Lorenz, utile a descrivere la distribuzione della ricchezza tra i partecipanti, sia il grafico a barre che analizza la relazione tra stipendi distribuiti nelle diverse fasce di età e la stabilità economica, offrendo così un'ulteriore prospettiva dei dati.

In questa scheda della dashboard quindi sono stati inseriti:

- **4 widget** - che descrivono dati principalmente di tipo demografico
- **2 widget con tendenze** - che descrivono le medie aggregate di spese e salari con i loro relativi trend
- **1 curva di Lorenz** - che descrive la distribuzione della ricchezza
- **1 grafico a colonne** - che descrive i salari e la stabilità delle varie fasce di età.

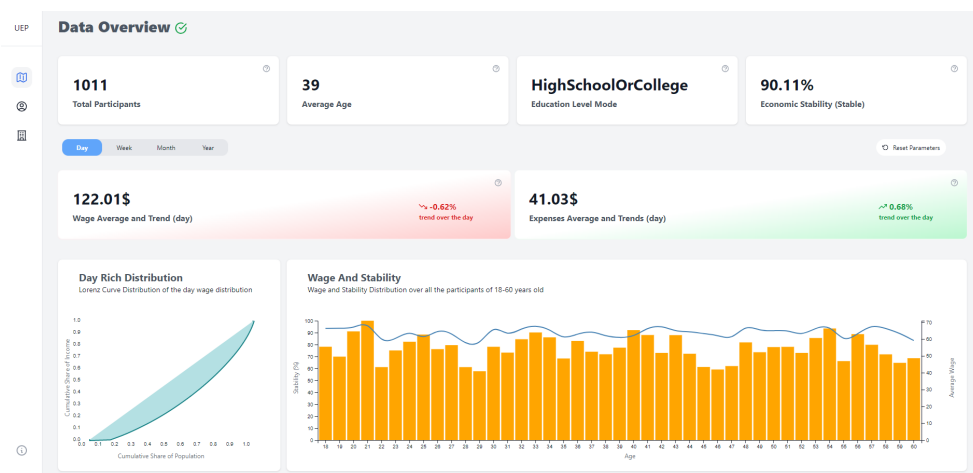


Figura 8: Scheda "Data Overview"

Nel contesto delle altre due schede della dashboard, si è adottato un approccio completo che abbraccia una panoramica dalla macro-esplorazione dei dati fino alla micro-esplorazione.

La vista denominata **"Participants Economics Profile"** (Capitolo 3.1.1) evidenzia il suo orientamento principale verso i partecipanti, consentendo così un'analisi approfondita degli aspetti economici sia a livello globale che individuale. (Fig. 9)

L'impiego di filtri mirati offre la possibilità di analizzare l'andamento dei valori salariali, delle spese alimentari, degli svaghi, degli affitti, delle spese educative e degli indici di risparmio sia su scala globale che a livello individuale.

Inoltre questa parte propone i dati in due visualizzazioni distinte: una a livello di "macro-esplorazione",

presentata come una sorta di "calendario", che mostra le aggregazioni medie per aspetti giornalieri, mensili, annuali o del periodo completo, e una a livello di "micro-esplorazione", mettendo in luce i dati di ogni singolo partecipante relativi al filtro selezionato.

Le due visualizzazioni, "*Calendar Distribution*" e "*Participant Distribution*", sono strettamente collegate. In particolare, la selezione di un giorno, mese o anno specifico nella vista "calendario" influisce sui dati visualizzati nella vista "partecipanti", mostrando solo le informazioni rilevanti alla data specifica selezionata. Questa connessione tra le due viste permette di esplorare in modo dinamico e dettagliato l'andamento dei dati economici in relazione a periodi specifici.

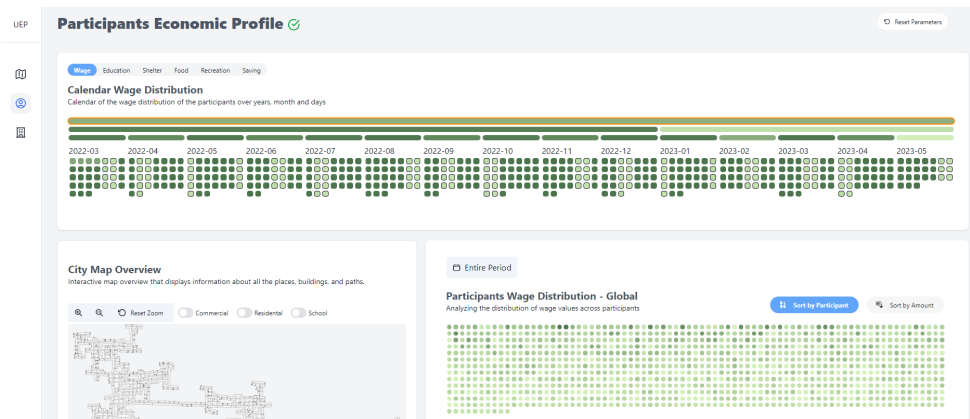


Figura 9: Vista "Calendario" e "Partecipanti" della scheda "Participants Economics Profile"

All'interno di questa prospettiva della dashboard, si apre anche la possibilità di esplorare in dettaglio i percorsi intrapresi (Fig. 10). Tale esplorazione si basa sul periodo di dati preso in considerazione, che può essere specifico per uno o per tutti gli utenti all'interno del range temporale selezionato. Attraverso questa analisi, è possibile identificare sia informazioni relative al singolo utente selezionato, sia i percorsi che gli utenti, singolarmente o collettivamente, hanno percorso con maggiore frequenza, cercando così di individuare schemi comportamentali ricorrenti. Inoltre, la dashboard offre la capacità di analizzare quale sia il percorso che ha generato la spesa più elevata. Un elemento distintivo di questa visualizzazione è la possibilità di mappare geograficamente i percorsi identificati. Integrando una mappa interattiva, gli utenti possono comprendere visivamente la distribuzione spaziale di tali percorsi, visualizzando inoltre i luoghi esatti in cui i partecipanti hanno fatto tappa.

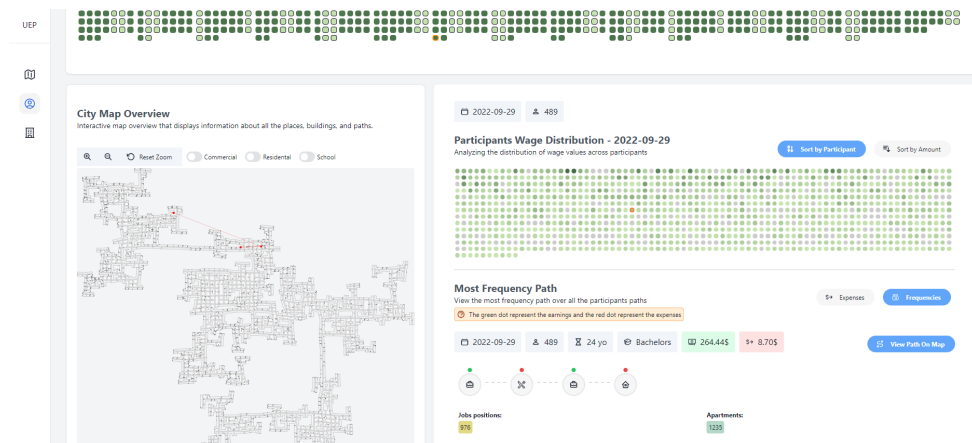


Figura 10: Vista "Most Frequency Path" e dati collegati.

A chiudere le informazioni presenti in questa scheda troviamo anche: (Fig. 11):

1. **Evoluzione dello Stipendio:** un grafico a linea che mostra l'andamento mese per mese degli stipendi di tutti o di uno solo dei partecipanti.
2. **Dieci Luoghi più Visitati o di Maggior Spesa:** visualizzazione che mostra i dieci luoghi con i maggiori valori di frequenza o di spesa di uno o tutti partecipanti in un dato periodo.
3. **Relazione tra Stipendio e Spese:** un grafico che mostra con delle colonne il totale delle spese e una linea che determina lo stipendio. Questi dati variano se si seleziona un periodo specifico o/e un partecipante.

Questa strategia di esplorazione dati mira a svelare insight significativi, fornendo al fruitore un quadro completo delle dinamiche finanziarie dei partecipanti. L'uso mirato di grafici e filtri specifici consente di evidenziare in modo efficace le relazioni chiave e di identificare aree di interesse per ulteriori analisi.

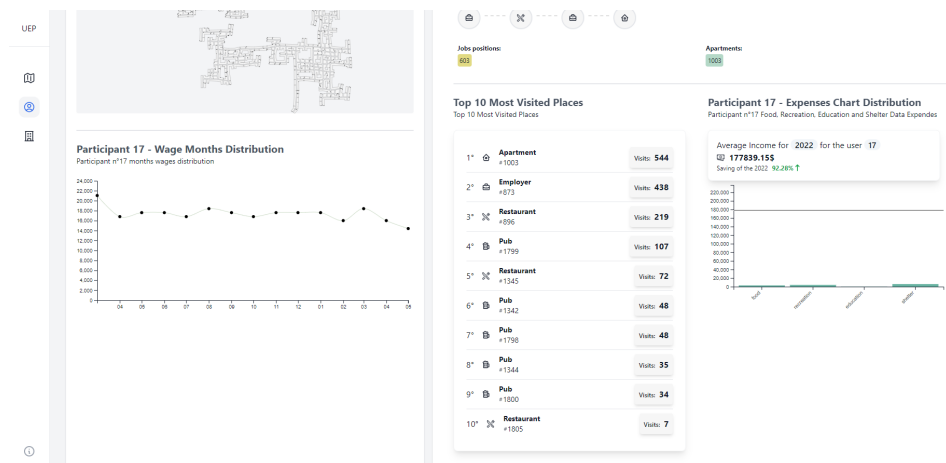


Figura 11: Vista delle informazioni aggiuntive della scheda "Participants Economic Profile".

La sezione finale della dashboard, denominata "**Businesses Economic Profile**" (Capitolo 3.1.1), è concepita come una focalizzazione incentrata sull'analisi avanzata dei dati relativi alle attività commerciali e ai datori di lavoro. In questa visualizzazione, viene offerta la possibilità di applicare filtri ai dati, includendo o escludendo esclusivamente le attività commerciali, compresi pub e ristoranti, oppure concentrarsi esclusivamente su pub, ristoranti o datori di lavoro.

Per quanto concerne l'analisi macroscopica delle attività commerciali, ci si concentra sul posizionamento e sull'evoluzione mensile nel periodo di riferimento. Questa analisi si basa sui dati degli incassi mensili e totali, fornendo una prospettiva dettagliata delle variazioni positive o negative nell'affluenza a specifici pub o ristoranti. Nel caso dei datori di lavoro, l'attenzione è rivolta all'importo complessivo erogato in stipendi mensili, con una particolare analisi dei mesi in cui si osserva un significativo turnover tra i dipendenti (Fig. 12).

In sintesi, questa sezione mira a fornire un approfondimento dettagliato sull'andamento economico delle attività commerciali, evidenziando le dinamiche finanziarie mensili e le tendenze chiave nei salari e nella stabilità occupazionale per i datori di lavoro inclusi nell'analisi.

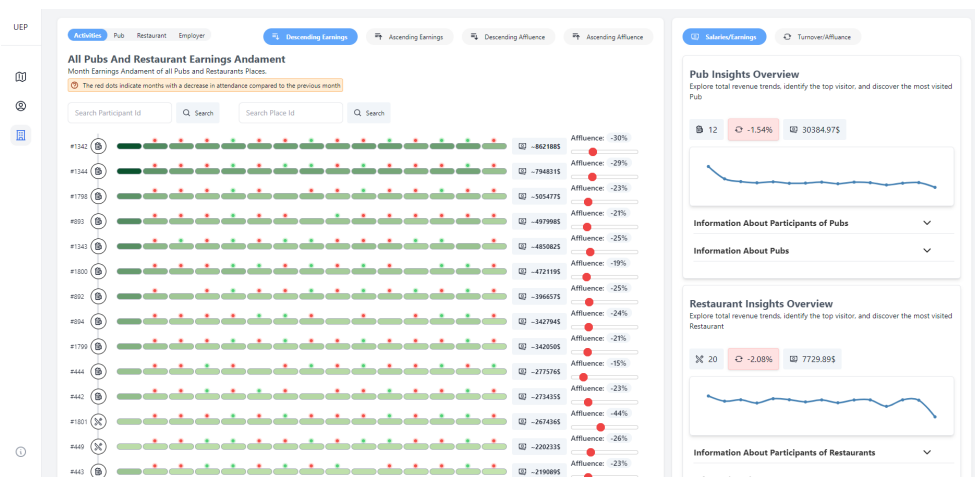


Figura 12: Vista dettagliata dell'evoluzione dei Pubs e dei Restaurants.

In relazione all'analisi più approfondita delle singole attività o posizioni lavorative, è possibile effettuare interazioni avanzate con l'elenco dei luoghi, visualizzando informazioni dettagliate sul luogo selezionato. Ciò include la capacità di esplorare le variazioni di affluenza nel corso del tempo, esponendo i dati relativi ai partecipanti che hanno frequentato quel luogo durante il periodo specificato (Fig. 13).

In particolare, è possibile distinguere tra coloro che hanno fatto ritorno in quel luogo, coloro che non hanno fatto ritorno e coloro che sono nuovi nell'ambito di quel contesto spazio-temporale.

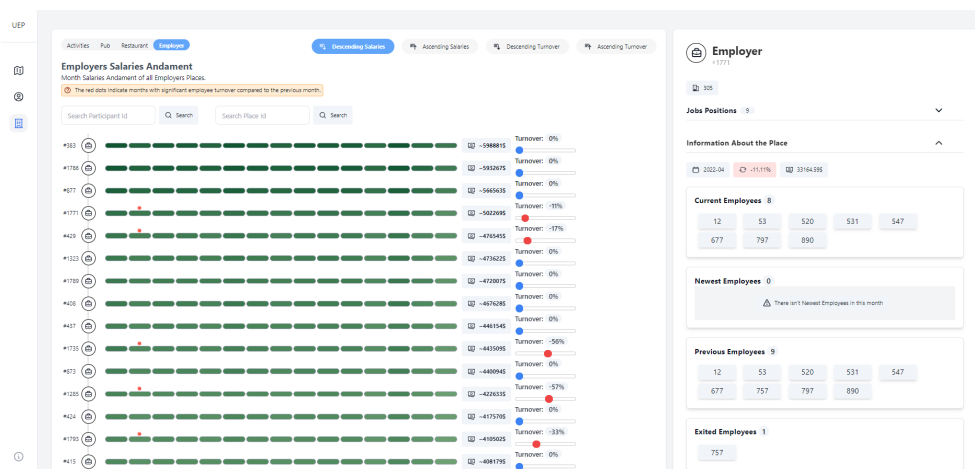


Figura 13: Vista dettagliata sulla sidebar di destra di un singolo Employer nel mese 2022-04