

SHOAP

CONSEGNA:

È richiesta la creazione di un software utilizzando uno dei pattern di programmazione.

Dopo un confronto col gruppo abbiamo deciso di creare un software gestionale per la gestione e l'amministrazione di un punto vendita. Il programma risultante è utilizzabile per gestire il magazzino e le vendite di suddetto punto vendita. Il negozio, denominato SHOAP (SHOP e SOAP) si occupa della vendita di saponi artigianali.

LA SCELTA DEL PATTERN

Cos'è un pattern?

Un **design pattern** può essere visto come un mattone utilizzato per costruire una casa, nel senso che i pattern in programmazione software servono e sono utilizzati come soluzione a quei **problemi strutturali** e di scrittura del codice che accomunano vari programmi. Il concetto di base può, volendo, essere riassunto con il motto "Don't reinvent the wheel" letteralmente "Non reinventare la ruota", cioè non riscrivere codici che risolvono problemi già risolti, ma di utilizzare **soluzioni già implementate**.

Questa tecnica aumenta anche la qualità del progetto, perché lo rende più facilmente comprensibile anche ad altri sviluppatori, dato che l'utilizzo dei pattern porta all'utilizzo di un linguaggio e di una **struttura di pensiero e ragionamento comune**. Inoltre, l'utilizzo dei pattern rende più semplice la manutenzione del codice.

Esistono diversi tipi di pattern, che sono stati classificati secondo vari criteri.

Un primo criterio di distinzione è **il loro scopo**, esistono, infatti, pattern:

- Creazionali
- Strutturali
- Comportamentali

I **pattern creazionali** si occupano della **creazione di oggetti**, ad esempio il singleton pattern che si occupa di creare una sola istanza all'interno di una classe.

I **pattern strutturali** vengono, invece, utilizzati per definire la **struttura del codice**, e sono pensati sui concetti base della programmazione Object Oriented, come l'ereditarietà ed il polimorfismo. Un esempio di pattern di questo tipo è l'adapter.

I **pattern comportamentali**, infine, si occupano delle **interazioni degli oggetti**, il strategy pattern è un ottimo esempio di pattern di tipo comportamentale.

Un secondo criterio di categorizzazione è **l'area di competenza del pattern**, cioè su cosa agisce.

Esistono, infatti, pattern che lavorano con le **classi** ed altri che lavorano con gli **oggetti**.

La principale differenza sta nel fatto che i pattern che **lavorano con le classi** e le loro relazioni solitamente lavorano col concetto di **ereditarietà**, e sono quindi **statici**, mentre quelli definiscono le **relazioni tra oggetti** sono basati sull'idea del **polimorfismo**, e sono, perciò più **dinamici**.

Tutti i pattern quando vengono pensati e creati vengono accomunati da alcune caratteristiche, le principali sono:

Un nome e una classificazione, cioè un nome che sia chiaramente identificativo del concetto di base del pattern e una sua classificazione in base ai criteri sopracitati;

Una motivazione, vale a dire uno scenario astratto a cui si può applicare il pattern;

L'applicabilità, ovvero in quali situazioni il pattern può essere applicato;

La struttura, ossia una descrizione grafica degli elementi che cooperano per la risoluzione del problema;

I partecipanti, cioè tutte quelle classi ed oggetti che fanno parte del pattern;

e tutte quelle informazioni per la sua implementazione a livello di scrittura del codice.

Per il nostro progetto abbiamo scelto di utilizzare il **Singleton Pattern**, a cui ora viene dedicato un approfondimento.

Il Singleton Pattern

Per lo sviluppo e come stile per la scrittura del codice del software viene utilizzato il singleton pattern.

Questo pattern utilizza come filosofia principale **l'unicità di oggetti per classe**, fa in modo, cioè, che per ogni classe possa essere istanziato un solo oggetto, il quale viene creato, per permettere ciò, all'interno della classe stessa e viene invocato come istanza statica quando necessario.

Per impedire che vengano istanziati più oggetti il costruttore della classe dichiara il modello "private", ed istanzia lui stesso l'oggetto al suo interno, è questo ad assicurare l'impossibilità la creazione di oggetti di una stessa classe.

Utilizzando il singleton la classe creata è accessibile globalmente dal programma.

Visto come viene generata l'istanza della classe viene reso **impossibile modificarla dall'esterno**, questo ne assicura il pieno controllo.

Un problema del Singleton Pattern è l'eventuale **modifica e la manutenzione del programma**, perché modificando il Singleton non sappiamo precisamente quali parti dell'applicativo sono interessate, anche la stessa eliminazione della classe singleton potrebbe creare dei buchi nel codice del programma.

Un altro svantaggio derivante dall'utilizzo di questo pattern è sul piano prestazionale, dato che **non permette un flusso regolare dei dati**.

Spesso questo pattern viene utilizzato quando si rende necessario risolvere **processi ripetitivi**, ad esempio per la scrittura dei log in file di testo.

All'interno del nostro progetto il Singleton Pattern può essere visto utile **nell'istanza del carrello**. Infatti, ogni utente può avere **uno e un solo carrello**, questo viene reso possibile dal pattern che **limita ed impedisce la creazione di nuove istanze**. Un altro esempio di utilizzo del Singleton Pattern tra le righe di codice del progetto risiede nella **connessione al database**, per lo stesso motivo di prima, e cioè che ne **può esistere solo una**.

Di seguito viene riportato un esempio di classe scritta utilizzando il singleton pattern, dove il singleton è la connessione al database. Il metodo getInstance salva la prima istanza e la richiama ogni qual volta si fa necessario.

```
// Il database attraverso il metodo getInstance permette all'utente  
l'accesso alla stessa istanza della connessione al database.
```

```
class Database is
```

```
// Viene definito static per permettere il salvataggio  
private static field instance: Database
```

```
// Il costruttore è definito private per impedire la creazione di nuove  
istanze dall'esterno
```

```
private constructor Database() is  
//
```

```
// Questo metodo serve per accedere all'istanza creata
```

```
public static method getInstance() is  
if (Database.instance == null) then  
    acquireThreadLock() and then  
        //Il metodo si assicura che l'istanza non sia già stata creata e  
        in caso la crea  
        if (Database.instance == null) then  
            Database.instance = new Database()  
return Database.instance
```

```
public method query(sql) is  
// Tutte le query passano attraverso questo metodo
```

```
class Application is
```

```
method main() is  
    Database foo = Database.getInstance()  
    foo.query("SELECT ...")  
    // ...  
    Database bar = Database.getInstance()  
    bar.query("SELECT ...")  
    // La variabile "bar" avrà lo stesso contenuto di quella "foo"
```

ANALISI REQUISITI

Per soddisfare le richieste si sviluppi un software per la **gestione di un negozio di saponi**.

Il software viene concettualmente diviso in 2 parti principali, il sito **e-shop** visibile a tutti gli utenti e potenziali clienti e **la parte amministrativa** accessibile attraverso login dall'admin/owner dove viene reso possibile aggiungere, modificare, eliminare i prodotti presenti e gestire le loro quantità in giacenza.

La web-app all'avvio si apre sulla parte visibile a **tutti gli utenti** registrati e non, qui viene visualizzata in alto una navbar che permette una comoda navigazione tra le pagine del sito e l'homepage, dove sono presenti i prodotti principali disponibili all'acquisto, e sul lato sinistro una breve panoramica delle recensioni generiche del sito e dei prodotti al loro interno.

Sulla navbar nella parte destra è presente un pulsante che permetta il **login**, o la **registrazione**, dell'utente.

Se si effettua il login sarà necessario inserire le credenziali d'accesso, cioè username e password, se, invece, si effettua la registrazione si rende necessario **inserire tutti i dati richiesti** per la creazione dell'utente. Per la gestione dell'utente, della sua registrazione e dei suoi futuri login vengono utilizzate le funzioni già presenti in wordpress.

Selezionando uno dei prodotti dal sito si apre la pagina del **dettaglio dei prodotti**, nella quale è presente una descrizione scritta del prodotto, le recensioni relative al prodotto, e la possibilità di **aggiungerle al carrello**.

È poi possibile visualizzare il carrello con tutti i prodotti presenti al suo interno, le relative quantità, i prezzi subtotali e totali, la possibilità di inserire o modificare l'indirizzo di spedizione e di **completare la prenotazione dell'ordine**.

Il pagamento dell'ordine, per scelta progettuale, non viene gestito e viene affidato ad un servizio esterno.

Nella **parte amministrativa** invece viene reso possibile all'utente con privilegi di accesso admin di **gestire i prodotti** presenti all'interno del negozio online, e quindi di creare nuovi prodotti, di modificare quelli già presenti o anche di eliminarli. Per effettuare l'operazione di creazione l'admin dovrà inserire le **informazioni richieste dal sistema**, e cioè un nome, una descrizione, un prezzo, la categoria e la quantità disponibile alla vendita. L'admin potrà poi attivare e rendere visibili i prodotti a proprio piacimento. La modifica di questi dati viene resa possibile anche in un secondo momento.

Un altro elemento che deve essere possibile gestire da parte dell'admin sono **le categorie dei prodotti**.

CONCEPT GRAFICI

Si riportano qui dei concept e l’idea generale che è sono stati seguiti per lo sviluppo della parte grafica del progetto.

Immagine 1, Homepage

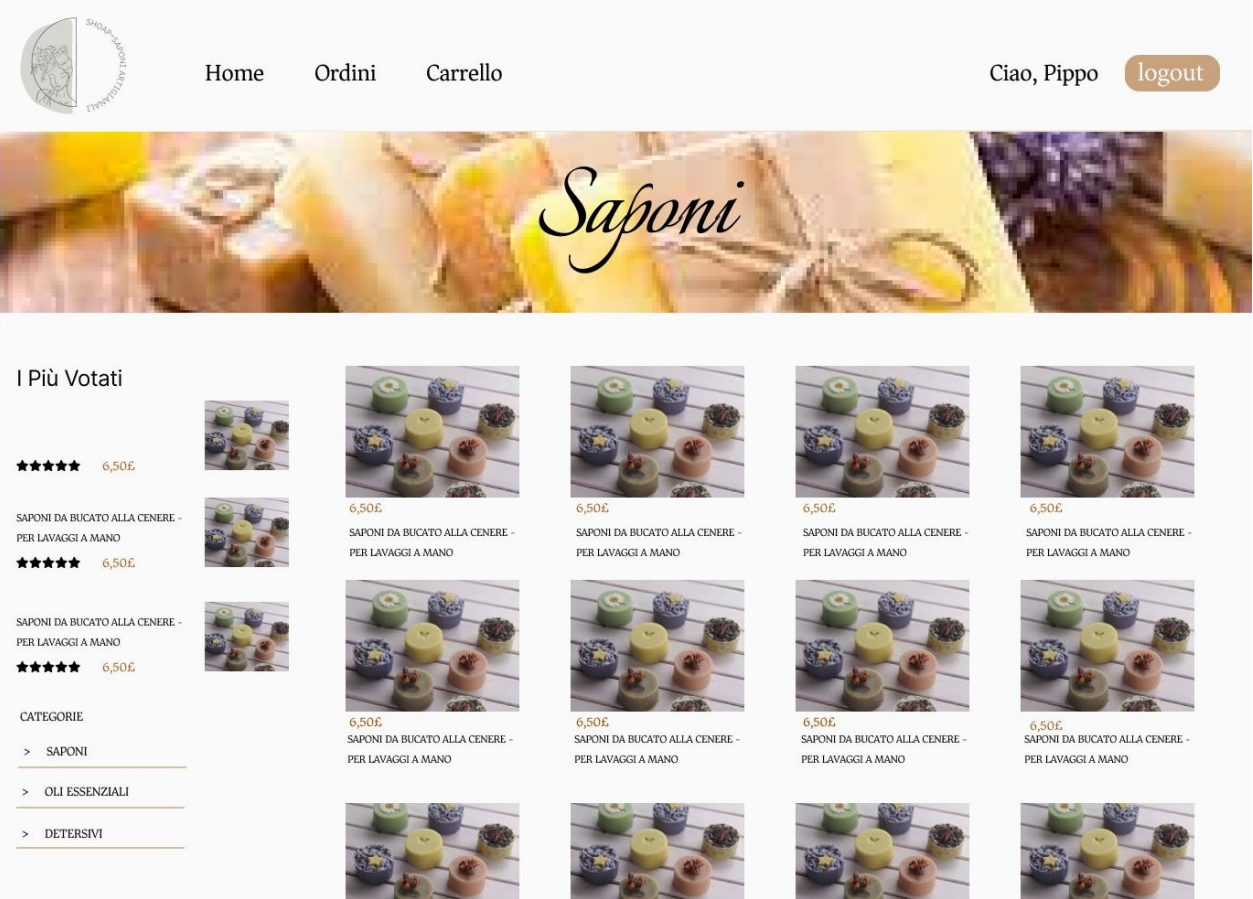



Immagine 2, Scheda specifica di un prodotto



Immagine 3, Carrello




Home


Ordini

Carrello

Ciao, Pippo


logout



	Prodotto	Prezzo	Quantità
<div><div>×</div><div></div></div>	Saponi da bucato alla cenere - per lavaggi a mano	6,50€	- 1 +


Ordina

Totale carrello

Prezzo	6,50€
Spedizione	Tariffa unica: 6,50€ Modifica indirizzo 
Totale	13,00€

Procedi con l'ordine

Immagine 4, Cassa




Home

Ordini

Carrello

Ciao, Pippo

logout



Dettagli di fatturazione

Il tuo ordine

Prezzo	6,50€
Spedizione	Tariffa unica: 6,50€ Modifica indirizzo
Totale	13,00€

Procedi con l'ordine

ANALISI FUNZIONALE

La web-app si apre sull'**homepage del sito**, qui trova, come da immagine 1 del concept grafico, i prodotti disponibili alla vendita, alcune loro recensioni e le varie categorie. L'utente può, infatti, aggiungere prodotti all'interno del suo carrello e per poi effettuarne e completarne la prenotazione, può anche, se registrato e se ha effettuato il login visualizzare lo **storico dei suoi ordini**, con tutte le **informazioni ad essi relative**.

All'interno della web-app ci troviamo una schermata dalla quale è possibile effettuare il **login** o la **registrazione** dell'utente all'interno del servizio messo a disposizione da wordpress.

A seconda della tipologia dell'utente che effettua il login risultano attive o meno alcune funzionalità.

All'utente standard è semplicemente possibile **navigare e visualizzare** le pagine dei prodotti disponibili alla vendita, divisi per categoria, e di effettuare la prenotazione degli stessi.

Per effettuare la prenotazione dei prodotti inseriti nel carrello al **checkout** viene richiesto all'utente un indirizzo e i **dati di spedizione**, che possono essere forniti inserendo le varie informazioni richieste nei corrispettivi campi di testo.

Una volta inserite tutte le informazioni necessarie si può procedere alla **prenotazione dell'ordine**.

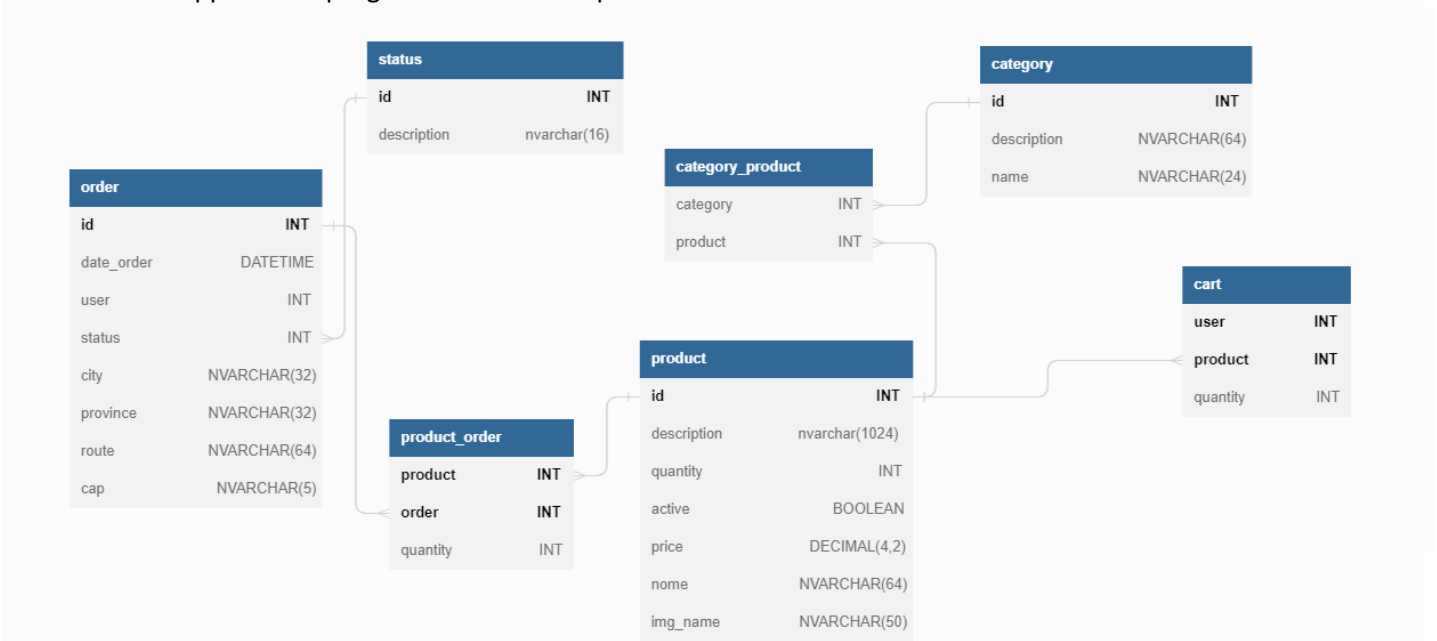
Nella parte amministrativa, alla quale può accedere un **account flaggato come admin**, l'utente con privilegi di admin può gestire vari aspetti dell'e-shop e dei prodotti. L'admin è abilitato ad effettuare tutte le **operazioni** di base **sui prodotti**, e quindi di crearne di nuovi, di modificare quelli esistenti o di eliminarne.

Un'altra operazione effettuabile dall'admin all'interno della sezione a lui dedicata è la **gestione delle categorie** di prodotti, e quindi di crearne di nuove, o di modificare o eliminare quelle esistenti.

Queste operazioni è possibile svolgerle premendo i corrispettivi **bottoni** presenti sulla schermata.

ANALISI TECNICA

Il **database** a supporto del programma è così composto:



Le **classi** presenti all'interno del progetto sono:

- + cart
- + category
- + category_product
- + order
- + product
- + product_order
- + status
- + wordpress (per la gestione dell'user)

Vengono ora riportate in particolare le **firme dei metodi** di ogni classe, i nomi sono esplicativi di ciò che il metodo fa.

cart:

- addCartProductQuantity
- deleteCartProduct
- deleteUserCart
- getArchiveCarts
- getUserCart
- setCart
- subtractCartProductQuantity

category:

- getActiveCategories
- getCategory
- setCategory
- updateCategory

category_product:

- getArchiveProductsCategories
- getProductsByCategory
- setProductCategory

order:

- getArchiveOrders
- getArchiveOrdersByUser
- getOrder
- getOrdersByStatus
- setOrder
- updateOrderStatus

product:

- addProductQuantity
- deleteProduct
- getActiveProducts
- getArchiveProducts
- getProduct
- getProductsByLikeName
- getProductsByName
- reactiveProduct
- setProduct
- subtarctProductQuantity
- updateProduct

product_order:

- getArchiveProductsOrders
- getProductsByOrder
- setProductOrder

status:

- getArchiveStatus
- getStatus
- setStatus
- updateStatus

wordpress:

- deleteUser

- getUser
- getArchiveUser
- getUser
- getUserByEmail
- reactiveUser

STRUMENTI UTILIZZATI

Per lo sviluppo sono stati utilizzati i seguenti software e strumenti:

- Visual Studio Code
- Github
- Xampp
- Figma

Per quanto riguarda il linguaggio di programmazione abbiamo utilizzato **PHP** e **JavaScript**, mentre come linguaggi di markup per lo sviluppo grafico della web-app e **HTML** e **CSS**.

Viene ora riportata una breve e sintetica descrizione di questi linguaggi.

PHP è un linguaggio di scripting **open-source** utilizzato per lo **sviluppo di applicazioni web** dinamiche.

PHP acronimo di "PHP Hypertext Preprocessor" è stato originariamente creato nel 1994 come un insieme di script CGI per monitorare le visite ad un sito web, da allora è cresciuto enormemente in popolarità e oggi è uno dei **linguaggi** di programmazione web **più utilizzati** al mondo.

PHP è noto per la sua facilità d'uso, **flessibilità** e **compatibilità** con la maggior parte dei sistemi operativi e dei web server. Viene spesso utilizzato insieme a MySQL o altri sistemi di gestione dei database per creare applicazioni web **dinamiche e interattive** come forum online, social network o e-commerce, proprio come nel nostro caso.

PHP è anche estremamente popolare tra i programmatori web grazie la grande comunità di sviluppatori che lo utilizzano e quindi degli ampi supporti per la documentazione e le **risorse disponibili in rete**, questo rende il processo di apprendimento del linguaggio relativamente semplice e accessibile a tutti coloro che vogliono iniziare a sviluppare applicazioni web dinamiche, nonché velocizza l'eventuale ricerca di soluzioni a problemi che si possono presentare in fase di sviluppo e di stesura del codice.

JavaScript è un linguaggio di programmazione interpretato **utilizzato** principalmente **per lo sviluppo di applicazioni web** lato client. È stato introdotto nel 1995 da Netscape ed è stato rapidamente adottato a livello mondiale come linguaggio di scripting per la programmazione web.

JavaScript è noto per la sua **versatilità** e per la sua capacità di creare **interattività e dinamicità** all'interno delle pagine, è anche utilizzato per sviluppare applicazioni web complesse come i giochi online, le applicazioni di streaming video e le applicazioni di product management.

Il linguaggio è estremamente popolare tra i programmatori web per la sua semplicità d'uso e per la **grande quantità di librerie e framework** disponibili, come React, Angular e Vue.js, che permettono di effettuare una vastissima gamma di azioni in modo semplice, veloce ed efficiente, **ottimizzando** così **l'utilizzo di risorse**, sia hardware che temporali. Inoltre, JavaScript è compatibile con la maggior parte dei browser moderni e viene costantemente aggiornato per mantenere il passo con le nuove tecnologie e le esigenze degli sviluppatori, rendendolo così uno strumento essenziale per lo sviluppo di applicazioni web moderne ed innovative.

HTML e **CSS** sono i fondamenti della creazione di pagine web.

HTML acronimo per "HyperText Markup Language" è un **linguaggio di markup** utilizzato per definire la struttura e il contenuto di una pagina web. CSS, invece, è acronimo per "Cascading Style Sheets" ed è utilizzato per **definire lo stile** e la presentazione visiva delle pagine web.

HTML è costituito da una **serie di tag e attributi** che definiscono l'aspetto del contenuto della pagina, come il testo, le immagini, i link e le tabelle. CSS, d'altra parte, viene utilizzato per **definire le proprietà visive** degli elementi HTML, come il **colore**, il **font**, la **posizione** e le **dimensioni**.

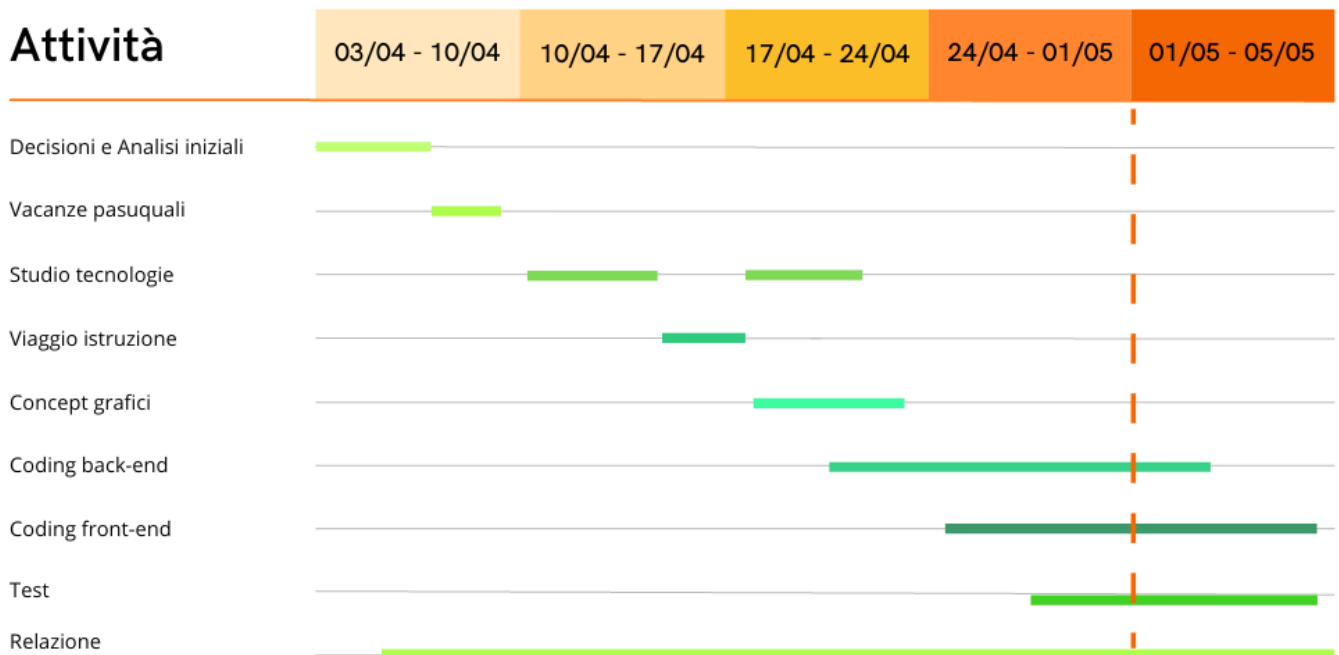
Insieme, HTML e CSS consentono agli sviluppatori web di creare pagine web di bel aspetto e funzionali, con una buona usabilità e una facile navigazione. HTML e CSS sono anche compatibili con una vasta gamma di dispositivi e browser web, garantendo che le pagine web siano accessibili **da qualsiasi dispositivo o sistema operativo**.

RENDICONTO LAVORO

Il lavoro è stato diviso equamente tra i membri del gruppo, considerando le capacità e i **punti forti di ognuno**, per cercare di ottenere il miglior risultato possibile. Il **diagramma di Gantt** di seguito riportato presenta il piano iniziale della gestione dei tempi, che non è stato del tutto rispettato, portando a finire il progetto circa una settimana dopo il previsto, rappresentata nel diagramma dall'ultima settimana colorata di arancione scuro, che è da vedere come una prosecuzione della settimana precedente.

SHOAP

Diagramma di Gantt



Sostanzialmente possiamo dire di aver impiegato circa **45 ore** totali per la realizzazione del progetto, così suddivise:

- **8 ore** (2 ore x 4 persone) per la decisione del **pattern di sviluppo**, la scelta del **progetto** e la stesura di una prima bozza dell'**analisi dei requisiti**;
- **6 ore** (1.5 ore x 4 persone) per lo **studio del pattern** e **delle tecnologie** necessarie allo sviluppo del progetto;
- **4 ore** (2 ore x 2 persone) per lo sviluppo dei **concept grafici**;
- **6 ore** per la stesura della relazione e la **ricerca di informazioni** e approfondimenti utili;
- **20 ore** (5 ore x 4 persone) per la **stesura del codice** ed il **debugging**;

Tutti han contribuito alla realizzazione del progetto, permettendo di arrivare alla data di consegna finale con un lavoro funzionante e a nostro avviso buono.

MIGLIORIE E UPDATE FUTURI

Avendo avuto più tempo a disposizione, o avendo gestito meglio quello dato a disposizione, si sarebbe potuta curare meglio la **parte grafica** della web-app, rendendola magari **più personalizzata** e meno anonima.

Un altro aspetto non gestito sono i **pagamenti** quando si effettua l'ordine, che si possono implementare, o per lo meno simulare, con gli strumenti per sviluppatori messi a disposizione dalle varie aziende di gestione dei pagamenti, come Nexi o PayPal.

Sempre per quanto riguarda la parte di checkout dell'ordine nelle versioni future può essere migliorata la **gestione degli indirizzi di spedizione**, dando la possibilità all'utente di **salvarne** alcuni tra i suoi preferiti e permettendo al programma di inserire automaticamente lo stesso indirizzo dell'**ultimo ordine effettuato**.