
Metodologie di programmazione del Web 2018/2019

Documento di progetto

20014021 Viktor Zahariev

23/06/2019

1.1 PLAGIO (NON RIMUOVERE QUESTA SEZIONE)

Consegnando questa relazione al docente, si dichiara implicitamente che il lavoro effettuato (codice del progetto e relativo documento) è frutto di un lavoro strettamente individuale.

Il plagio (cioè spacciare per proprio un lavoro fatto in realtà da altri) comporta l'annullamento immediato dell'esame, con obbligo di saltare un appello.

1.2 INTRODUZIONE

Il progetto consiste nella realizzazione di un sito e-commerce “Nuova Avventura” per conto di un committente: il gestore del sito. Questi vuole vendere online dei prodotti d'intrattenimento (libri, album musicali, film e videogiochi) gestendo il catalogo prodotti e gli ordini dei clienti.

Lo scopo è impegnare lo studente in una sfida multidisciplinare, che permetta di combinare in un unico progetto le nozioni apprese in diversi corsi del piano di studi. Si parte dalla realizzazione di un database la cui entità principale, i prodotti, sono suddivisi in più categorie con caratteristiche proprie. Le tabelle del DB richiedono una mappatura ad oggetti adeguata per non ripetersi nella scrittura del codice e per effettuare le query con efficacia. I controller del pattern architetturale MVC devono implementare i requisiti funzionali catturando tutte le direttive e limiti che il DB ed il modello ad oggetti non sono riusciti a imporre. Infine vi è la parte di presentazione che ha l'obiettivo di rendere facile e gradevole la navigazione dei clienti.

1.3 ANALISI DEI REQUISITI

Tabella 1 Analisi dei requisiti

| Id. | O / D | Descrizione formale e non ambigua | Eventuale ipotesi |
|------|---------------------------------------|--|--|
| RU01 | O | Il sito permette agli utenti di effettuare solamente degli acquisti (in altre parole, un utente non può vendere a sua volta; può solo acquistare) | NA |
| RU02 | O | Il gestore del sito è unico. | NA |
| RU03 | O | Il gestore può mettere on-line un articolo specificando, attraverso una pagina dedicata, i seguenti attributi: un identificativo, la sua categoria commerciale (libri, fumetti, videogiochi, Cd, vinile e Dvd), la foto del prodotto e una sua descrizione, il costo, il costo di spedizione (a prezzi diversi, a seconda del tipo di spedizione: posta ordinaria o corriere). | L'identificativo è uno SKU (Stock Keeping Unit) scelto dal gestore. Le categorie sono quattro: libri (di cui alcuni fumetti), album musicali (CD o vinili), film (DVD o Bluray) e videogiochi. |
| RU04 | O | Il prodotto è descritto da attributi diversi a seconda del tipo. Per esempio, un libro avrà un autore, un titolo, un editore, un numero ISBN, il numero di pagine e l'anno di uscita. Un Cd e il | NA |

| | | | |
|------|---|--|--|
| | | corrispondente vinile avranno l'autore e il titolo, i titoli delle canzoni e la durata di ciascuna, l'anno di uscita. Il Dvd avrà un titolo, un direttore e i nomi di almeno 3 attori (ma non più di 6), l'anno di uscita e il genere. | |
| RU05 | O | Qualora la giacenza di un prodotto scenda sotto le X (valore a scelta dello studente) unità, la pagina deve visualizzare un messaggio del tipo "Solo X istanze di questo prodotto sono disponibili". Nel caso in cui il prodotto sia esaurito, la pagina deve mostrare un messaggio corrispondente, ma deve offrire all'utente la possibilità di prenotarlo. | NA |
| RU06 | O | Quando la giacenza del prodotto sia nuovamente maggiore di zero, l'utente che si era prenotato deve ricevere un e-mail, che lo avvisa che ora è possibile effettuare l'acquisto. | NA |
| RU07 | O | L'acquisto di un prodotto avviene esclusivamente per mezzo di carta di credito. Quando il gestore riceve un ordine d'acquisto, invia gli estremi della carta di credito (proprietario, numero di carta, scadenza e codice sicurezza) a un gestore di carte di credito separato, che approva o rifiuta il pagamento. Le ragioni di un rifiuto sono: attributi errati o superamento del tetto mensile. | Il gestore di carte di credito deve essere simulato. Si considera come causa di rifiuto anche la mancanza di credito sufficiente per effettuare l'ordine. |
| RU08 | O | Un utente può registrarsi memorizzando i propri dati anagrafici, l'indirizzo di spedizione e la carta di credito (numero, data scadenza, codice di tre cifre e intestatario). | NA |
| RU09 | O | Un utente anonimo può visionare il catalogo offerto dal gestore, effettuando un semplice browsing per categoria, oppure specificando un prodotto (anche parzialmente specificato): per esempio il nome di un autore, o di un attore, o una canzone, un videogioco, o il personaggio di un fumetto. La ricerca può essere riordinata per "prezzo: crescente", "prezzo: decrescente", "nome: A-Z", "nome Z-A", "media recensioni clienti" e "Ultimi arrivi". | La ricerca tramite termini di ricerca restituisce risultati precisi e di numero limitato, perciò ha poco senso raffinarla ulteriormente. È più utile nella navigazione per categoria, dove il numero di prodotti elencati è assai maggiore. Con media recensioni clienti si mettono per primi i prodotti con miglior valutazione. |

| | | | |
|------|---|--|--|
| RU10 | O | Una volta che l'utente accede a un articolo, deve poter visionare le recensioni di altri clienti (una recensione ha da 1 a 4 stelle e un testo obbligatorio di lunghezza finita, minimo 80 caratteri e massimo 512). Se lo desidera, deve poter ordinare le recensioni per stelle sia in ordine crescente che decrescente. | NA |
| RU11 | O | Un utente registrato può lasciare una recensione su di un prodotto se lo desidera (non si controlla se l'utente ha acquistato l'articolo o meno). | NA |
| RU12 | O | Per effettuare l'acquisto di un prodotto, l'utente deve essere registrato. Ovviamente, è possibile acquistare più prodotti per volta editando un apposito carrello. | NA |
| RU13 | D | Gestione spedizioni. | Semplicemente si marca l'ordine come spedito e si indica la data di spedizione |
| RU14 | D | Storico degli acquisti per un utente registrato. | NA |

1.4 ANALISI FUNZIONALE

1.4.1 Il Data Base

Le entità principali sono i *prodotti*, *utenti* e *ordini*. I prodotti presentano le categorie *album*, *film*, *libri* e *videogiochi*. Però solo gli album hanno la tabella collegata *canzoni*. Mentre attori e autori sono considerati come un unico campo i cui valori sono separati da virgola.

Gli utenti possono recensire, prenotare e mettere nel proprio carrello i prodotti. Possiedono una *carta di credito* e hanno indicato un *indirizzo* di residenza. Possono effettuare ordini, cioè acquistare tramite carta di credito i propri prodotti in carrello.

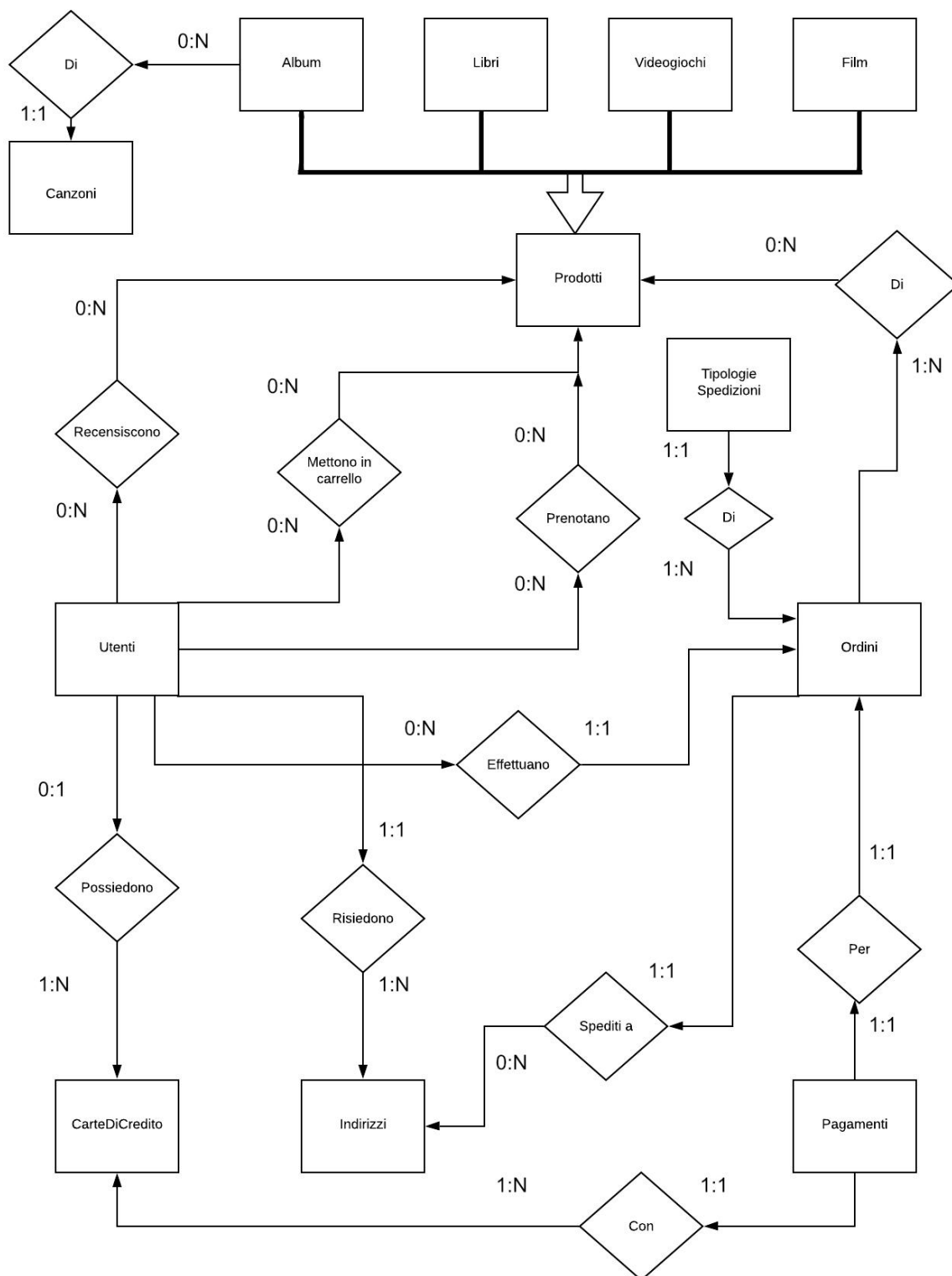
Ogni ordine consiste di più prodotti, ha associata la *tipologia di spedizione* (posta o corriere), la carta di credito utilizzata, che può essere specificata a tempo di acquisto e l'indirizzo di spedizione. Anche questo ultimo può essere cambiato, a tempo d'acquisto, da quello di residenza dell'utente. Infine c'è il *pagamento* che è univoco: se rifiutato si deve fare un nuovo ordine.

Tramite un booleano si specifica se il libro è un fumetto. Invece tramite degli *enum* si specificano le tipologie di prodotti: album (CD, VINILE) e film (DVD, BLURAY).

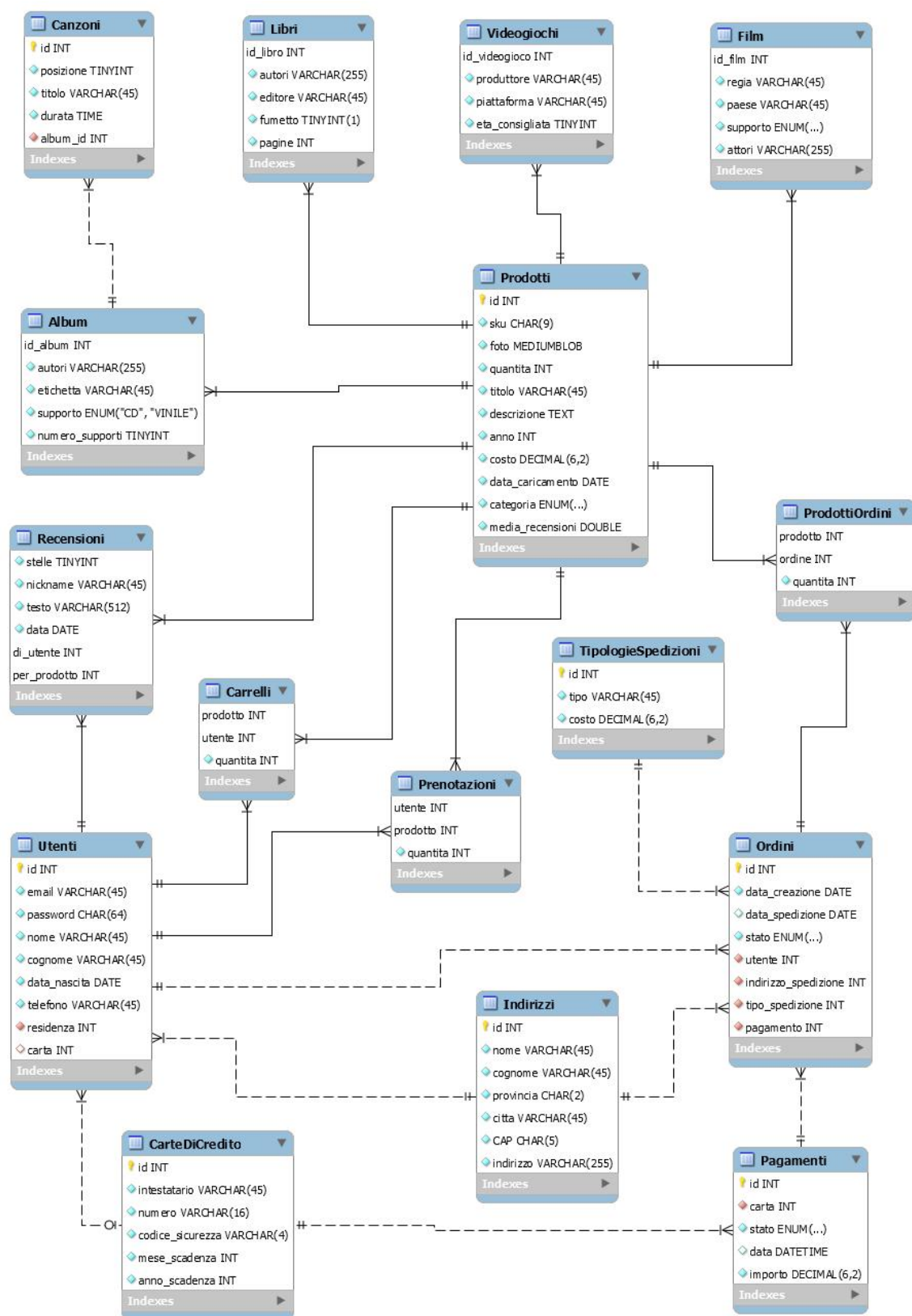
Anche lo stato dell'ordine è specificato da enum: IN_CORSO, PRONTO (pagamento accettato), SPEDITO, ANNULLATO.

Lo stesso vale per i pagamenti: IN_ATTESA, ACCETTATO, DATI_SCORRETTI, SUPERAMENTO_TETTO (mensile), CREDITO_INSUFFICIENTE.

1.4.1.1 Modello EER



1.4.1.2 Modello Logico-Relazionale



1.4.2 Corrispondenza dei requisiti utente

Tabella 2 Requisiti funzionali

| Requisito utente | Requisito funzionale | Descrizione | Eventuali ipotesi |
|------------------|----------------------|---|--|
| RU01 | RF01 | <p>Quando l'utente, durante la navigazione clicca su un prodotto, se la quantità disponibile è superiore a zero, visualizza il bottone "Aggiungi al carrello". Al click viene portato alla pagina di checkout se è loggato, altrimenti alla pagina di login. Nella pagina di checkout può inserire l'indirizzo di spedizione, scegliere il tipo di spedizione, e inserire i dati della carta di credito. I campi da compilare sono:</p> <ul style="list-style-type: none"> ● Nome, cognome, città, intestatario: required e maxlength = 45. ● Indirizzo: required e maxlength = 255 ● CAP: required, min = 10000 e max = 99999 ● Provincia: required e length = 2 ● Numero (carta) = required e espressione regolare: "\d{13} \d{16}" ● Cod. sicurezza: req. e min= 100, max= 9999 ● Data scadenza: req. E min = mese corrente | NA |
| RU02 | RF02 | Tutte le pagine di gestione sono sotto la cartella /admin. Questa è protetta da username e password non salvate nel DB, piuttosto nel file tomcat-users.xml. Le security constraints specificate in web.xml garantiscono che solo il gestore possa aver accesso alla parte amministrativa. | NA |
| RU03 | RF03 | <p>Nella pagina "Aggiungi prodotto" il gestore può specificare:</p> <ul style="list-style-type: none"> ● Titolo, editore, etichetta, nome canzone, regia, paese, produttore, piattaforma: req. e maxlength = 45 ● Autori e attori: req. e maxlength = 255 ● SKU: req. e length = 9 | Non si può aggiungere un prodotto con SKU già presente nel DB. |

| | | | |
|------|------|---|--|
| | | <ul style="list-style-type: none"> ● Foto, descrizione, durata canzone: required ● Anno: req. e min = 1000 ● Quantita: req. e min = 0 ● Costo: req. e min = 0, max = 9999.99 ● Numero pagine: req. e min = 1 ● Numero supporti: req. e min = 1, max = 127 ● Età consigliata: req. e min = 3, max = 18 | |
| RU04 | RF04 | I dettagli prodotto sono sempre aggiunti nella pagina del requisito 03. A seconda della categoria scelta viene caricato il form appropriato tramite codice Javascript nel file prodotto.js. | I limiti per il numero di attori sono controllati lato server. |
| RU05 | RF05 | Se meno di 10 prodotti disponibili lo si segnala all'utente nella pagina del prodotto, se non più disponibile si sostituisce il bottone "Aggiungi al carrello" con "Prenta ora". Il gestore può vedere per ogni prodotto prenotato la somma delle quantità richieste da più utenti. | NA |
| RU06 | RF06 | Quando il gestore nella pagina "Modifica prodotto" aumenta la quantità e questa era precedentemente zero, allora viene inviata una mail a tutti gli utenti prenotati. | Si assume che le email siano sotto il dominio gmail.com |
| RU07 | RF07 | La simulazione del gestore di carte di credito avviene generando un numero random da 1 a 10. Se compreso tra 1 e 7 il pagamento viene accettato, altrimenti gli altri casi (di rifiuto) hanno ugual possibilità di avvenire. | NA |
| RU08 | RF08 | <p>Un utente si registra specificando solo i dati anagrafici e l'indirizzo di residenza.</p> <p>Alcune delle validazioni (nome, cognome, indirizzo, città, CAP, provincia) le abbiamo già viste nella pagina di checkout. Di nuovo c'è:</p> <ul style="list-style-type: none"> ● Email: req., UNIQUE e maxlength = 45 ● Password: req. e minlength = 6 ● Data nascita e numero di telefono: req. | Per non richiedere troppe info in fase di registrazione i dati della carta si richiedono solo prima di fare un acquisto. |

| | | | |
|------|------|---|--|
| RU09 | RF09 | Il browsing per categoria si effettua cliccando sulla categoria nella navbar del sito. In essa è anche presente un box di ricerca. Il riordinamento dei prodotti avviene scegliendo il campo voluto in una select. I sottotipi (es. solo vinili per album) si scelgono tramite radio button. | NA |
| RU10 | RF10 | Le recensioni non sono in una pagina separata, ma in quella del prodotto. L'ordinamento per numero di stelle è effettuato lato client con script JS. I campi sono validati come da requisito. | NA |
| RU11 | RF11 | Sempre nella pagina prodotto, cliccando su "Scrivi una recensione" si apre un modal dove poter compilare i campi, tra cui un nickname se non vuole indicare il nome vero per privacy. | Utente lascia una sola recensione per prodotto |
| RU12 | RF12 | Per effettuare l'acquisto (così come l'aggiunta al carrello e la prenotazione) l'utente deve essere registrato e deve aver effettuato l'accesso. Quando si modifica la quantità desiderata nel carrello (obbligatoriamente minore di quella disponibile) bisogna cliccare su "Aggiorna" per salvare la modifica anche nel DB. | Tra carrello e checkout la quantità disponibile potrebbe scendere perciò si ripete il controllo lato server. |
| RU13 | RF13 | Quando il gestore riceve un ordine, questo è in corso. Se il pagamento va a buon fine diventa pronto. Solo in quel caso il gestore può marcare l'ordine come spedito. | NA |
| RU14 | RF14 | Dal menu di navigazione un utente registrato e loggato può vedere l'elenco degli ordini precedenti. | Se non ancora pagato l'ordine si può annullare. |

1.5 ANALISI TECNICA

1.5.1 Tecnologia utilizzata

Il DBMS utilizzato è MySQL 5.7, il vantaggio fornito è la disponibilità del tool MySQL Workbench che ha permesso la generazione fisica del DB dallo schema Logico-Relazionale. Inoltre è stato possibile sincronizzare con il DB tutte le successive modifiche allo schema con pochi click, evitando la scrittura manuale di codice SQL. Il web server utilizzato è Tomcat 9.0, scelta quasi inevitabile visto che l'intero sito, a parte qualche script JavaScript, è stato realizzato con tecnologia Java EE, in particolare Servlets e JSP. Quasi inevitabile visto che è il più utilizzato e di conseguenza si è trovato molto materiale di supporto online. Peccato solo per i

problemi avuti nel codificare il testo in formato UTF8. L'interfaccia utente è stata costruita con il framework Bootstrap 4 (notevole risparmio di tempo), non il tema di default ma: <https://bootswatch.com/flatly/>

1.5.2 Mappatura dei requisiti funzionali Tecnologia utilizzata

Purtroppo è impossibile descrivere l'implementazione di tutti i requisiti funzionali, anche cercando di essere brevi, rimanendo nei limiti di lunghezza imposti. Perciò se si vuole andare in dettaglio sarebbe meglio farlo durante la dimostrazione, analizzando direttamente il codice.

Vorrei però lo stesso sottolineare alcuni aspetti. Per ogni entità del DB vi è la corrispondente classe (es. Prodotto, Recensione) nel package *business*. Il package *data* contiene le rispettive classi di accesso al DB (es. ProdottoDB, RecensioneDB). I controller invece sono solo quattro. *GestioneController* implementa tutte le funzionalità di amministrazione svolte dal gestore, è mappato con l'URL: "admin/GestioneController" e insieme a tutti i file .jsp che si trovano nella cartella admin, l'accesso è riservato solamente al gestore. *NavigazioneController* è accessibile a chiunque e si occupa della navigazione del sito e della ricerca e riordinamento dei prodotti. *AutenticazioneController* si occupa della registrazione e accesso dei clienti: quando un cliente effettua l'accesso si salva nella sua sessione il suo id. *AcquistoController* infine si occupa di tutte le operazioni che portano all'acquisto e che richiedono l'accesso dell'utente, es. mettere un prodotto nel carrello, effettuare o annullare l'ordine, vedere lo storico degli ordini, lasciare recensioni, ecc.

Il package *util* contiene le classi *Base64ImageEncoder* che codifica le immagini in formato binario salvate nel DB in stringhe, affinché possano essere visualizzate dal browser. *CharacterSetFilter* filtra tutte le richieste e si assicura che il formato di testo utilizzato sia UTF8. *PaymentSimulator* è stato già descritto in RF07. *MailUtilGmail* invia una mail agli utenti prenotati se il prodotto diventa disponibile, come da nome funziona solo se si utilizza una mail *Gmail*: per lo scopo ho creato l'account nuovaavventura.sales@gmail.com

L'unica cosa degna di nota è che non sono stato in grado di implementare la funzione di ricerca in modo completo. Ho creato degli indici FULLTEXT sui campi desiderati, però la mia implementazione ricerca solamente il titolo e la descrizione del prodotto, senza andare a guardare nei dettagli per categoria (es. autore di libro, produttore di videogioco). Ho cambiato diverse volte prima di arrivare all'implementazione finale delle classi dei prodotti. Ora la classe *Prodotto* ha il campo *DettagliProdotto* (classe astratta) da cui ereditano le classi *Libro*, *Album*, *Film* e *Videogioco*. Perciò ho usato la composizione piuttosto che l'ereditarietà. Non ho avuto grandi problemi ad implementare le varie funzionalità tranne che nel caso della ricerca. Forse è perché so come creare query che dipendono da più variabili. Cioè selezionare i prodotti per categoria è semplice. Così come per categoria e limite (es. solo 10), categoria e tipo (es. film bluray), categoria e ordinamento (es. libri per prezzo decrescente). Ho utilizzato metodi separati di accesso al DB. Ma quando si tratta di combinare più variabili insieme es. categoria, tipo, limite e ordinamento la query si complica molto e ci sono tanti casi da considerare. Proprio come l'aggiungere i dettagli prodotto tra i campi in cui cercare. Forse una soluzione pulita non esiste e avrei dovuto sporcarmi le mani realizzando la funzionalità, anche con un po' di spaghetti code, però credo che avrei perso il controllo e ho preferito rinunciare.