



Il Villaggio Turistico

Documento di Progettazione

PROGETTAZIONE

PROGETTAZIONE ARCHITETTURALE

Requisiti non funzionali

Nell'Analisi del Problema (Tabella Vincoli) sono emersi tre requisiti non funzionali che impongono dei vincoli al sistema:

- Tempo di risposta
- Usabilità
- Sicurezza

Nello specifico caso in esame, Usabilità e Sicurezza hanno pochi conflitti a parte l'eventuale richiesta di un ulteriore login se per caso scade la sessione di lavoro. L'Usabilità impatta molto di più la struttura delle interfacce che andranno progettate in modo tale da mantenere nelle stesse View le informazioni necessarie alle funzionalità richieste

Diversa la questione che riguarda Tempo di risposta e Sicurezza, aggiungere strati (layer) e meccanismi di cifratura per migliorare la sicurezza ovviamente porta ad un peggioramento delle prestazioni del sistema, occorre quindi trovare un bilanciamento tra i due aspetti.

Considerando la tipologia di sistema che deve essere sviluppato, si ritiene maggiormente critico l'aspetto di sicurezza dei dati in quanto la "Tabella Valutazione Beni" mette in luce che nel caso di attacchi al sistema andati a buon fine si rischia un'esposizione molto alta con perdite finanziarie e di immagine. Inoltre, gli utenti principali di tale sistema sono operatori umani che spesso non sono in grado di percepire se il Sistema impiega qualche frazione di secondo in più o in meno nella risposta, non si hanno vincoli real-time da soddisfare.

Scelta dell'architettura

Dal punto di vista architetturale, l'architettura più idonea per questo tipo di sistema è un'*architettura client/server a tre livelli*.

L1- Client:

Al fine di rispettare il vincolo del "minimo privilegio" si è deciso di sviluppare un client diverso per ogni diversa funzionalità Core:

- Un client per la funzionalità "Hotel"
- Un client per le funzionalità relative ai Punti Vendita e alla Catena Punti Vendita
- Un client per la funzionalità di gestione dei Log

L2 - Server:

Per quel che riguarda il lato server, si è deciso di adottare server diversi per la gestione delle diverse funzionalità

- Un server per la funzionalità "Hotel"
- Un server per le funzionalità relative ai Punti Vendita e alla Catena Punti Vendita
- Un server per la funzionalità di gestione dei Log
- Un server per la gestione del login

L3 - Persistenza:

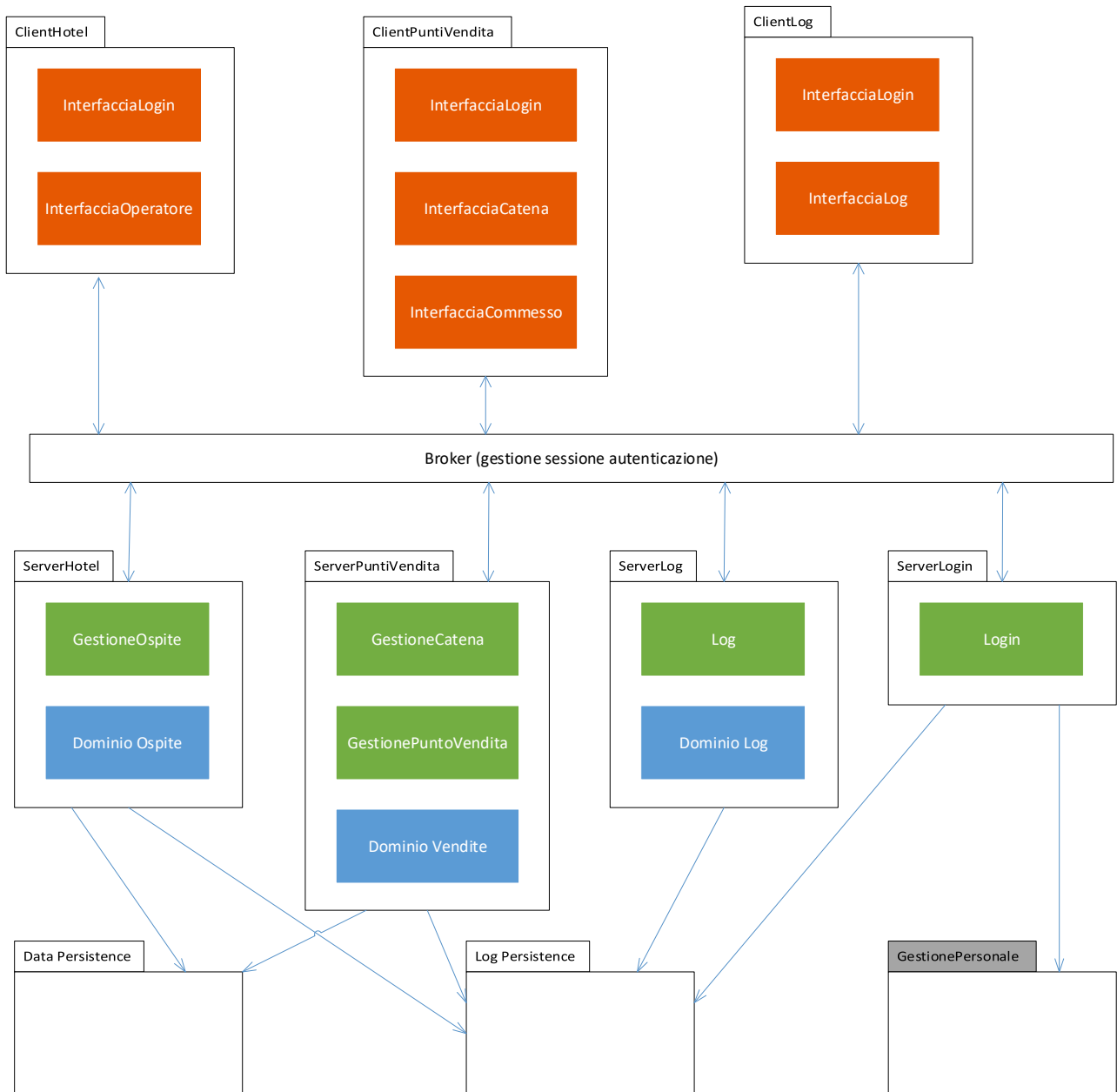
Per la gestione della persistenza si avrà un server dedicato nel quale sarà installato un opportuno DBMS che gestirà la base dati del Villaggio Turistico (per lo schema di tale base dati si veda oltre) e un ulteriore server sarà dedicato alla memorizzazione dei log. Tenere separati log e db è molto importante in quanto eventuali compromissioni di una macchina non dovrebbero alterare anche l'altra. Ci deve essere sempre un modo per riportare i dati del sistema a uno stato consistente ed integro.

L'interfacciamento con il DBMS avverrà attraverso la metodologia "forza bruta" utilizzando i metodi CRUD.

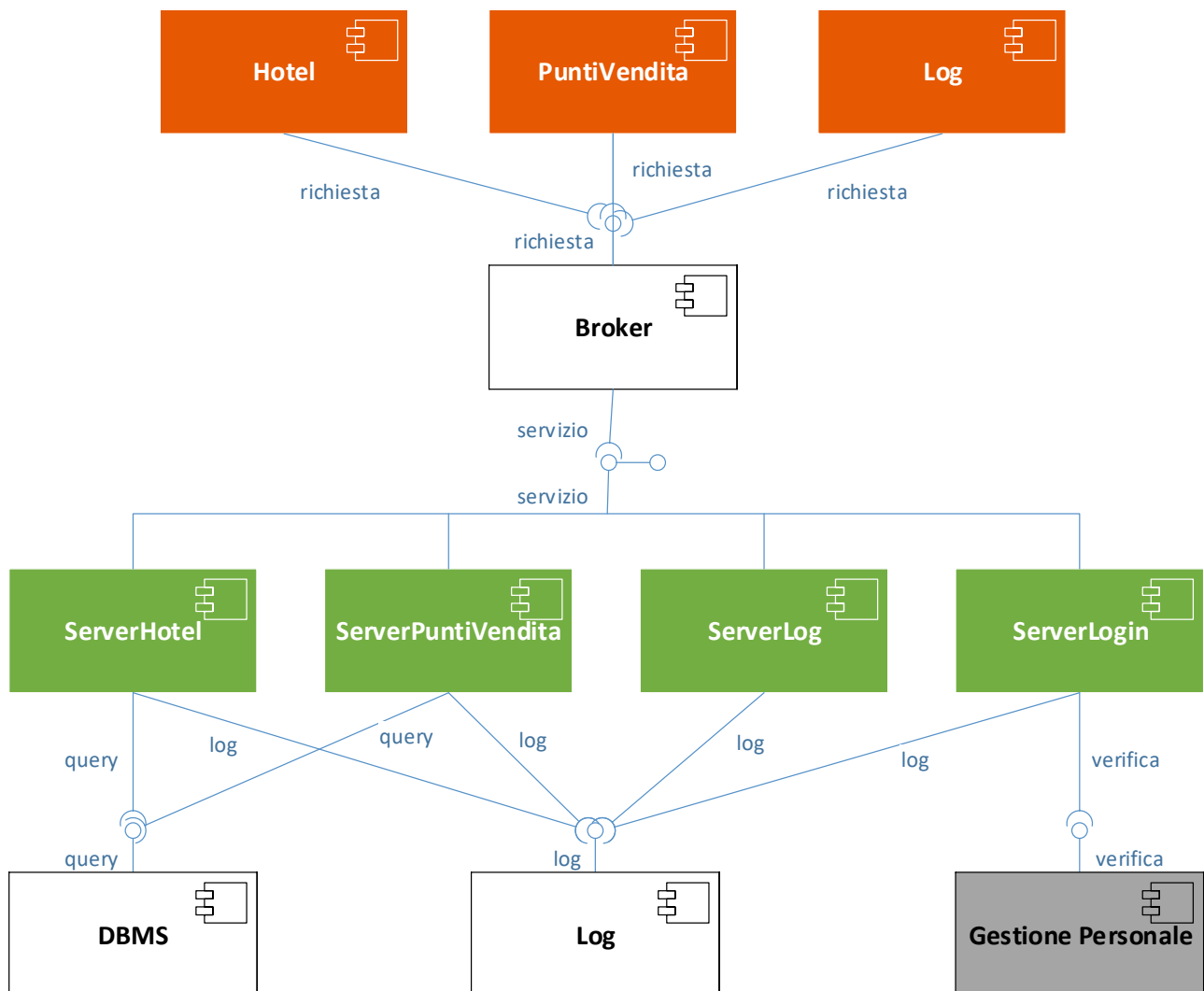
Si è inoltre deciso di adottare il *Pattern Broker* per la gestione della sessione e per proteggere ulteriormente lo strato dei server. Tale scelta inoltre permette di applicare *The Dependency Inversion Principle*: disaccoppiando i client dai server, la struttura dei server è "nascosta" dal broker e diventa abbastanza semplice "sostituire" un server oppure aggiungerne uno nuovo (*design for change*). Per esempio se si volesse introdurre il servizio di Prenotazioni on-line sarebbe sufficiente aggiungere un nuovo server e applicare una piccola modifica al broker.

Infine, per *garantire la sicurezza* nella comunicazione client/server si è deciso di utilizzare il protocollo *TLS* (transport layer security). TLS è un protocollo crittografico che permette una comunicazione sicura dalla sorgente al destinatario (end-to-end) su reti TCP/IP (come ad esempio, Internet) fornendo autenticazione, integrità dei dati e cifratura, operando al di sopra del livello di trasporto.

Nella figura sottostante è riportata l'Architettura del Sistema organizzata attraverso un diagramma dei package.



Nella figura sottostante è riportata l'Architettura del Sistema organizzata attraverso un diagramma dei componenti.



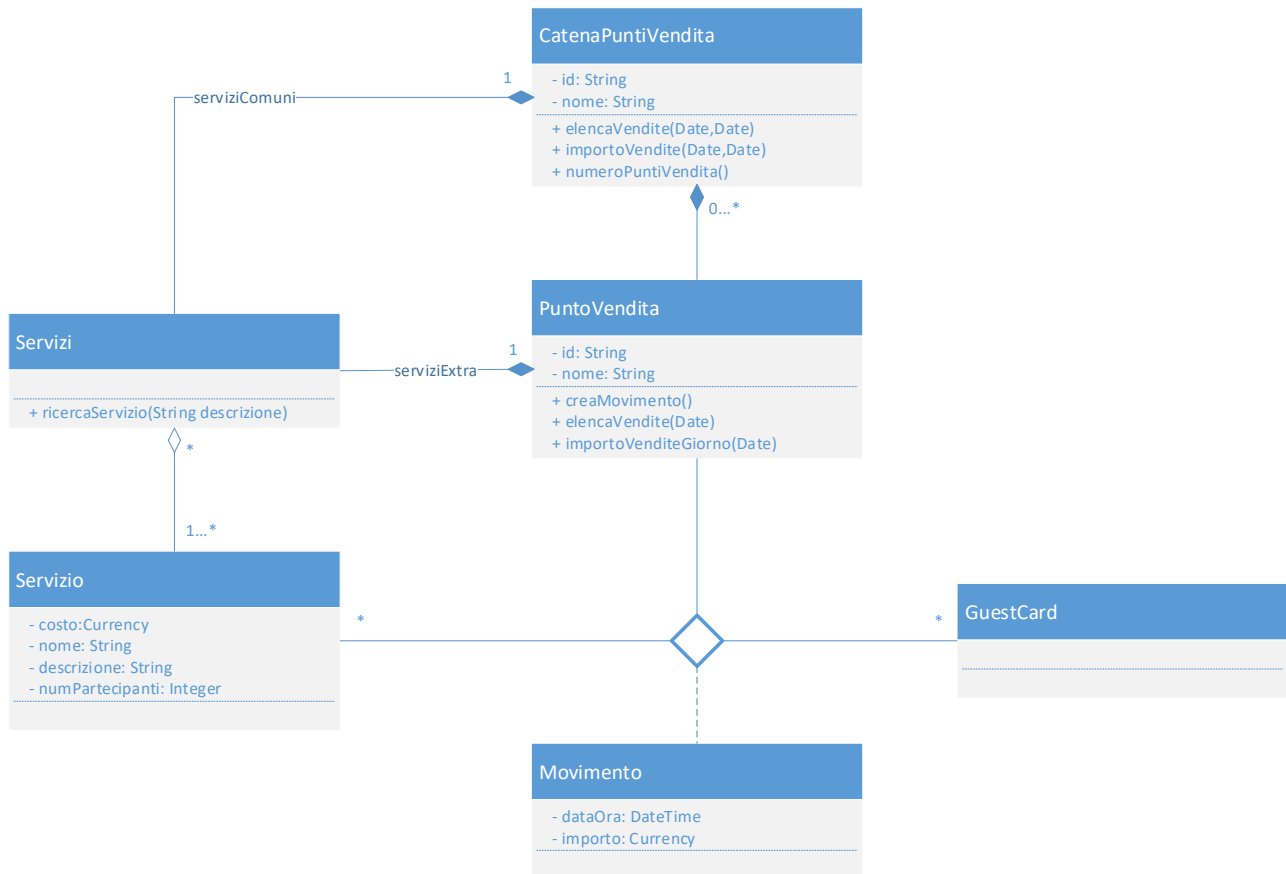
Si notino i collegamenti attraverso le interfacce.

PROGETTAZIONE DI DETTAGLIO

Nel seguito si riportano i diagrammi di dettaglio delle varie parti del Sistema.

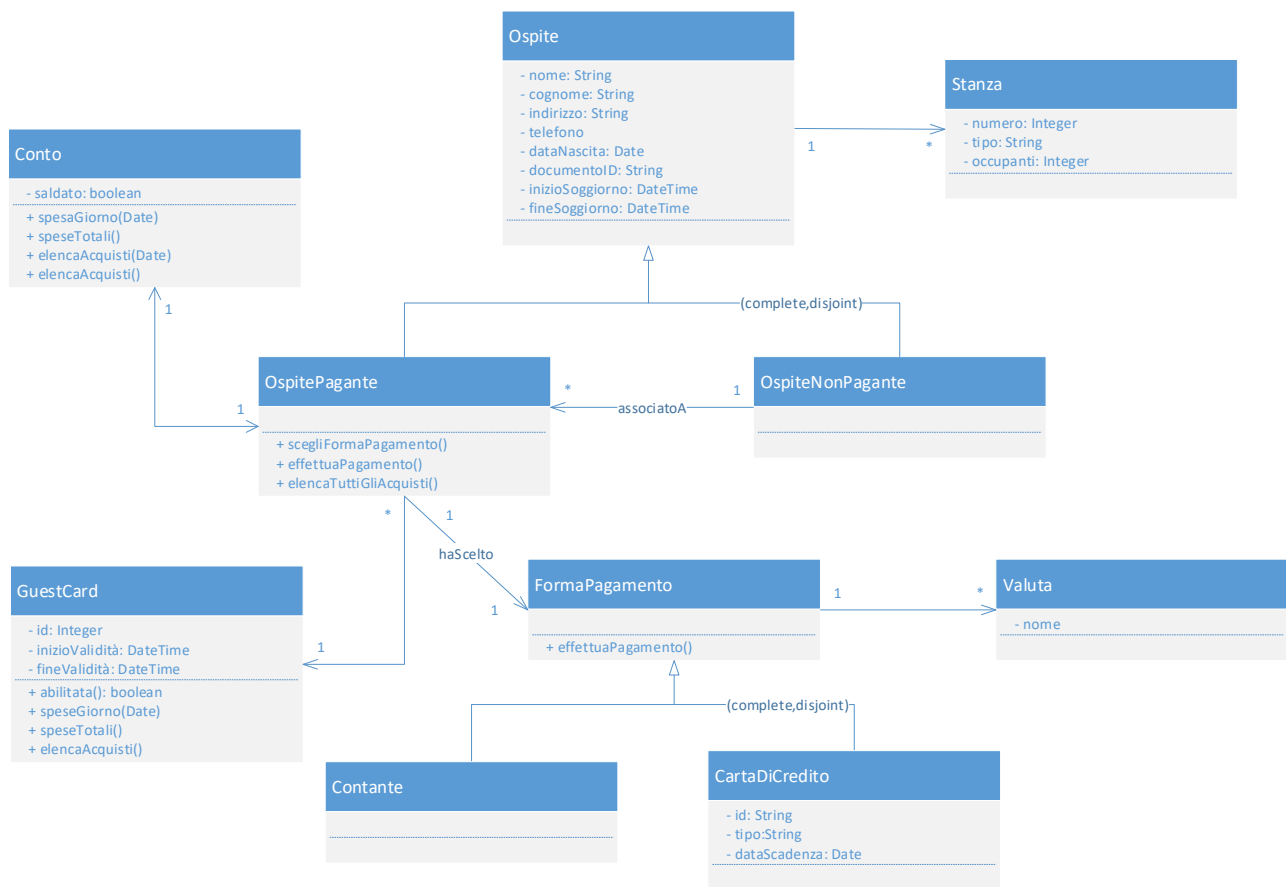
Struttura

Diagramma di Dettaglio: Dominio - Vendite



Come scelta progettuale si è deciso di classificare meglio i Servi offerti organizzandoli in “Servizi Comuni” a tutti i Punti Vendita e “Servizi Extra” che sono disponibili solo presso uno specifico Punto Vendita.

Diagramma di Dettaglio: Dominio- Ospite



Come scelta progettuale si è deciso di eliminare la Generalizzazione “Maggiorenne” / “Minorenne” perché non portava nessuna informazione utile. Se per caso un OspiteNonPagante passa da “Minorenne” a “Maggiorenne” deve essere cura dell'OspitePagante segnalare la cosa alla gestione del Villaggio Turistico ed eventualmente chiedere una nuova GuestCard.

L'associazione bi-direzionale Conto-OspitePagante permette di risalire alle GuestCard associate all'OspitePagante e quindi di calcolare le spese e richiedere l'elenco acquisti.

Diagramma di Dettaglio: Dominio- Log

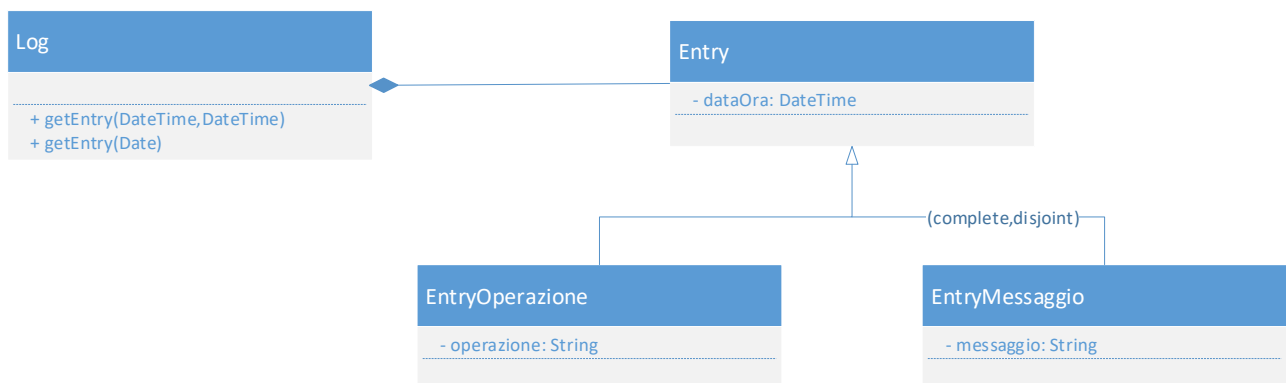
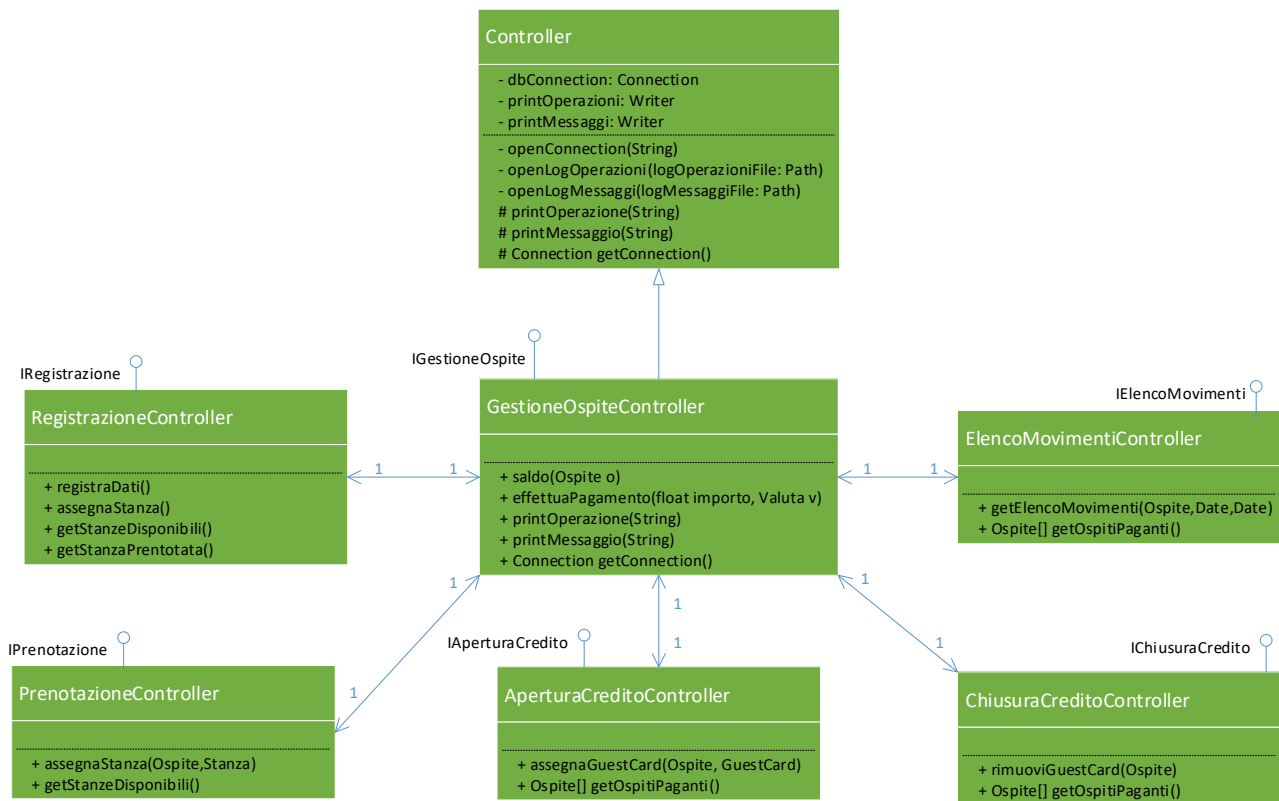


Diagramma di Dettaglio: Interfacce nei server



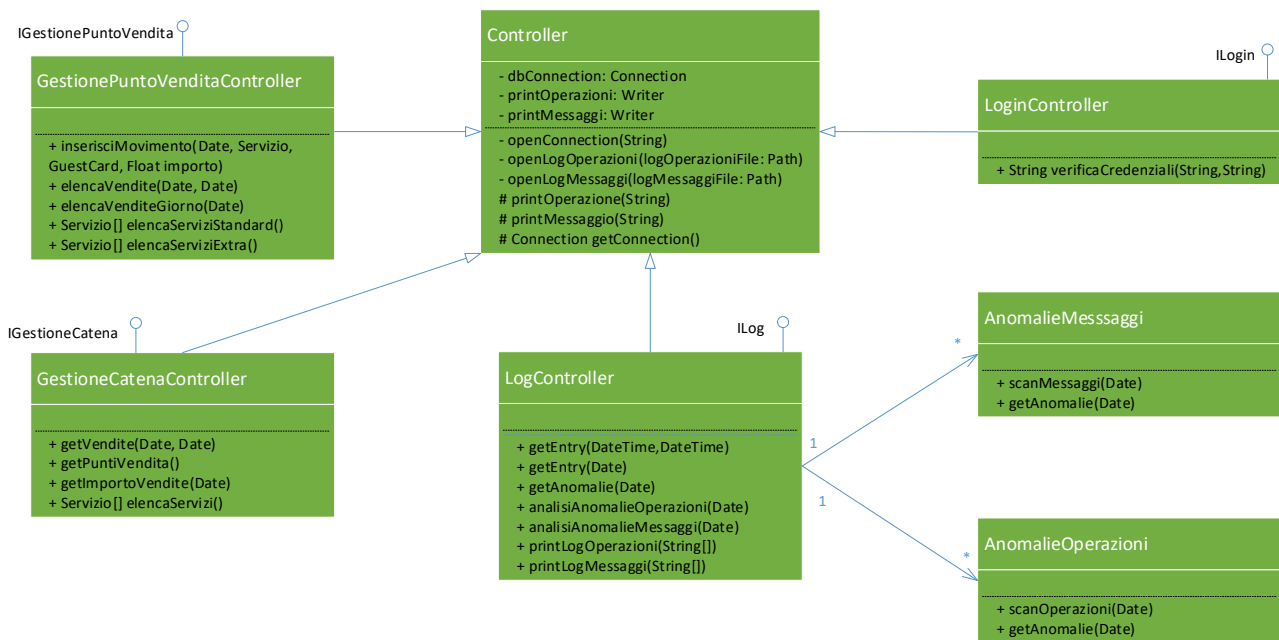
Queste interfacce permettono di applicare *The Dependency Inversion Principle*. Da notare l'interfaccia IPrenotazione, è stata inserita per poter gestire la gestione delle prenotazioni come richiesto nel Piano di Lavoro tra gli sviluppi futuri.

Diagramma di Dettaglio: GestioneOspite



Si è deciso di fattorizzare nella classe **Controller** tutta la gestione della persistenza dei dati in modo da non dover replicare la scrittura/lettura su db e sui log. Attenzione che solo **GestioneOspiteController** eredita da **Controller**, gli altri controller accedono ai servizi di **Controller** attraverso le associazioni bi-direzionali.

Diagramma di Dettaglio: GestioneCatena / GestionePuntoVendita/ Log / Login

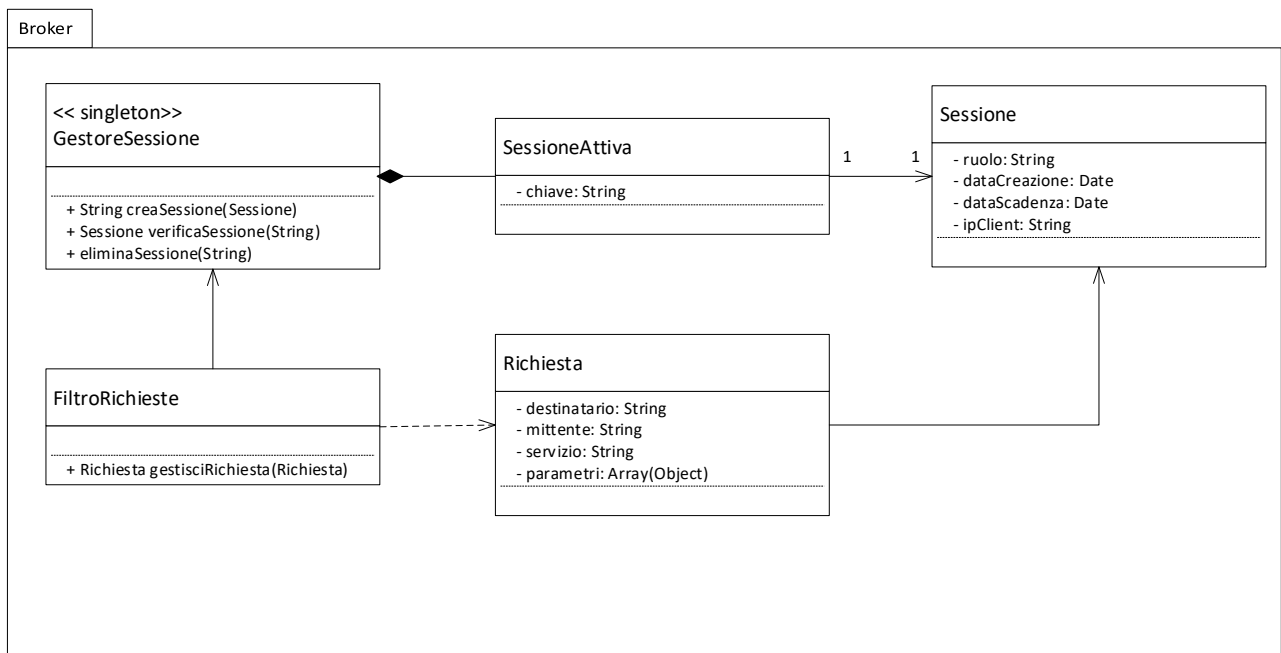


Come per GestioneOspite anche questi controller ereditano da Controller per la gestione degli accessi al db e per la scrittura dei Log.

Da notare che le funzionalità di LogController sono state estese al fine di supportare la scrittura dei Log da parte dei diversi Client. In particolare, sono comparse le funzioni “printLogOperazione” e “printLogMessaggio”. Si è deciso di operare in tale modo per fattorizzare la scrittura dei log da parte dei diversi Client, l’alternativa sarebbe stata quella di creare un ulteriore server per la scrittura dei log dei Client. Tale scelta però avrebbe portato a un inutile spreco di risorse per l’installazione di un server dedicato e si è preferito procedere per altra strada.

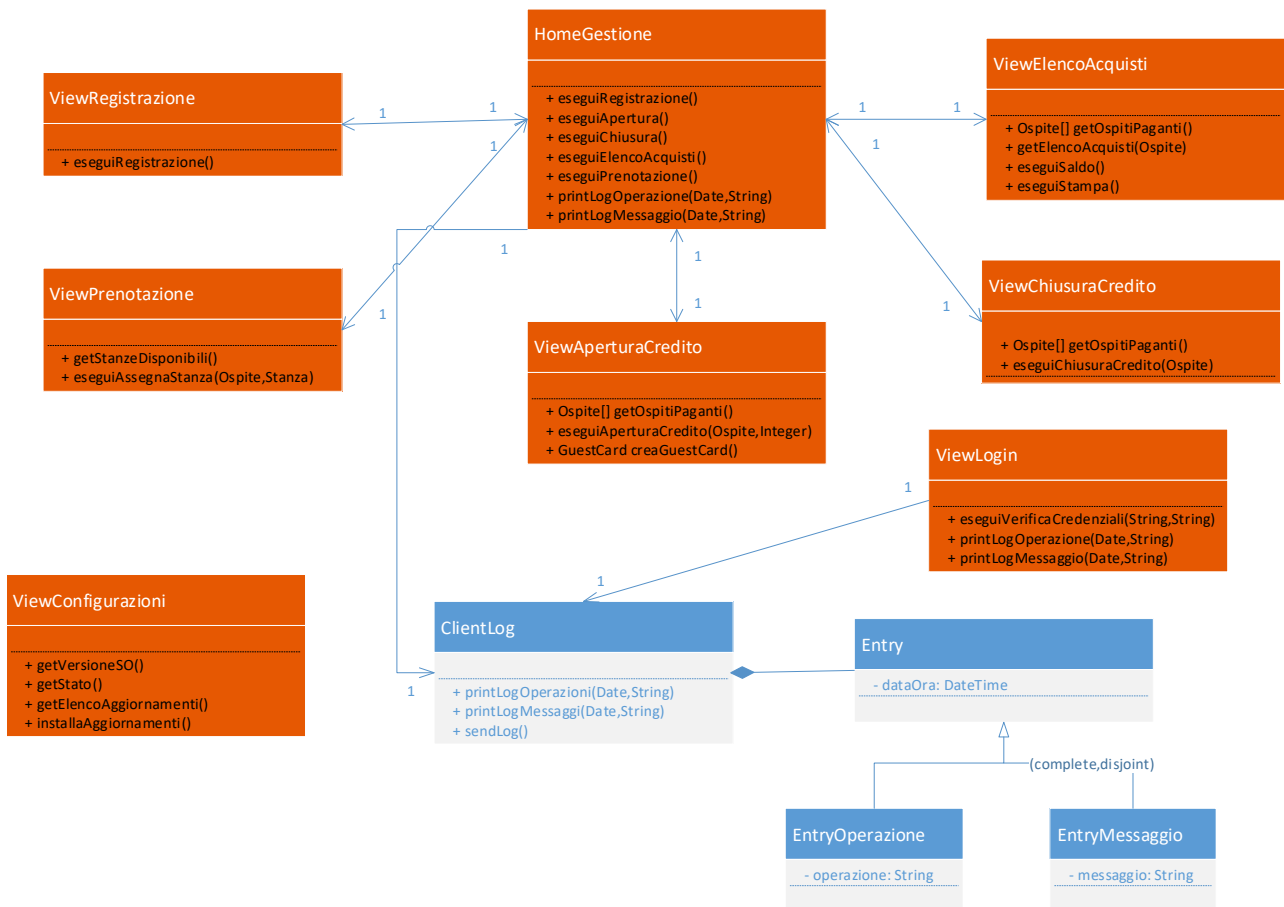
I meccanismi di scrittura dei log da parte dei Client saranno discussi successivamente.

Diagramma di Dettaglio: Broker



Qui si ha la struttura del Broker: l'entrypoint è **FiltroRichieste**, che manipola le richieste che arrivano dai client aggiungendo la sessione e verificando che la sessione sia attiva, smista poi le richieste al server corretto e restituisce le risposte ai Client.

Diagramma di Dettaglio: ClientHotel



Questo diagramma rappresenta il dettaglio delle classi presenti nel client per la gestione dell'Hotel del Villaggio Turistico. In particolare, le classi "rosse" rappresentano le interfacce utente che dovranno essere realizzate, mentre "View Configurazioni" è una nuova interfaccia inserita a seguito delle valutazioni che saranno mostrate nella "Progettazione per il Deployment". La classe "ClientLog" si occupa della memorizzazione locale dei Log nel client. Tali log sono poi inviati periodicamente (vedi diagramma di sequenza successivo) al ServerLog. Per fare una corretta Analisi delle anomalie è necessario disporre anche dei log dei client al fine di identificare tentativi di attacchi "man in the middle", o di verificare eventuali compromissioni nelle comunicazioni o tentativi di svolgere operazioni diverse da quelle svolte solitamente.

Al fine di semplificare la creazione delle interfacce, nel seguito viene riportata una versione schematica dei componenti necessari nelle diverse interfacce.

Interfaccia Login

Sistema di Gestione Villaggio Turistico

username

password

Accedi

Sistema di Gestione Ospiti

Registra

Apertura Credito

Chiusura Credito

Elenco Acquisti

Prenota

Apertura Credito

Ospite Pagante

Num GuestCard

Apri

Annulla

Registrazione

Nome

Cognome

Indirizzo

Telefono

Data nascita

Estremi Documento

Data Inizio

Data Fine

Stanza

Carta Credito

Pagante

☐

Registra

Annulla

Chiusura Credito

Ospite Pagante

Elenco Acquisti

Chiudi

Annulla

Gestione Pagamento

Totale da Pagare

Scegli Valuta

Emetti Fattura

Annulla

Elenco Acquisti

Ospite Pagante

Data	Punto Vendita	Servizio	Importo
------	---------------	----------	---------

Stampa

Salda

Chiudi

Gestione Prenotazione

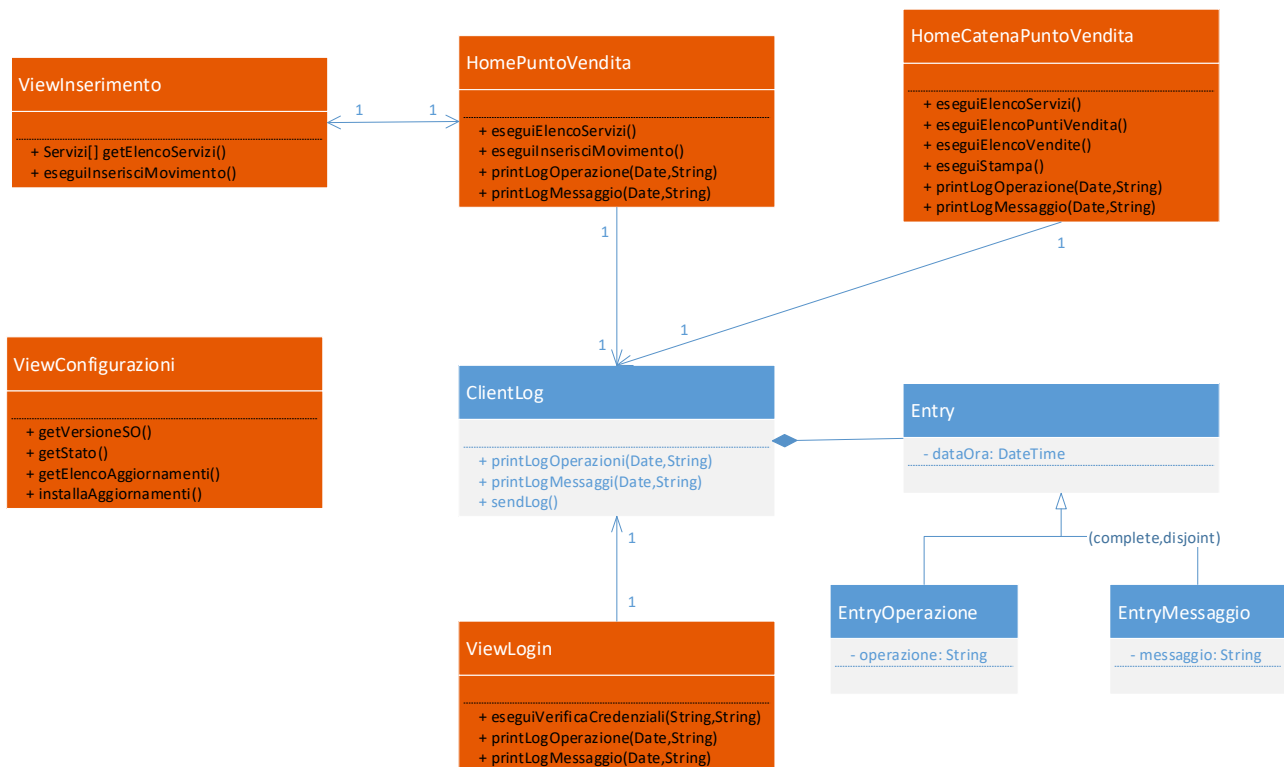
Cognome Ospite

Scegli Stanza

Prenota

Annulla

Diagramma di Dettaglio: ClientPuntiVendita



Valgono le stesse considerazioni fatte per il ClientHotel.

Di seguito si riportano gli schemi delle interfacce utente a esclusione dell'interfaccia di Login che è stata mostrata in precedenza.



Sistema di Gestione Catena

Elenco Servizi

Elenco Negozi

Elenco Vendite

Elenco Servizi

Servizio	Costo	Partecipanti
----------	-------	--------------

Chiudi

Elenco Punti Vendita

Punto Vendita

Chiudi

Elenco Vendite

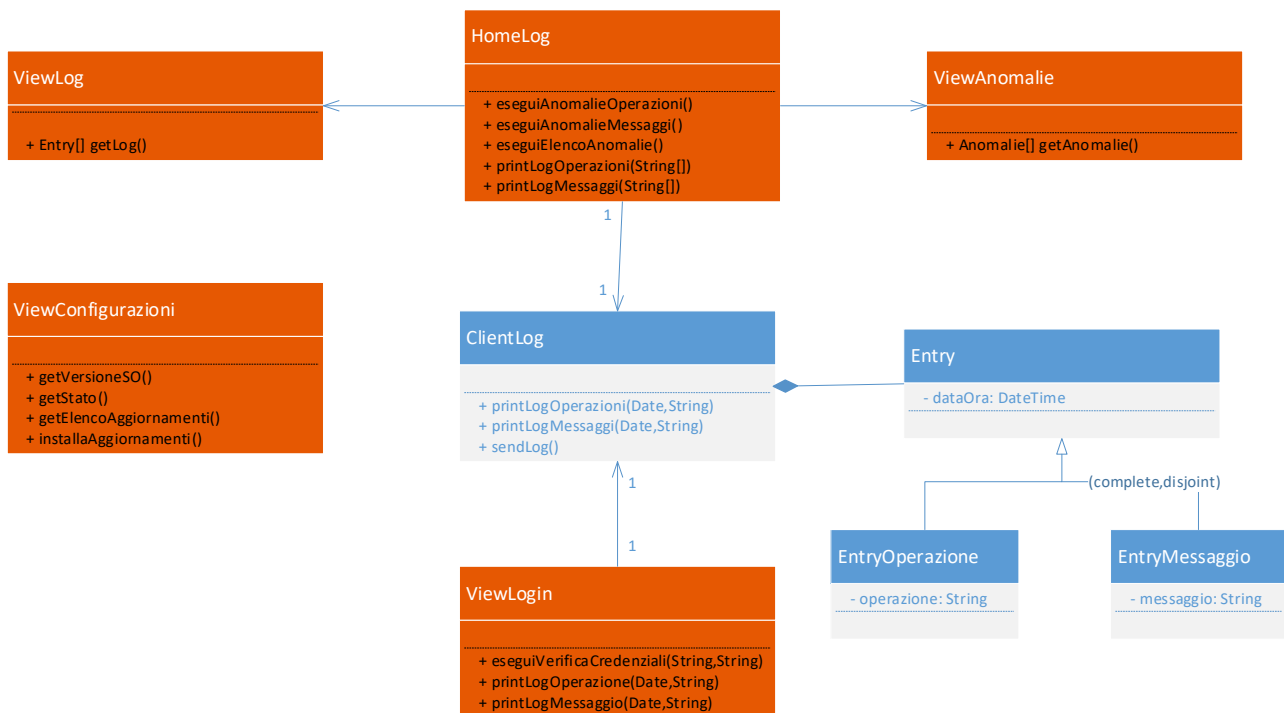
PuntoVendita

Data	Servizio	Importo
------	----------	---------

Stampa

Chiudi

Diagramma di Dettaglio: ClientLog



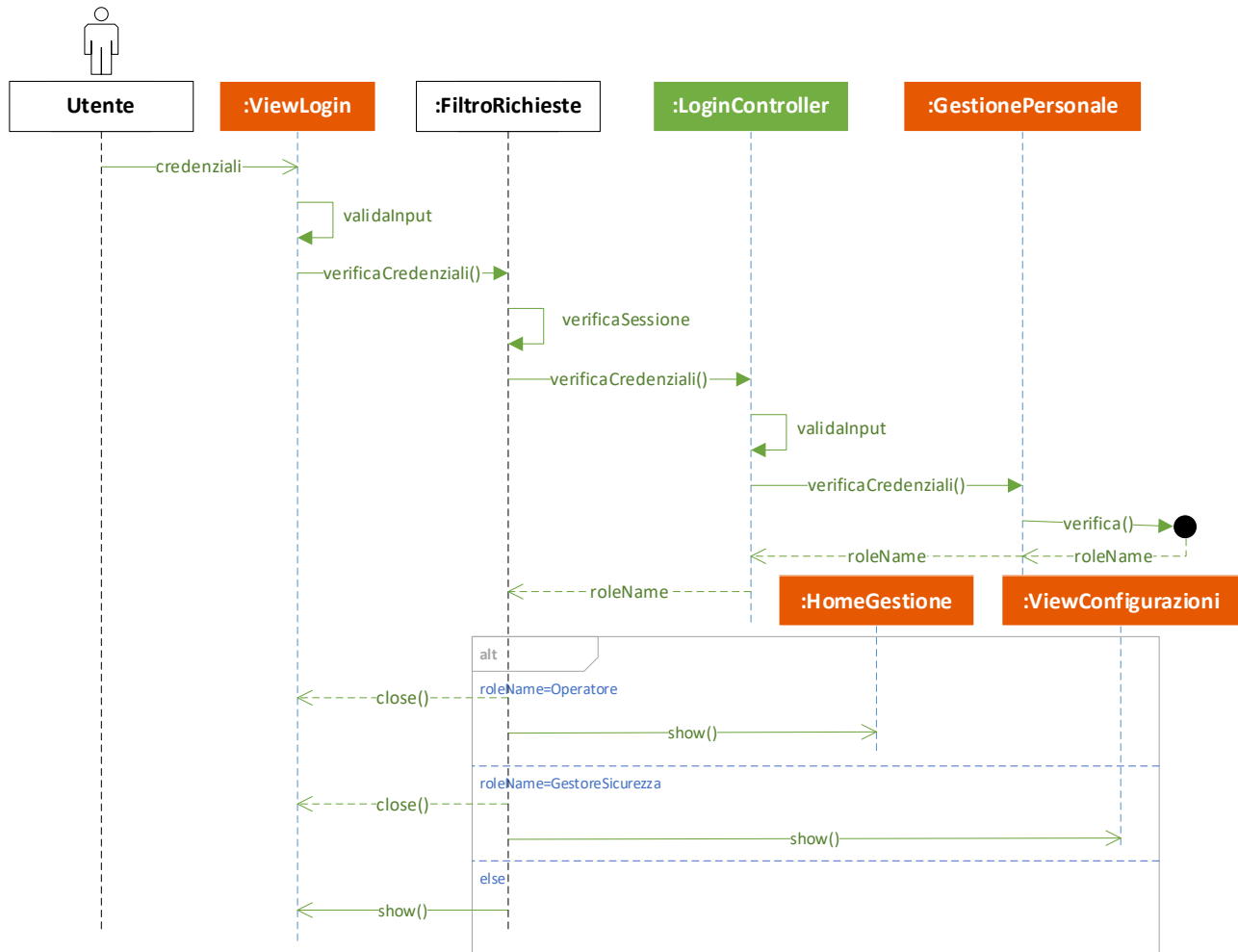
Valgono le stesse considerazioni fatte per il ClientHotel.

Di seguito si riportano gli schemi delle interfacce utente a esclusione dell'interfaccia di Login che è stata mostrata in precedenza.



INTERAZIONE

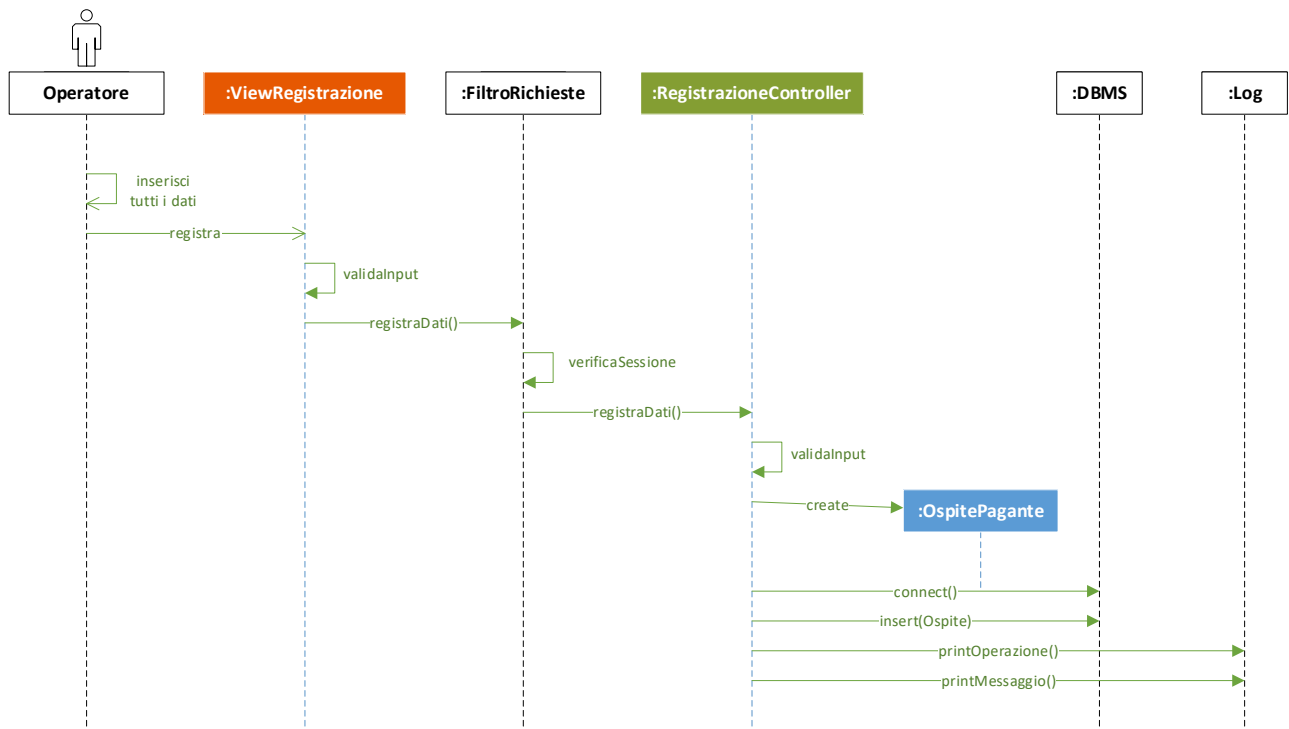
Diagramma di Sequenza: Login – ClientHotel



Tale diagramma mostra il protocollo per il login nel caso del ClientHotel, per gli altri client la procedura è analoga. Da notare la “validazione” degli input sia lato client che lato server al fine di verificare che gli input non siano in un qualche modo variati durante la comunicazione. Ovviamente a questo livello si può verificare solo che le “dimensioni” e il “tipo” dei parametri rispettino le indicazioni date nella fase di analisi del problema, non si è in grado di entrare nel merito della semantica dei parametri, questo aspetto sarà valutato durante l’analisi delle anomalie.

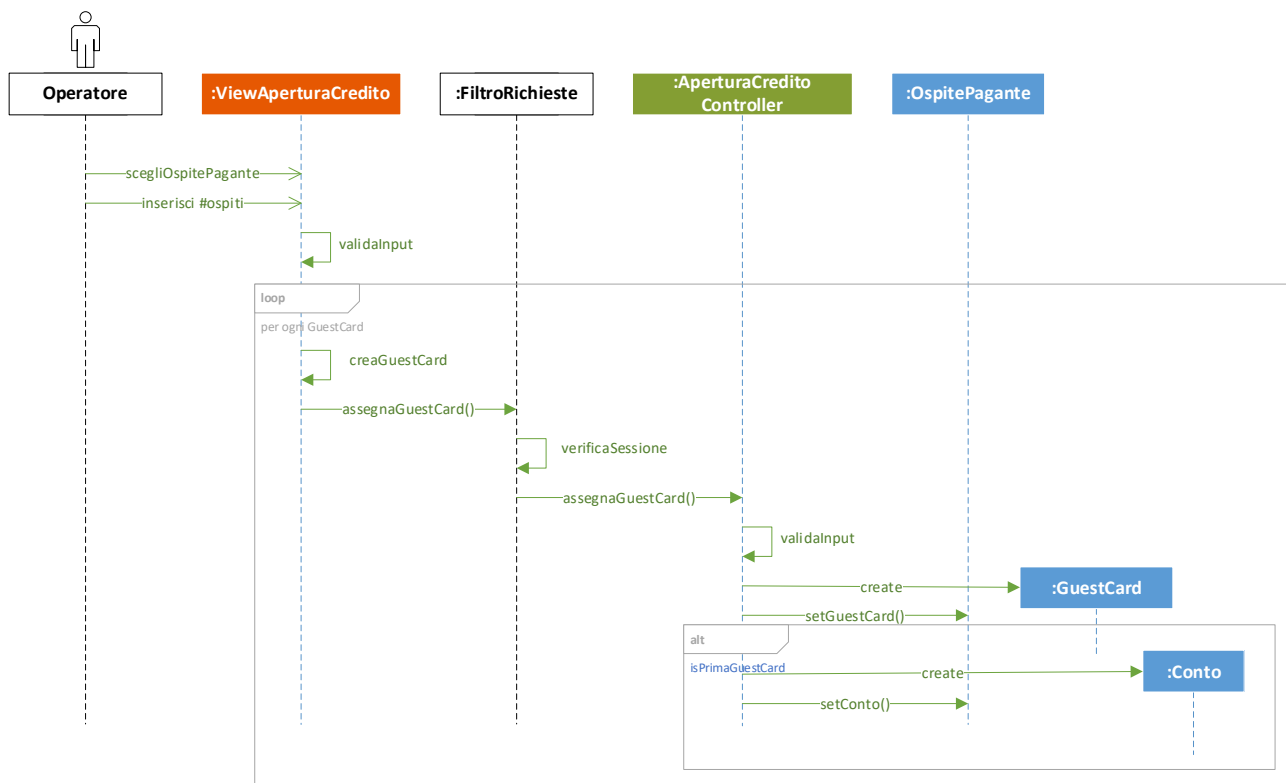
Da notare che nel caso di ClientPuntiVendita, ci sarà una ulteriore condizione di controllo sul “roleName” in modo da mostrare l’interfaccia relativa al Commesso oppure al Gestore Catena. Anche se non è indicato nel diagramma per motivi di spazio, ovviamente ci sarà la scrittura dei log.

Diagramma di Sequenza: Registrazione Nuovo Ospite



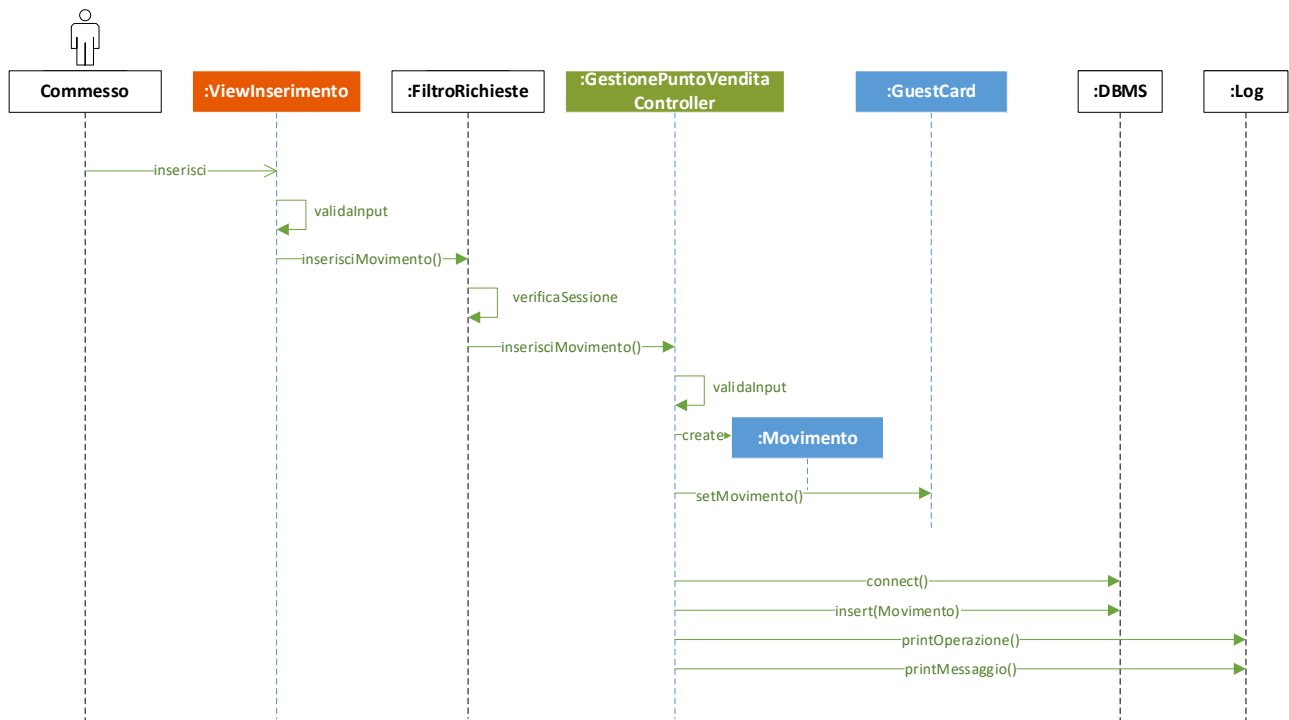
In questa fase, non si evidenziano i ritorni dei messaggi sincroni.

Diagramma di Sequenza: Apertura Credito



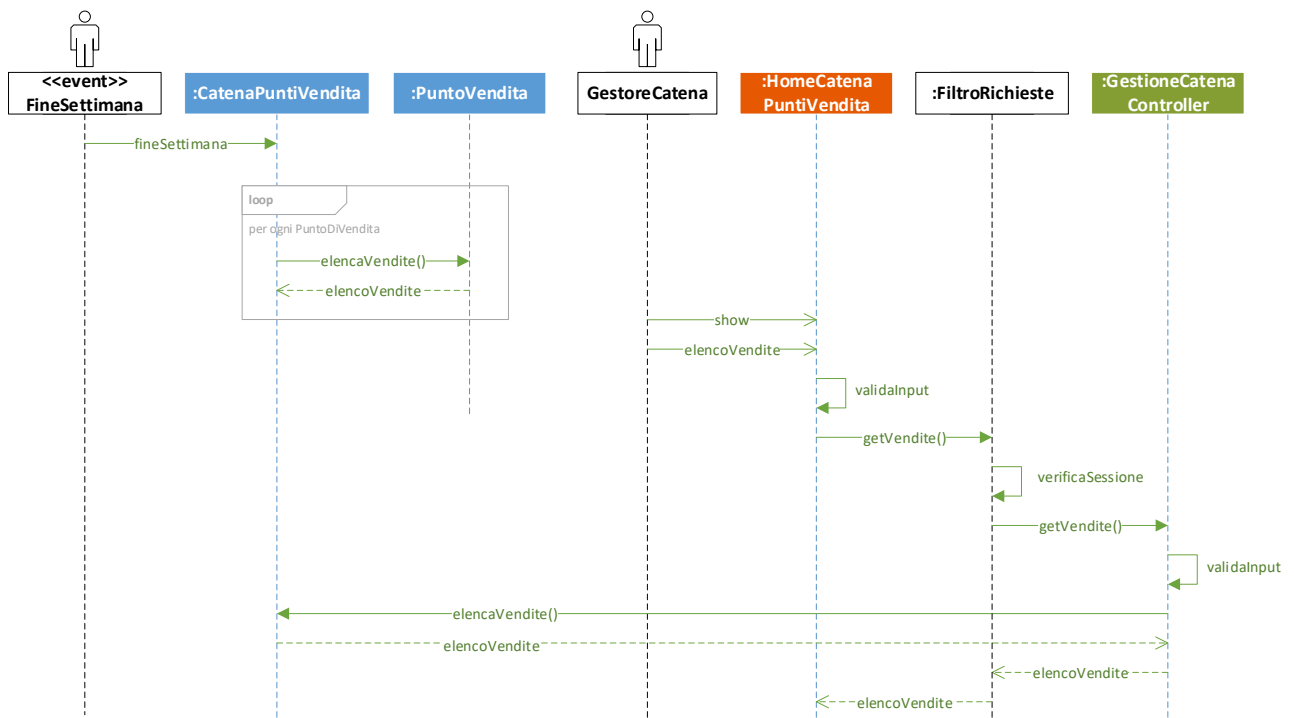
In questo diagramma è bene commentare che “creaGuestCard” è un’operazione interna del ClientHotel. Si è deciso che la scelta migliore per la creazione delle GuestCard sia quella di utilizzare lato client un apposito dispositivo esterno che permetta di “stampare” le GuestCard e inserire le apposite informazioni nel chip di ogni scheda fisica. Ogni scheda è dotata di un apposito identificativo che sarà poi l’id che verrà inviato al server per l’associazione tra OspitePagante e GuestCard. Anche se non è indicato nel diagramma, per motivi di spazio, ovviamente ci sarà la memorizzazione dei dati sul DB e la scrittura dei log.

Diagramma di Sequenza: Inserimento Movimento



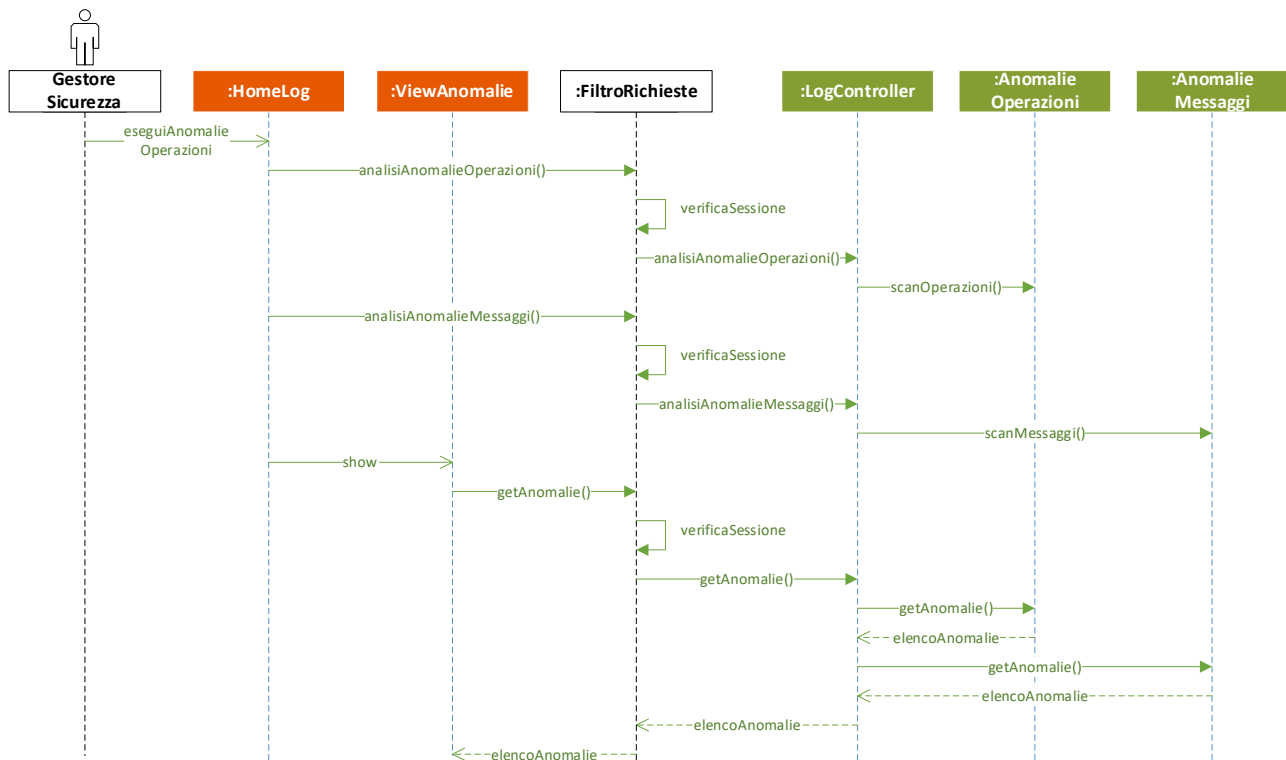
Per ogni PuntoDiVendita dovrà essere acquistato un opportuno Lettore di RFID che permetta di leggere i dati dalla GuestCard fisica per poi essere inviati al server per la memorizzazione

Diagramma di Sequenza: ElencoVendite



Anche se non indicata nel diagramma per motivi di spazio, ovviamente ci sarà la memorizzazione dei dati sul DB e la scrittura dei log.

Diagramma di Sequenza: Anomalie



Anche se non indicata nel diagramma per motivi di spazio, ovviamente ci sarà la memorizzazione dei dati sul DB e la scrittura dei log.

Diagramma di Sequenza: Scrittura Log Client

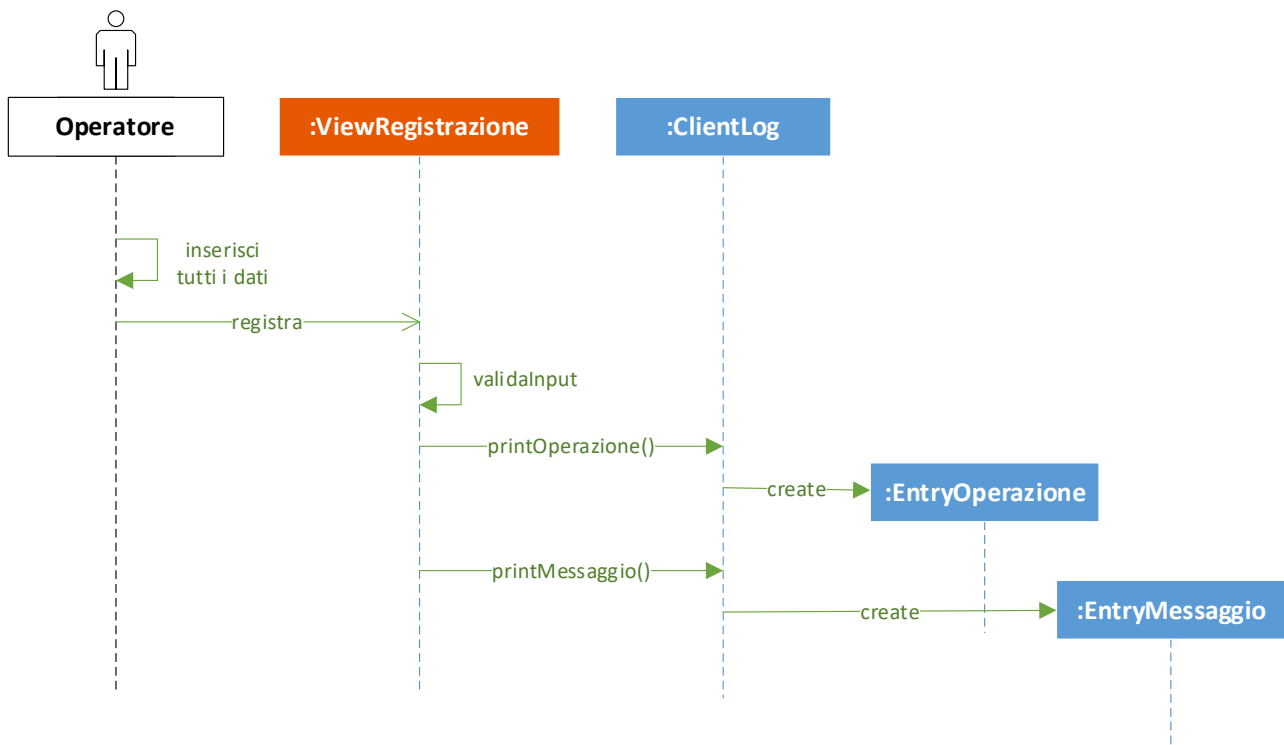


Diagramma di Sequenza: Client-Server Log

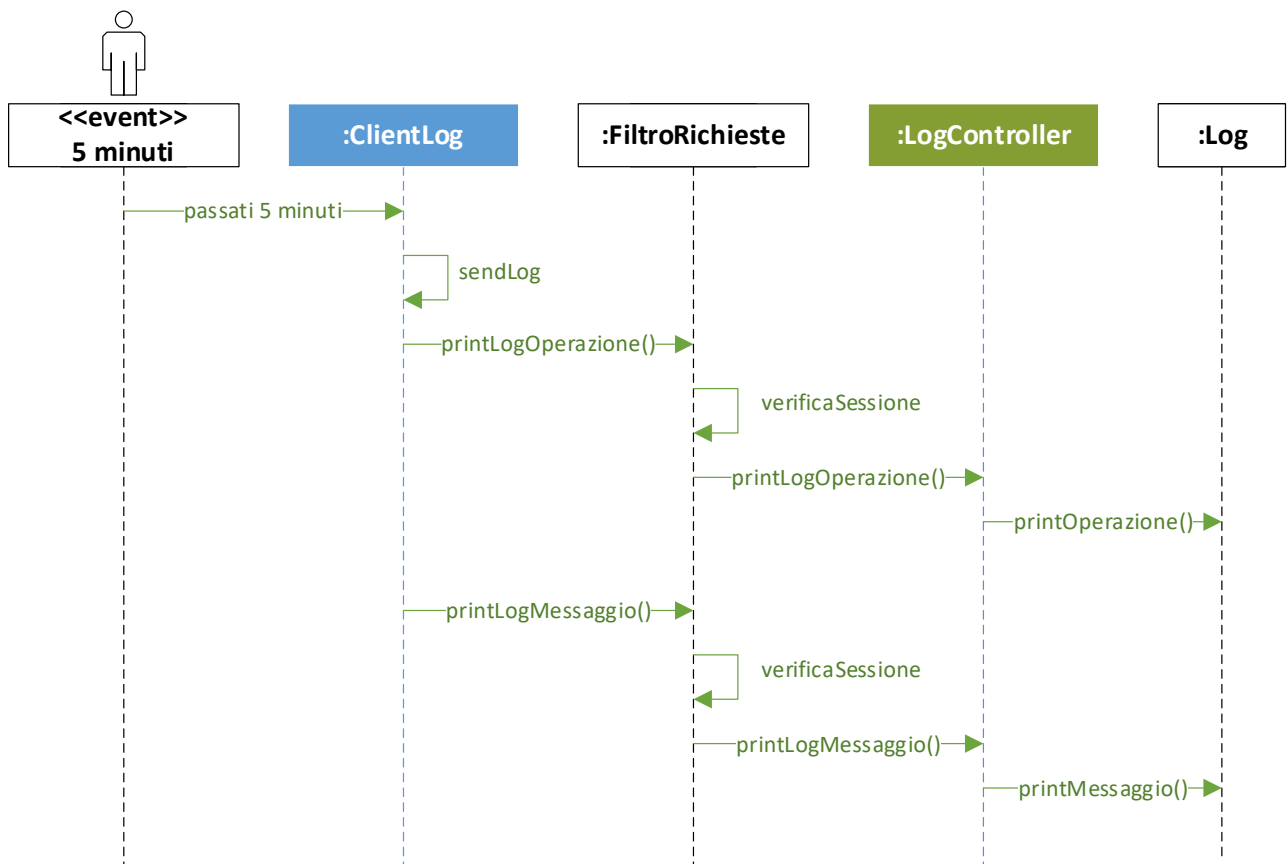
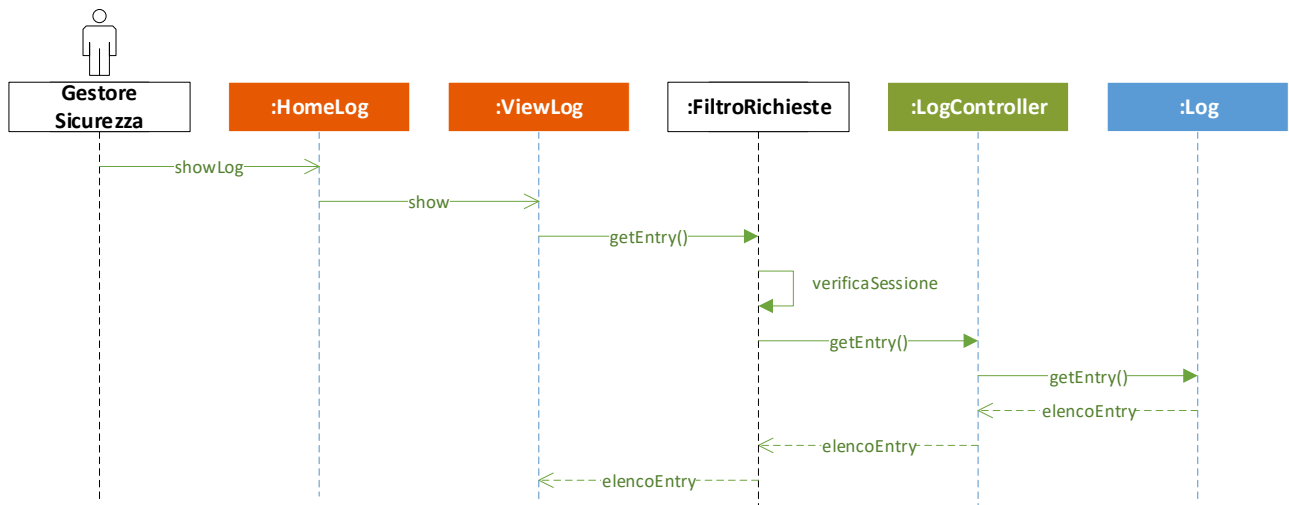


Diagramma di Sequenza: Visualizzazione Log



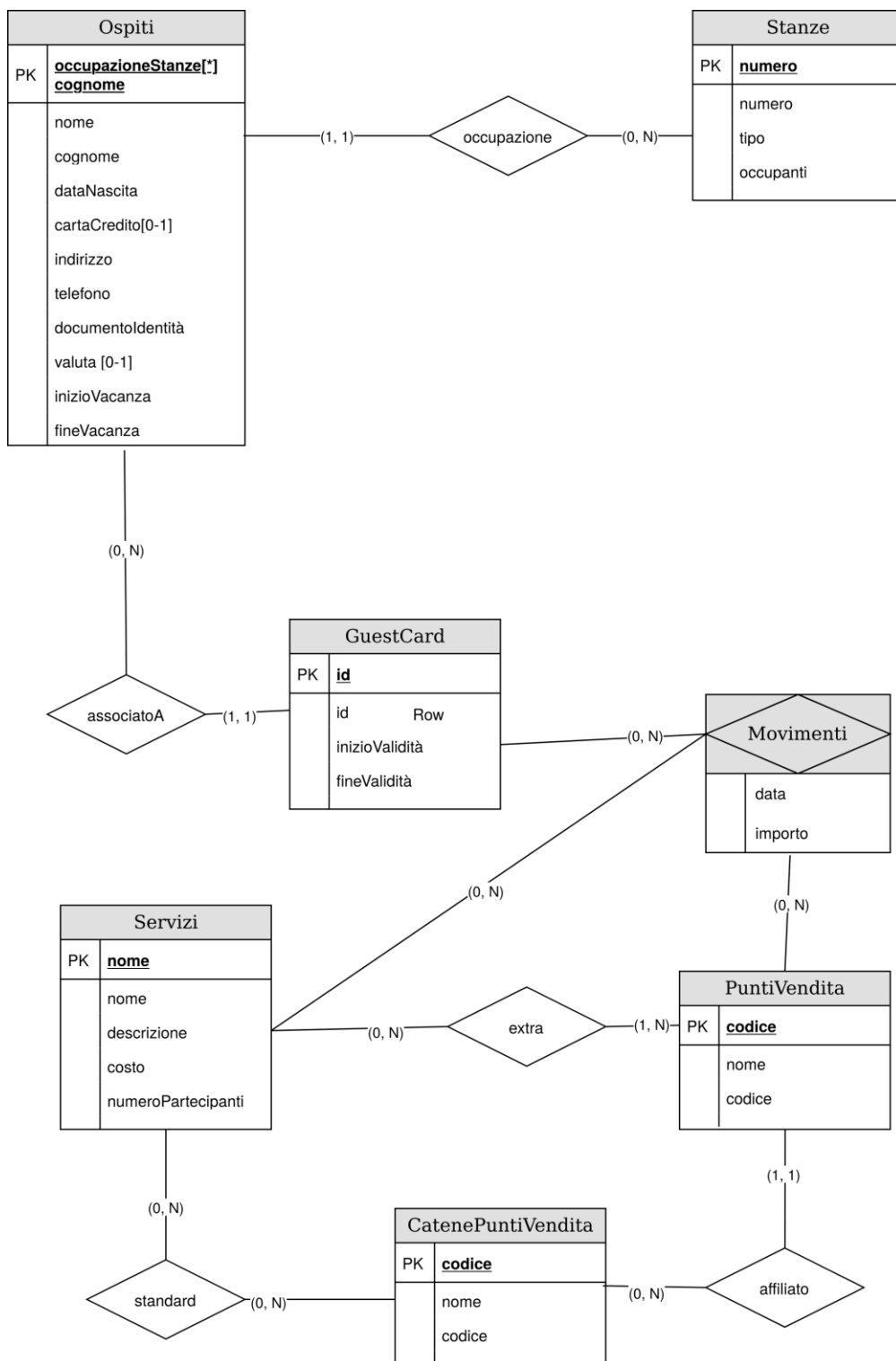
Anche se non indicata nel diagramma per motivi di spazio, ovviamente ci sarà la memorizzazione dei dati sul DB e la scrittura dei log.

COMPORTAMENTO

Non sono sviluppati ulteriori diagrammi di stato. Quello fatto in fase di analisi del problema risulta sufficiente.

PROGETTAZIONE DELLA PERSISTENZA

Diagramma E-R:



La presenza di un ciclo tra Servizi, PuntiVendita (e CatenePuntiVendita) implica un vincolo sull'associazione movimento, in quanto un movimento può riferirsi solamente a un Servizio fornito da un PuntoVendita, o tra quelli standard della CatenaPuntiVendita associata, o tra quelli extra.

Formato File Log

- Formato file per Log delle operazioni
DataOra - operazione - esecutore
- Formato file per Log dei messaggi
DataOra - messaggio protetto - invio/ricezione - autore

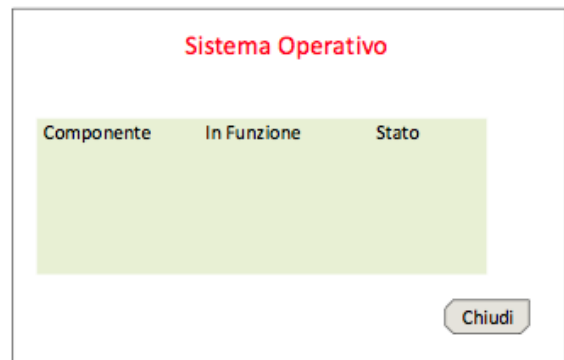
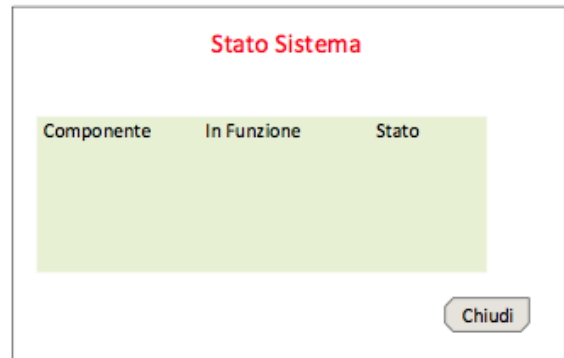
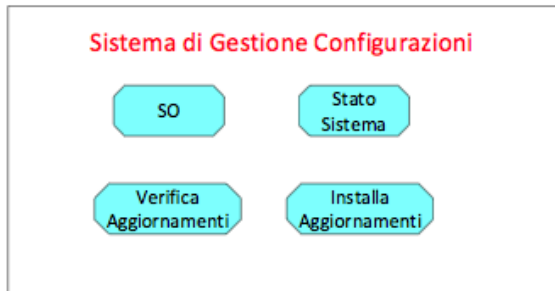
PROGETTAZIONE DEL COLLAUDO

Vanno completati i test per tutte le classi.

PROGETTAZIONE PER IL DEPLOYMENT

Come già accennato in precedenza, è stata inserita una ulteriore interfaccia che permette di visionare le configurazioni solo del lato client del sistema.

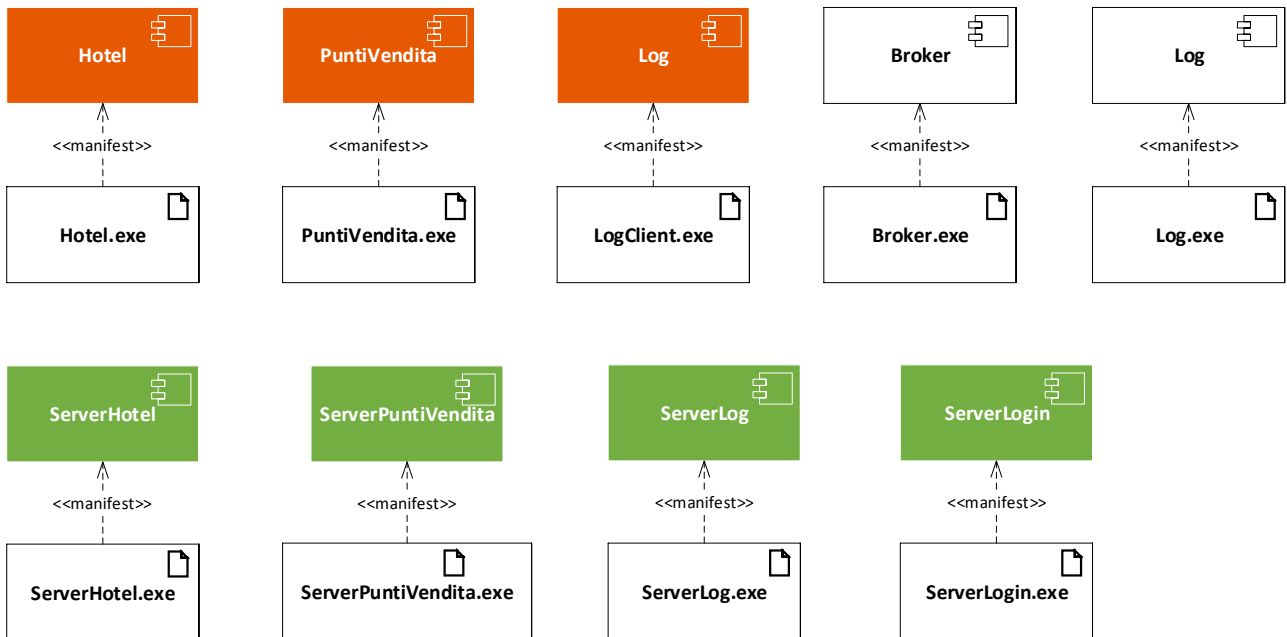
Le interfacce previste lato client sono le seguenti:



Lato server: i server dovranno essere installati su macchine all'interno di una rete privata opportunamente protetta da un firewall a cifratura di pacchetti. L'unico punto di contatto verso l'esterno è il Broker.

DEPLOYMENT

Artefatti



Deployment Type-Level

