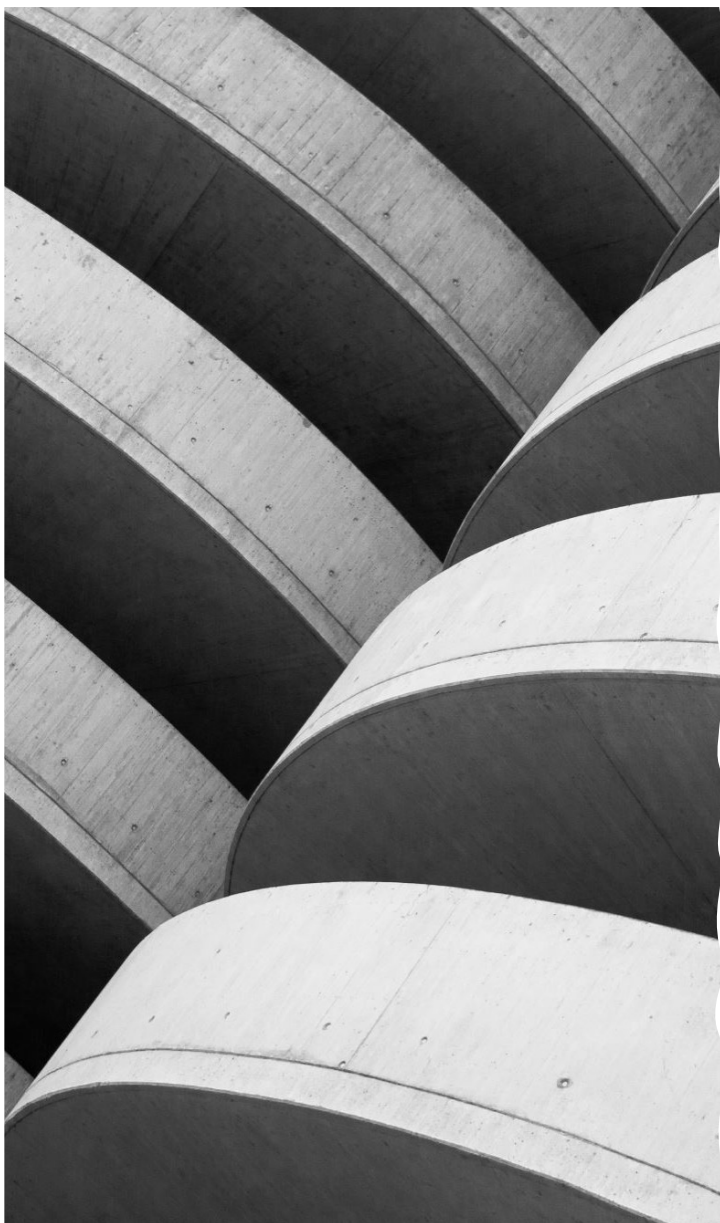


ESERCITAZIONE SU
ARCHITETTURE SOFTWARE

Prof. Mariacarla Staffa



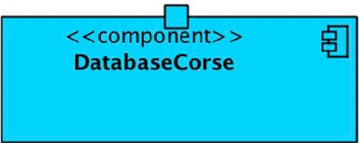
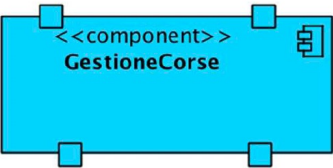
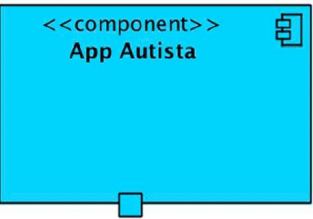
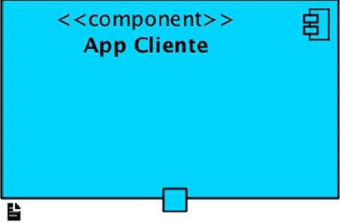
ESERCIZI SU ARCHITETTURE SW

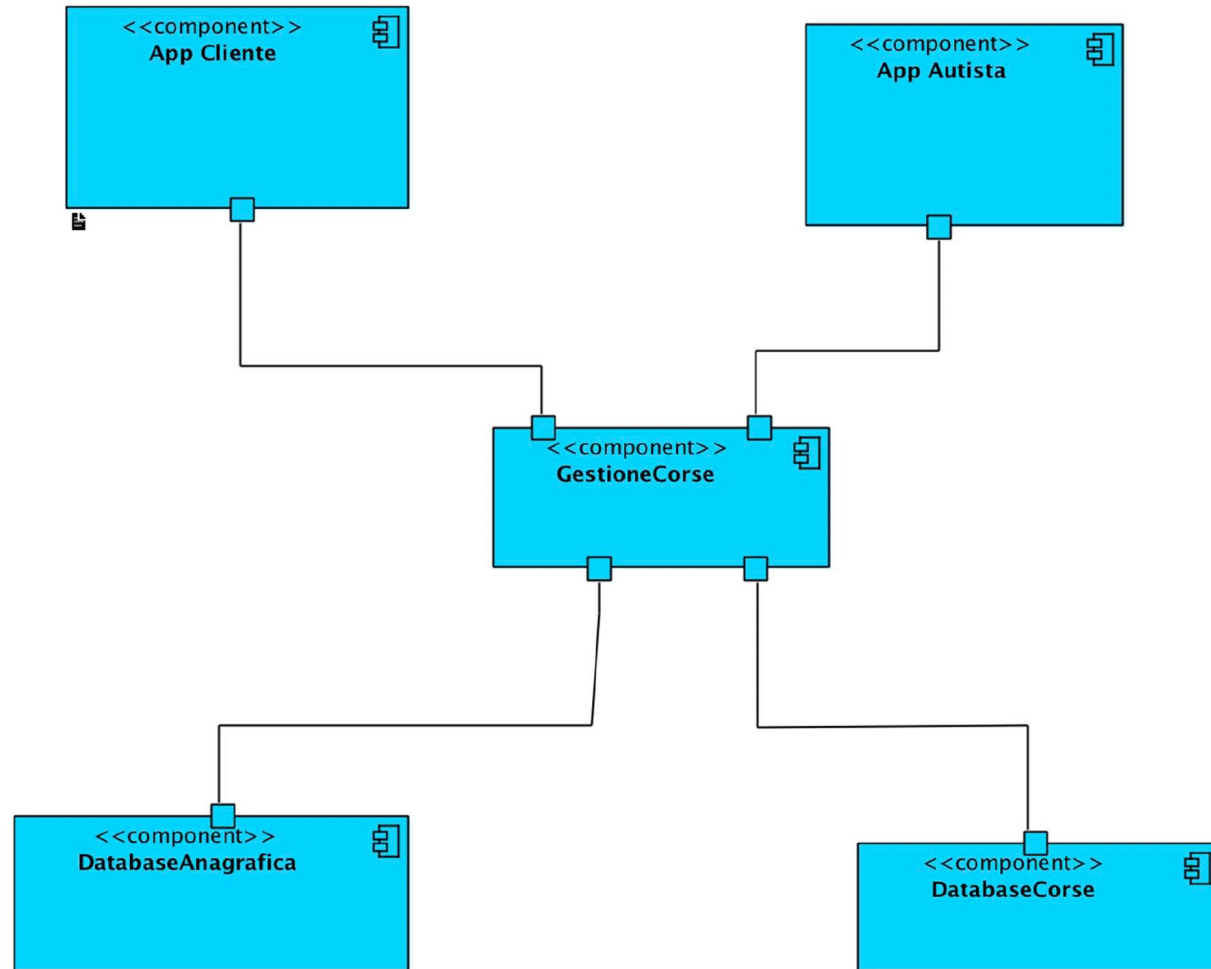
ES1: REBU

Si assuma che sia stata definita un'architettura con i seguenti componenti:

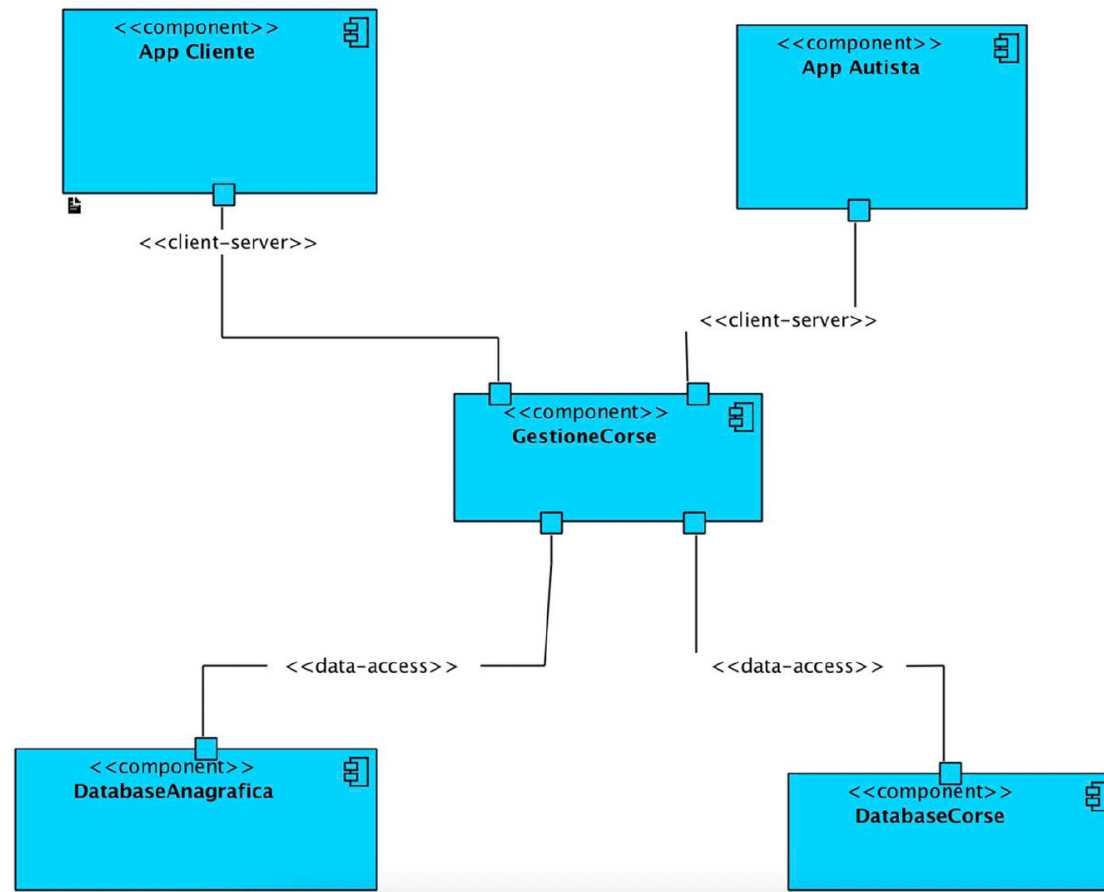
- app_cliente (che implementa l'app usata dal cliente)
- sistema_prenotazione (che implementa la business logic del sistema)
- db_anagrafica (che raccoglie dati anagrafici su autisti e clienti)
- db_corse (che raccoglie dati su prenotazioni, corse assegnate, corse effettuate)
- app_autista (che implementa l'app usata dall'autista)

Si disegni il diagramma C&C, specificando gli stereotipi corrispondenti al tipo di interazione che si immagina sui connettori.

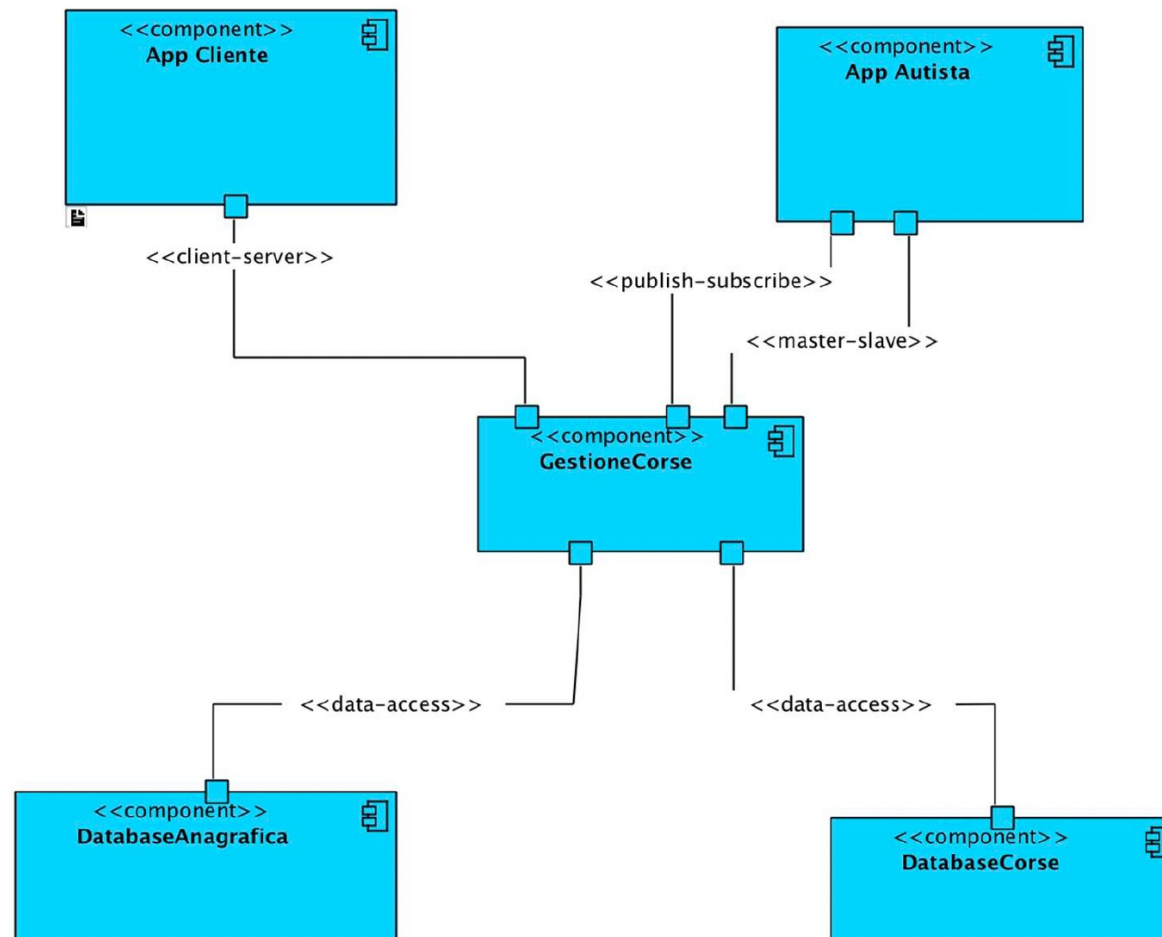




Prima soluzione



Altra soluzione

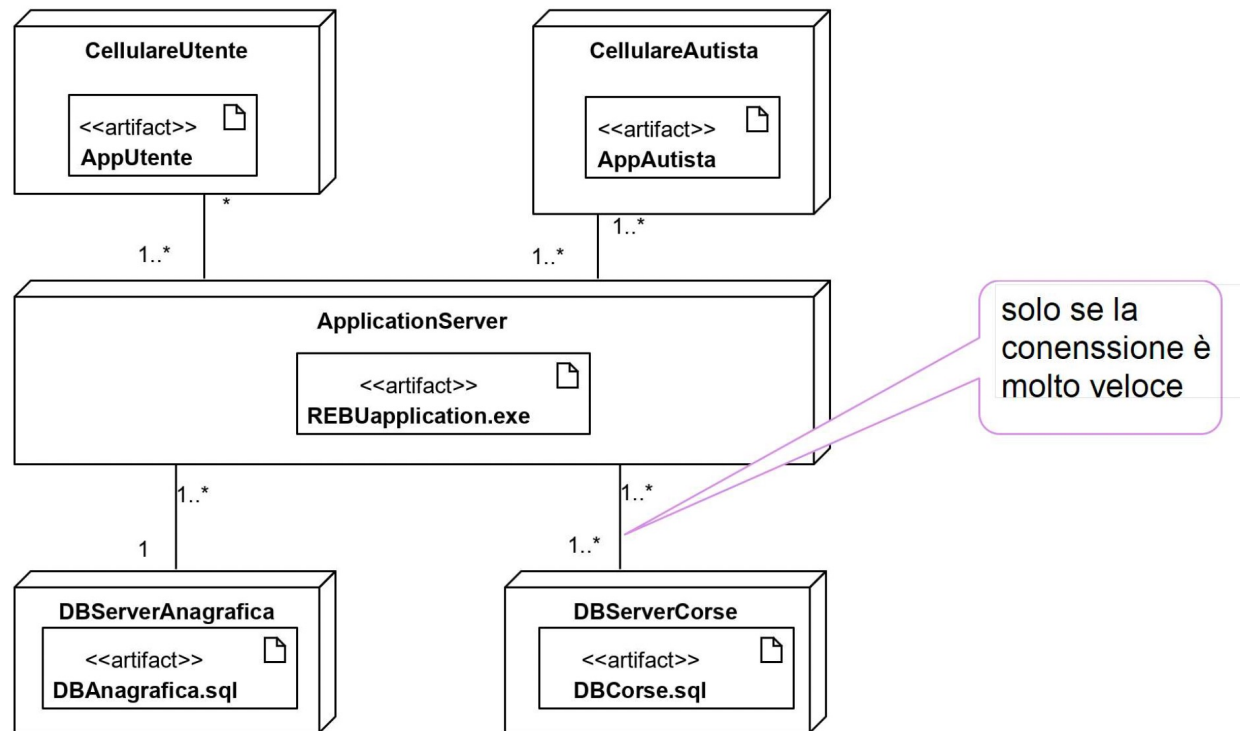


ES2: REBU

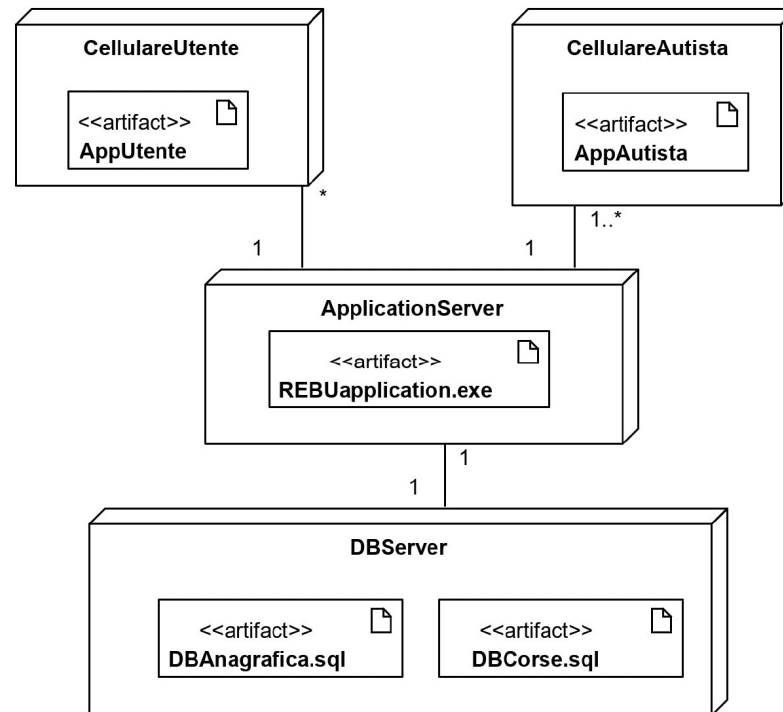
Sotto le stesse assunzioni dell'esercizio 1, e assumendo un artefatto (omonimo) per componente, si disegni un diagramma di dislocazione (deployment) per il sistema.

Si assuma, che il db_anagrafica venga aggiornato poco frequentemente (solo quando si iscrivono nuovi autisti e nuovi clienti), e che venga interrogato con query semplici, mentre il db_corse sia soggetto ad aggiornamenti più frequenti, e debba eseguire query complesse (per esempio, per verificare quali autisti saranno liberi a un dato orario occorre incrociare le ore di inizio e fine servizio di tutte le corse attualmente prenotate e delle altre richieste pendenti).

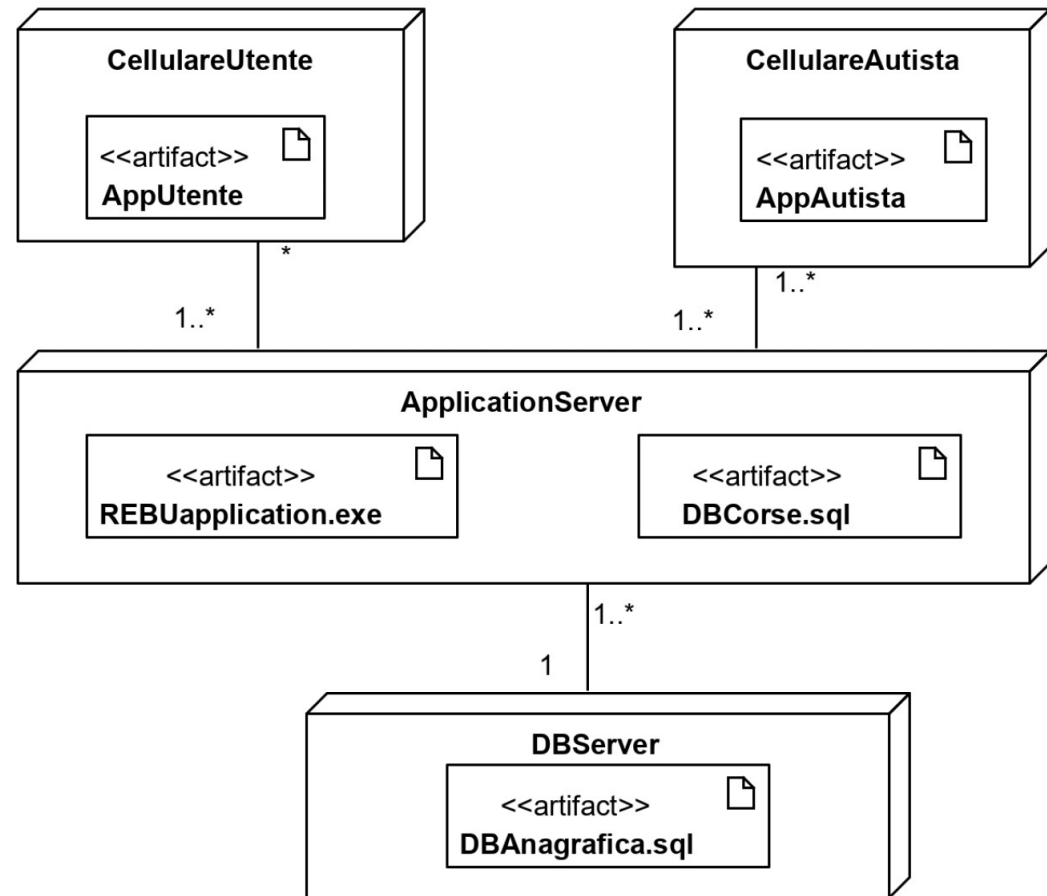
PRIMA SOLUZIONE



SECONDA SOLUZIONE (QUASI MINIMA)



TERZA SOLUZIONE



ES3: REBU, ESTENSIONE AEROPORTI

- Nuova feature: **associare un numero di volo**, in caso di corse da/per un aeroporto.
- **Verso un aeroporto**: al momento della richiesta, il cliente specifica il numero del volo e con quanto anticipo, in minuti, desidera arrivare in aeroporto rispetto alla partenza del volo.
 - Il sistema si interfaccia con i servizi dell'aeroporto e ottiene l'orario di partenza. Si interfaccia quindi con un servizio di calcolo di percorso stradale (tipo google maps) per ottenere i tempi di percorrenza tra la posizione di ritrovo e l'aeroporto, e stabilire un orario di partenza per la corsa. A questo punto procede come con una normale prenotazione.
- Nel caso di spostamenti **da un aeroporto**, al momento della richiesta, il cliente specifica il numero del volo e se avrà del bagaglio imbarcato da attendere dopo l'atterraggio. Il sistema si interfaccia con i servizi dell'aeroporto, ottiene l'orario di atterraggio del volo, stabilisce un orario di partenza per la corsa, e procede come con una normale prenotazione.
- Nelle ultima due ore prima dell'inizio della corsa, REBU monitora eventuali ritardi dei voli. In caso di ritardo superiore ai 15 minuti, ricalcola l'orario di partenza della corsa, cerca un nuovo autista per la corsa ritardata (allo stesso prezzo della corsa già pattuita con l'utente), e cancella la precedente. Quindi informa l'utente del nuovo orario.

EX3: REBU, ESTENSIONE AEROPORTI

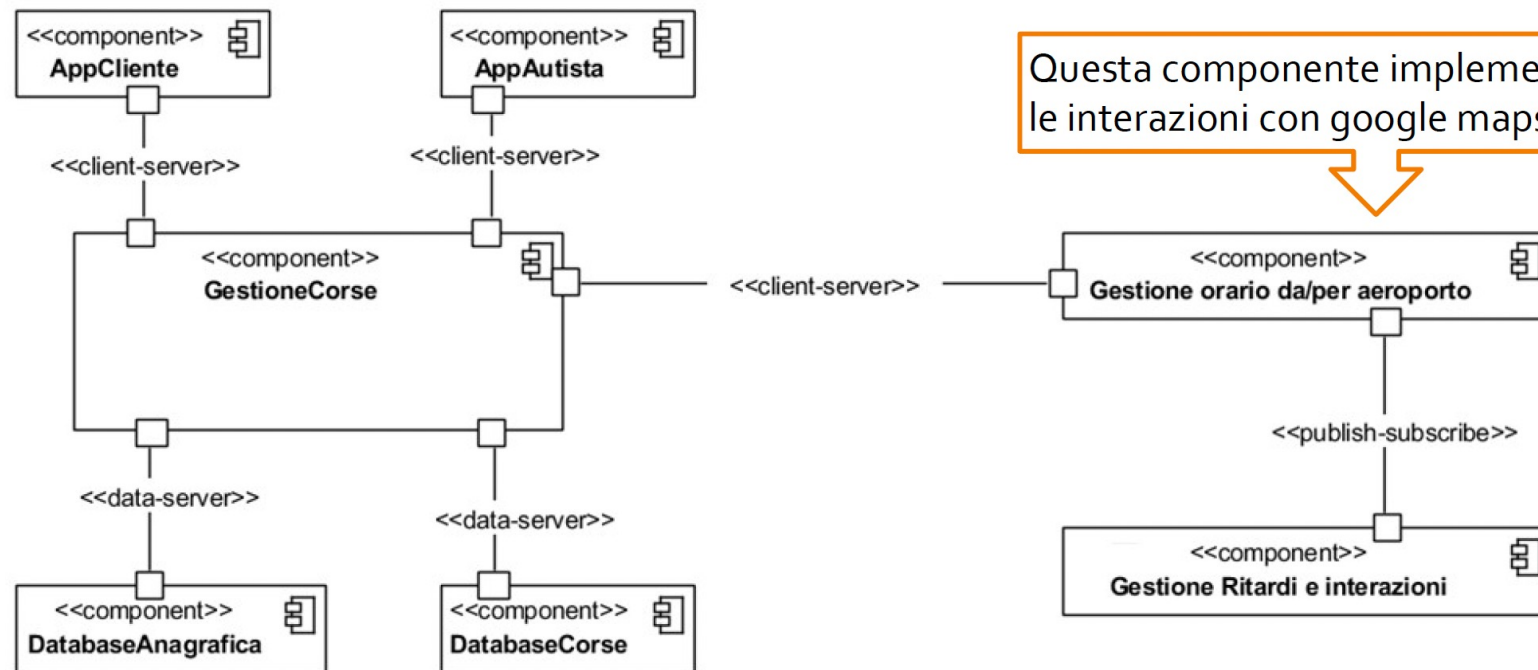
La componente **GestioneRitardi** si occupa di monitorare eventuali ritardi dei voli e, se necessario, modificare la corsa.

I servizi offerti dai vari aeroporti non sono uniformi: alcuni offrono la possibilità di abbonarsi alle informazioni su un volo e informano REBU di eventuali ritardi (**modalità PUSH**), altri offrono solo un servizio di informazioni domanda/risposta (**modalità PULL**). Nella modalita' PULL si ipotizzi che il sistema REBU si aggiorni sui voli ogni 20 minuti.

Dare la vista C&C e descrivere, con un diagramma di sequenza, le interazioni tra GestioneRitardi e l'aeroporto, nei due casi.

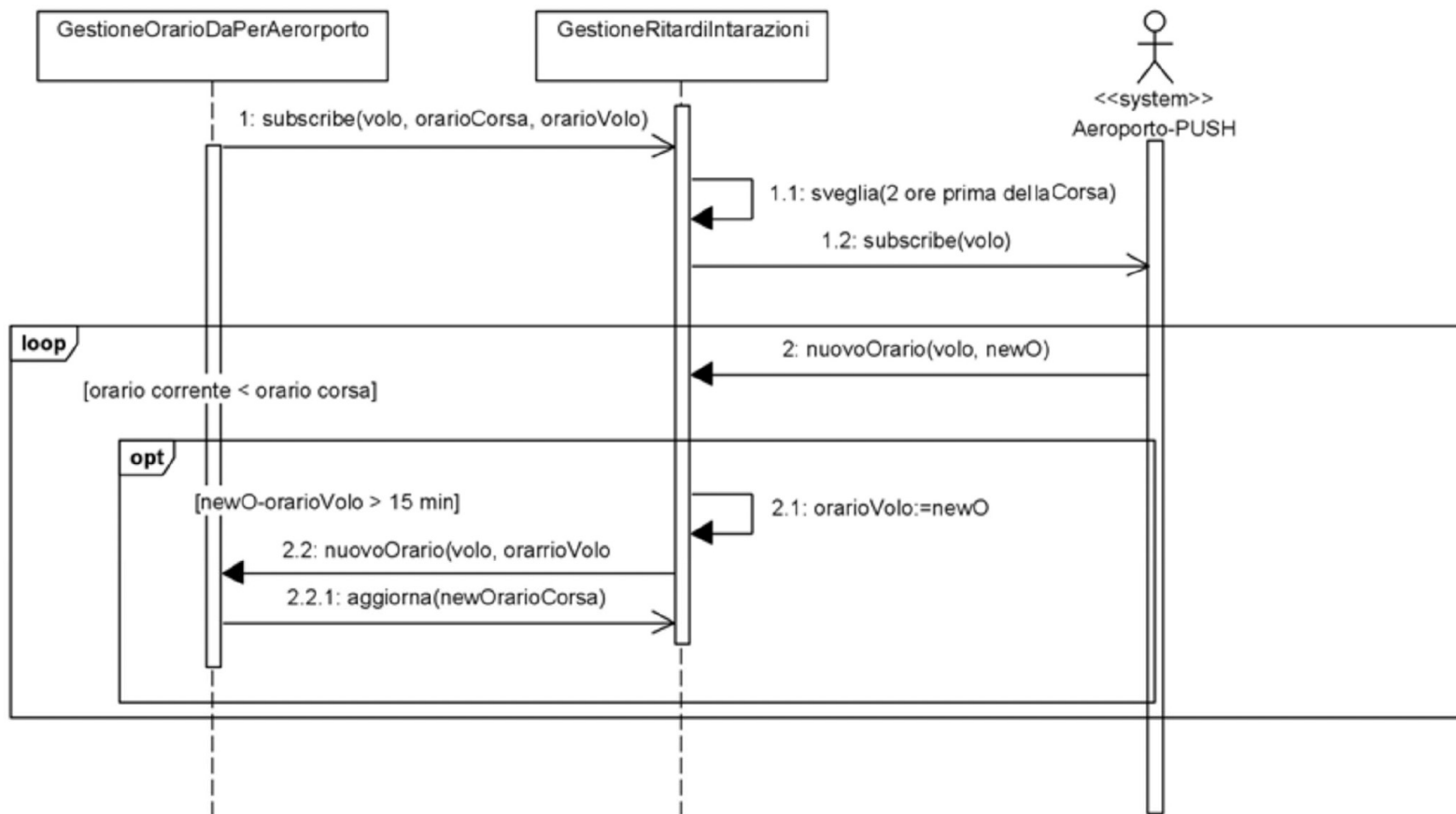
VISTA C&C ???

VISTA C&C

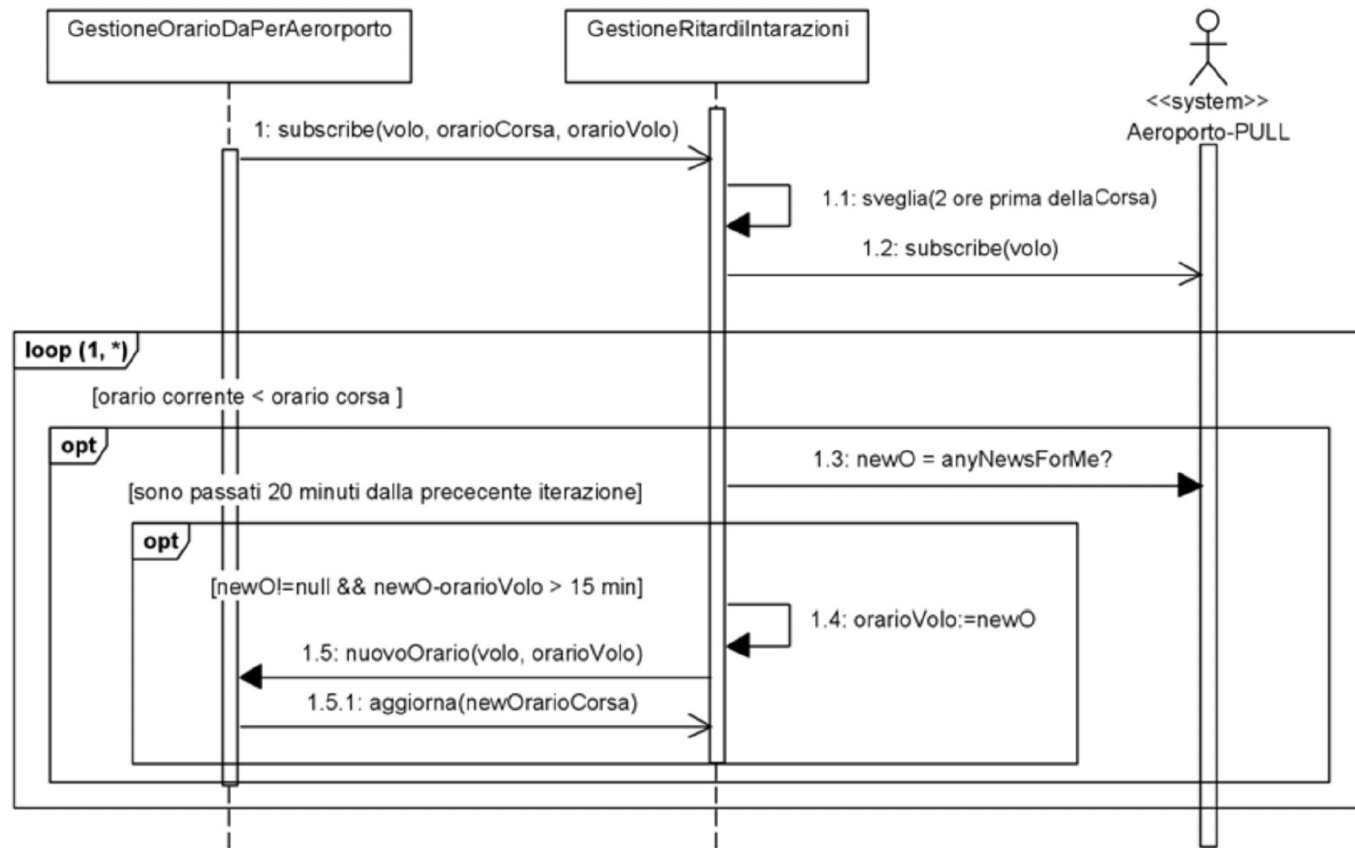


PUSH&PULL CON I SEQUENCE DIAGRAM???

PUSH



PULL



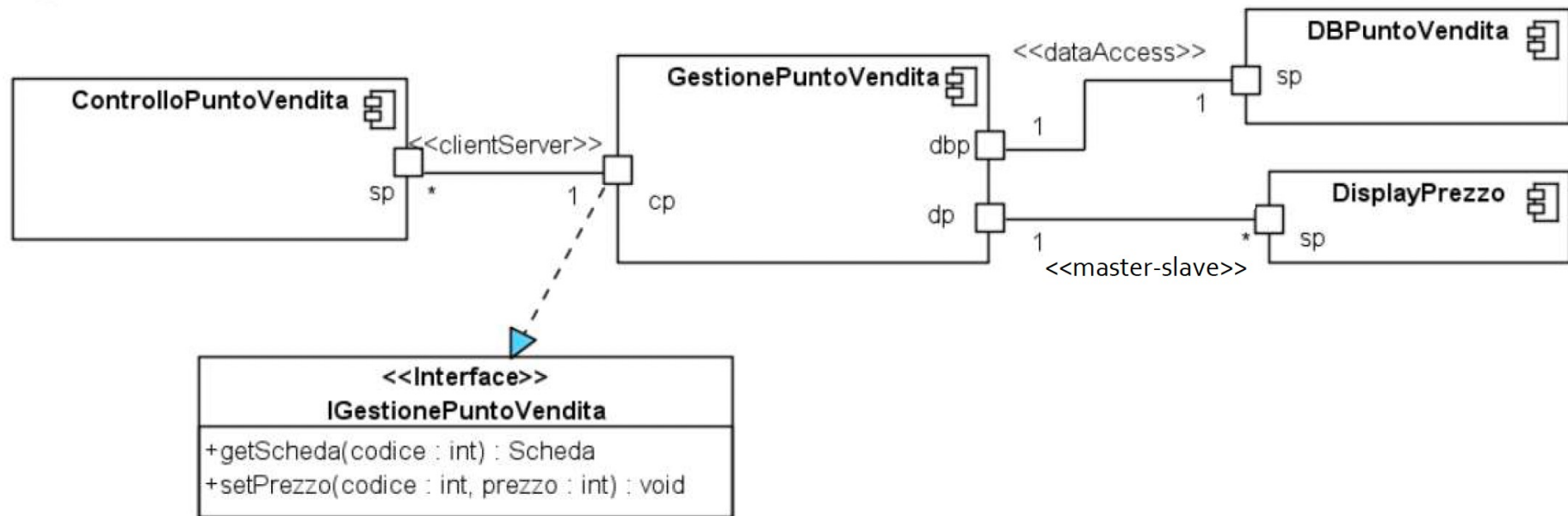
ES. 4. AGGIORNAMENTO PREZZI

Fornire la vista C&C considerando la seguente descrizione per le componenti, e mostrare l'interfaccia che GestionePuntoVendita deve offrire per realizzare il caso d'uso.

Componente	Responsabilità
DBPuntoVendita	Memorizza i dati sui prodotti in vendita.
GestionePuntoVendita	Gestisce l'aggiornamento dei prezzi del punto vendita.
ControlloPuntoVendita	Permette al ResponsabilePuntoVendita di accedere al sistema per aggiornare i prezzi del punto vendita.
DisplayPrezzo	Gestisce la visualizzazione dei prezzi sugli scaffali.

AGGIORNAMENTO PREZZI: VISTA C&C???

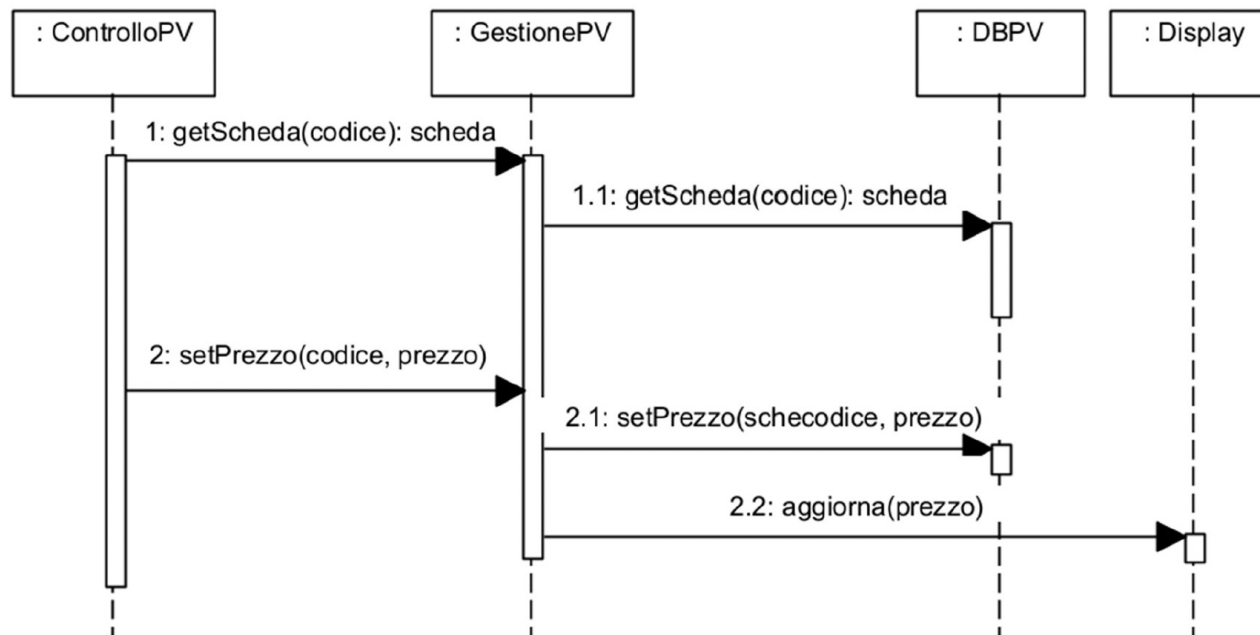
AGGIORNAMENTO PREZZI: VISTA C&C



AGGIORNAMENTO PREZZI: DIAGRAMMA DELLE SEQUENZE???

AGGIORNAMENTO PREZZI: DIAGRAMMA DELLE SEQUENZE

Diagramma di sequenza che mostra come le componenti e i connettori individuati realizzano il caso d'uso



ES. 5 COPS & ROBBERS

Scopo del progetto è realizzare Cops And Robbers un semplice gioco multiplayer. In Cops And Robbers il mondo virtuale è popolato da un certo numero Ladri e da un insieme di Guardie.

Per l'amministrazione del gioco, CopsAndRobbersAdmin:

- 2.1.deve ricevere le richieste di connessione al gioco da parte dei giocatori;
- 2.2.quando ha ricevuto esattamente k richieste, deve
 - 2.2.1.inviare ai giocatori la posizione degli oggetti all'interno della mappa, e
 - 2.2.2.notificare a ogni giocatore che può iniziare il gioco;
- 2.3.deve inviare ai giocatori che ne fanno richiesta la lista dei giocatori che partecipano al gioco, con la loro energia;
- 2.4.deve notificare la vincita/perdita ai giocatori.

Per permettere lo svolgimento del gioco, PeerCopsAndRobbers:

- 3.1.deve dar corso alle richieste del giocatore, e notificargliene l'esito;
- 3.2.deve muovere le guardie assegnategli.

ESERCIZIO

Fornire:

- *1. una vista C&C*
- *2. un diagramma delle attività del giocatore*
- *3. un diagramma degli stati del controllore*
- *4. una vista strutturale dell'architettura*

VISTA C&C

Visual Paradigm for UML Standard Edition (University of Pisa)

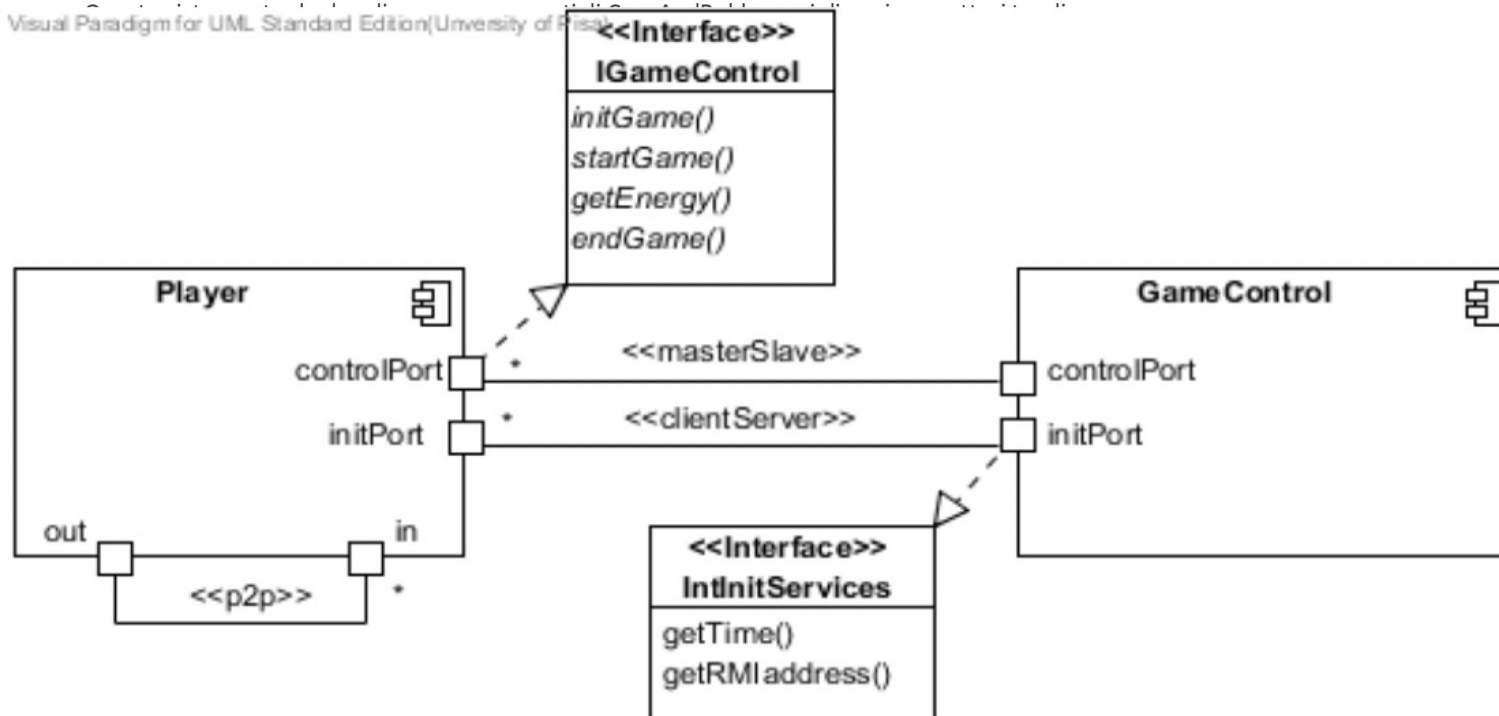


DIAGRAMMA DELLE ATTIVITA' DEL GIOCATORE

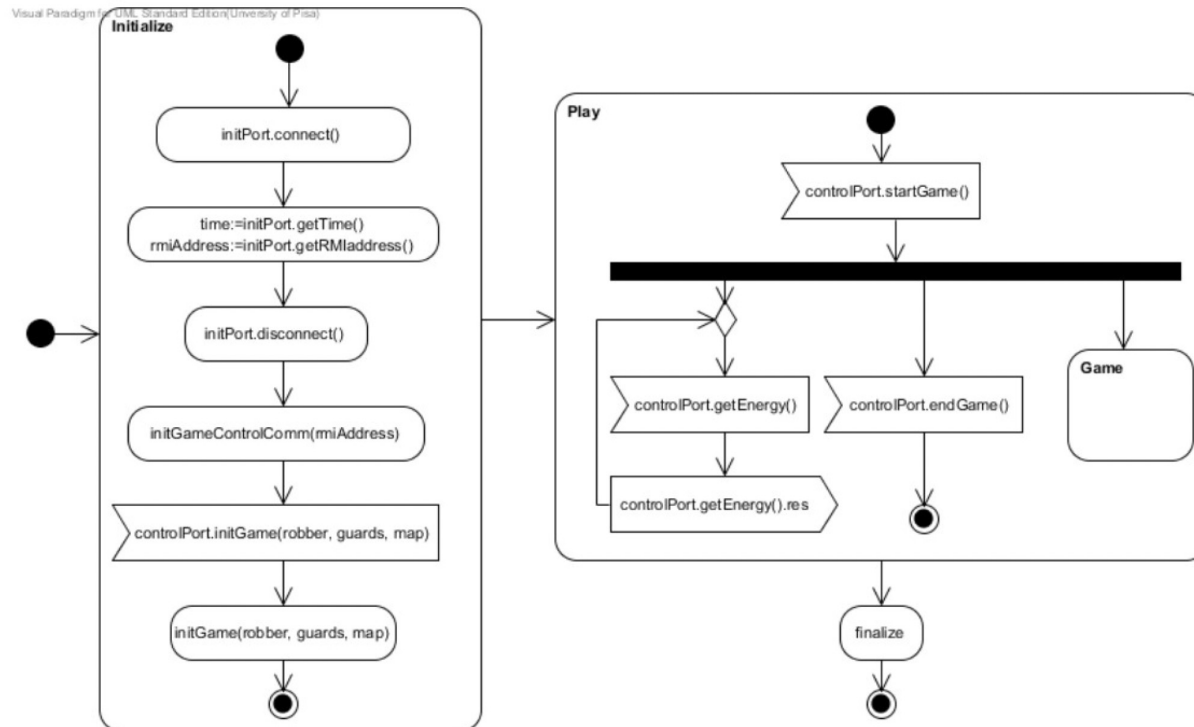
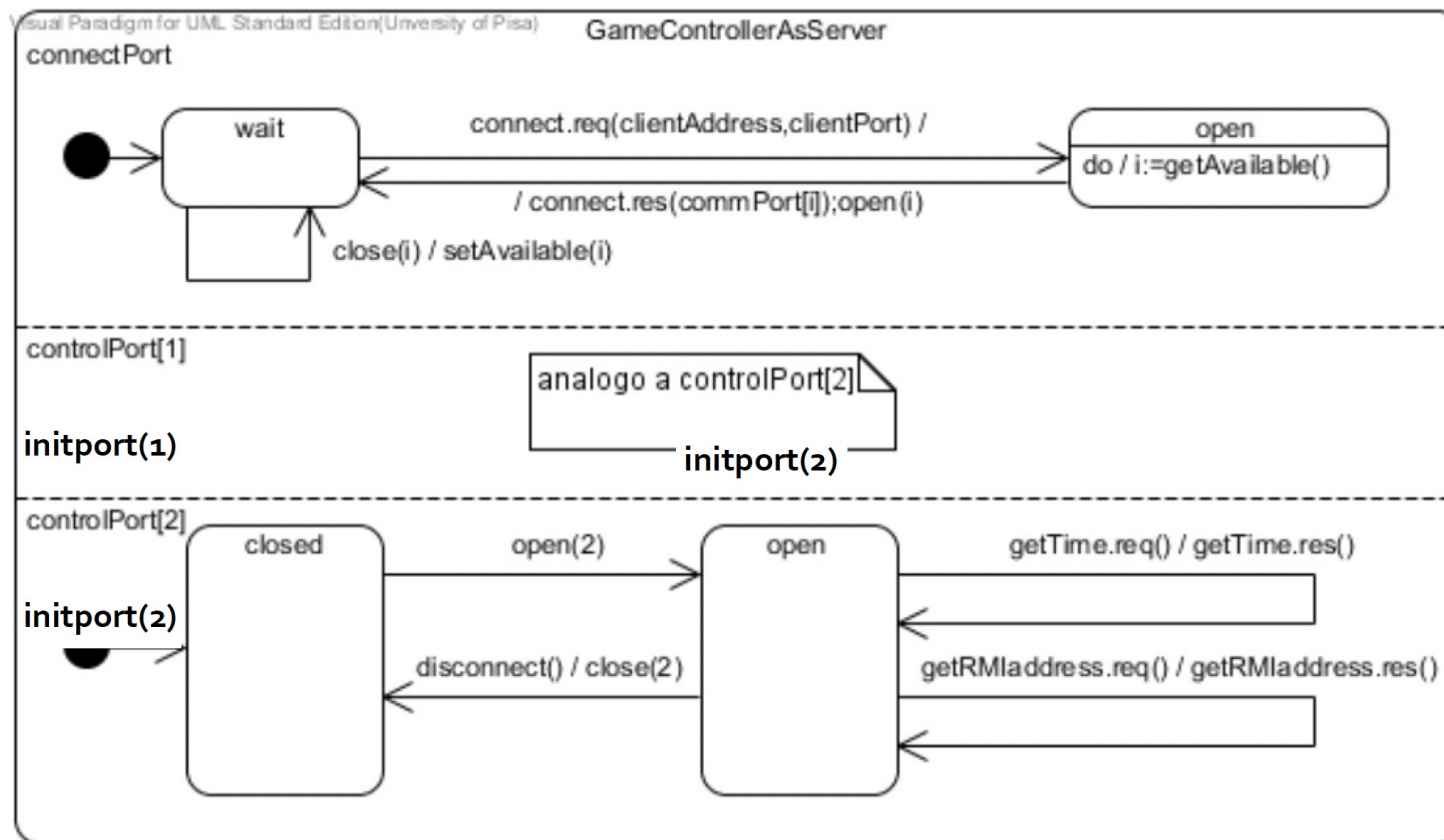


DIAGRAMMA DEGLI STATI DEL CONTROLLORE



VISTA STRUTTURALE DELL'ARCHITETTURA

