

ANALISI TECNICA

Nome Progetto	Device Shop
Cliente	Device Shop S.R.L.
Versione Documento	1.0
Autore	Daniele Cocuzza
Data Creazione	15/02/2023

Business Requirement

Creare un sistema a microservizi client-server, in cui il client contatta per il server per la configurazione di componenti PC. Mettere delle regole per la configurazione del PC: il sistema non deve permettere di aggiungere dispositivi incompatibili. Il sistema deve tenere la lista degli acquisti del client. Implementare un sistema di Circuit Breaker in modo da evitare le chiamate quando il sistema non funziona bene. L'utente deve visualizzare i dispositivi disponibili, aggiungere un dispositivo al carrello, visualizzare il suo carrello e cercare un dispositivo.

Tecnologie di Implementazione

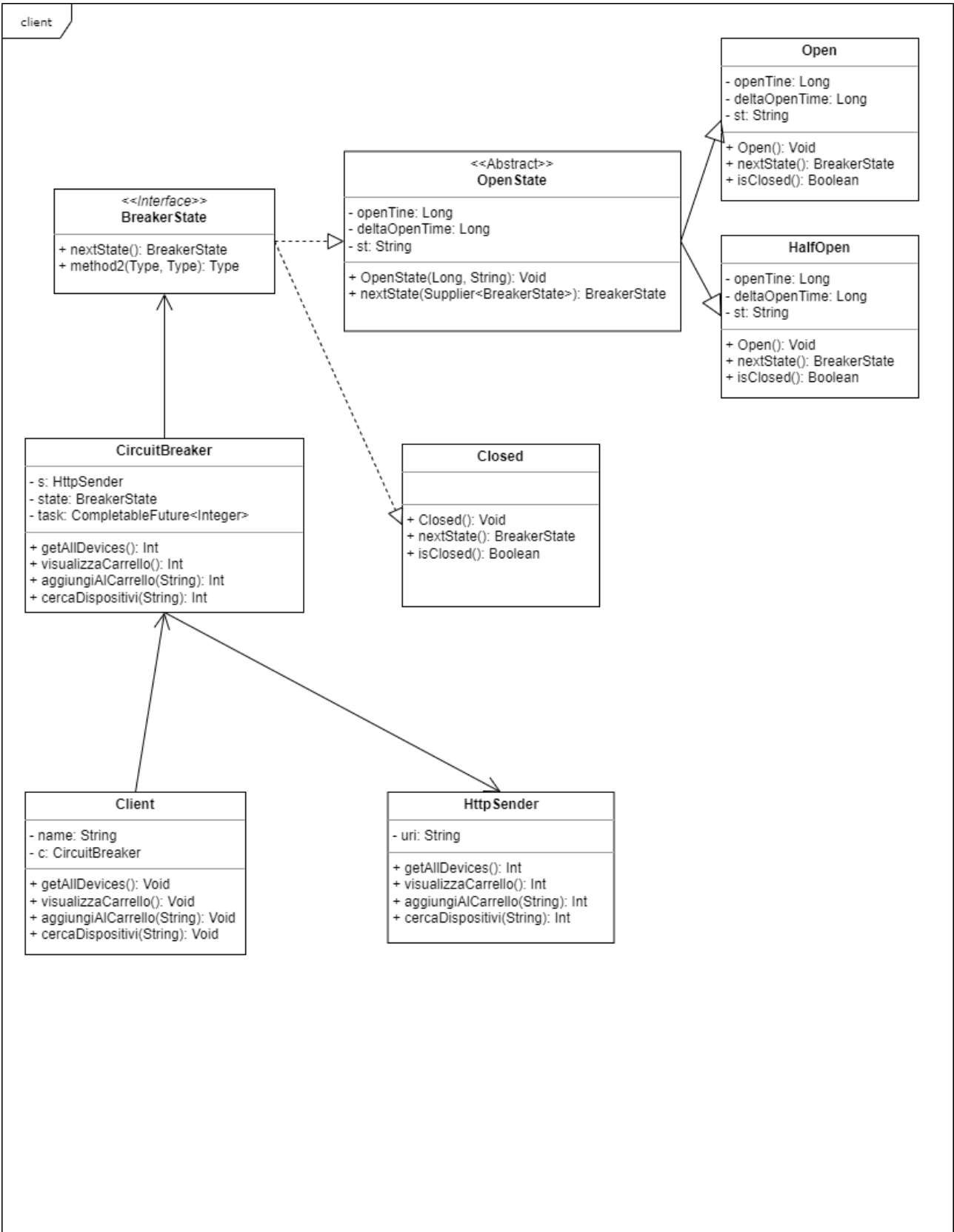
Linguaggio	Java 19
Framework	Spring Boot
Comunicazione HTTP	API RESTful

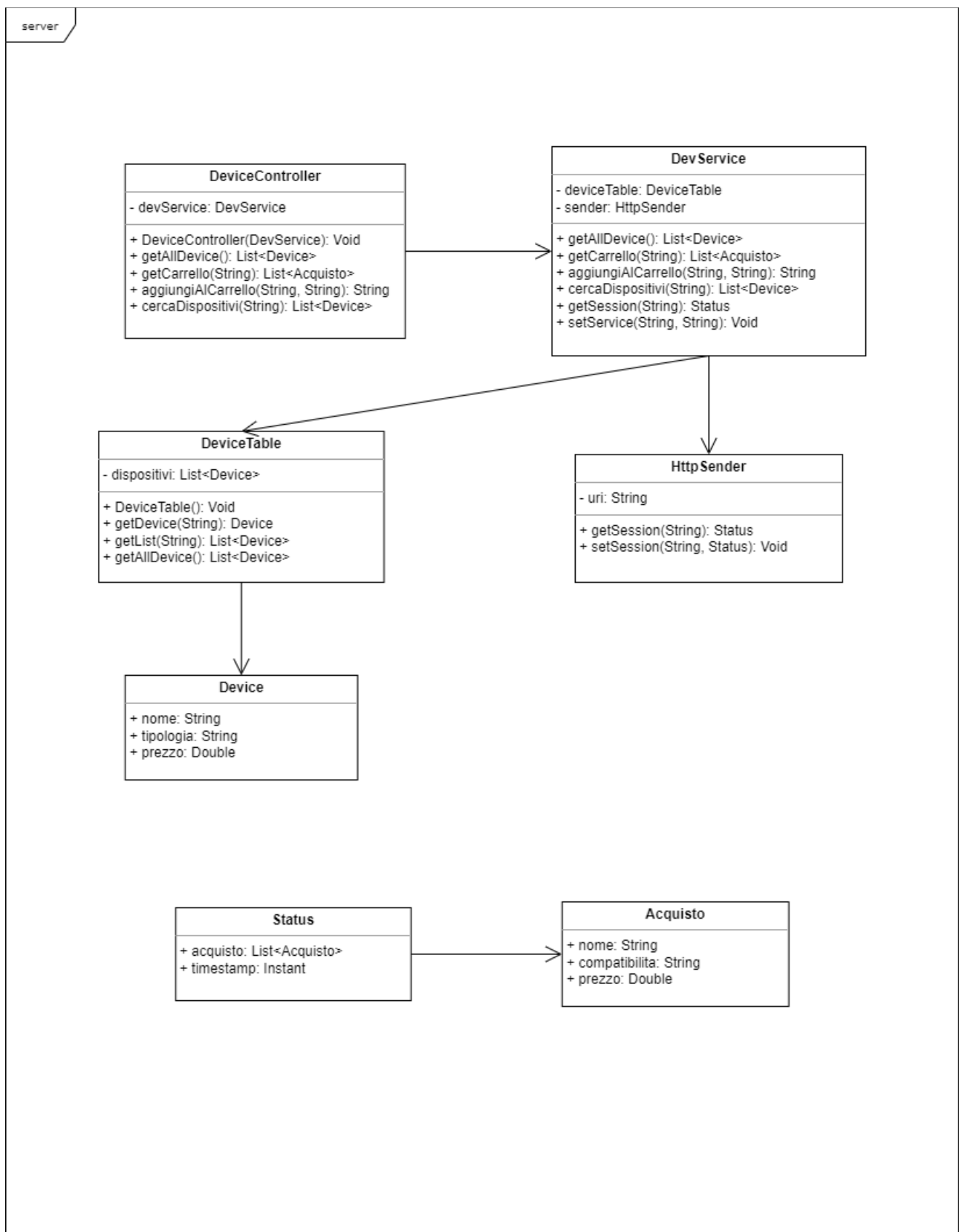
Microservizi

Ho implementato i seguenti microservizi:

- **Client:** permette all'utente di inserire il nome con cui sarà salvata la sessione. Contiene il Circuit Breaker tramite il quale effettuiamo le chiamate REST API. Il Circuit Breaker ha un riferimento alla classe `HttpRequest` per mandare le richieste al server. Il client può visualizzare la lista dei componenti disponibili, può cercare un componente scrivendo una parola contenuta nel nome di esso, può acquistare un componente e visualizzare la lista dei suoi acquisti.
- **Server:** riceve le richieste dal Client. Il Server contiene i dati dei dispositivi in vendita, permette la ricerca dei dispositivi e verifica la compatibilità. In caso che il client chiede di visualizzare il suo carrello o chiede di acquistare un dispositivo il Server inoltra la richiesta al microservizio che si occupa di gestire la sessione.
- **Sessione:** si occupa di gestire la sessione dei client. Contiene una mappatura `<nome, stato>` in cui ogni nome del client è associata la lista degli acquisti effettuati. Lo stato viene serializzato e salvato nel file system del microservizio così da salvare i dati dell'utente anche quando esso non è collegato.

Diagramma UML





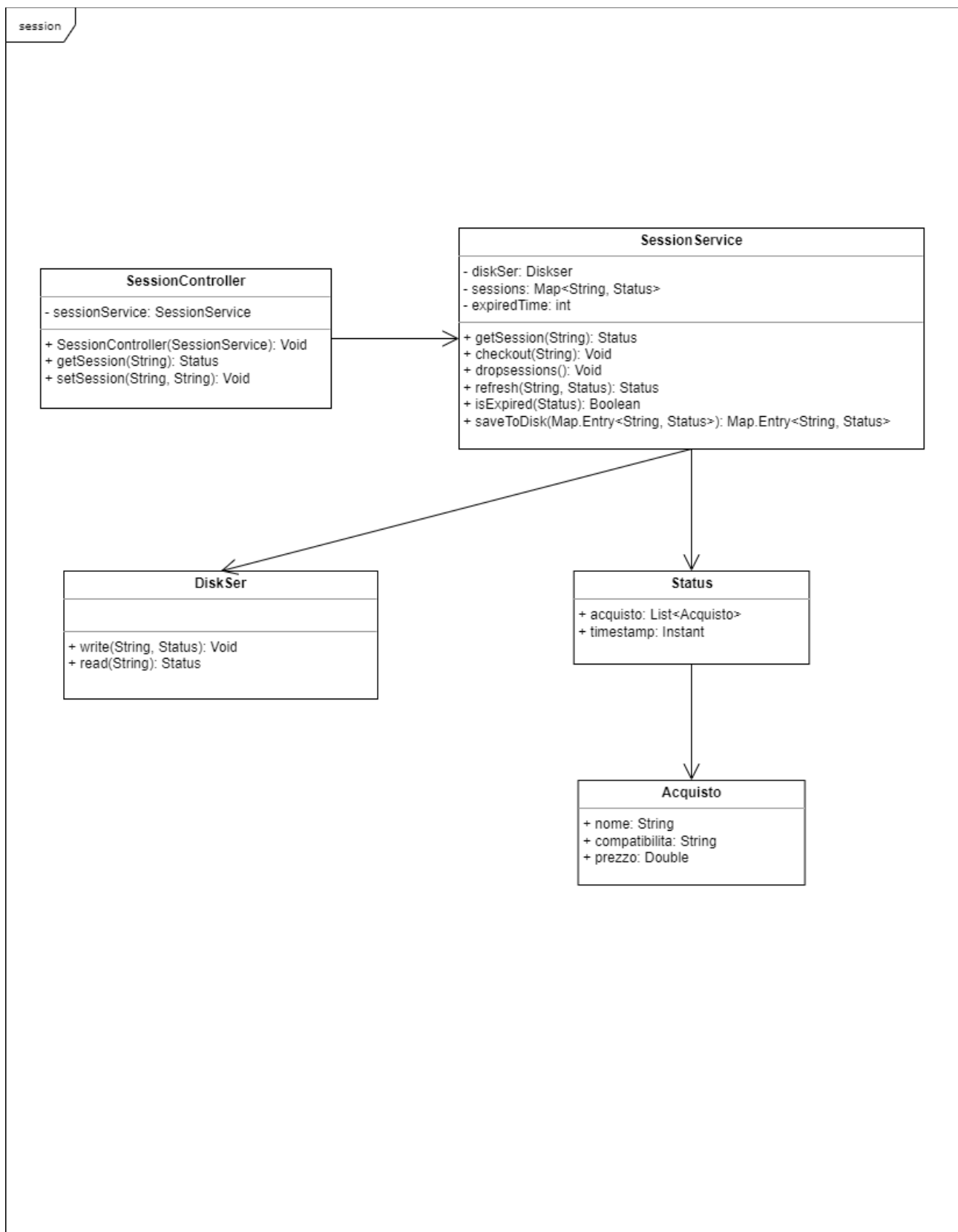


Diagramma di Sequenza: visualizzazione del carrello

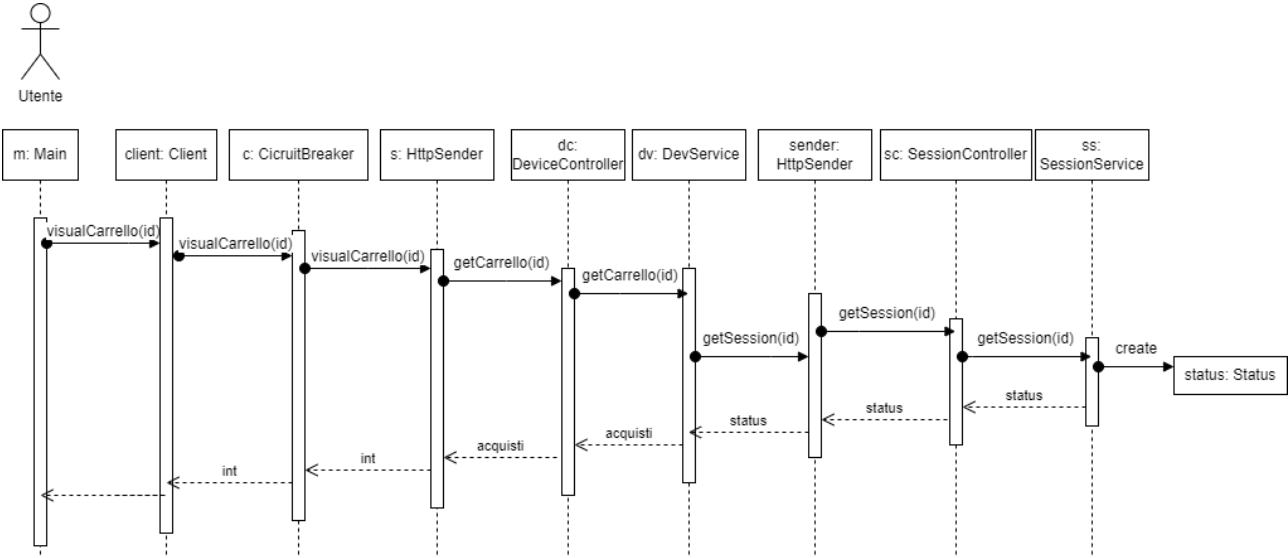
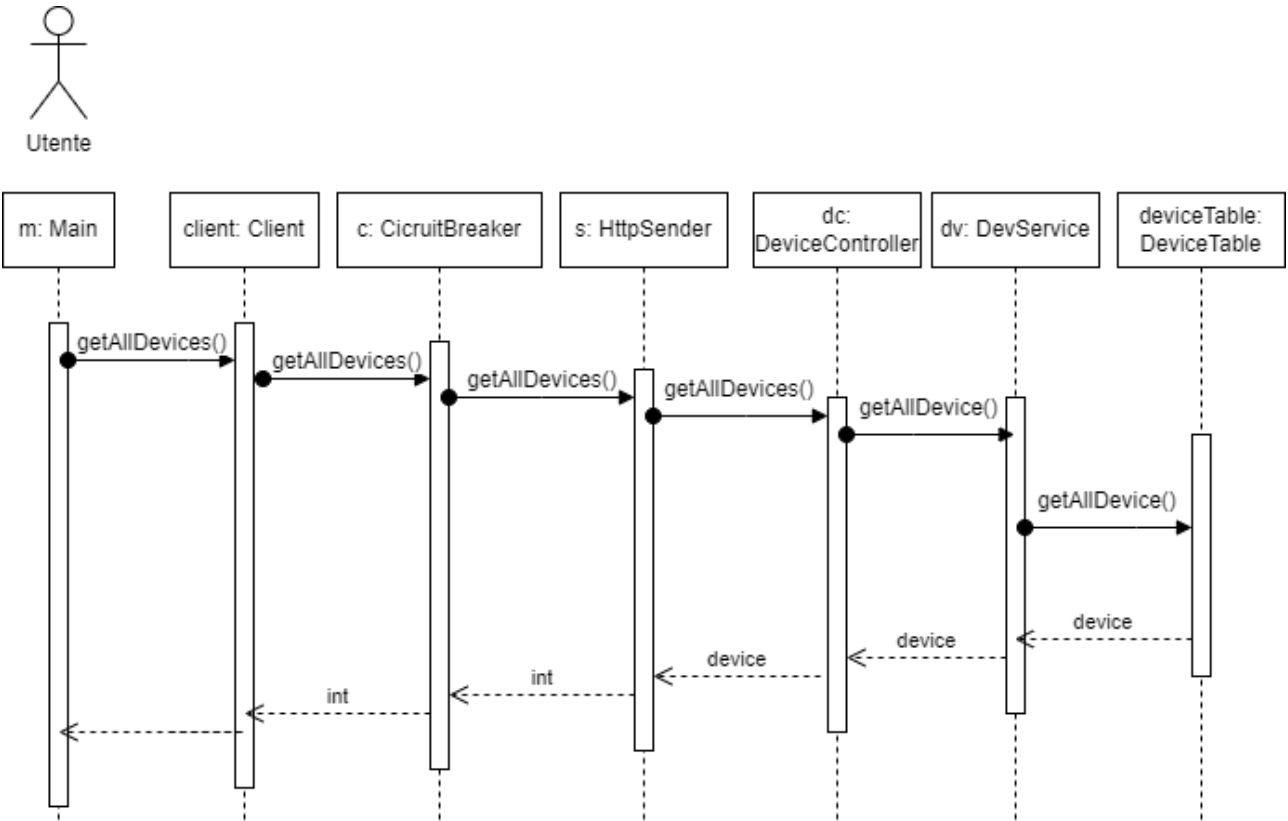


Diagramma di Sequenza: visualizza dispositivi



Microservizio Client: classi

- **Client:** La classe client ha come attributi una Stringa per il nome dell'utente e un riferimento al Circuit Breaker, nel costruttore avviene l'inizializzazione del nome e del Circuit Breaker. I metodi che permettono di interagire col server vengono chiamati dall'oggetto CircuitBreaker e viene stampato a video l'esito della richiesta.
- **CircuitBreaker:** Ha come attributi un HttpSender che si occupa di mandare le richieste HTTP, un BreakerState per indicare lo stato del CircuitBreaker e un CompletableFuture<Integer>.
Nei metodi prima di effettuare le richieste chiamando i metodi di HttpSender lo stato del CircuitBreaker deve essere "chiuso" altrimenti il metodo ritorna -1, se durante la richiesta si verificano delle eccezioni il CircuitBreaker si apre.
- **HttpSender:** Permette di mandare richieste HTTP al server, ha una Stringa per indicare l'indirizzo degli endpoint.

Microservizio Server: classi

- **DeviceController:** Implementa gli endpoint necessari per la comunicazione con il Client, ha un riferimento al DeviceService.
- **DeviceService:** Implementa la logica di business del Server. Ha un riferimento al DeviceTable e al HttpSender, tramite il DeviceTable permettiamo il ritorno della lista dei dispositivi e la funzione per cercare i dispositivi. L'oggetto HttpSender serve a mandare le richieste di accesso alla sessione al Microservizio Session.
- **HttpSender:** permette la ricezione e l'invio dello Status della sessione.
- **DeviceTable:** ha una lista di Device e permette le operazioni di ricerca dei dispositivi, di ricerca dei dispositivi e ritorno della lista dei dispositivi.
- **Device:** record che contiene i dati del singolo dispositivo.
- **Acquisto:** record che contiene i dati del singolo acquisto.
- **Status:** record che contiene la lista degli acquisti e il timestamp per la gestione della sessione.

Microservizio Session: classi

- SessionController: Implementa gli endpoint necessari per la comunicazione con il Server per le operazioni di invio e ricezione dello Status della sessione.
- SessionService: Implementa la logica di business per la gestione della sessione degli utenti, ha una mappatura <String, Status> per permettere l'associazione del nome utente con il suo stato. L'oggetto della classe DiskSer permette il salvataggio della sessione nel file system del servizio Session.
- DiskSer: si occupa della serializzazione dello stato della sessione.
- Acquisto: record che contiene i dati del singolo acquisto.
- Status: record che contiene la lista degli acquisti e il timestamp per la gestione della sessione.