

Specifica Tecnica

2024-02-20 — v0.0.8



overture.unipd@gmail.com

Destinatari	Prof. Tullio Vardanega Prof. Riccardo Cardin <i>Zextras</i> Gruppo <i>Overture</i>
Responsabile	Francesco Costantino Bulychov
Redattori	Riccardo Bonavigo Francesco Costantino Bulychov Riccardo Fabbian Alex Vedovato
Verificatori	Michele Bettin Riccardo Fabbian Francesco Furno

Registro delle modifiche

Versione	Data	Autori	Verificatori	Dettaglio
0.0.8	2024-02-20	Francesco Costantino Bulychov	Francesco Furno	Aggiunta la sezione 'Database'.
0.0.7	2024-02-20	Alex Vedovato	Riccardo Fabbian	Aggiunta la sezione 'Architettura di deployment'.
0.0.6	2024-02-19	Riccardo Bonavigo, Riccardo Fabbian	Michele Bettin	Aggiunta la sezione 'Architettura logica'.
0.0.5	2024-02-16	Riccardo Bonavigo	Michele Bettin	Aggiunta la sezione 'Tecnologie per il testing'.
0.0.4	2024-02-16	Riccardo Fabbian	Francesco Furno	Aggiunta la sezione 'Tecnologie per l'analisi del codice'.
0.0.3	2024-02-15	Francesco Costantino Bulychov	Riccardo Fabbian	Aggiunta la sezione 'Tecnologie per la codifica'.
0.0.2	2024-02-15	Alex Vedovato	Francesco Furno	Aggiunta la sezione 'Stato dei requisiti funzionali'.
0.0.1	2024-02-12	Riccardo Fabbian, Alex Vedovato	Michele Bettin	Struttura di base del documento e introduzione.

Indice

1) Introduzione	5
1.1) Scopo del documento	5
1.2) Glossario	5
1.3) Riferimenti	5
1.3.1) Riferimenti normativi	5
1.3.2) Riferimenti informativi	5
2) Tecnologie	7
2.1) Tecnologie per la codifica	7
2.1.1) Linguaggi	7
2.1.2) Strumenti e servizi	7
2.1.3) Framework	7
2.1.4) Librerie	7
2.2) Tecnologie per l'analisi del codice	8
2.2.1) Analisi statica	8
2.2.2) Analisi dinamica	8
2.3) Tecnologie per il testing	9
2.3.1) Linguaggi	9
2.3.2) Framework	9
3) Architettura	10
3.1) Architettura logica	10
3.2) Architettura di deployment	11
3.2.1) Struttura a monolite vs struttura a microservizi	11
3.2.2) Docker e containerizzazione dell'applicazione	11
3.3) Database	12
3.3.1) Scelta di RethinkDB	12
3.3.2) Funzionalità di RethinkDB	12
3.3.3) Conclusioni	13
4) Stato dei requisiti funzionali	14
4.1) Grafici riassuntivi	23

Lista della immagini

Figura 1: Architettura logica del prodotto	10
Figura 2: Stato dei requisiti funzionali totali	23
Figura 3: Stato dei requisiti funzionali obbligatori	23
Figura 4: Stato dei requisiti funzionali desiderabili	24
Figura 5: Stato dei requisiti funzionali opzionali	24

Lista delle tabelle

Tabella 1: Linguaggi di programmazione usati per la codifica	7
Tabella 2: Strumenti e servizi usati per la codifica	7
Tabella 3: Framework usati per la codifica	7
Tabella 4: Librerie usate per la codifica	7
Tabella 5: Tecnologie usate per l'analisi statica del codice	8
Tabella 6: Tecnologie usate per l'analisi dinamica del codice	8
Tabella 7: Linguaggi di programmazione usati per il testing	9
Tabella 8: Framework utilizzati per il testing	9
Tabella 9: Stato dei requisiti funzionali	14

1) Introduzione

1.1) Scopo del documento

Lo scopo di questo documento è quello di elencare e motivare le scelte architettureali che il gruppo Overture ha intrapreso per la realizzazione dell'infrastruttura informatica richiesta. Il documento comprende anche i diagrammi delle classi e dei package al fine di spiegare in maniera più chiara e dettagliata il software sviluppato.

1.2) Glossario

Per evitare ambiguità o incomprensioni riguardanti la terminologia usata nel documento, è stato deciso di adottare un glossario in cui vengono riportate le varie definizioni. In questa maniera in esso verranno riportati tutti i termini specifici del dominio d'uso con relativi significati.

La presenza di un termine all'interno del Glossario viene indicata applicando [questo stile](#).

1.3) Riferimenti

1.3.1) Riferimenti normativi

- Norme di Progetto v1.0.0:
https://overture-unipd.github.io/docs/rtb/interni/norme_di_progetto_v1.0.0.pdf
- **PD2 - Regolamento del progetto didattico**
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/PD2.pdf>
- **Capitolato d'appalto C8: JMAP**, il nuovo protocollo standard per la comunicazione email
<https://www.math.unipd.it/~tullio/IS-1/2023/Progetto/C8.pdf>

1.3.2) Riferimenti informativi

- Glossario v1.0.0:
https://overture-unipd.github.io/docs/rtb/interni/glossario_v1.0.0.pdf
- Analisi dei Requisiti v1.1.0:
https://overture-unipd.github.io/docs/rtb/esterni/analisi_dei_requisiti_v1.1.0.pdf
- Progettazione: le dipendenze fra le componenti, Prof. Riccardo Cardin
<https://www.math.unipd.it/~rcardin/swear/2023/Object-Oriented%20Programming%20Principles%20Revised.pdf>
- Progettazione e programmazione: diagrammi delle classi (UML), Prof. Riccardo Cardin
<https://www.math.unipd.it/~rcardin/swear/2023/Diagrammi%20delle%20Classi.pdf>
- Progettazione: i *pattern* architettureali, Prof. Riccardo Cardin
<https://www.math.unipd.it/~rcardin/swear/2022/Software%20Architecture%20Patterns.pdf>
- Progettazione: il *pattern Dependency Injection*, Prof. Riccardo Cardin
<https://www.math.unipd.it/~rcardin/swear/2022/Design%20Pattern%20Architettureali%20-%20Dependency%20Injection.pdf>
- Progettazione *software* (T6), Prof. Tullio Vardanega
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/T6.pdf>
- Progettazione: i *pattern* creazionali (GoF), Prof. Riccardo Cardin
<https://www.math.unipd.it/~rcardin/swear/2022/Design%20Pattern%20Creazionali.pdf>
- Progettazione: i *pattern* strutturali (GoF), Prof. Riccardo Cardin
<https://www.math.unipd.it/~rcardin/swear/2022/Design%20Pattern%20Strutturali.pdf>
- Progettazione: i *pattern* di comportamento (GoF), Prof. Riccardo Cardin
https://www.math.unipd.it/~rcardin/swear/2021/Design%20Pattern%20Comportamentali_4x4.pdf

-
- Programmazione: SOLID *programming*, Prof. Riccardo Cardin
https://www.math.unipd.it/~rcardin/swea/2021/SOLID%20Principles%20of%20Object-Oriented%20Design_4x4.pdf

2) Tecnologie

In questa sezione vengono elencate tutte le tecnologie utilizzate per l'implementazione del prodotto richiesto dal capitolato.

2.1) Tecnologie per la codifica

2.1.1) Linguaggi

Nome	Versione	Descrizione
Java	21 LTS	Linguaggio di programmazione ad alto livello, orientato agli oggetti e a tipizzazione statica. Fortemente consigliato dall'azienda proponente Zextras dato che è il linguaggio principale nello stack tecnologico di Carbonio.

Tabella 1: Linguaggi di programmazione usati per la codifica

2.1.2) Strumenti e servizi

Nome	Versione	Descrizione
Docker	25.0.0	Strumento di containerizzazione richiesto esplicitamente dall'azienda proponente. Ogni contenitore viene eseguito in modo isolato dagli altri. Si integra facilmente con Gradle tramite un plugin.
Docker-compose	2.24.0	Strumento di containerizzazione utilizzato per la definizione di servizi multi-container. Semplifica la gestione e la scalabilità delle applicazioni.
RethinkDB	2.4.4	Database NoSQL che supporta subqueries e changefeed. Facilita la gestione e le conversioni degli oggetti Json e consente di rappresentare facilmente dati complessi e annidati.
Gradle	8.6	Strumento di automazione della build e di gestione delle dipendenze, progettato per la compilazione, il testing e la distribuzione di progetti software. Utilizza un linguaggio DSL basato su Groovy o Kotlin e supporta la build incrementale.
Caddy	2.7.0	Web server open-source che converte il traffico da HTTP a HTTPS. Si distingue per la sua facilità d'uso, configurazione automatizzata e supporto nativo per HTTPS. Necessario in quanto richiesto dai client.

Tabella 2: Strumenti e servizi usati per la codifica

2.1.3) Framework

Nome	Versione	Descrizione
Spark	3.5.0	Lightweight framework open-source per lo sviluppo di applicazioni web in Java. Ideale per progetti agili che cercano di mantenere la complessità sotto controllo.

Tabella 3: Framework usati per la codifica

2.1.4) Librerie

Nome	Versione	Descrizione
Java JMAP iNPUTmice Library	0.8.13	Libreria consigliata dall'azienda proponente che sincronizza i dati tra client e server utilizzando il JSON Meta Application Protocol. Definisce classi per ogni operazione possibile, funzioni per operare su di esse e conversioni da e verso Json.

Tabella 4: Librerie usate per la codifica

2.2) Tecnologie per l'analisi del codice

2.2.1) Analisi statica

Nome	Versione	Descrizione
Compilatore Java	JDK 21	Traduce il codice sorgente scritto in linguaggio di programmazione Java in un formato eseguibile, generalmente chiamato bytecode Java. Il bytecode Java può essere eseguito da una macchina virtuale Java (JVM) su qualsiasi piattaforma che abbia una JVM disponibile, rendendo il codice Java altamente portabile.
Spotless	2.42.0	Plugin Gradle open-source utilizzato per applicare automaticamente le convenzioni di formattazione del codice a progetti Java. Garantisce che il codice sorgente sia formattato secondo regole specifiche, migliorandone la leggibilità e la manutenibilità.

Tabella 5: Tecnologie usate per l'analisi statica del codice

2.2.2) Analisi dinamica

Nome	Versione	Descrizione
Junit	5.10.0	Framework open source per l'automazione di unit testing per il linguaggio di programmazione Java. Offre un ambiente di sviluppo strutturato per la scrittura dei test, consentendo agli sviluppatori di verificare il comportamento delle singole unità di codice in modo efficiente e affidabile.
Mockito	5.10.0	Framework open source di testing per il linguaggio di programmazione Java. È utilizzato per la creazione, configurazione e gestione di oggetti mock nei test di unità. Consente agli sviluppatori di simulare il comportamento di oggetti reali durante l'esecuzione dei test.
TestContainers	1.19.4	Framework open source che fornisce un'interfaccia per l'integrazione di container Docker nelle attività di testing. È particolarmente utile per lo sviluppo di test di integrazione che coinvolgono componenti dipendenti come database o qualsiasi altro servizio che può essere eseguito in un container Docker.

Tabella 6: Tecnologie usate per l'analisi dinamica del codice

2.3) Tecnologie per il testing

2.3.1) Linguaggi

Nome	Versione	Descrizione
Python	3.12.1	Linguaggio di programmazione ad alto livello, interpretato e general-purpose. È noto per la sua sintassi chiara e leggibile, che lo rende molto adatto per sviluppare rapidamente script e applicazioni.

Tabella 7: Linguaggi di programmazione usati per il testing

2.3.2) Framework

Nome	Versione	Descrizione
Locust	2.23.1	Framework open source di testing di carico e stress delle applicazioni. È scritto in Python e permette agli sviluppatori di scrivere test di carico simulando il comportamento di migliaia di utenti concorrenti.

Tabella 8: Framework utilizzati per il testing

3) Architettura

La descrizione dell'architettura del prodotto adotta un approccio top-down, partendo dalla struttura generale per poi scendere nel dettaglio.

3.1) Architettura logica

Nell'architettura logica che abbiamo scelto di adottare, il server di posta è organizzato in un modello esagonale che riflette una suddivisione chiara delle responsabilità e delle funzionalità delle varie componenti, andando a porre al centro la business logic, la quale non andrà così a dipendere da altre parti del sistema riguardanti, per esempio, logiche di persistenza.

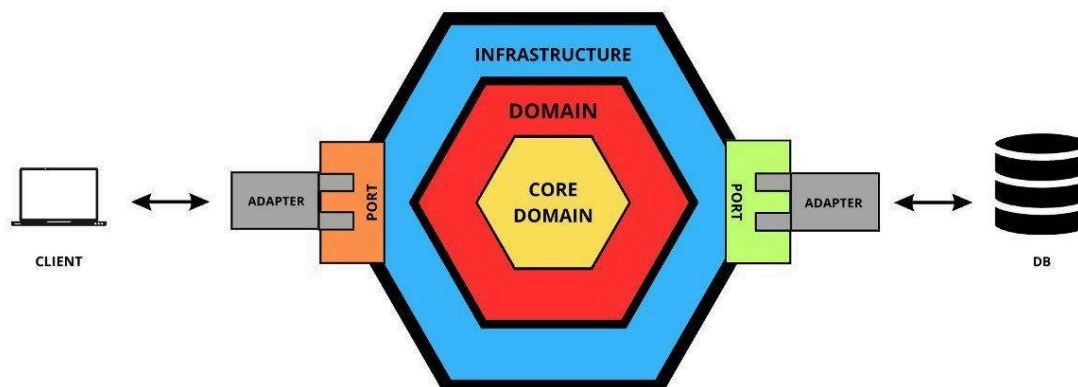


Figura 1: Architettura logica del prodotto

All'esterno dell'esagono, in entrata, è presente una componente dedicata alla gestione delle richieste provenienti dai client, fornendo un'interfaccia per l'ingresso di quest'ultime nel sistema. Questa classe è responsabile di gestire gli oggetti JSON forniti dagli utilizzatori adattandoli per garantire una coerenza con la logica di business del sistema. È il punto di contatto tra il server di posta elettronica e il mondo esterno, quindi i client che interagiscono con il nostro prodotto.

Al centro dell'esagono risiede poi la business logic del server di posta elettronica. Questa è la parte centrale del sistema, dove avviene l'elaborazione delle richieste in arrivo. Qui si trovano quindi le implementazioni delle funzionalità come la gestione delle email, la gestione delle caselle di posta, la gestione delle condivisioni di quest'ultime ed altro ancora, oltre a tutti gli oggetti della libreria specifici del dominio del nostro prodotto. La business logic costituisce il cuore del server di posta elettronica, garantendo il corretto funzionamento del sistema e la coerenza delle operazioni svolte.

Infine, all'esterno dell'esagono, in uscita, troviamo l'insieme di componenti che si occupano dell'interfacciamento con il database. Queste classi hanno il ruolo di gestire la persistenza dei dati necessari per il funzionamento del sistema, inclusi salvataggio delle email, memorizzazione delle cartelle, persistenza degli account e così via. Assicurano che i dati vengano salvati e recuperati in modo affidabile e efficiente.

L'adozione di questa architettura esagonale favorisce una gestione modulare e scalabile del server di posta elettronica. Ogni componente svolge un ruolo specifico e ben definito, facilitando la manutenzione, l'aggiornamento e l'espansione del sistema nel tempo. Inoltre, la chiara separazione delle responsabilità e delle funzionalità promuove la testabilità del sistema, consentendo una maggiore fiducia nella robustezza e nella stabilità complessiva del prodotto.

Un grande vantaggio di questa architettura è la facilità nel cambiare, per esempio, il database sottostante. Supponendo infatti che si voglia aggiornare il sistema di persistenza dei dati, grazie alla net-

ta separazione delle componenti e all'interfacciamento ben definito con il database è relativamente semplice farlo senza dover apportare modifiche significative al resto del sistema. Questa flessibilità consente di adattare il server di posta elettronica alle esigenze future e alle evoluzioni tecnologiche, garantendo una maggiore longevità e versatilità del prodotto.

3.2) Architettura di deployment

3.2.1) Struttura a monolite vs struttura a microservizi

Dal momento in cui il prodotto software che dobbiamo andare a realizzare ha lo scopo di fornire una implementazione per un nuovo protocollo, per poi successivamente testarne le performance con degli stress test, la scelta di un architettura a monolite rispetto che ad altre, come quella a microservizi, è motivata da una vasta gamma di fattori.

Prima di tutto, come detto prima, il software non necessiterà di particolari espansioni future, una volta forniti i risultati degli stress test essa verrà utilizzata dal committente per effettuare ulteriori test o al massimo per osservare come abbiamo implementato certe funzionalità di jmap nel caso volessero integrarle nei loro sistemi.

Fatte queste premesse, l'approccio monolitico è quindi preferito per la sua rapidità e semplicità. L'applicazione è destinata ad essere di dimensioni limitate e a svolgere un compito specifico, il team dunque intende semplificare lo sviluppo senza la necessità di gestire la complessità aggiuntiva introdotta da un'architettura a microservizi.

Inoltre, in un progetto dove il team di sviluppo ha un'esperienza di programmazione scarsa e necessita di modificare e correggere continuamente alcune parti del software, l'architettura monolitica risulta essere più gestibile in quanto permette di semplificare il processo di sviluppo riducendo la necessità di coordinazione tra diversi servizi, come richiesto dalle architetture a microservizi.

3.2.1.1) Conclusioni

L'architettura monolitica è raccomandata per applicazioni semplici e per prototipi, semplificando lo sviluppo senza la necessità di integrare molteplici servizi. D'altra parte l'architettura a microservizi si adatta meglio a sistemi complessi offrendo un'aggiunta flessibile di nuove funzionalità.

Per la scelta dell'architettura è anche fondamentale conoscere le competenze generali del team di sviluppo. Per lo sviluppo con i microservizi sono generalmente essenziali conoscenze tecniche specifiche riguardanti cloud, API e containerizzazione.

In secondo luogo la scelta dell'architettura è anche condizionata dall'infrastruttura con cui si ha a che fare: le applicazioni monolitiche operano su un singolo server, i microservizi traggono maggiore beneficio dall'ambiente cloud, offrendo scalabilità e distribuzione.

Per concludere, il team *Overture* riconosce i vantaggi che alcune architetture più complesse, come quella a microservizi, porterebbero sicuramente al prodotto sviluppato, ma per una serie di ragioni legate alla complessità intrinseca delle architetture stesse riteniamo che nel nostro scenario i vantaggi non coprano assolutamente gli sforzi e i tempi necessari nel presente per adottare queste architetture. Per questo abbiamo optato per una architettura monolitica.

3.2.2) Docker e containerizzazione dell'applicazione

L'uso della containerizzazione nell'ambito dell'architettura di deployment è considerato interessante per i seguenti motivi (i quali ci hanno fatto comprendere il motivo per cui il committente aveva inserito la containerizzazione dell'applicazione come requisito obbligatorio):

- Portabilità: i container forniscono un ambiente isolato che include tutte le dipendenze. Questo rende l'applicazione altamente portabile tra diversi ambienti, eliminando le preoccupazioni legate alle differenze di configurazione tra i sistemi di sviluppo e produzione.

- **Consistenza:** la containerizzazione garantisce che l'applicazione venga eseguita in un ambiente consistente, indipendentemente dalla macchina host. Ciò riduce i problemi legati a differenze di configurazione tra le diverse fasi di sviluppo e deployment, migliorando la coerenza del processo.
- **Scalabilità:** i container sono leggeri e possono essere avviati rapidamente. Questo facilita la scalabilità orizzontale, consentendo l'esecuzione di più istanze di un'applicazione su più container.
- **Gestione delle risorse:** i container condividono il kernel del sistema operativo host, riducendo l'overhead rispetto a soluzioni di virtualizzazione tradizionali. Ciò consente di utilizzare in modo più efficiente le risorse hardware, riducendo i costi e migliorando le prestazioni complessive del sistema.

Complessivamente, la containerizzazione offre numerosi vantaggi nell'ambito dell'architettura di deployment, migliorando l'efficienza, la portabilità e la gestione delle applicazioni.

3.2.2.1) Come lo abbiamo utilizzato noi

Se andiamo sulla cartella root del nostro progetto noteremo che c'è un file chiamato "docker-compose.yml". Questo definisce una configurazione di Docker Compose per orchestrare i container di tre servizi diversi: web server, database e caddy.

Il servizio web server è attivato mediante l'immagine custom `overture-unipd/jmap:latest` ricavata dall'immagine di base `openjdk:21-jdk-slim` e esposta nella porta 8000.

Il servizio di database è creato mediante l'immagine `rethinkdb:2.4.2-bullseye-slim` disponibile direttamente sul repository pubblico al seguente link https://hub.docker.com/_/rethinkdb mappando le porte 9000, 29015 e 28015 del container con rispettivamente le porte 8080, 29015 e 28015 dell'host e configurando i volumi in modo tale da consentire la persistenza dei dati del database tra le esecuzioni del container.

Il servizio di caddy invece lo attiviamo con l'immagine custom `overture-unipd/caddy:latest` ricavata dall'immagine di base `caddy:latest` ma integrata con il plugin per Duck DNS. Anche qui vengono mappate le porte 80 e 443 del container con quelle dell'host e vengono configurati i volumi Docker per condividere dati tra il container e l'host, ad esempio per persistere i dati del server web Caddy e per fornire un file di configurazione Caddy personalizzato.

Si nota che tutti i volumi che i servizi montano, sono riportati alla fine del file "docker-compose.yml" dopo il record `volumes`.

3.3) Database

Come già citato nella sezione **Tecnologie** del documento, il nostro prodotto utilizza RethinkDB come database NoSQL per la gestione dei dati. Il database viene inizializzato con la creazione delle collezioni richieste (account, email, mailbox, attachment, update...) e l'inserimento di dati di esempio. Successivamente, viene utilizzato per l'aggiunta di nuovi dati o la sostituzione di quelli esistenti.

3.3.1) Scelta di RethinkDB

RethinkDB è stata la nostra scelta principale per molteplici motivi che rispecchiano le esigenze uniche del progetto: la necessità di un sistema altamente flessibile, scalabile e performante, in grado di adattarsi a diverse condizioni operative, che includono sia situazioni normali che di elevato carico e sovraccarico.

3.3.2) Funzionalità di RethinkDB

- **Modello di dati flessibile:** la natura NoSQL di questo database ci consente di modellare i dati in modo flessibile, senza vincoli rigidi di schema. Ciò significa che possiamo memorizzare email, cartelle, metadati e altri dati associati in un formato che si adatta alle esigenze specifiche del nostro sistema, consentendoci di gestire la complessità del nostro dominio in modo efficiente;

- **Scalabilità Orizzontale:** RethinkDB è progettato per scalare orizzontalmente, consentendo al nostro sistema di gestire volumi crescenti di dati e carichi di lavoro variabili. Questa funzionalità è fondamentale per assicurare che il sistema possa crescere in modo fluido con l'aumentare del carico, senza compromettere le prestazioni o la disponibilità del servizio;
- **Real time:** RethinkDB è stato progettato per supportare applicazioni che richiedono aggiornamenti in tempo reale dei dati grazie alla sua capacità di fornire un flusso continuo di cambiamenti ai dati attraverso i feed dei cambiamenti: questo lo rende particolarmente adatto per l'integrazione con un server di posta elettronica.

RethinkDB offre, inoltre, un potente sistema di query che semplifica l'accesso e la manipolazione dei dati, inclusi strumenti come subqueries e changefeed.

- **Subqueries:** sono query annidate all'interno di altre query e consentono agli sviluppatori di scrivere query più complesse e efficienti per soddisfare le esigenze specifiche delle loro applicazioni;
- **Changefeeds:** permettono agli sviluppatori di tracciare le modifiche nei dati e di ricevere notifiche istantanee quando avvengono cambiamenti nel database, facilitando lo sviluppo di applicazioni reattive.

3.3.3) Conclusioni

L'adozione di RethinkDB nel nostro server di posta elettronica rappresenta un elemento chiave nella nostra strategia di gestione dei dati. Grazie alla sua flessibilità, scalabilità e capacità di query avanzate, siamo in grado di offrire un servizio di posta elettronica affidabile, efficiente e altamente performante. Questo ci permette di ottenere i migliori risultati possibili durante i test di carico (stress test) richiesti dal proponente.

4) Stato dei requisiti funzionali

Codice	Tipo	Descrizione	Stato
R-001-F-2	Desiderabile	L'utente che utilizza un client di posta elettronica per interagire con il server deve autenticarsi all'interno del sistema.	Non soddisfatto
R-002-F-2	Desiderabile	È necessario che il client fornisca all'interno della richiesta l'indirizzo email personale dell'utente per procedere con l'autenticazione.	Non soddisfatto
R-003-F-2	Desiderabile	È necessario che il client fornisca all'interno della richiesta la password associata all'indirizzo email personale dell'utente per procedere con l'autenticazione.	Non soddisfatto
R-004-F-2	Desiderabile	Se la fase di autenticazione è fallita allora è necessario che il client riceva dal server una risposta con eventuali dettagli che ne indicano il motivo.	Non soddisfatto
R-005-F-1	Obbligatorio	Il client deve essere in grado di reperire la risorsa JMAP Session, contenente informazioni sulle capacità del server, dettagli sull'account dell'utente e le URL per le richieste API future, in modo da poter interagire con dati e servizi offerti dal server.	Non soddisfatto
R-006-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "unknownCapability" in caso di esecuzione di una richiesta con proprietà "using" non supportata dal server.	Non soddisfatto
R-007-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "notJSON" se il contenuto di una richiesta inviata al server non era application/json o se la richiesta non è stata interpretata dal server come I-JSON.	Non soddisfatto
R-008-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "notRequest" se una richiesta JSON non ha corrisposto alla firma di tipo dell'oggetto di richiesta (Request).	Non soddisfatto
R-009-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "limit" in caso di inserimento di una richiesta che supera uno dei limiti definiti sull'oggetto di capacità, come maxSizeRequest, maxCallsInRequest o maxCurrentRequests.	Non soddisfatto
R-010-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "serverUnavailable" in caso di inserimento di una richiesta che necessita di alcune risorse interne del server momentaneamente non disponibili.	Non soddisfatto
R-011-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "serverFail" in caso si verifichi un errore inaspettato o sconosciuto durante l'elaborazione di una sua richiesta dal server.	Non soddisfatto
R-012-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "serverPartialFail" e proceda risincronizzando i dati in caso si verifichi un errore inaspettato o sconosciuto durante l'elaborazione di una sua richiesta dal server.	Non soddisfatto
R-013-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "unknownMethod" in caso di inserimento di una richiesta contenente un metodo non riconosciuto dal server.	Non soddisfatto
R-014-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "invalidArguments" se uno degli argomenti di un metodo fornito all'interno di una richiesta al server è di tipo errato, non valido o, nel caso in cui sia obbligatorio, è assente.	Non soddisfatto

R-015-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "invalidResultReference" se uno degli argomenti di un metodo fornito all'interno di una richiesta al server ha utilizzato un riferimento di risultato che non è stato possibile risolvere da parte del server.	Non soddisfatto
R-016-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "forbidden" in caso utilizzi, all'interno di una richiesta al server, un metodo la cui esecuzione violerebbe una Access Control List (ACL) o un'altra policy di autorizzazione.	Non soddisfatto
R-017-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "accountNotFound" se l'"accountID" fornito all'interno di una richiesta al server non corrisponde a un account valido.	Non soddisfatto
R-018-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "accountNotSupportedByMethod" se all'interno di una richiesta al server è presente un metodo o tipo di dato non supportato dall'"accountID" fornito.	Non soddisfatto
R-019-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "accountReadOnly" se all'interno di una richiesta al server è presente un metodo che tenta di modificare lo stato nonostante l'account sia in sola lettura.	Non soddisfatto
R-020-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "cannotCalculateChanges" se, in seguito all'inserimento di una richiesta, il server non possa calcolare le modifiche dello stato dalla stringa di stato fornita dal client.	Non soddisfatto
R-021-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "overQuota" se una richiesta inserita nel server richiede la creazione di oggetti che per dimensione o quantità superano il limite imposto dal server.	Non soddisfatto
R-022-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "notFound" se una richiesta inserita nel server fornisce degli ID che non possono essere trovati.	Non soddisfatto
R-023-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "willDestroy" se ha richiesto che un oggetto fosse sia aggiornato che distrutto all'interno della stessa richiesta al server.	Non soddisfatto
R-024-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "tooLarge" se una richiesta inserita nel server richiede la creazione di un oggetto che supera il limite definito dal server per la dimensione massima per un oggetto di quel tipo.	Non soddisfatto
R-025-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "rateLimit" se una richiesta inserita nel server comporta la creazione di un oggetto per cui sono stati creati troppi oggetti quel tipo di recente, raggiungendo un limite di frequenza definito dal server.	Non soddisfatto
R-026-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "invalidPatch" se una richiesta inserita nel server fornisce un PatchObject non valido per modificare il record.	Non soddisfatto
R-027-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "invalidProperties" se una richiesta inserita nel server fornisce un record non valido.	Non soddisfatto
R-028-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "singleton" se una richiesta inserita nel server tentasse di agire erroneamente su un tipo singleton.	Non soddisfatto

R-029-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "requestTooLarge" se una richiesta inserita nel server contiene un numero di azioni che supera il massimo che il server è disposto a elaborare in una singola chiamata di metodo interna alla richiesta.	Non soddisfatto
R-030-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "stateMismatch" se una richiesta inserita nel server contiene un argomento ifnState e questo non corrisponde allo stato attuale.	Non soddisfatto
R-031-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "blobNotFound" se una richiesta inserita nel server contiene almeno un ID blob fornito per una parte del corpo dell'email che non esiste.	Non soddisfatto
R-032-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "tooManyKeywords" se una richiesta inserita nel server modifica un numero di parole chiave dell'email superiore al limite massimo definito dal server.	Non soddisfatto
R-033-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "tooManyMailboxes" se una richiesta inserita nel server modifica un numero di cartelle a cui appartiene l'email superiore al limite massimo definito dal server.	Non soddisfatto
R-034-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "alreadyExists" se una richiesta inserita in un server che vieta i duplicati contiene un record già esistente nell'account di destinazione.	Non soddisfatto
R-035-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "fromAccountNotFound" se una richiesta inserita nel server contiene un fromAccountId che non corrisponde a nessun account valido.	Non soddisfatto
R-036-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "fromAccountNotSupportedByMethod" se una richiesta inserita nel server contiene un fromAccountId che non supporta un tipo di dato utilizzato.	Non soddisfatto
R-037-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "anchorNotFound" se una richiesta inserita nel server contiene un argomento di ancoraggio che non è stato trovato nei risultati della query.	Non soddisfatto
R-038-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "unsupportedSort" se una richiesta inserita nel server presenta una clausola di ordinamento non supportata o un metodo di collezione non riconosciuto dal server.	Non soddisfatto
R-039-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "unsupportedFilter" se una richiesta inserita nel server contiene un filtro che il server non è grado di elaborare.	Non soddisfatto
R-040-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "tooManyChanges" se una richiesta inserita nel server contiene un numero di modifiche superiore all'argomento maxChanges inserito del client.	Non soddisfatto
R-041-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "mailboxHasChild" se una richiesta inserita nel server desidera rimuovere una cartella(Mailbox) che ha ancora almeno una cartella figlia.	Non soddisfatto
R-042-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "mailboxHasEmail" se una richiesta inserita nel server, con l'argomento onDestroyRemoveEmails impostato su false, desidera	Non soddisfatto

		rimuovere una cartella(Mailbox) che ha al suo interno almeno una email.	
R-043-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "invalidEmail" se una richiesta inserita nel server contiene un'email da inviare non valida.	Non soddisfatto
R-044-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "tooManyRecipients" se una richiesta inserita nel server contiene un envelope (insieme di destinatari) che ha più destinatari di quanti il server consenta.	Non soddisfatto
R-045-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "noRecipients" se una richiesta inserita nel server contiene un envelope (insieme di destinatari) che non presenta alcun destinatario.	Non soddisfatto
R-046-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "invalidRecipients" se una richiesta inserita nel server contiene un envelope (insieme di destinatari) con almeno un indirizzo email destinatario non valido.	Non soddisfatto
R-047-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "forbiddenMailFrom" se una richiesta è inserita in un server che non consente all'utente di inviare un messaggio con quel indirizzo mittente nell'envelope (From address).	Non soddisfatto
R-048-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "forbiddenFrom" se una richiesta è inserita in un server che non consente all'utente di inviare un messaggio con il campo di intestazione From del messaggio da inviare.	Non soddisfatto
R-049-F-1	Obbligatorio	È necessario che il client riceva in risposta l'errore "forbiddenToSend" se una richiesta è inserita in un server che non consente all'utente di inviare un messaggio in quel momento.	Non soddisfatto
R-050-F-1	Obbligatorio	L'utente che utilizza un client di posta elettronica per interagire con il server deve avere la possibilità di inviare email.	Non soddisfatto
R-051-F-1	Obbligatorio	È necessario che il client inserisca all'interno della richiesta l'identificazione delle capacità necessarie all'invio di una email.	Non soddisfatto
R-052-F-1	Obbligatorio	È necessario che il client inserisca all'interno della richiesta le chiamate di metodo necessarie all'invio di una email, con relativi parametri (sia quelli comuni, come l'identificativo dell'account da utilizzare, sia quelli specifici del caso) e un identificatore univoco associato.	Non soddisfatto
R-053-F-1	Obbligatorio	È necessario che il client inserisca all'interno della richiesta le proprietà dell'oggetto Email da creare (cartelle in cui è contenuta, mittente, destinatari, oggetto, corpo del messaggio ed altri dettagli definiti dall'RFC5322).	Non soddisfatto
R-054-F-1	Obbligatorio	È necessario che il client inserisca all'interno della richiesta le proprietà dell'oggetto EmailSubmission da creare (l'identificativo dell'email creata in precedenza ed ora da inviare, le informazioni necessarie per l'invio ed altri dettagli).	Non soddisfatto
R-055-F-1	Obbligatorio	È possibile che il client inserisca all'interno della richiesta eventuali azioni da compiere in seguito al corretto invio dell'email.	Non soddisfatto
R-056-F-1	Obbligatorio	È necessario che il client riceva una risposta che contiene l'esito dell'operazione di invio dell'email con relativi parametri (come il nuovo stato, gli oggetti creati ed eventuali errori).	Non soddisfatto

R-057-F-1	Obbligatorio	L'utente che utilizza un client di posta elettronica per interagire con il server deve avere la possibilità di ricevere email e visualizzarne il dettaglio.	Non soddisfatto
R-058-F-1	Obbligatorio	È necessario che il client inserisca all'interno della richiesta l'identificazione delle capacità necessarie alla ricezione di una email.	Non soddisfatto
R-059-F-1	Obbligatorio	È necessario che il client inserisca all'interno della richiesta le chiamate di metodo necessarie alla ricezione di una email, con relativi parametri (sia quelli comuni, come l'identificativo dell'account da utilizzare, sia quelli specifici del caso) e un identificatore univoco associato.	Non soddisfatto
R-060-F-1	Obbligatorio	È necessario che il client inserisca all'interno della richiesta l'identificativo delle email da ricevere	Non soddisfatto
R-061-F-1	Obbligatorio	È possibile che il client inserisca all'interno della richiesta le proprietà specifiche delle email che è interessato a ricevere	Non soddisfatto
R-062-F-1	Obbligatorio	È necessario che il client riceva una risposta che contiene l'esito dell'operazione di ricezione dell'email con relativi parametri (come lo stato corrente dei dati di tipo Email sul server, la lista delle email richieste ed eventuali errori)	Non soddisfatto
R-063-F-1	Obbligatorio	L'utente che utilizza un client di posta elettronica per interagire con il server deve avere la possibilità di eliminare email.	Non soddisfatto
R-064-F-1	Obbligatorio	È necessario che il client inserisca all'interno della richiesta l'identificazione delle capacità necessarie all'eliminazione di una email.	Non soddisfatto
R-065-F-1	Obbligatorio	È necessario che il client inserisca all'interno della richiesta le chiamate di metodo necessarie all'eliminazione di una email, con relativi parametri (sia quelli comuni, come l'identificativo dell'account da utilizzare, sia quelli specifici del caso) e un identificatore univoco associato.	Non soddisfatto
R-066-F-1	Obbligatorio	È necessario che il client inserisca all'interno della richiesta la lista degli identificativi delle email da eliminare.	Non soddisfatto
R-067-F-1	Obbligatorio	È necessario che il client riceva una risposta che contiene l'esito dell'operazione di eliminazione delle email con relativi parametri (come il nuovo stato, gli identificativi degli oggetti eliminati ed eventuali errori)	Non soddisfatto
R-068-F-1	Obbligatorio	L'utente che utilizza un client di posta elettronica per interagire con il server deve avere la possibilità di ricevere cartelle e visualizzarne il dettaglio.	Non soddisfatto
R-069-F-1	Obbligatorio	È necessario che il client inserisca all'interno della richiesta l'identificazione delle capacità necessarie alla ricezione di una cartella.	Non soddisfatto
R-070-F-1	Obbligatorio	È necessario che il client inserisca all'interno della richiesta le chiamate di metodo necessarie alla ricezione di una cartella, con relativi parametri (sia quelli comuni, come l'identificativo dell'account da utilizzare, sia quelli specifici del caso) e un identificatore univoco associato.	Non soddisfatto
R-071-F-1	Obbligatorio	È necessario che il client inserisca all'interno della richiesta l'identificativo delle cartelle da ricevere	Non soddisfatto
R-072-F-1	Obbligatorio	È possibile che il client inserisca all'interno della richiesta le proprietà specifiche delle cartelle che è interessato a ricevere	Non soddisfatto

R-073-F-1	Obbligatorio	È necessario che il client riceva una risposta che contiene l'esito dell'operazione di ricezione della cartella con relativi parametri (come lo stato corrente dei dati di tipo Mailbox sul server, la lista delle cartelle richieste ed eventuali errori)	Non soddisfatto
R-074-F-1	Obbligatorio	L'utente che utilizza un client di posta elettronica per interagire con il server deve avere la possibilità di creare cartelle.	Non soddisfatto
R-075-F-1	Obbligatorio	È necessario che il client inserisca all'interno della richiesta l'identificazione delle capacità necessarie alla creazione di una cartella.	Non soddisfatto
R-076-F-1	Obbligatorio	È necessario che il client inserisca all'interno della richiesta le chiamate di metodo necessarie alla creazione di una cartella, con relativi parametri (sia quelli comuni, come l'identificativo dell'account da utilizzare, sia quelli specifici del caso) e un identificatore univoco associato.	Non soddisfatto
R-077-F-1	Obbligatorio	È necessario che il client inserisca all'interno della richiesta le proprietà dell'oggetto Mailbox da creare (nome, eventuale genitore, ruolo ed altri dettagli).	Non soddisfatto
R-078-F-1	Obbligatorio	È necessario che il client riceva una risposta che contiene l'esito dell'operazione di creazione della cartella con relativi parametri (come lo stato corrente del server, la lista delle cartelle create ed eventuali errori)	Non soddisfatto
R-079-F-1	Obbligatorio	L'utente che utilizza un client di posta elettronica per interagire con il server deve avere la possibilità di modificare cartelle esistenti.	Non soddisfatto
R-080-F-1	Obbligatorio	È necessario che il client inserisca all'interno della richiesta l'identificazione delle capacità necessarie alla modifica di una cartella.	Non soddisfatto
R-081-F-1	Obbligatorio	È necessario che il client inserisca all'interno della richiesta le chiamate di metodo necessarie alla modifica di una cartella, con relativi parametri (sia quelli comuni, come l'identificativo dell'account da utilizzare, sia quelli specifici del caso) e un identificatore univoco associato.	Non soddisfatto
R-082-F-1	Obbligatorio	È necessario che il client inserisca all'interno della richiesta le modifiche da apportare all'oggetto Mailbox che l'utente desidera modificare.	Non soddisfatto
R-083-F-1	Obbligatorio	È necessario che il client riceva una risposta che contiene l'esito dell'operazione di modifica della cartella con relativi parametri (come lo stato corrente del server, la lista delle cartelle modificate ed eventuali errori)	Non soddisfatto
R-084-F-1	Obbligatorio	L'utente che utilizza un client di posta elettronica per interagire con il server deve avere la possibilità di eliminare cartelle esistenti.	Non soddisfatto
R-085-F-1	Obbligatorio	È necessario che il client inserisca all'interno della richiesta l'identificazione delle capacità necessarie alla eliminazione di una cartella.	Non soddisfatto
R-086-F-1	Obbligatorio	È necessario che il client inserisca all'interno della richiesta le chiamate di metodo necessarie alla eliminazione di una cartella, con relativi parametri (sia quelli comuni, come l'identificativo dell'account da utilizzare, sia quelli specifici del caso) e un identificatore univoco associato.	Non soddisfatto
R-087-F-1	Obbligatorio	È necessario che il client inserisca all'interno della richiesta la lista degli identificativi delle cartelle da eliminare.	Non soddisfatto

R-088-F-1	Obbligatorio	È necessario che il client specifichi all'interno della richiesta il comportamento desiderato da parte del server quando si cerca di eliminare una cartella che contiene ancora delle email.	Non soddisfatto
R-089-F-1	Obbligatorio	È necessario che il client riceva una risposta che contiene l'esito dell'operazione di eliminazione della cartella con relativi parametri (come lo stato corrente del server, la lista degli identificativi delle cartelle eliminate ed eventuali errori)	Non soddisfatto
R-090-F-1	Obbligatorio	L'utente che utilizza un client di posta elettronica per interagire con il server deve avere la possibilità di gestire i contenuti di una cartella.	Non soddisfatto
R-091-F-1	Obbligatorio	È necessario che il client inserisca all'interno della richiesta l'identificazione delle capacità necessarie alla gestione dei contenuti di una cartella.	Non soddisfatto
R-092-F-1	Obbligatorio	È necessario che il client inserisca all'interno della richiesta le chiamate di metodo necessarie alla gestione dei contenuti di una cartella, con relativi parametri (sia quelli comuni, come l'identificativo dell'account da utilizzare, sia quelli specifici del caso) e un identificatore univoco associato.	Non soddisfatto
R-093-F-1	Obbligatorio	È necessario che il client inserisca all'interno della richiesta le modifiche da apportare agli oggetti Email che l'utente desidera aggiungere ad una o più cartelle.	Non soddisfatto
R-094-F-1	Obbligatorio	È necessario che il client inserisca all'interno della richiesta le modifiche da apportare agli oggetti Email che l'utente desidera rimuovere da una o più cartelle.	Non soddisfatto
R-095-F-1	Obbligatorio	È necessario che il client inserisca all'interno della richiesta le modifiche da apportare agli oggetti Email che l'utente desidera spostare da una o più cartelle ad una o più cartelle.	Non soddisfatto
R-096-F-1	Obbligatorio	È necessario che il client riceva una risposta che contiene l'esito delle operazioni di gestione dei contenuti di una cartella con relativi parametri (come lo stato corrente del server, la lista delle email modificate ed eventuali errori)	Non soddisfatto
R-097-F-2	Desiderabile	L'utente che utilizza un client di posta elettronica per interagire con il server deve avere la possibilità di condividere le sue cartelle con altri utenti.	Non soddisfatto
R-098-F-2	Desiderabile	È necessario che il client inserisca all'interno della richiesta l'identificazione delle capacità necessarie alla creazione della condivisione di una cartella.	Non soddisfatto
R-099-F-2	Desiderabile	È necessario che il client inserisca all'interno della richiesta le chiamate di metodo necessarie alla creazione della condivisione di una cartella, con relativi parametri (sia quelli comuni, come l'identificativo dell'account da utilizzare, sia quelli specifici del caso) e un identificatore univoco associato.	Non soddisfatto
R-100-F-2	Desiderabile	È necessario che il client inserisca all'interno della richiesta le proprietà dell'oggetto Principal da creare (nome, tipo, descrizione, gli identificativi degli account a cui vogliamo condividere le cartelle e altri dettagli).	Non soddisfatto
R-101-F-2	Desiderabile	È necessario che il client inserisca all'interno della richiesta le modifiche da apportare all'oggetto Mailbox che l'utente desidera condividere, specificando quali diritti hanno su quest'ultimo i membri del principale a cui si sta condividendo.	Non soddisfatto
R-102-F-2	Desiderabile	È necessario che il client riceva una risposta che contiene l'esito delle operazioni di creazione della condivisione di una cartella con relativi	Non soddisfatto

		parametri (come lo stato corrente dei dati sul server, la lista dei principali creati, la lista delle cartella modificate ed eventuali errori)	
R-103-F-2	Desiderabile	L'utente che utilizza un client di posta elettronica per interagire con il server deve avere la possibilità di modificare principali esistenti.	Non soddisfatto
R-104-F-2	Desiderabile	È necessario che il client inserisca all'interno della richiesta l'identificazione delle capacità necessarie alla modifica di un principale.	Non soddisfatto
R-105-F-2	Desiderabile	È necessario che il client inserisca all'interno della richiesta le chiamate di metodo necessarie alla modifica di un principale, con relativi parametri (sia quelli comuni, come l'identificativo dell'account da utilizzare, sia quelli specifici del caso) e un identificatore univoco associato.	Non soddisfatto
R-106-F-2	Desiderabile	È necessario che il client inserisca all'interno della richiesta le modifiche da apportare all'oggetto Principal che l'utente desidera modificare.	Non soddisfatto
R-107-F-2	Desiderabile	È necessario che il client riceva una risposta che contiene l'esito dell'operazione di modifica del principale con relativi parametri (come lo stato corrente del server, la lista dei principali modificati ed eventuali errori)	Non soddisfatto
R-108-F-2	Desiderabile	L'utente che utilizza un client di posta elettronica per interagire con il server deve avere la possibilità di eliminare principali esistenti.	Non soddisfatto
R-109-F-2	Desiderabile	È necessario che il client inserisca all'interno della richiesta l'identificazione delle capacità necessarie alla eliminazione di un principale.	Non soddisfatto
R-110-F-2	Desiderabile	È necessario che il client inserisca all'interno della richiesta le chiamate di metodo necessarie alla eliminazione di un principale, con relativi parametri (sia quelli comuni, come l'identificativo dell'account da utilizzare, sia quelli specifici del caso) e un identificatore univoco associato.	Non soddisfatto
R-111-F-2	Desiderabile	È necessario che il client inserisca all'interno della richiesta la lista degli identificativi dei principali da eliminare.	Non soddisfatto
R-112-F-2	Desiderabile	È necessario che il client riceva una risposta che contiene l'esito dell'operazione di eliminazione del principale con relativi parametri (come lo stato corrente del server, la lista degli identificativi dei principali eliminati ed eventuali errori)	Non soddisfatto
R-113-F-2	Desiderabile	L'utente che utilizza un client di posta elettronica per interagire con il server deve avere la possibilità di modificare la condivisione di una cartella (compresa l'eliminazione di quest'ultima).	Non soddisfatto
R-114-F-2	Desiderabile	È necessario che il client inserisca all'interno della richiesta l'identificazione delle capacità necessarie alla modifica della condivisione di una cartella (compresa l'eliminazione di quest'ultima).	Non soddisfatto
R-115-F-2	Desiderabile	È necessario che il client inserisca all'interno della richiesta le chiamate di metodo necessarie alla modifica della condivisione di una cartella (compresa l'eliminazione di quest'ultima), con relativi parametri (sia quelli comuni, come l'identificativo dell'account da utilizzare, sia quelli specifici del caso) e un identificatore univoco associato.	Non soddisfatto
R-116-F-2	Desiderabile	È necessario che il client inserisca all'interno della richiesta le modifiche da apportare all'oggetto Mailbox di cui l'utente desidera modificare/eliminare la condivisione, specificando i nuovi diritti che	Non soddisfatto

		hanno su quest'ultimo i membri del principale a cui si sta condividendo.	
R-117-F-2	Desiderabile	È necessario che il client riceva una risposta che contiene l'esito dell'operazione di modifica della condivisione di una cartella (compresa l'eliminazione di quest'ultima) con relativi parametri (come lo stato corrente del server, la lista degli identificativi delle cartelle modificate ed eventuali errori)	Non soddisfatto
R-118-F-3	Opzionale	Un client di posta elettronica utilizzato da un utente per interagire con il server deve avere la possibilità di mantenersi sincronizzato con gli ultimi aggiornamenti per quanto riguarda le email.	Non soddisfatto
R-119-F-3	Opzionale	È necessario che il client inserisca all'interno della richiesta l'identificazione delle capacità necessarie alla sincronizzazione delle email.	Non soddisfatto
R-120-F-3	Opzionale	È necessario che il client inserisca all'interno della richiesta le chiamate di metodo necessarie alla sincronizzazione delle email, con relativi parametri (sia quelli comuni, come l'identificativo dell'account da utilizzare, sia quelli specifici del caso) e un identificatore univoco associato.	Non soddisfatto
R-121-F-3	Opzionale	È necessario che il client inserisca all'interno della richiesta il suo stato corrente per quanto riguarda le email, con lo scopo di sincronizzarsi.	Non soddisfatto
R-122-F-3	Opzionale	È necessario che il client inserisca all'interno della richiesta il numero massimo di identificatori di email che desidera ricevere come risposta, con lo scopo di sincronizzarsi.	Non soddisfatto
R-123-F-3	Opzionale	È necessario che il client riceva una risposta che contiene le informazioni di cui ha bisogno per sincronizzarsi (come lo stato corrente del server, un flag booleano che indica se ci sono ulteriori cambiamenti nel server relativi ai dati di tipo Email, oltre a quelli già restituiti nella risposta corrente, e la lista delle email da creare/modificare/eliminare per sincronizzarsi)	Non soddisfatto
R-124-F-3	Opzionale	Un client di posta elettronica utilizzato da un utente per interagire con il server deve avere la possibilità di mantenersi sincronizzato con gli ultimi aggiornamenti per quanto riguarda le cartelle.	Non soddisfatto
R-125-F-3	Opzionale	È necessario che il client inserisca all'interno della richiesta l'identificazione delle capacità necessarie alla sincronizzazione delle cartelle.	Non soddisfatto
R-126-F-3	Opzionale	È necessario che il client inserisca all'interno della richiesta le chiamate di metodo necessarie alla sincronizzazione delle cartelle, con relativi parametri (sia quelli comuni, come l'identificativo dell'account da utilizzare, sia quelli specifici del caso) e un identificatore univoco associato.	Non soddisfatto
R-127-F-3	Opzionale	È necessario che il client inserisca all'interno della richiesta il suo stato corrente per quanto riguarda le cartelle, con lo scopo di sincronizzarsi.	Non soddisfatto
R-128-F-3	Opzionale	È necessario che il client inserisca all'interno della richiesta il numero massimo di identificatori di cartelle che desidera ricevere come risposta, con lo scopo di sincronizzarsi.	Non soddisfatto
R-129-F-3	Opzionale	È necessario che il client riceva una risposta che contiene le informazioni di cui ha bisogno per sincronizzarsi (come lo stato corrente del server, un flag booleano che indica se ci sono ulteriori cambiamenti nel server relativi ai dati di tipo Mailbox, oltre a quelli	Non soddisfatto

		già restituiti nella risposta corrente, e la lista delle cartelle da creare/ modificare/eliminare per sincronizzarsi)	
--	--	--------------------------------------------------------------------------------------------------------------------------	--

Tabella 9: Stato dei requisiti funzionali

4.1) Grafici riassuntivi

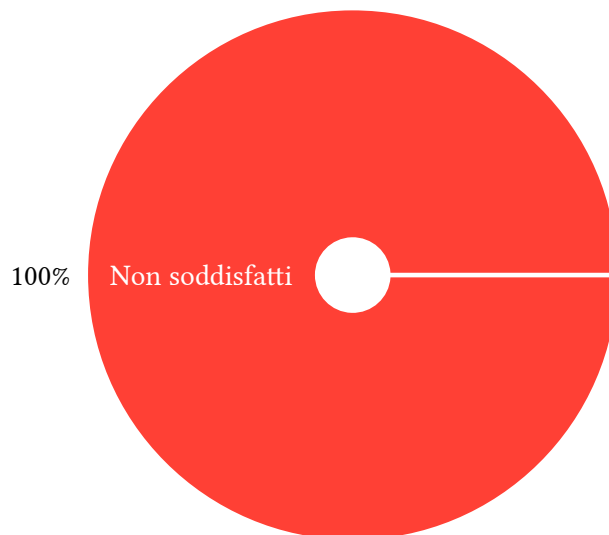


Figura 2: Stato dei requisiti funzionali totali

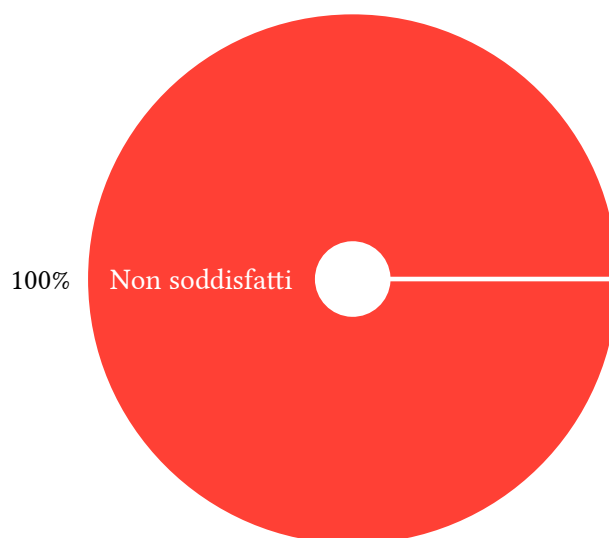


Figura 3: Stato dei requisiti funzionali obbligatori

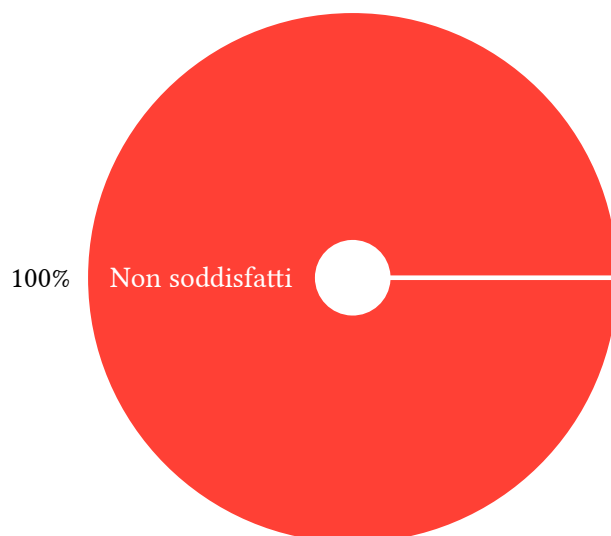


Figura 4: Stato dei requisiti funzionali desiderabili

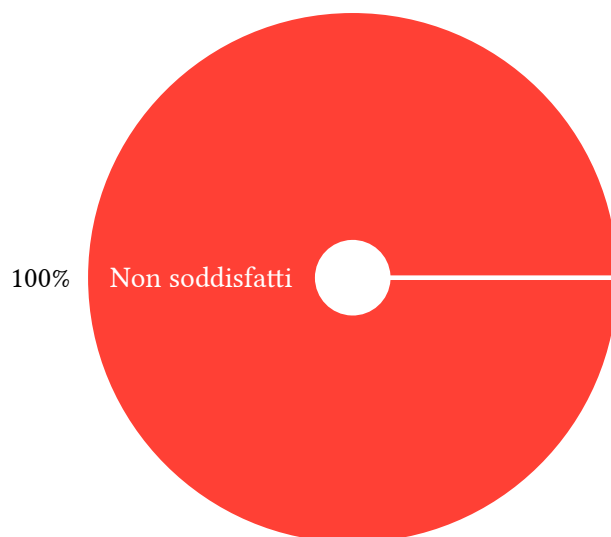


Figura 5: Stato dei requisiti funzionali opzionali