

# ANALISI DEI REQUISITI

**RF** = Requisiti Funzionali

**RNF** = Requisiti Non Funzionali

ID	Requisito Funzionale
<b>RF-1.0.1</b>	Il sistema deve permettere l'accesso tramite email e password, reindirizzando automaticamente l'utente all'interfaccia corretta in base al suo ruolo (Studente/Esercente).
<b>RF-1.0.2</b>	Il sistema deve impedire l'accesso se le credenziali non sono valide, fornendo un feedback visivo immediato.
<b>RF-1.3.1</b>	Il sistema deve recuperare e visualizzare in una vista dedicata i dati anagrafici e di contatto attualmente associati all'account dell'utente loggato.
<b>RF-1.3.2</b>	Il sistema deve rendere modificabili i campi informativi dell'utente (es. telefono, indirizzo, password), permettendo la sovrascrittura dei valori esistenti.
<b>RF-1.3.3</b>	Il sistema deve validare la correttezza formale dei nuovi dati inseriti (es. sintassi email valida, complessità password) prima di procedere al salvataggio.

<b>RF-1.3.4</b>	Il sistema deve aggiornare in modo persistente le informazioni dell'utente, garantendo che le modifiche siano immediatamente effettive per le sessioni future.
<b>RF-1.3.5</b>	Il sistema deve fornire un feedback visivo esplicito al termine dell'operazione (messaggio di conferma)
<b>RF-1.1.1</b>	L'utente deve poter aggiungere, rimuovere o modificare la quantità dei prodotti nel carrello prima dell'ordine.
<b>RF-1.1.2</b>	Il sistema deve garantire il vincolo di unicità dell'esercente: deve essere impossibile aggiungere un prodotto di un Esercente B se il carrello contiene già prodotti dell'Esercente A (a meno che il carrello non venga svuotato).
<b>RF-1.1.3</b>	Il sistema deve permettere all'utente di modificare la quantità dei singoli prodotti già inseriti o di rimuoverli completamente dalla lista.
<b>RF-1.1.4</b>	Il sistema deve ricalcolare e aggiornare visivamente il totale complessivo dell'ordine in tempo reale ad ogni singola variazione (aggiunta, rimozione o modifica quantità).

<b>RF-1.1.5</b>	Il sistema deve inibire l'aggiunta al carrello di prodotti che, seppur visibili nel menu, risultano temporaneamente non disponibili.
<b>RF-1.5.1</b>	Il sistema deve impedire l'avvio della procedura di ordine se il carrello è vuoto.
<b>RF-1.5.2</b>	Il sistema deve verificare in tempo reale lo stato dell'esercente al momento della conferma, bloccando l'ordine se il locale è chiuso o se mancano meno di 10 minuti alla chiusura.
<b>RF-1.5.3</b>	Il sistema deve validare la disponibilità di ogni singolo prodotto nel carrello prima di registrare l'ordine. In caso di prodotti non disponibili, il sistema deve impedire l'ordine, segnalare l'anomalia e aggiornare il carrello.
<b>RF-1.5.4</b>	Il sistema deve registrare l'ordine nel database associandolo all'utente e all'esercente, impostando lo stato iniziale: "In Attesa di Approvazione"
<b>RF-1.5.5</b>	Il sistema deve rendere immediatamente visibile il nuovo ordine nella dashboard dell'esercente destinatario.
<b>RF-1.5.6</b>	Il sistema deve garantire che la funzione di invio ordine sia accessibile

	esclusivamente agli utenti autenticati (Login effettuato).
<b>RF-1.6.1</b>	Il sistema deve verificare lo stato di giacenza attuale di tutti i prodotti presenti nel carrello nell'esatto momento in cui viene richiesto l'ordine.
<b>RF-1.6.2</b>	Il sistema deve segnalare la presenza o la mancanza dei prodotti.
<b>RF-1.7.1</b>	Il sistema deve fornire un campo di input testuale dedicato (es. area di testo) all'interno della vista del carrello per l'inserimento di richieste speciali.
<b>RF-1.7.2</b>	Il sistema deve trattare il campo commento come opzionale, permettendo all'utente di procedere con l'ordine anche se il campo viene lasciato vuoto.
<b>RF-1.7.3</b>	Il sistema deve accettare e validare l'input testuale dell'utente, imponendo un limite massimo di caratteri (es. 250) per garantire la leggibilità da parte dell'esercente.
<b>RF-1.7.4</b>	Il sistema deve associare in modo persistente il testo inserito all'oggetto "Ordine" corrente, garantendo che la

	nota sia visualizzabile nei dettagli dell'ordine ricevuto dall'Esercente.
<b>RF-1.8.1</b>	Il sistema deve recuperare e far visualizzare all'utente, l'elenco completo di tutti gli ordini, includendo sia quelli conclusi (passati) che quelli eventualmente in corso (attuali).
<b>RF-1.8.2</b>	Il sistema deve presentare la lista degli ordini ordinata cronologicamente (es. dal più recente al meno recente) per facilitare la consultazione.
<b>RF-1.8.3</b>	Per ogni voce in elenco, il sistema deve mostrare le informazioni sintetiche essenziali: Nome Esercente, Data/Ora dell'ordine, Importo Totale e Stato Corrente.
<b>RF-1.8.4</b>	Nel caso in cui non esistano ordini precedenti associati all'account (storico vuoto), il sistema deve visualizzare un messaggio informativo specifico (es. "Nessun ordine trovato") anziché una schermata vuota.
<b>RF-1.9.1</b>	Il sistema deve permettere all'utente di selezionare un singolo ordine dalla lista dello storico per espanderne le informazioni.

<b>RF-1.9.2</b>	Il sistema deve recuperare e visualizzare tutte le informazioni specifiche dell'ordine selezionato, inclusi: elenco dei prodotti, quantità per prodotto, prezzi singoli, totale complessivo e la data/ora di effettuazione.
<b>RF-1.9.3</b>	Il sistema deve mostrare chiaramente lo stato attuale di lavorazione dell'ordine.
<b>RF-2.0.1</b>	Il sistema deve permettere all'Esercente loggato di visualizzare l'elenco completo dei propri prodotti presenti nel menù.
<b>RF-2.0.2</b>	Il sistema deve permettere all'Esercente di modificare la <b>disponibilità</b> di un singolo prodotto (renderlo ordinabile o nascondere) in qualsiasi momento, indipendentemente dallo stato di apertura dell'esercizio.
<b>RF-2.0.3</b>	Il sistema deve permettere all'Esercente di <b>Aggiungere, Modificare</b> (nome, descrizione, prezzo) o <b>Eliminare</b> un prodotto <b>esclusivamente</b> quando lo stato dell'esercizio è impostato su "CHIUSO".
<b>RF-2.0.4</b>	Il sistema deve bloccare (disabilitare i comandi di interfaccia e rifiutare le richieste API) le operazioni di aggiunta, modifica strutturale ed eliminazione

	prodotti quando l'esercizio è in stato "APERTO".
<b>RF-2.0.5</b>	Il sistema deve validare i dati inseriti dall'Esercente prima del salvataggio, impedendo l'inserimento di prezzi negativi, nulli o di campi obbligatori vuoti.
<b>RF-2.1.1</b>	Il sistema deve permettere all'Esercente di visualizzare i dati attuali del proprio profilo pubblico (Nome, Descrizione, Indirizzo e Logo corrente).
<b>RF-2.1.2</b>	Il sistema deve permettere all'Esercente di modificare i campi testuali informativi (Nome, Descrizione, Indirizzo) che verranno mostrati agli utenti.
<b>RF-2.1.3</b>	Il sistema deve permettere all'Esercente di effettuare l'upload di una nuova immagine da utilizzare come Logo dell'attività.
<b>RF-2.1.4</b>	Il sistema deve validare il file immagine caricato, accettando esclusivamente formati grafici supportati (es. JPG, PNG) e rifiutando formati non validi (es. PDF) o file che superano la dimensione massima consentita.

<b>RF-2.1.5</b>	Il sistema deve impedire il salvataggio delle modifiche se uno o più campi obbligatori (es. Nome Esercizio) risultano vuoti.
<b>RF-2.1.6</b>	Il sistema deve aggiornare in tempo reale la scheda dell'esercente visibile nella lista pubblica (lato Cliente) dopo il salvataggio delle modifiche.
<b>RF-2.2.1</b>	Il sistema deve permettere all'Esercente di visualizzare una lista dedicata contenente esclusivamente gli ordini nello stato "In attesa di approvazione", completi di dettagli (elenco piatti, totale e orario ricezione).
<b>RF-2.2.2</b>	Il sistema deve permettere all'Esercente di <b>Accettare</b> un ordine specifico. A seguito di questa azione, il sistema deve aggiornare lo stato dell'ordine in "Accettato" e abilitare il passaggio alla fase successiva (preparazione).
<b>RF-2.2.3</b>	Il sistema deve permettere all'Esercente di <b>Rifiutare</b> un ordine specifico. A seguito di questa azione, il sistema deve aggiornare lo stato dell'ordine in "Rifiutato" e impedire ulteriori modifiche su di esso.
<b>RF-2.2.4</b>	Il sistema deve rendere immediatamente visibile il cambio di stato (da "In attesa" a "Accettato" o



	"Rifiutato") nella scheda dell'ordine visualizzata dal Cliente.
<b>RF-2.3.1</b>	Il sistema deve permettere all'Esercente di visualizzare l'elenco degli ordini attivi in lavorazione.
<b>RF-2.3.2</b>	Il sistema deve permettere all'Esercente di avanzare lo stato di un ordine da "Accettato" a "In Preparazione" per segnalare l'inizio della produzione.
<b>RF-2.3.3</b>	Il sistema deve permettere all'Esercente di avanzare lo stato di un ordine da "In Preparazione" a "Pronto per il ritiro".
<b>RF-2.3.4</b>	Al momento del passaggio allo stato "Pronto per il ritiro", il sistema deve <b>generare automaticamente un codice univoco</b> di ritiro e renderlo visibile al Cliente nei dettagli dell'ordine.
<b>RF-2.3.5</b>	Il sistema deve fornire all'Esercente una funzionalità per <b>verificare il codice di ritiro</b> fornito dal Cliente. Se il codice è corretto, il sistema deve permettere il passaggio allo stato finale "Ritirato".
<b>RF-2.3.6</b>	Il sistema deve aggiornare in tempo reale la vista del Cliente ad ogni cambio

	di stato ("In Preparazione", "Pronto", "Ritirato").
<b>RF-2.3.7</b>	Il sistema deve permettere all'Esercente di chiudere forzatamente un ordine impostando lo stato su "Non Ritirato" nel caso in cui il Cliente non si presenti entro l'orario di chiusura.
<b>RF-1.2.1</b>	Il sistema deve permettere al Cliente loggato di visualizzare il listino completo di un esercente selezionato, mostrando per ogni prodotto: nome, prezzo e descrizione.
<b>RF-1.2.2</b>	Il sistema deve presentare i prodotti organizzati visivamente per <b>categorie</b> (es. Primi, Secondi, Bevande) per facilitare la consultazione.
<b>RF-1.2.3</b>	Il sistema deve permettere al Cliente di <b>filtrare</b> i prodotti visualizzati in base alla categoria.
<b>RF-1.2.4</b>	Il sistema deve distinguere graficamente i prodotti <b>Disponibili</b> da quelli <b>Non Disponibili</b> (esauriti), inibendo la possibilità di aggiungere questi ultimi al carrello.
<b>RF-1.2.5</b>	Il sistema deve permettere al Cliente di selezionare un prodotto per

	visualizzarne la scheda di dettaglio, che deve includere informazioni aggiuntive come ingredienti e allergeni.
<b>RF-1.2.6</b>	Nel caso in cui l'esercente selezionato sia in stato "Chiuso", il sistema deve mostrare il menu in modalità "Sola Lettura", visualizzando un avviso informativo e disabilitando tutte le funzioni di acquisto.
<b>RF-1.2.7</b>	Nel caso in cui il menu dell'esercente risulti vuoto (lista prodotti vuota), il sistema deve fornire un feedback visivo esplicito all'utente ("Nessun prodotto disponibile").
<b>RF-1.10.1</b>	Il sistema deve permettere la registrazione di un nuovo utente richiedendo l'inserimento di nome, cognome, email istituzionale e password.
<b>RF-1.10.2</b>	Il sistema deve validare il dominio dell'indirizzo email inserito, consentendo la registrazione esclusivamente agli indirizzi appartenenti al dominio universitario
<b>RF-1.10.3</b>	Il sistema deve verificare che l'indirizzo email non sia già associato a un

	account esistente (controllo di unicità), impedendo la creazione di duplicati.
<b>RF-1.10.4</b>	Nel caso di tentativo di registrazione con email già esistente, il sistema deve bloccare l'operazione e mostrare un messaggio che suggerisce all'utente di effettuare il login o recuperare la password.
<b>RF-1.10.5</b>	Il sistema deve validare la correttezza formale dei dati inseriti (es. campi obbligatori mancanti, password troppo breve) fornendo un messaggio di errore specifico in corrispondenza del campo non valido.
<b>RF-1.10.6</b>	Al termine di una registrazione conclusa con successo, il sistema deve effettuare automaticamente il login dell'utente, rendendo attiva la sessione senza richiedere un nuovo inserimento delle credenziali.
<b>RF-1.4.1</b>	Il sistema deve permettere al Cliente loggato di visualizzare l'elenco completo degli esercenti convenzionati presenti nella piattaforma.
<b>RF-1.4.2</b>	Il sistema deve mostrare, per ogni esercente in lista, le informazioni riepilogative essenziali: Nome

	dell'attività, Logo/Immagine e breve descrizione.
<b>RF-1.4.3</b>	Il sistema deve indicare visivamente lo stato di apertura corrente di ogni esercente (es. Aperto/Chiuso) direttamente nella lista, per orientare la scelta dell'utente.
<b>RF-1.4.4</b>	Il sistema deve filtrare la lista mostrando esclusivamente i ristoranti marcati come "Attivi" nel database, nascondendo quelli dismessi o sospesi.
<b>RF-1.4.5</b>	Il sistema deve permettere all'utente di selezionare un esercente dalla lista per accedere alla visualizzazione del relativo menu (collegamento al caso d'uso Consultazione Menu).

---

<b>ID</b>	<b>Requisito Non Funzionale</b>
<b>RNF-1.1</b> (Tecnologici e Architetture)	Il Backend deve essere sviluppato in PHP nativo senza l'utilizzo di framework (es. Laravel). Il Frontend deve essere sviluppato in HTML, CSS e JavaScript (è consentito jQuery), senza framework reattivi come React o Angular.

<b>RNF-1.2</b> (Tecnologici e Architetture)	L'applicazione lato client deve rispettare rigorosamente il pattern architetturale Model-View-Presenter (MVP), separando la logica di presentazione dalla struttura HTML e dai dati.
<b>RNF-1.3</b> (Tecnologici e Architetture)	La comunicazione tra Frontend e Backend deve avvenire esclusivamente tramite RESTful API asincrone, utilizzando il formato JSON per lo scambio dati.
<b>RNF-1.4</b> (Tecnologici e Architetture)	È vietata la generazione di codice HTML lato server (Server-Side Rendering). Il PHP deve restituire solo dati grezzi; la costruzione dell'interfaccia è demandata al JavaScript client-side.
<b>RNF-1.5</b> (Tecnologici e Architetture)	Il sistema deve essere fornito all'interno di container Docker configurati (file docker-compose.yml), garantendo l'esecuzione su qualsiasi ambiente ospite senza installazioni manuali.
<b>RNF-1.6</b> (Tecnologici e Architetture)	Il codice sorgente e la documentazione devono essere gestiti tramite repository GitHub, dimostrando un corretto flusso di lavoro collaborativo.
<b>RNF-2.1</b> (Sicurezza)	Il sistema deve gestire l'autenticazione tramite Token (Stateless) generati dal server e memorizzati nel client. È vietato l'uso delle sessioni server-side native di PHP.
<b>RNF-2.2</b> (Sicurezza)	Le password degli utenti devono essere memorizzate nel database esclusivamente sotto forma di hash crittografico sicuro.

<b>RNF-2.3</b> (Sicurezza)	Tutti gli endpoint API che permettono la modifica dei dati o l'accesso a informazioni personali devono richiedere un token di autenticazione valido nell'header della richiesta HTTP.
<b>RNF-3.1</b> (Usabilità e Interfaccia)	L'interfaccia utente deve essere Responsive, adattando automaticamente il layout per dispositivi mobili, desktop e tablet.
<b>RNF-3.2</b> (Usabilità e Interfaccia)	Il sistema deve fornire un feedback visivo in risposta a qualsiasi azione dell'utente (es. messaggi di errore, conferma inserimento carrello).
<b>RNF-4.1</b> (Affidabilità e Correttezza)	Il sistema deve garantire la consistenza dei dati in caso di accessi concorrenti, impedendo l'ordine di prodotti esauriti se richiesti simultaneamente da più utenti.
<b>RNF-4.2</b> (Affidabilità e Correttezza)	Il Backend deve implementare una validazione ridondante di tutti i dati in ingresso, non affidandosi esclusivamente ai controlli effettuati dal JavaScript lato client.
<b>RNF-5.1</b> (Dominio)	Il sistema non deve gestire transazioni monetarie elettroniche; il pagamento è previsto esclusivamente in loco al momento del ritiro.

---

## MATRICE DI TRACCIABILITA'

Area Pubblica (Cliente UniPR)

<b>ID Caso d'Uso</b>	<b>Nome Caso d'Uso</b>	<b>Requisiti Funzionali (RF)</b>	<b>Requisiti Non Funzionali (RNF)</b>
<b>UC-1.0</b>	<b>Login</b>	RF-1.0.1, RF-1.0.2	RNF-2.1 (Generazione Token), RNF-2.2 (Hash Password), RNF-1.3 (API JSON), RNF-3.2 (Feedback)
<b>UC-1.10</b>	<b>Registrazione</b>	RF-1.10.1, RF-1.10.2, RF-1.10.3, RF-1.10.4, RF-1.10.5, RF-1.10.6	RNF-2.2 (Hash Password), RNF-4.2 (Validazione Backend), RNF-3.2 (Feedback)
<b>UC-1.3</b>	<b>Gestire Profilo</b>	RF-1.3.1, RF-1.3.2, RF-1.3.3, RF-1.3.4, RF-1.3.5	RNF-2.1 (Token richiesto), RNF-2.3 (Protezione API PUT), RNF-4.2 (Validazione dati)
<b>UC-1.4</b>	<b>Consultare Lista Esercenti</b>	RF-1.4.1, RF-1.4.2, RF-1.4.3, RF-1.4.4, RF-1.4.5	RNF-2.1 (Token richiesto), RNF-3.1 (Responsive Mobile), RNF-1.4 (No Server HTML), RNF-1.3 (API GET)
<b>UC-1.2</b>	<b>Consultazione Menu</b>	RF-1.2.1, RF-1.2.2, RF-1.2.3, RF-1.2.4, RF-1.2.5, RF-1.2.6, RF-1.2.7	RNF-2.1 (Token richiesto), RNF-4.1 (Disponibilità Real-time), RNF-3.1 (Responsive), RNF-1.2 (MVP View)
<b>UC-1.1</b>	<b>Gestire Carrello</b>	RF-1.1.1, RF-1.1.2, RF-1.1.3, RF-1.1.4, RF-1.1.5	RNF-2.1 (Token richiesto), RNF-1.2 (Logica JS Presenter), RNF-3.2 (Feedback immediato), RNF-1.4 (Calcolo locale/API)



<b>UC-1.7</b>	<b>Aggiungere Commento</b>	RF-1.7.1, RF-1.7.2, RF-1.7.3, RF-1.7.4	RNF-2.1 (Token richiesto), RNF-4.2 (Validazione lunghezza Backend)
<b>UC-1.5</b>	<b>Ordinare</b>	RF-1.5.1, RF-1.5.2, RF-1.5.3, RF-1.5.4, RF-1.5.5, RF-1.5.6	RNF-2.1 (Token richiesto), RNF-2.3 (Protezione API POST), RNF-4.1 (Transazione/Concorrenza), RNF-5.1 (Pagamento in loco)
<b>UC-1.6</b>	<b>Verificare disponibilità prodotti</b>	RF-1.6.1, RF-1.6.2	RNF-2.1 (Token richiesto), RNF-4.1 (Integrità Dati concorrenti)
<b>UC-1.8</b>	<b>Visualizzare Storico ordini</b>	RF-1.8.1, RF-1.8.2, RF-1.8.3, RF-1.8.4	RNF-2.1 (Token richiesto), RNF-1.3 (API JSON)
<b>UC-1.9</b>	<b>Visualizzare dettagli ordine</b>	RF-1.9.1, RF-1.9.2, RF-1.9.3	RNF-2.1 (Token richiesto), RNF-3.1 (Layout Responsive)

### Area Gestionale (Esercente)

<b>ID Caso d'Uso</b>	<b>Nome Caso d'Uso</b>	<b>Requisiti Funzionali (RF)</b>	<b>Requisiti Non Funzionali (RNF)</b>
<b>UC-2.1</b>	<b>Modifica Info Esercente</b>	RF-2.1.1, RF-2.1.2, RF-2.1.3, RF-2.1.4, RF-2.1.5, RF-2.1.6	RNF-2.1 (Token richiesto), RNF-2.3 (Protezione API), RNF-4.2 (Validazione Formato File), RNF-3.2 (Feedback salvataggio)

<b>UC-2.0</b>	<b>Gestire Menu</b>	RF-2.0.1, RF-2.0.2, RF-2.0.3, RF-2.0.4, RF-2.0.5	RNF-2.1 (Token richiesto), RNF-4.2 (Validazione Prezzi), RNF-1.3 (CRUD API)
<b>UC-2.2</b>	<b>Gestire Richieste Ordini</b>	RF-2.2.1, RF-2.2.2, RF-2.2.3, RF-2.2.4	RNF-2.1 (Token richiesto), RNF-3.2 (Feedback immediato), RNF-3.1 (Desktop View), RNF-4.1 (Consistenza Stato)
<b>UC-2.3</b>	<b>Gestire Stato Ordini</b>	RF-2.3.1, RF-2.3.2, RF-2.3.3, RF-2.3.4, RF-2.3.5, RF-2.3.6, RF-2.3.7	RNF-2.1 (Token richiesto), RNF-1.3 (Aggiornamento asincrono), RNF-2.3 (Protezione Azioni), RNF-4.1 (Sequenza stati corretta)