

# Regression Testing

## QuantuMoonLight

### Team

Matteo Cicalese - 0522501516  
Luca Contrasto – 0522501512

### Repo

[github.com/CicaMatt/QML-IGES](https://github.com/CicaMatt/QML-IGES)  
[github.com/Robertales/QuantuMoonLight](https://github.com/Robertales/QuantuMoonLight)

## Sommario

1.	Regression testing.....	3
1.1	Approccio.....	3
1.2	Esito.....	4

# 1. Regression testing

## 1.1 Approccio

Il processo di testing di regressione è stato eseguito sulla [fork](#) creata per lo sviluppo del progetto, contenente tutte le Change Request sviluppate. La decisione di mantenere tutte le CR in un'unica branch e non effettuare suddivisioni è dettata dal fatto che il lavoro necessario allo sviluppo delle change request presentava un certo grado di correlazione, in seguito al quale, adottando un approccio con branch separate, sarebbe stato necessario il completamento di una CR prima di iniziare a lavorare su un'altra. Di conseguenza, abbiamo ritenuto che la suddivisione in branch separati avrebbe rappresentato più un ostacolo che un vantaggio.

In particolare, la CR2 (download di modelli e altri file per gli esperimenti) è collegata alla CR1 (aggiornamento delle librerie per il download di modelli) per l'abilitazione del download di modelli quantistici. Allo stesso modo, la CR2 è correlata alla CR3 (cifratura e decifratura dei file scaricati), poiché una gestione corretta della cifratura e decifratura era necessaria durante la generazione e il download dei file degli esperimenti.

Tuttavia, ogni modifica effettuata alla piattaforma è sempre stata effettuata in maniera atomica per una singola CR. Inoltre, per tenere traccia delle CR rispetto alle quali venivano fatti dei progressi, i commit di sviluppo presentano una descrizione seguita dal contrassegno CR#.

Dal punto di vista pratico, seguendo le assunzioni precedenti, i test sono stati eseguiti in tre fasi. Inizialmente, sono stati eseguiti i test di unità, seguiti dai test di integrazione in maniera bottom-up e infine dai test di sistema. Questi test includono sia quelli preesistenti, adattati e migliorati per riflettere le modifiche introdotte dalle CR, sia quelli specificamente sviluppati per le nuove funzionalità a livello di unità e sistema.

È importante notare che, durante l'adeguamento dei test esistenti, descritto nel documento **Pre-CR System Testing**, sono stati affrontati principalmente problemi legati a mancanza di assert nei test e discrepanza fra implementazione e testing riscontrata durante l'analisi del codice.

Tali problematiche non sono direttamente associate alle funzionalità introdotte dalle CR, in quanto erano presenti prima dell'implementazione delle stesse.

In conclusione, il processo di testing ha garantito una copertura completa delle funzionalità esistenti e nuove, assicurando la robustezza e l'integrità del sistema nell'ambito delle modifiche apportate.

## 1.2 Esito

Dopo aver apportato le necessarie correzioni per risolvere i problemi individuati durante i test delle Change Request e descritti nel documento **Post-CR Test Incident Report**, è stata condotta una revisione completa dei test, includendo quelli precedenti alle CR, come precedentemente specificato. Tale revisione è stata eseguita tramite l'utilizzo del file *test\_all*, il quale fa uso della funzionalità di autodiscovery dei test, come dettagliato nel **Pre-CR System Testing**. Questo approccio consente di eseguire automaticamente tutti i test di unità e integrazione presenti nel repository, che in totale sono 53.

L'esecuzione del file *test\_all* ha restituito un esito positivo, confermando che l'introduzione delle nuove funzionalità non ha generato nuovi difetti o errori nel sistema, con tutti i test che hanno avuto esito positivo. Questo risultato sottolinea l'efficacia delle correzioni apportate e la stabilità complessiva del software.