Relazione APL 2022/2023

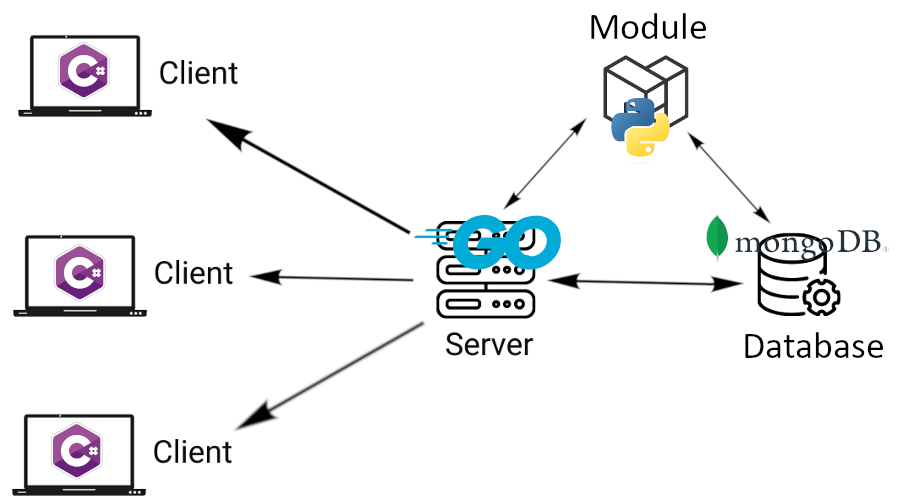
Luca Dimarco - Antonino Salemi

# Consegna

L’idea alla base del progetto consiste in un videogioco a quiz, basato su minigiochi. L’applicazione realizzata consente di giocare a diverse modalità suddivise per categoria, su cui verteranno le domande. Le modalità sono:

* Classico: 30 domande con 4 opzioni di risposta. Se si sbaglia il gioco finisce.
* A tempo: 1 minuto di gioco e 30 domande.
* Vero o Falso: 30 domande con vero o falso come opzioni di risposta. Se si sbaglia il gioco finisce.
* Impiccato: 5 vite per trovare la risposta alla domanda indovinando le sue lettere. Ogni tentativo errato toglie una vita.

Oltre al gioco in sé e per sé, l’applicazione consente tutta una serie di funzionalità aggiuntive, tra le quali un negozio dove acquistare categorie addizionali, il profilo dove visualizzare o modificare i propri dati e statistiche, e le classifiche dove vengono mostrati i migliori risultati ottenuti da tutti i giocatori.



# Client

Il Client è stato scritto interamente usando il linguaggio di programmazione C#. Il programma è stato organizzato in Windows Forms, con l’ausilio di User Controls.

Il codice C# è stato strutturato in namespace. Il namespace relativo ai Forms comprende le schermate principali del programma, quindi le homepage, il gioco, le schermate di registrazione e accesso, il negozio e le classifiche. Mentre, tutte le sottosezioni del programma sono state realizzate con degli User Control che si sostituiscono dinamicamente (ad es. nella schermata di gioco è stato creato un unico modello con testo e pulsanti di risposta, che poi è stato replicato per tutte le domande e inserito nel form del gioco). I restanti namespaces sono relativi a strutture dati, funzioni utils e relativi alla connessione.

L’unica libreria aggiuntiva è NewtonsoftJson.

Il primo form che si apre all’avvio dell’applicazione è il form del Login, dove l’utente può effettuare l’accesso inserendo i propri dati o può selezionare un link di reindirizzamento al form di registrazione, qualora non fosse ancora registrato. Entrambi i form (Login e Registrazione), prevedono una serie di campi, ognuno con i relativi controlli sulla correttezza degli stessi. Inoltre, ci si può autenticare come amministratore o come player. L’amministratore e il player avranno delle homepage differenti.

## AdminHome

L’home relativa all’amministratore è organizzata in tab controls. Le pagine della tab controls sono:

* Categorie: per visualizzare, modificare il prezzo o cancellare le categorie presenti nel database.
* Domande: per visualizzare, modificare o cancellare le domande presenti nel database.
* Nuova Domanda: per inserire una nuova domanda nel database, selezionando una categoria già esistente o creandone una nuova.
* Utenti: per visualizzare, modificare i crediti o cancellare gli utenti presenti nel database.
* Statistiche: per visualizzare dei grafici riguardanti le partite giocate, i punteggi, i punteggi medi suddivisi per categorie e le categorie più acquistate.

Anche in questo caso sono stati inseriti tutti i controlli. Ad esempio, non è possibile modificare i codici di categorie o domande; se viene eliminata una categoria si eliminano tutte le domande ad essa associate; quando viene creata una nuova categoria essa viene inserita solamente se ha domande sufficienti (31 nello specifico).

## PlayerHome (Home)

L’home relativa ai player presenta 5 pulsanti:

* Gioca: cliccando su gioca si apre un form di selezione della modalità e successivamente della categoria. Una volta scelti entrambi inizia la partita. Nella modalità classica e a tempo, vi è la possibilità di usufruire di aiuti.
* Profilo: nel profilo, l’utente può modificare i campi relativi al nome utente, all’email e alla password, o semplicemente visualizzarli, così come il proprio livello, i crediti e la propria esperienza.
* Negozio: nel negozio vengono visualizzati i crediti disponibili e le categorie acquistabili.
* Classifiche: le classifiche sono suddivise in pagine di una tab control. Ogni pagina è relativa ad una categoria diversa dove viene visualizzata la tabella con i migliori risultati dei vari giocatori.
* Esci: il pulsante esci serve semplicemente per effettuare il logout.

Come per il resto del programma, anche in questa sezione vi sono tutti i controlli su ogni funzione. Infine, nella schermata del gioco è possibile anche visualizzarne le regole.

# Server

Il server è stato realizzato nel linguaggio di programmazione Go.

È strutturato in package e package principali sono service, utils e data.

In service ci sono funzioni per quanto riguarda servizi dell’utente, dell’admin e del gioco.

In utils, invece, ci sono le funzioni che permettono le modifiche di dati nel database, il token generator e le funzioni utili per l’instaurazione della comunicazione tra server e client e tra server e Python.

Infine, il package data contiene le struct utilizzate nel programma.

Il server, inoltre, prevede una funzione di startup. Tale funzione crea le collections necessarie per il funzionamento del programma e i dati principali di default all’interno del database.

# Comunicazione Client - Server

La comunicazione tra server e client è stata realizzata tramite SocketTCP. Vengono inviati dei messaggi in bytes. Innanzitutto, viene inviata la lunghezza del messaggio, in maniera tale da poter calcolare la dimensione del messaggio in arrivo. Il messaggio contenente la dimensione ha una lunghezza fissa di 16 bytes. Il messaggio che trasporta il contenuto, invece, è strutturato in maniera seguente: Numero di servizio | Spazio | Corpo | Ritorno a capo.

Si fa uso di id di protocollo per richiedere i diversi servizi e i messaggi sono organizzati in Json, tramite metodi di serializzazione e deserializzazione.

Al momento del login, il server assegna un token univoco al client. Tramite il token e gli id, il server è in grado di identificare il tipo di operazione richiesta dal client, se si tratta di gestione, inserimento o richiesta di dati dal database. Sempre tramite il token il server riconosce un utente normale da un amministratore.

# Modulo Python

Il modulo Python sostanzialmente realizza, su richiesta del client, tramite il server, i grafici relativi alle statistiche. La funzione main rimane in ascolto di eventuali messaggi inviati dal server ed in base a ciò che riceve, realizza il grafico apposito. Il main chiama uno switch case, nel quale vengono creati degli oggetti della classe stats, dove sono presenti le funzioni per generare i grafici.

I grafici vengono generati tramite i dati prelevati dal database e mediante l’ausilio di librerie quali pymongo, matplotlib e pandas. Vi sono due sottocategorie di grafici: istogrammi e tabelle.

I grafici realizzati sono quelli descritti nella tab statistiche dell’AdminHome e nelle tab classifiche di PlayerHome.

# Comunicazione Server-Python

Il server invia a Python una richiesta contenente il tipo di grafico voluto. La richiesta perviene dal client e contiene un id e la categoria scelta.

La risposta è strutturata in maniera analoga a quanto descritto per la comunicazione tra client e server, ovvero in messaggi di byte contenenti la lunghezza e il contenuto.

L’immagine generata da Python e passata dal server al client, viene poi rigenerata in quest’ultimo tramite una funzione di conversione Base64ToImage.

# Database

Per il database, la scelta progettuale è stata di utilizzare il database non relazionale MongoDB, strutturato in collections. Il database si chiama “apl\_database”. Le collections create sono:

* utenti: contiene i documenti di ogni utente registrato e dell’admin.
* categories: contiene i documenti di ogni categoria creata dall’admin.
* questions: contiene i documenti di ogni domanda presente nel gioco.
* records: contiene i documenti dei record realizzati da ogni utente.
* stats: contiene i documenti sulle statistiche legate ad ogni categoria.
* payments: contiene i documenti relativi agli acquisti di un utente.