Scelte Implementative

Ingegneria del Software

Stefano Ravetta, Simone Renzo, Matteo Scarpone, Mattia Tollari

29 Febbraio, 2016

1 Introduzione

1.1 Informazioni su questo ducumento

In questo documento verranno trattate e discusse le scelte implementative che sono state prese per la realizzazione del progetto "Team Diary Management" e le motivazioni delle stesse .

1.2 Contenuti

Si porra' l'attenzione in modo particolare su

- Software per la creazione dei diagrammi
- Software per la stesura della documentazione
- Linguaggio di programmazione
- Formato e struttura della base di dati
- Librerie per la creazione dell'interfaccia utente
- Struttura dei sistemi di sicurezza adottati

2 Linguaggio

2.1 Introduzione

É stato scelto il linguaggio di programmazione Python per alcune caratteristiche piuttosto interessanti che lo possono far risaltare nell'ecosistema dei linguaggi di programmazione odierno.

2.2 Alto livello

Python consente di programmare ad un livello molto alto: con poche righe riesce a gestire in modo semplice e diretto operazioni complesse senza che il codice risulti troppo complesso da leggere e comprendere.

Questo ha permesso al team di sviluppo di concentrarsi di più sulle parti più astratte e progettuali (come i design pattern e la rimozione di vari code smell) che su questioni più vicine alla comprensione del funzionamento del linguaggio.

2.3 Filosofia

Una parte di un famoso "easter egg" inserito nell'interprete Python (che può apparire appena si prova ad importare la libreria "this") recita:

> If the implementation is hard to explain, it's a bad idea.
>
> If the implementation is easy to explain, it may be a good idea.

Queste righe sembrano denotare in modo sarcastico la filosofia del linguaggio. Uno degli obiettivi del linguaggio è appunto far sì che il codice scritto abbia il miglior compromesso tra stringatezza e comprensibilità. Si intende quindi che codice estremamente compatto non è sinonimo di efficienza a lungo termine in quanto provocherà quasi sicuramente un dilatamento delle tempistiche sia fasi di sviluppo successive sia in fase di testing ed ottimizzazione. Un esmpio che può far notare il modo in cui questo tipo di filosofia inserito nella struttura del linguaggio possa effettivamente influire sul codice è quello dell'indentazione: per forzare il programmatore ad indentare sempr e il codice (e a farlo in modo corretto) questo linguaggio non usa caratteri di begin ed end nella definzione di classi, funzioni e procedure, espressioni condizionali, espressioni iterative (come fa ad esempio Java con le parentesi graffe) ma usa solo l'indentazione. In tal modo da una parte i programmatori che di norma non indentano il loro codice risulteranno "costretti" a farlo, e dall'altra i programmatori che gia‘ di norma indentano il proprio codice non dovranno inserire altri simboli per esprimere ciò che concettualmente può essere già presente nell'indentazione.

Secondo il nostro team di sviluppo questa caratteristica risultava molto interessante per la progettazione di un progetto in cui èrichiesta particolare attenzione alla qualità del codice prodotta. Inoltre a livello pratico tutto ciò èrisultato addirittura comodo dato che ogni membro del gruppo aveva un proprio stile di stesura del codice e si eòtata la necessità di uno stile univoco con cui lavorare.

References