



ARGO

Verbale Riunione 09-03-2024

Gruppo Argo

Informazioni sul documento

Versione	1.0.0
Approvazione	Sebastiano Lewental
Uso	Interno
Distribuzione	Prof. Tullio Vardanega Prof. Riccardo Cardin Gruppo Argo



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Registro delle modifiche

Versione	Data	Autori/Autrici	Ruolo	Descrizione
1.0.0	15-03-2024	Sebastiano Lewental	<i>Responsabile di Progetto</i>	Approvazione
0.1.0	14-03-2024	Mattia Zecchinato, Martina Dall'Amico	<i>Verificatore</i>	Revisione completa
0.0.1	11-03-2024	Riccardo Cavalli	<i>Redattore</i>	Stesura del documento

Indice

1	Informazioni	3
1.1	Descrizione	3
1.2	Partecipanti	3
2	Riunione	4
2.1	Ordine del Giorno	4
2.2	Discussione e decisioni	4
2.2.1	Organizzazione dei repository GitHub	4
2.2.2	Github deep dive	5
2.2.2.1	Labeling	5
2.2.2.2	Merging	5
2.2.2.3	Linking issue - pull request	5
2.2.2.4	Review	6
2.2.2.5	Project board	6
2.2.3	LaTeX e Docker	6
3	Todo / In Progress	7

1 Informazioni

- **Inizio incontro:** 9:00
- **Fine incontro:** 12:00
- **Tipo incontro:** remoto (Discord)

1.1 Descrizione

Durante l'incontro, il gruppo ha discusso la configurazione e l'organizzazione dell'ambiente di lavoro su GitHub. Inoltre, si è registrata una sessione collaborativa di apprendimento, condotta da Mattia Zecchinato, per allineare i componenti del team sulle funzionalità base di LaTeX e Docker. Parallelamente a ciò, il gruppo ha definito alcune modifiche da apportare al template LaTeX.

1.2 Partecipanti

- **Argo:**
 - Tommaso Stocco
 - Marco Cristo
 - Raul Pianon
 - Sebastiano Lewental
 - Mattia Zecchinato
 - Martina Dall'Amico
 - Riccardo Cavalli

2 Riunione

2.1 Ordine del Giorno

- Organizzazione dei Repository GitHub;
- Test degli strumenti di GitHub attraverso esempi concreti;
- Panoramica sulle funzionalità base di LaTeX e Docker.

2.2 Discussione e decisioni

2.2.1 Organizzazione dei repository GitHub

Il gruppo ha confermato la decisione di convertire l'account GitHub in un'organizzazione, a cui ciascun componente può unirsi come membro ufficiale. Inoltre, si è deciso di associare tre repository all'organizzazione Argo:

- Docs, contenente la documentazione di progetto (inclusi i sorgenti e i template) da verificare, validare e versionare;
- argo-swe.github.io, contenente la pagina di presentazione accessibile via web e la documentazione in formato pdf;
- Un repository contenente il codice del prodotto software inerente al capitolato scelto.

In un primo momento, il repository Docs conterrà solamente la cartella Candidatura, al cui interno saranno presenti i seguenti file e subdirectory:

- Lettera di candidatura per l'aggiudicazione dell'appalto;
- Stima dei costi e assunzione impegni: documento composto da un preventivo dei costi, una scadenza di consegna, una distribuzione delle ore in base ai ruoli di progetto, un elenco dei rischi e delle azioni di mitigazione;
- Valutazione dei capitolati: documento di analisi dei capitolati disponibili.
- Verbalì:
 - Interni: riunioni circoscritte ai membri del gruppo fornitore;
 - Esterni: incontri tra cliente e fornitore.

Si è deciso inoltre di creare una "Branch protection rule" in modo da bloccare i commit diretti verso il ramo principale. Di conseguenza, ogni membro del team dovrà lavorare in un ramo di sviluppo o di feature, per poi aprire una pull request e richiedere la validazione delle modifiche al verificatore. All'interno del Repository argo-swe.github.io è stata configurata la sezione GitHub Pages, per rendere accessibile via web la pagina di presentazione. Il processo di build and deployment avviene a partire dal main branch, poiché protetto da una Branch protection rule.

2.2.2 Github deep dive

2.2.2.1 Labeling Il gruppo ha avviato una prima fase di testing degli strumenti offerti da GitHub, in particolare gli issue e le pull request. Per gli issue si è deciso di utilizzare prevalentemente le seguenti label:

- “Enhancement”: nuove feature o miglioramenti;
- “Bug”: problemi da risolvere con priorità immediata;
- Etichette personalizzate, da aggiornare durante tutto l’avanzamento del progetto, come ad esempio la label “Documentazione”.

Le issue e le pull request vengono associate a una milestone (pietra miliare), così da tenere traccia di obiettivi intermedi e scadenze.

2.2.2.2 Merging Per quanto riguarda le pull request, si sono discusse le modalità di merging e chiusura delle pull request

- “Create a merge commit”: tutti i commit della pull request vengono aggiunti al ramo base tramite un merge commit;
- “Squash and merge”: tutti i commit della pull request vengono compressi in un unico commit e aggiunti al ramo base;
- “Rebase and merge”: tutti i commit della pull request vengono aggiunti individualmente al ramo base senza un merge commit.

Il gruppo ha deciso di privilegiare la seconda opzione, “squash and merge”, in modo da mantenere la cronologia del progetto più pulita. Inoltre, si è discussa la possibilità di convertire pull request in draft e viceversa. Nel caso in cui un verificatore rilevi la necessità di apportare delle modifiche prima di approvare una richiesta, è possibile convertire la pull request in una bozza “work in progress”. Nessuno può integrare le modifiche nel ramo base finché la richiesta è ancora in corso. Quando le modifiche sono pronte per la verifica, la bozza deve essere contrassegnata come “ready for review”, così da convertirla nuovamente in pull request.

2.2.2.3 Linking issue - pull request Nella sezione “Development” delle pull request è possibile legare uno o più issue, che verranno chiusi in modo automatico una volta effettuato con successo il merge nel ramo base. In alternativa, Git offre la possibilità di collegare un issue a una pull request inserendo, nel messaggio di commit o nella descrizione della pull request, una delle seguenti keyword seguita dall’ID univoco dell’issue:

- close;
- closes;
- closed;
- fix;
- fixes;
- fixed;

- resolve;
- resolves;
- resolved.

2.2.2.4 Review La sezione relativa alle pull request è sostanzialmente divisa in tre parti: l'area di conversazione, la history dei commit e l'elenco dei file modificati. Per ogni file verificato, il revisore può selezionare una porzione di codice e lasciare un commento, positivo o negativo. Dopo aver esaminato la richiesta, il revisore deve "Completare la review", lasciando un commento riassuntivo e selezionando una delle seguenti opzioni:

- "Comment": fornisce un feedback generale senza approvare esplicitamente la pull request;
- "Approve": valida la richiesta. Ovviamente gli autori delle pull request non possono approvare le proprie modifiche;
- "Request changes": richiede modifiche e azioni correttive.

2.2.2.5 Project board Per avere una visione d'insieme delle attività, il gruppo ha deciso di creare una vista Project da associare a ciascun Repository. Visualizzando così in un unico Backlog gli issue e le pull request di tutti e tre i Repo, suddivisi in "Todo", "In Progress" e "Done". Inoltre, GitHub mette a disposizione layout che garantiscono una visualizzazione di alto livello del progetto, come ad esempio le roadmap. Nelle roadmap, le attività sono posizionate su una sequenza temporale, allo scopo di monitorarne lo stato e l'avanzamento nel tempo.

2.2.3 LaTeX e Docker

Mattia Zecchinato, che ha già esperienza con Docker, ha introdotto il resto del team alle funzionalità essenziali e ai vantaggi offerti da questo software, come la gestione degli applicativi in container isolati e distribuibili. In aggiunta, il team ha esaminato attentamente il template LaTeX, apportando leggere modifiche per personalizzarlo e renderlo più ordinato. Successivamente, il gruppo ha definito la struttura dei file LaTeX, con particolare attenzione a elementi come la copertina, il logo, l'above the fold e il registro delle modifiche. Si è deciso inoltre di organizzare le directory in tools, assets, docs e modelli, e di mantenere le firme in una cartella separata, cercando di adottare sin da subito un approccio quanto più disciplinato e mantenibile. Infine, Mattia Zecchinato ha mostrato un esempio di verbale generato tramite LaTeX, spiegando al gruppo i comandi per avviare il container Docker e generare il pdf dal codice LaTeX.

3 Todo / In Progress

Incarico	Incaricato/a
Creazione sondaggio per la scelta finale del logo	Riccardo Cavalli
Migrazione dei documenti su LaTeX	Tommaso Stocco
Setting dell'ambiente GitHub (apertura di issue, milestone, pull request e feature di sviluppo)	Riccardo Cavalli
Caricamento della vetrina di GitHub accessibile via web	Riccardo Cavalli
Revisione e riordinamento delle domande da porre alle aziende	Tutti i membri del gruppo

Luogo e Data:
Padova (PD) 15/03/2024

Firma: _____


Responsabile: Sebastiano Lewental