



Cyber 13

STUDIO DI FATTIBILITÀ

Gruppo Cyber13 - Progetto P2PCS

Informazioni sul documento

Versione	1.0.0
Data Redazione	00/00/2000
Responsabile	RESPONSABILE
Redazione	REDATTORI Bonolo Marco Pace Giulio Pezzuto Francesco
Verifica	VERIFICATORI Bonolo Marco Pace Giulio Pezzuto
Approvazione	Sanna Giovanni
Uso	Interno
Destinatari	Cyber 13 Prof. Tullio Vardanega Prof. Riccardo Cardin
Mail di contatto	<i>cyber13@gmail.com</i>

Diario delle modifiche

Versione	Data	Descrizione	Autore	Ruolo
0.02	07/03/2019	Stesura documento	Elena Pontecchiani	Da definire
0.01	07/03/2019	Creazione scheletro documento	Elena Pontecchiani	Da definire

Indice

1	Introduzione	3
1.1	Scopo del documento	3
1.2	Glossario	3
1.3	Riferimenti	3
1.3.1	Riferimenti normativi	3
1.3.2	Riferimenti informativi	3
2	Capitolato C1	5
2.1	Informazioni sul capitolato	5
2.2	Descrizione	5
2.3	Studio del dominio	5
2.4	Esito finale	6
3	Capitolato C2	7
3.1	Informazioni sul capitolato	7
3.2	Descrizione	7
3.3	Studio del dominio	7
3.4	Esito finale	7
4	Capitolato C3	9
4.1	Informazioni sul capitolato	9
4.2	Descrizione	9
4.3	Studio del dominio	9
4.4	Esito finale	9
5	Capitolato C4	11
5.1	Informazioni sul capitolato	11
5.2	Descrizione	11
5.3	Studio del dominio	11
5.4	Esito finale	12
6	Capitolato C5	13
6.1	Informazioni sul capitolato	13
6.2	Descrizione	13
6.3	Studio del dominio	13
6.4	Esito finale	13
7	Capitolato C6	15
7.1	Informazioni sul capitolato	15
7.2	Descrizione	15
7.3	Studio del dominio	16
7.4	Esito finale	16

1 Introduzione

1.1 Scopo del documento

Il presente documento ha il compito di descrivere le motivazioni che hanno portato i componenti del team Cyber13 alla scelta dello svolgimento del capitolato C5. La scelta è stata effettuata tenendo conto dei seguenti fattori:

- Le proposte dei vari progetti e quello che richiedono di implementare;
- Le tecnologie che il progetto richiede per il suo sviluppo. Considerando che il gruppo si colloca nel secondo lotto e che la maggioranza degli studenti ha esami arretrati, si è preferito avvantaggiare progetti che proponessero tecnologie almeno in parte conosciute dai componenti;
- I progetti ancora disponibili. I progetti non più a disposizione sono stati scartati, sebbene fossero stati reputati interessanti dai componenti del team.

1.2 Glossario

Onde evitare ambiguità o incomprensioni di natura lessicale, si allega il glossario v 1.0.0. All'interno del documento saranno presenti parole di ambito specifico, uso raro che potrebbero creare incomprensioni. Per una maggiore leggibilità tali parole sono riconoscibili all'interno dei vari documenti in quanto scritte in corsivo e con un 'g' corsiva a pedice tra barre orizzontali (per esempio *Glossario_g*)

1.3 Riferimenti

1.3.1 Riferimenti normativi

- Norme di progetto v 1.0.0;

1.3.2 Riferimenti informativi

- Capitolato C1:
<https://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C1.pdf>
- Capitolato C2:
<https://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C2.pdf>
- Capitolato C3:
<https://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C3.pdf>
- Capitolato C4:
<https://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C4.pdf>

- Capitolo C5:
<https://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C5.pdf>
- Capitolo C6:
<https://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C6.pdf>
- AMPLIARE CON RIFERIMENTI ALLE TECNOLOGIE USATE

2 Capitolato C1

2.1 Informazioni sul capitolato

- Nome del progetto: Butterfly
- Proponente: Imola Informatica
- Committente: Prof. Tullio Vardanega, Prof. Riccardo Cardin.

2.2 Descrizione

L'obiettivo del capitolato è lo sviluppo di una piattaforma software che permetta l'accentramento e la standardizzazione delle segnalazioni generate da software utilizzati nell'ambito dello sviluppo, più precisamente in quello delle pratiche di Continuous Integration e Continuous Delivery. Il problema sollevato dal proponente risiede nella necessità di accedere a molteplici piattaforme diverse per visionare tali segnalazioni, ciascuna con una propria struttura specifica e in alcuni casi con limitate capacità di configurazione. Al fine di facilitare la gestione e l'accesso alle informazioni contenute nelle segnalazioni è quindi stato proposto il progetto oggetto del capitolato. Viene proposto di realizzare, attraverso un design-pattern di tipo publisher-subscriber, una serie di componenti che si interfaccino direttamente con i diversi strumenti software da cui vengono generate le segnalazioni per poi procedere a redistribuirle agli utenti destinatari nella forma da loro desiderata, come ad esempio una e-mail.

2.3 Studio del dominio

- Dominio applicativo: Il capitolato fa riferimento all'ambito dello sviluppo software tramite l'utilizzo di strumenti per l'implementazione delle pratiche di Continuous Integration e Continuous Delivery. Tali strumenti producono statistiche e segnalazioni automatiche contenenti informazioni riguardanti problemi riscontrati durante lo sviluppo. Queste sono solitamente di interesse per specifici utenti, i quali hanno la necessità di consultarle in modo rapido.
- Dominio tecnologico:
 - RedMine: Software per la pianificazione di progetti e gestione delle segnalazioni tramite interfaccia web.
 - GitLab: Piattaforma per la gestione di repository GIT.
 - SonarQube: Strumento per l'analisi statica del codice.
 - Telegram e Slack: Applicazioni per invio e ricezione di messaggi tra utenti, i messaggi generati dal software devono poter essere inoltrati su entrambi i software.

- Linguaggi Java, python o nodejs per lo sviluppo del software.
- Apache Kafka: Una piattaforma open source utilizzabile come Broker, ovvero il componente dell'applicativo che si occupa di raccogliere le segnalazioni come messaggi e inoltrarli all'utente nel software scelto.

2.4 Esito finale

- Aspetti positivi:
 - La stretta relazione del progetto con strumenti molto utilizzati nello sviluppo software come SonarQube e GitLab è stata vista come un'ottima opportunità per acquisire conoscenze che potrebbero tornare utili in futuro.
 - La possibilità di utilizzare il linguaggio Java, con cui tutti i membri del gruppo hanno già una discreta familiarità.
 - Lo studio e la messa in pratica del design pattern publisher-subscriber per i componenti del software è stato ritenuto interessante.
- Fattori di rischio:
 - La grande quantità di software esterni con cui si deve interfacciare l'applicativo potrebbe risultare in una gestione problematica dello sviluppo, non sostenibile in termini di tempo dal gruppo.
 - Alcune Tecnologie come Apache Kafka sono completamente sconosciute ai membri del gruppo. Lo studio necessario per apprendere ad utilizzarle in modo sufficientemente efficace potrebbe richiedere troppo tempo.
- Conclusioni:
 - Sebbene il capitolato sia stato giudicato positivamente dai componenti del gruppo, la quantità di software con cui il progetto deve interfacciarsi e per cui il gruppo deve adeguatamente documentarsi è stata considerata non sostenibile.
 - Scelta: rigettato.

3 Capitolato C2

3.1 Informazioni sul capitolato

- Nome del progetto: piattaforma per raccolta dati mediante esercizi grammaticali;
- Proponente: MIVAQ;
- Committente: Prof. Tullio Vardanega, Prof. Riccardo Cardin.

3.2 Descrizione

L'obiettivo del capitolato è lo sviluppo di una piattaforma collaborativa di raccolta dati in cui gli utenti possano creare e/o svolgere esercizi di grammatica (per esempio esercizi di analisi grammaticale). I dati raccolti devono essere utilizzabili da sviluppatori e ricercatori al fine di insegnare ad un elaboratore a svolgere i medesimi esercizi mediante tecniche di apprendimento automatico supervisionato.

3.3 Studio del dominio

- Dominio applicativo: Il dominio applicativo a cui Mivoq fa riferimento è quello del commercio di strumenti che si occupano dell'analisi automatica del testo per decidere come si debba leggere correttamente. Questi strumenti si applicano nell'ambito dell'istruzione per facilitare il compito degli insegnanti e per migliorare l'apprendimento degli studenti.
- Dominio tecnologico: La piattaforma può essere realizzata sia come applicazione web che come applicazione mobile. Per immagazzinare i dati si suggerisce di appoggiarsi ad un servizio esistente, come Firebase [Google(2011-2018)], in modo da semplificare al massimo le problematiche di installazione, ma possono essere utilizzati anche altri sistemi. Si suggerisce l'utilizzo di un software opensource per lo svolgimento degli esercizi, come Hunpos [Mivoq(2014-2018)] o FreeLing [TALP Research Center, UPC(2008-2018a)].

3.4 Esito finale

- Aspetti positivi: L'idea di creare un software utile per l'apprendimento automatico è molto interessante perchè quest'ultima serve a facilitare sia il compito dell'insegnante, sia lo svolgimento e l'apprendimento degli esercizi da parte degli studenti.
- Fattori di rischio:
 - Le tecnologie fornite sono sconosciute ai partecipanti del gruppo;

- Il tempo a disposizione è poco per lo studio delle tecnologie;
 - Il capitolato non è più disponibile.
- Conclusioni
 - Il gruppo non ha la possibilità di scegliere il capitolato;
 - Esito:rigettato.

4 Capitolato C3

4.1 Informazioni sul capitolato

- Nome del progetto: GB: monitoraggio intelligente di processi DevOps;
- Proponente: Zucchetti;
- Committente: Prof. Tullio Vardanega, Prof. Riccardo Cardin.

4.2 Descrizione

L'azienda che commissiona il capitolato presenta un prodotto Open Source, Grafana, con cui monitora i propri sistemi; il quale dà la possibilità di creare plug-in per personalizzarne l'uso. Il capitolato infatti richiede lo sviluppo di quest'ultimo che grazie alle reti Bayesiane ha lo scopo di prevedere quali potrebbero essere gli interventi da eseguire e le zone di intervento nella linea di produzione del software. La scelta dell'utilizzo delle reti Bayesiane deriva dal seguente motivo, la competenza degli esperti viene raccolta in un sistema probabilistico, collegata poi ai dati effettivamente raccolti sul campo per determinare infine quali eventi non ancora presentatosi saranno più probabili.

4.3 Studio del dominio

- Dominio applicativo: Nel dettaglio l'azienda ha lo scopo di utilizzare tale servizio per monitorare allarmi e segnalazioni tra gli operatori del servizio Cloud e la linea di produzione del software;
- Dominio tecnologico:
 - Grafana: conoscenza dell'ambiente di sviluppo del plug-in;
 - Javascript: linguaggio su cui si basa il plug-in;
 - Reti Bayesiane.

4.4 Esito finale

- Aspetti positivi: Il progetto si presenta molto interessante dal punto di vista degli ambiti d'uso e dell'ambiente di sviluppo, la presentazione risulta molto dettagliata quindi si ha la chiara visione del lavoro da svolgere;
- Fattori di rischio: Progetto non disponibile;
- Conclusioni:

- Sebbene il capitolato sia stato giudicato complessivamente molto interessante dai componenti del gruppo, esso non è più disponibile;
- Scelta: rigettato.

5 Capitolato C4

5.1 Informazioni sul capitolato

- Nome del progetto: MegAlexa;
- Proponente: ZERO12;
- Committente: Prof Tullio Vardanega, Prof. Riccardo Cardin.

5.2 Descrizione

Il capitolato ha come scopo lo sviluppo di skill per Alexa di Amazon in grado di avviare dei workflow (routine di micro-funzioni) creati dagli utenti tramite interfaccia web o mobile app per iOS e Android. I workflow sono personalizzabili completamente dagli utenti, in modo tale che una volta creati possono soddisfare a pieno tutte le varie esigenze dell'utente finale senza che ci sia la necessità di dover fare manualmente la routine tutte le volte. Per poter avviare i workflow creati (da web o mobile) basta pronunciare "Alexa, esegui la routine nome workflow" dove il nome del workflow viene assegnato nel momento della sua creazione. Il workflow può eseguire le micro-funzioni anche in modo combinato, per esempio si può utilizzare una micro-funzione che legge un feed rss combinata con un'altra micro-funzione che filtra le notizie del feed rss interessate. Per fare ciò è necessario creare una piattaforma (web e mobile) che offre delle micro-funzioni che possono essere collegate tra di loro creando il workflow voluto dall'utente.

5.3 Studio del dominio

- Dominio applicativo: il capitolato fa riferimento all'interazione con i servizi di Amazon;
- Dominio tecnologico:
 - Amazon Web Services: è un insieme di servizi di cloud computing che compongono la piattaforma on demand offerta dall'azienda Amazon;
 - API Gateway: è un servizio completamente gestito che semplifica agli sviluppatori la creazione, la pubblicazione, la manutenzione, il monitoraggio e la protezione delle API su qualsiasi scala;
 - Lambda: consente di eseguire codice senza dover effettuare il provisioning né gestire server;
 - DynamoDB: è un database che supporta i modelli di dati di tipo documento e di tipo chiave-valore che offre prestazioni di pochi millisecondi a qualsiasi livello;

- Node.js: è una piattaforma Open source event-driven per l'esecuzione di codice JavaScript Server-side;
- Framework Node.js Express: web framework per Node.js;
- Twitter bootstrap: è una raccolta di strumenti liberi per la creazione di siti e applicazioni per il Web. Essa contiene modelli di progettazione basati su HTML, CSS e JavaScript;
- Alexa developer: il capitolato si basa sullo sviluppo di una skill per Alexa;

5.4 Esito finale

- Aspetti positivi:
 - Apprezzata l'idea di creare una skill per Amazon Alexa;
 - Lo studio e l'utilizzo dei servizi Amazon è reputato molto interessante dai componenti del gruppo;
 - L'utilizzo dei framework Node.js Express e bootstrap facilitano lo sviluppo del codice;
 - Interessante l'utilizzo del codice lato server.
- Fattori di rischio:
 - La maggior parte delle tecnologie risulta sconosciuta ai componenti del gruppo, inoltre si presentano innumerevoli casi d'uso da gestire per poter offrire all'utente un'ampia gamma di micro-funzioni.
- Conclusioni
 - Sebbene il capitolato sia stato giudicato complessivamente molto interessante dai componenti del gruppo, il tempo necessario per la gestione di tutti i casi d'uso per gestire le micro-funzioni è stato considerato non sostenibile;
 - Scelta: rigettato.

6 Capitolato C5

6.1 Informazioni sul capitolato

- Nome del progetto: P2PCS, "Car sharing Peer to Peer";
- Proponente: GaiaGo;
- Committente: Prof Tullio Vardanega, Prof. Riccardo Cardin.

6.2 Descrizione

L'azienda richiede di arricchire l'applicazione già esistente che opera nell'ambito di car sharing condominiale. Nello specifico si desidera implementare un servizio di condivisione peer to peer che integri dinamiche di gaming per incentivare l'utente all'uso dell'applicazione.

6.3 Studio del dominio

- Dominio applicativo: il capitolato fa riferimento a un'applicazione Android rivolta ad utenti interessati alla condivisione della propria auto o a richiederne una in noleggio.
- Dominio tecnologico:
 - Kotlin: come linguaggio di sviluppo;
 - AWS: come piattaforma cloud;
 - Bitrise: come ambiente di test;
 - Espresso: framework per l'ambiente di test;
 - Android studio (???????);
 - API Google (?????????).

6.4 Esito finale

- Aspetti positivi:
 - Le tecnologie scelte sono ampiamente utilizzate e quindi reputate didatticamente interessanti dai componenti de gruppo;
 - I componenti del gruppo sono interessati ad appr9ofondire l'aspetto della gamification integrata all'interno di un'applicazione.
- Fattori di rischio:

- I componenti del gruppo non possiedono conoscenze specifiche in merito alle piattaforme utilizzate dall'azienda, pertanto sarà necessario uno studio preparatorio.
- Conclusioni
 - Alla luce dei capitolati disponibili e a quanto essi propongono il capitolato in questione è stato reputato il più interessante da parte del gruppo;
 - Scelta: accettato.

7 Capitolato C6

7.1 Informazioni sul capitolato

- Nome del progetto: Soldino;
- Proponente: Red Babel;
- Committente: Prof Tullio Vardanega, Prof. Riccardo Cardin.

7.2 Descrizione

Il capitolato ha come scopo lo sviluppo di diverse D-apps (ovvero applicazioni che usano contratti intelligenti per il loro trattamento) in esecuzione sulle EVM (Ethereum Virtual Machine). La piattaforma desiderata avrà tre attori principali, ognuno dei quali con funzionalità ben definite:

- Il governo: responsabile della coniazione e distribuzione di denaro ai cittadini e alle imprese. All'interno del sistema la valuta usata è un token compatibile con ECR20, detta Cubit. Inoltre contiene un registro nelle quali sono presenti le aziende che sono autorizzate ad eseguire transazioni monetarie all'interno del sistema;
- Gli imprenditori: se intendono eseguire transazioni all'interno del sistema devono registrare la propria attività all'interno del registro detenuto dal governo. Una volta che l'attività viene registrata l'impresa può comprare e/o offrire beni. Inoltre l'azienda su base trimestrale è obbligata a pagare una tassa, IVA, al governo;
- I cittadini: acquistano beni e servizi dalle aziende attraverso i Cubit. Inoltre i cittadini interessati a aprire la propria attività possono iscriversi alla lista detenuta dal governo e diventare imprenditori.

L'interazione tra le varie componenti avverrà tramite un insieme di contratti intelligenti. Pertanto il progetto sarà composto da due macro moduli: -WEB/UI: deve contenere un insieme di pagine Web che consentano all'utente di interfacciarsi con la EVM, e quindi di eseguire le operazioni che gli sono consentite (a seconda della tipologia di utente si avranno diverse operazioni consentite, come descritto sopra); -Smart contract: contengono le transazioni che vengono sviluppate con un meccanismo di deposito in garanzia dalla rete Ethereum. Le transazioni vengono memorizzate in una struttura dati detta Blockchain, che sostanzialmente è una catena di blocchi contenenti appunto le transazioni. La validazione delle singole transazioni è affidata a un meccanismo di consenso distribuito su tutti i nodi della rete Ethereum

7.3 Studio del dominio

- Dominio applicativo: il capitolato fa riferimento al contesto del e-commerce e pagamento di tasse tramite criptovaluta;
- Dominio tecnologico:
 - Github: come piattaforma di versionamento;
 - Ethereum: piattaforma per consentire agli utenti di scrivere D-apps che utilizzano tecnologia blockchain;
 - EVM (Ethereum Virtual Machine): piattaforma che permette la gestione della rete Ethereum, lo stato interno e il calcolo. Consente quindi verificare e eseguire codice sulla blockchain. Il codice in esecuzione sulla EVM è contenuto nei cosiddetti "contratti intelligenti", scritti in linguaggio Solidity7;
 - Framework Truffle: framework di sviluppo per Ethereum che ne facilita lo sviluppo. Ad esso sono delegati la maggior parte dei compiti di routine.
 - Meta Mask: plugin browser che consente di eseguire le dApp di Ethereum usando un nodo pubblico invece che eseguire un nodo Ethereum completo. Include inoltre un vault di identità protetto che fornisce all'utente l'interfaccia per gestire la sua identità su diversi siti e firmare transazioni Ethereum in blockchain;
 - Ropsten: la rete di test ufficiale, creata da The Ethereum Foundation;
 - Raiden Network: strato infrastrutturale sopra la blockchain di Ethereum. Ha lo scopo di consentire pagamenti quasi istantanei, a basso costo e scalabili sulla rete Ethereum. Funziona con qualsiasi token compatibile a ERC20.

7.4 Esito finale

- Aspetti positivi:
 - Studio della piattaforma Ethereum, reputata molto interessante dai componenti del gruppo;
 - Studio del linguaggio orientato agli oggetti Solidity per la creazione di Smart Contracts;
 - Il framework truffle facilita lo sviluppo, in quanto facilita e automatizza tutte le operazioni di routine;
 - Divisione dello sviluppo software in diversi ambienti (local, test, staging, production), ognuno dei quali ha fornisce un insieme di risorse.
- Fattori di rischio:

- La maggior parte delle tecnologie è completamente sconosciuta ai componenti del gruppo, e lo studio di un numero di tecnologie così alto potrebbe non essere sostenibile dal gruppo.
- Conclusioni
 - Sebbene il capitolato sia stato giudicato complessivamente molto interessante dai componenti del gruppo, il tempo necessario per lo studio dei framework utilizzati da Soldino è stato considerato non sostenibile;
 - Scelta: rigettato.