



ISTITUTO DI ISTRUZIONE SUPERIORE  
“A. BADONI”

TESI DI MATURITÀ

**Alternanza scuola-lavoro estesa  
presso Agomir S.p.A.**

**Sviluppo di applicazioni multiplatforma per  
dispositivi mobili**

*Claudio Maggioni*

18 aprile 2018

TRADUZIONE IN ITALIANO

**REVISIONE 5**

## Indice

<b>1</b>	<b>Informazioni generali sul lavoro svolto</b>	<b>1</b>
<b>2</b>	<b>Gestione dei progetti</b>	<b>1</b>
<b>3</b>	<b>Il progetto principale: InteGRa Mobile</b>	<b>1</b>
3.1	Architettura software . . . . .	2
3.2	Stato del progetto . . . . .	3
3.3	Fasi del progetto . . . . .	3
<b>4</b>	<b>Il progetto secondario: Guac Remote</b>	<b>4</b>
4.1	Architettura software . . . . .	5
4.2	Stato del progetto . . . . .	6
4.3	Fasi del progetto . . . . .	6

## 1 Informazioni generali sul lavoro svolto

A partire dal giorno 19/10/2017, collaboro con l'azienda Agomir S.p.A<sup>1</sup>, che produce software, sistemi e servizi per piccole e medie imprese, svolgendo un'attività di alternanza scuola-lavoro estesa con cadenza settimanale (nello specifico giovedì e venerdì pomeriggio). Mi occupo di sviluppo software di tipo gestionale, nello specifico di applicazioni per smartphone multiplatforma (cioè compatibili sia con Android che con iOS) utilizzando strumenti come *Ionic Framework*<sup>2</sup> e *Apache Cordova*<sup>3</sup>.

Tale approccio al mondo *mobile* facilita lo sviluppo, perché al posto di usare API e meccanismi legati alla piattaforma è possibile utilizzare tecnologie note e standard come HTML e Javascript. Naturalmente tali applicazioni richiedono più risorse e tendono ad essere meno fluide, ma questo aspetto è meno rilevante in contesti gestionali come quelli affrontati da Agomir, non legati ad esempio al mondo dei videogiochi o all'elaborazione real-time.

## 2 Gestione dei progetti

Non ho diretto controllo manageriale su ciò che sviluppo in azienda, in quanto ho un ruolo simile a quello di un dipendente. L'incarico di gestire l'andamento dei progetti spetta a Mario Goretti, A.D. dell'azienda e capo del settore di sviluppo software gestionale (SWG), e ai suoi collaboratori.

In generale sviluppo i progetti da solo. Collaboro con il collega Daniele Crippa per l'interfacciamento con i software aziendali esistenti e per l'organizzazione di nuovi progetti, nonché consigli e dritte varie.

Nonostante non abbia controllo totale è comunque mia responsabilità fare stime orarie sul lavoro da svolgere nonché definire passi e *milestone* per i vari progetti.

Per aumentare la forza lavoro per lo sviluppo di applicazioni *mobile* ho gestito momenti di formazione ad alcuni dipendenti nei quali ho mostrato il principale funzionamento delle tecnologie che uso.

## 3 Il progetto principale: InteGRa Mobile

Questa applicazione, una volta completata, dovrebbe permettere ad utenti in mobilità di interfacciarsi con alcune funzioni del prodotto ERP di punta di Agomir: il gestionale *InteGRa*<sup>4</sup>. Nel dettaglio, sarà possibile accedere alle seguenti sezioni:

**Ordini cliente** per registrare ordini di prodotti a clienti;

**Soggetti** per consultare informazioni anagrafiche di clienti e fornitori;

**Agenda** per consultare e aggiungere eventi nel calendario presente nell'ERP, il quale si può integrare con *Outlook*<sup>5</sup>;

---

<sup>1</sup>Agomir S.p.A.: <https://www.agomir.com/>

<sup>2</sup>Ionic Framework: <https://ionicframework.com/>

<sup>3</sup>Apache Cordova: <https://cordova.apache.org/>

<sup>4</sup>InteGRa ERP: <https://integra.agomir.com/>

<sup>5</sup>Microsoft Outlook: <https://products.office.com/it-it/outlook>

**Magazzini** gestire e inventariare scorte in magazzino.

In aggiunta, sarà possibile anche registrare le ore per interventi in trasferta, funzione già implementata nell'applicazione *InteGRa.Service*, sviluppata nei periodi di alternanza precedenti. Tale lavoro non è direttamente implementabile in *InteGRa Mobile* a causa di differenze consistenti nelle architetture dei due prodotti.

Inoltre, l'applicazione sarà in grado di funzionare in modo limitato anche senza connessione diretta ad *InteGRa*, permettendo la sincronizzazione dei dati modificati con il gestionale in un momento futuro.

### 3.1 Architettura software

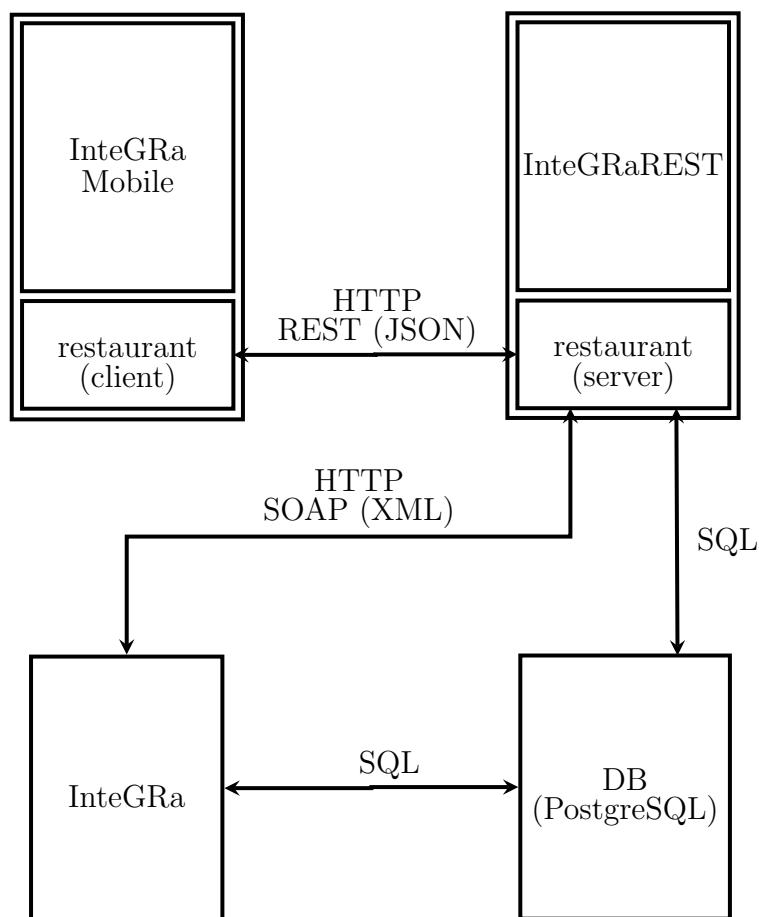


Figura 1: L'architettura di *InteGRa Mobile* raffigurata con un diagramma

Nella figura 1 si può notare come l'insieme dei componenti software all'interno del progetto siano organizzati e comunichino tra loro. Nel dettaglio:

- L'applicazione non comunica direttamente con il gestionale, ma tramite una serie di chiamate REST[4] ad un endpoint lato server (chiamato *InteGRaREST*) che svolge il ruolo di intermediario. Nello specifico, i dati in input vengono trasmessi come `application/x-www-form-urlencoded` mentre i dati in output vengono trasmessi come JSON[3], e le sessioni vengono identificate con un token presente nella query string di ciascuna richiesta;

- Tale componente server è in grado di effettuare autonomamente query SQL al database del gestionale per recuperare velocemente informazioni aventi una struttura dati semplice;
- Nel caso sia necessario interagire con record complessi, *InteGRaREST* comunica direttamente con *InteGRa* attraverso chiamate SOAP[5];
- La comunicazione tra *InteGRaREST* e *InteGRa Mobile* avviene tramite *restaurant*, libreria che implementa e rende standard il protocollo di comunicazione e sincronizzazione dati.

### 3.2 Stato del progetto

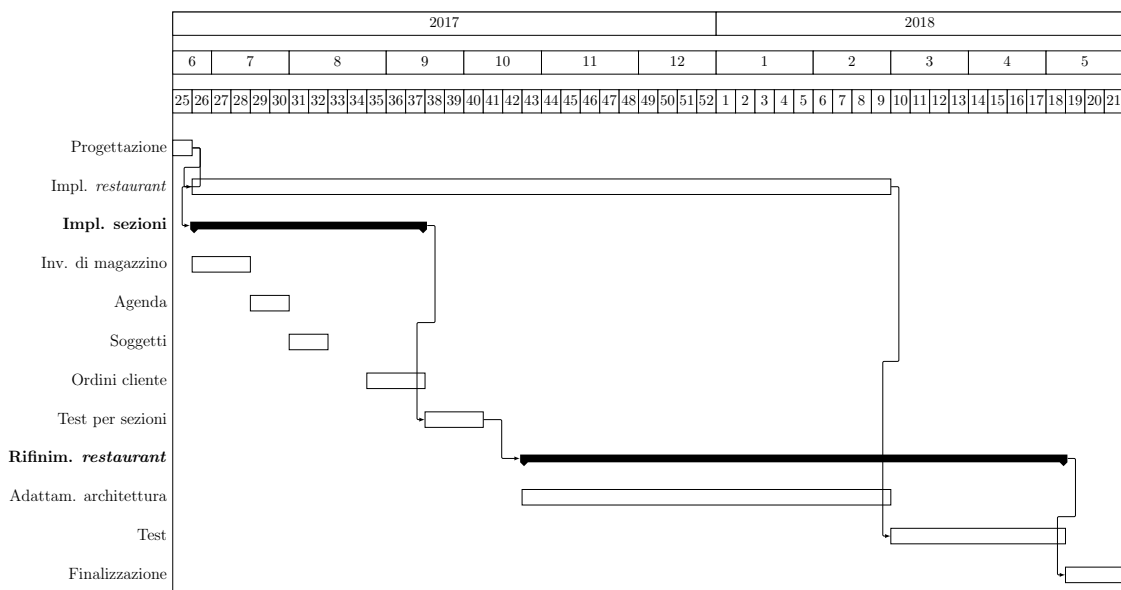


Figura 2: Il diagramma di Gantt del progetto *InteGRa Mobile*

Nella lettura del diagramma di Gantt in figura 2 è necessario tenere a mente che il completamento dell'intera applicazione non è certo per la fine di maggio. Ciò che dovrà essere necessariamente completato è la parte più importante del progetto: *restaurant*, il motore di sincronizzazione online/offline e libreria generica che permette di implementare velocemente nuove sezioni dell'applicazione.

### 3.3 Fasi del progetto

Le fasi del progetto, di cui date di inizio e fine sono state specificate nel diagramma in figura 2, sono:

**Progettazione** delineazione delle funzionalità da implementare nell'applicazione e prime bozze dell'interfaccia utente; //

**Implementazione restaurant** implementazione della libreria, su cui si basano tutte le sezioni del programma. Vista l'importanza di questo componente, gli sviluppi vengono continuati in parallelo con l'implementazione delle sezioni;

**Implementazione sezioni** creazione di interfaccia e logica per le funzionalità previste, cioè:

- Magazzini;
- Agenda;
- Soggetti;
- Ordini cliente;

**Test per le sezioni** messa alla prova degli algoritmi e della UI per efficacia ed efficienza, nonché eventuali correzioni;

**Rifinimento restaurant** messa alla prova delle scelte architetturali e algoritmi usati nella libreria e rifinimento di essi; Fase divisa in:

**Miglioramento architettura** analisi del protocollo per individuare punti di debolezza e possibili ottimizzazioni;

**Test per restaurant** messa alla prova dei cambiamenti fatti;

**Finalizzazione** task finali del progetto, tra cui branding, operazioni pre-rilascio, e gestione di eventuali personalizzazioni richieste dai clienti.

## 4 Il progetto secondario: Guac Remote

Questo progetto è destinato ad un'importante azienda del territorio, produttrice di macchine equilibratrici. Tale applicazione dovrebbe fungere da client di desktop remoto (come *TeamViewer*<sup>6</sup>) per il PC presente nel loro prodotto di punta, un sistema di calibrazione delle ruote di autoveicoli funzionante mediante telecamere. Il programma dovrebbe permettere all'operatore di tale prodotto di interagire con il software presente nel macchinario, senza scendere dal veicolo.

A causa di urgenza del committente, lo sviluppo di questo progetto ha interrotto e attualmente interrompe gli sviluppi per *InteGRa Mobile*.

---

<sup>6</sup>TeamViewer: <https://www.teamviewer.com/it/>

## 4.1 Architettura software

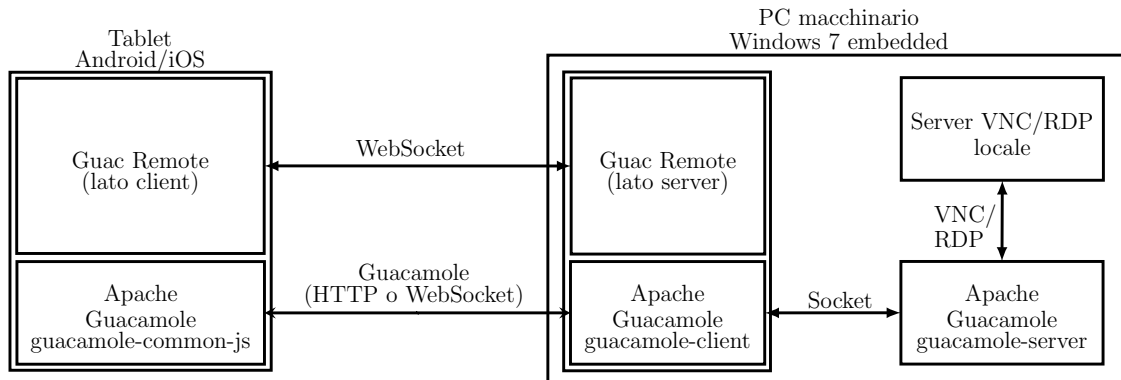


Figura 3: L'architettura di *Guac Remote*

*Guac Remote* è basata su un'applicativo e libreria per la connessione a computer remoto chiamato *Apache Guacamole*<sup>7</sup>. Tale software è costituito da due parti: [1]

**guacamole-server (o guacd)** un servizio scritto in C che funge da adattatore tra il protocollo *guacamole* e i protocolli VNC, RDP o SSH, utilizzando questi ultimi per stabilire connessioni con gli host remoti;

**guacamole-client** una WebApplication scritta tramite *Java servlet* che fornisce un'interfaccia web per interagire con *guacamole-server* e connettersi ai PC remoti.

Data la natura open-source del progetto, sia *guacamole-server* che *guacamole-client* possono essere usati come libreria per la realizzazione di software derivati. In particolare, *guacamole-client* può essere scomposto nelle librerie *guacamole-common-js*, che contiene il codice Javascript per il client, e *guacamole-common*, che fornisce classi Java per la connessione con *guacamole-server*. [2]

In *Guac Remote*, *Apache Guacamole* è usato per fornire accesso remoto al PC presente nel macchinario, che contiene l'applicativo per la calibrazione, all'applicazione installata sul tablet. Nel dettaglio, *guacamole-common-js* è usato nell'applicazione *mobile* fornire un'interfaccia touchscreen per interagire con l'host remoto, mentre *guacamole-common*, tramite una piccola WebApplication, assieme a *guacamole-server* sono installati sul PC.

In aggiunta, tablet e PC possono comunicare informazioni aggiuntive tramite una *WebSocket* creata al momento della connessione, necessaria per alcune estensioni al protocollo richieste dal cliente.

<sup>7</sup>*Apache Guacamole*: <https://guacamole.apache.org>

## 4.2 Stato del progetto

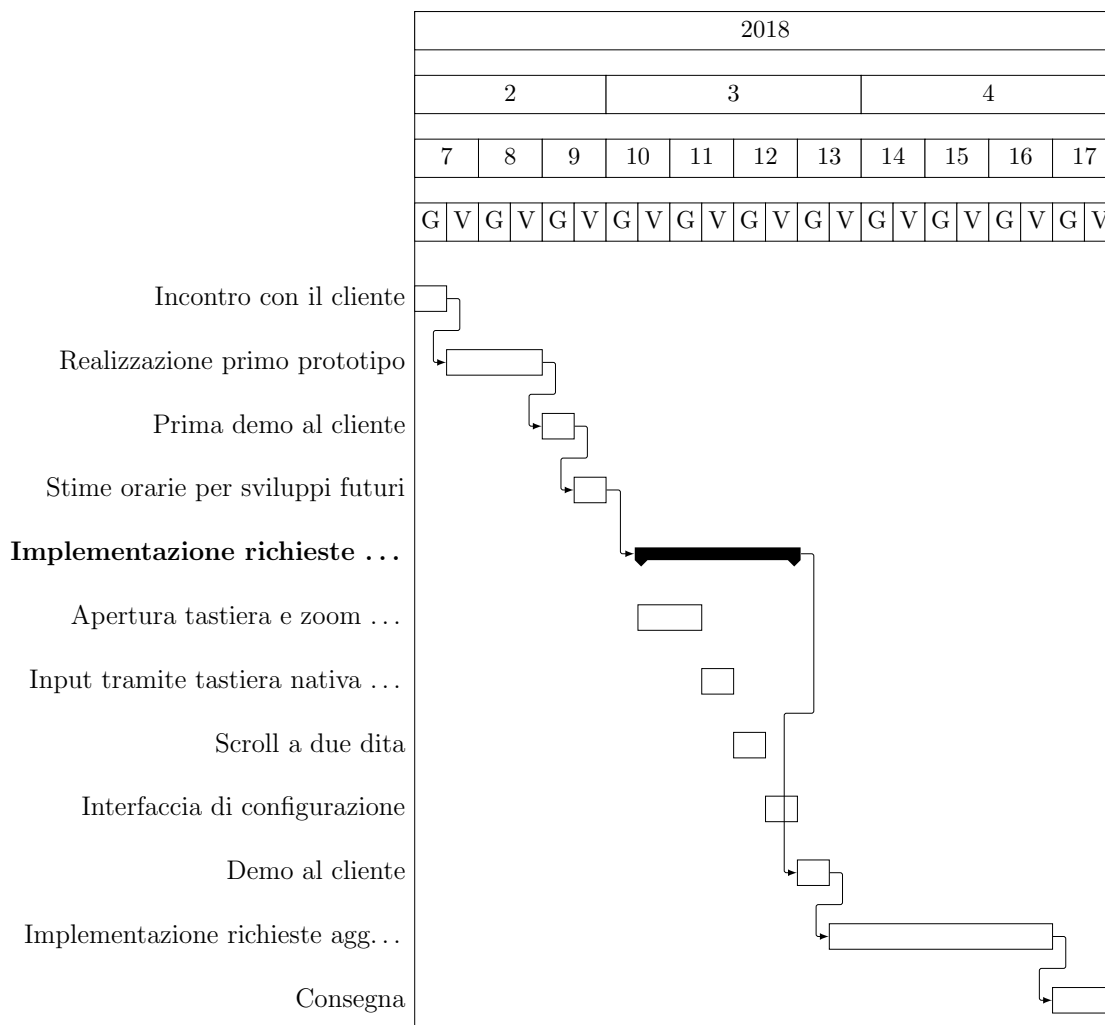


Figura 4: Il diagramma di Gantt del progetto *Guac Remote*

Questo progetto, al giorno 06/04, non ha ritardi. L'applicazione é stata mostrata in demo il giorno 29/03, ed il cliente si ritiene soddisfatto del lavoro fatto fino ad ora.

## 4.3 Fasi del progetto

Le fasi del progetto, di cui le date di inizio e di fine sono indicate nel diagramma della figura 4, sono:

**Incontro con il cliente** primo scambio di informazioni per capire gli obiettivi del progetto;

**Realizzazione primo prototipo** dimostrazione dell'efficacia del protocollo *guacamole* tramite un prototipo del prodotto;

**Prima demo al cliente** demo del prototipo al cliente;



**Stime orarie per sviluppi futuri** delineazione della tabella di marcia per gli sviluppi futuri;

**Implementazione richieste del cliente** sviluppo delle estensioni al protocollo richieste. Nel dettaglio, esse sono:

**Apertura tastiera e zoom al “focus” di un input** alla pressione di un campo di testo, l'applicazione *mobile* deve automaticamente ingrandire l'area selezionata e mostrare una tastiera;

**Input tramite tastiera nativa Android o iOS** sostituzione della tastiera su schermo di *guacamole-common-js* con quella nativa;

**Scroll a due dita** supporto della gesture di scroll verticale a due dita;

**Interfaccia di configurazione** creazione di una piccola finestra di configurazione, in cui inserire IP e porta del PC;

**Demo al cliente** dimostrazione degli sviluppi fatti al cliente e eventuale definizione di richieste aggiuntive;

**Implementazione delle richieste aggiuntive del cliente**

**Consegna** operazioni finali del progetto, tra cui branding e compilazione per rilascio.

## Elenco delle figure

1	L'architettura di <i>InteGRa Mobile</i> raffigurata con un diagramma . . .	2
2	Il diagramma di Gantt del progetto <i>InteGRa Mobile</i> . . . . .	3
3	L'architettura di <i>Guac Remote</i> . . . . .	5
4	Il diagramma di Gantt del progetto <i>Guac Remote</i> . . . . .	6

## Riferimenti bibliografici

[1] The Apache Software Foundation. *Guacamole Manual, Chapter 1. Implementation and architecture*. [Online; in data 9-aprile-2018, traduzione propria]. URL: <https://guacamole.apache.org/doc/gug/guacamole-architecture.html>.

[2] Apache Guacamole. *API Documentation*. [Online; in data 9-aprile-2018, traduzione propria]. URL: <https://guacamole.apache.org/api-documentation/>.

[3] Wikipedia. *JavaScript Object Notation — Wikipedia, L'enciclopedia libera*. [Online; in data 9-aprile-2018]. 2018. URL: [https://it.wikipedia.org/w/index.php?title=JavaScript\\_Object\\_Notation&oldid=94928405](https://it.wikipedia.org/w/index.php?title=JavaScript_Object_Notation&oldid=94928405).

[4] Wikipedia. *Representational State Transfer — Wikipedia, L'enciclopedia libera*. [Online; in data 9-aprile-2018]. 2018. URL: [https://it.wikipedia.org/w/index.php?title=Representational\\_State\\_Transfer&oldid=95970771](https://it.wikipedia.org/w/index.php?title=Representational_State_Transfer&oldid=95970771).

[5] Wikipedia. *SOAP — Wikipedia, L'enciclopedia libera*. [Online; in data 9-aprile-2018]. 2017. URL: <https://it.wikipedia.org/w/index.php?title=SOAP&oldid=85936136>.