



UNIVERSITA' DI PISA
FACOLTA' DI INGEGNERIA

**TESI DI LAUREA MAGISTRALE IN INGEGNERIA
INFORMATICA PER LA GESTIONE D'AZIENDA**

**Progettazione e sviluppo di
un'applicazione mobile
multiplatforma per l'ottimizzazione
dei processi di interrompibilità e
certificazione in Terna Rete Italia**

RELATORI

Prof.ssa Gigliola Vaglini

Ing. Mario G.C.A. Cimino

LAUREANDA

Domenica Boccuzzi

INDICE

INTRODUZIONE	5
1. L'AZIENDA	7
1.1 IL GRUPPO SOCIETARIO	7
1.2 IL SISTEMA ELETTRICO NAZIONALE	8
2. I PROCESSI DI STUDIO	11
2.1 L'INTERROMPIBILITA'	11
2.1.1 BMI	12
2.1.2 UPDC	12
2.2 LA CERTIFICAZIONE	15
2.2.1 INTERVENTI PROGRAMMATI	16
2.2.2 MALFUNZIONAMENTI	17
2.3 LA CERTIFICAZIONE SEMPLIFICATA	18
3. AMBITO DEL PROGETTO	19
3.1 SETTING DEL PROBLEMA	19
4. ANALISI DEI PROCESSI DI BUSINESS	21
4.1 BUSINESS PROCESS MANAGEMENT	21
4.2 I VANTAGGI DEL BPM	23
4.3 MODELLIZZAZIONE E DOCUMENTAZIONE DEI PROCESSI DI BUSINESS	24
4.4 ANALISI DEL PROCESSO DI CERTIFICAZIONE AS-IS	26
4.5 CRITICITA' E PUNTI DI MIGLIORAMENTO DEL PROCESSO DI CERTIFICAZIONE	31
4.6 VINCOLI ESTERNI	34
5. SOLUZIONE: OBIETTIVO DEL PROGETTO	35
5.1 I VANTAGGI DI UNA SOLUZIONE SOFTWARE	36
5.2 HTML5 VS CODICE NATIVO	38
5.3 PHONEGAP	42
5.4 SQLITE	45
5.5 SOAP VS REST	48
5.6 IL PROTOCOLLO SOAP	51
5.4.1 STRUTTURA DI UN MESSAGGIO SOAP	52
5.4.2 HEADER SOAP	53
5.4.3 BODY SOAP	54
5.4.4 SOAP ENCODING	54

5.4.5	<i>ESECUZIONE DI RPC CON SOAP</i>	55
5.4.6	<i>IL FILE WSDL E UDDI</i>	58
5.7	IL PROCESSO DI CERTIFICAZIONE TO-BE	58
6.	PROGETTAZIONE DELLA SOLUZIONE	62
6.1	REQUISITI FUNZIONALI LATO CLIENT	62
6.1.1	<i>AUTENTICAZIONE CERTIFICATORE</i>	62
6.1.2	<i>GESTIONE CALENDARIO</i>	63
6.1.3	<i>GESTIONE INTERVENTI</i>	64
6.1.4	<i>SCRITTURA ESITI</i>	65
6.2	REQUISITI FUNZIONALI LATO SERVER	66
6.2.1	<i>RICEZIONE ESITI</i>	67
6.3	REQUISITI NON FUNZIONALI	67
6.4	DIAGRAMMA DEI CASI D'USO	68
6.5	CASI D'USO LATO CLIENT	69
6.6	CASI D'USO LATO SERVER	75
6.7	DIAGRAMMI DI SEQUENZA	76
6.8	PROGETTAZIONE DEL DATABASE	84
6.8.1	<i>GLOSSARIO DEL DATABASE</i>	84
6.8.2	<i>ATTRIBUTI, VERIFICHE E PROPRIETA'</i>	91
6.8.3	<i>PROGETTAZIONE CONCETTUALE</i>	91
6.8.4	<i>PROGETTAZIONE FISICA</i>	94
6.9	PROGETTAZIONE INTERFACCE GRAFICHE	96
7.	SVILUPPO DELLA SOLUZIONE	98
7.1	SVILUPPO APPLICAZIONE CLIENT	98
7.1.1	<i>VISUALIZZA CALENDARIO E APPUNTAMENTI</i>	98
7.1.2	<i>VISUALIZZA ELENCO ATTIVITA'</i>	99
7.1.3	<i>VISUALIZZA ELENCO TASK</i>	100
7.1.4	<i>VISUALIZZA RIEPILOGO CHECKLIST E INVIO</i>	101
7.2	CREAZIONE FILE XML	102
7.3	SVILUPPO WEB SERVICES	106
7.4	CREAZIONE FILE PDF	112
7.5	CREAZIONE FILE EXCEL	114
7.6	OPERAZIONI SU DATABASE	115
8.	CONCLUSIONI	117
	BIBLIOGRAFIA	118

INDICE DELLE FIGURE

Figura 1.1 - Operatori su un traliccio gestito da Terna.....	7
Figura 1.2 - Rete Elettrica Nazionale.....	8
Figura 2.1 - Architettura del sistema BMI.....	12
Figura 2.2 - Architettura del Sistema UPDC.....	14
Figura 2.3 - Procedura BMI Terna.....	16
Figura 4.1 - Ciclo vita di un Processo	23
Figura 4.2 - Macro-attività del processo certificazione	26
Figura 4.3 - Processo certificazione AS-IS - Diagramma di flusso a swimlane.....	29
Figura 4.4 - Processo Certificazione BPMN AS-IS part1.....	30
Figura 4.5 - Processo Certificazione BPMN AS-IS part2.....	31
Figura 5.1 - Soluzione proposta	36
Figura 5.2 - Market Share Q2 2013.....	38
Figura 5.3 - App. Native vs App. Ibride	40
Figura 5.4 - Applicazioni Mobili in HTML5 e PhoneGap	44
Figura 5.5 - PhoneGap Packaging Process	44
Figura 5.6 - Struttura Applicazione con PhoneGap	45
Figura 5.7 - Architettura SQLite	47
Figura 5.8 - Funzionamento protocollo SOAP	51
Figura 5.9 - Struttura messaggio SOAP.....	53
Figura 5.10 - Header SOAP.....	54
Figura 5.11 - Body SOAP	54
Figura 5.12 - Messaggio SOAP di richiesta ad un web service di prova	57
Figura 5.13 - Schematizzazione funzionamento SOAP	58
Figura 5.14 - Processo certificazione TO-BE - Diagramma di flusso a swimlane.....	59
Figura 5.15 - Processo Certificazione BPMN TO-BE part1	60
Figura 5.16 - Processo Certificazione BPMN TO-BE part2	61
Figura 6.1 - Diagramma dei casi d'uso	69
Figura 6.2 - Diagramma di Sequenza: Login	77
Figura 6.3 - Diagramma di Sequenza VisualizzaCalendario	78

Figura 6.4 - Diagramma di Sequenza VisualizzaElencoAttività.....	78
Figura 6.5 - Diagramma di sequenza VisualizzaElencoTask.....	79
Figura 6.6 - Diagramma di Sequenza InviaChecklist.....	80
Figura 6.7 - Diagramma di sequenza InserisciNota.....	81
Figura 6.8 - Diagramma Classi lato Client.....	82
Figura 6.9 - Diagramma Classi lato Server.....	83
Figura 6.10 - Struttura fisica del database lato client.....	90
Figura 6.11 - Schema concettuale E-R.....	92
Figura 6.12 - Attributo composto "Indirizzo".....	93
Figura 6.13 - Attributi associati all' Entità "Impianto".....	93
Figura 6.14 - Struttura fisica del database lato server.....	95
Figura 6.15 - Mockup visualizzazione agenda.....	
Figura 6.16 - Mockup visualizzazione checklist.....	96
Figura 6.17 - Mockup visualizzazione riepilogo checklist.....	
Figura 6.18 - Mockup visualizzazione attività di verifica.....	97
Figura 7.1 - Vista Calendario e Appuntamenti.....	98
Figura 7.2 - Vista Elenco Attività.....	99
Figura 7.3 - Vista Elenco Task per Attività.....	100
Figura 7.4 - Vista Visualizza Riepilogo Checklist.....	101
Figura 7.5 - Web Service: GestoreDati.....	110
Figura 7.6 - WSDL associato al Web Service GestoreDati.....	111
Figura 7.7 - Struttura file excel.....	115

INTRODUZIONE

Uno degli obiettivi dell'azienda Terna, gestore e proprietario principale della Rete di Trasmissione Nazionale di energia elettrica ad Alta Tensione, è quello di ridurre, in tempo reale, gli assorbimenti dei clienti "Interrompibili" allacciati in Alta Tensione al fine di mantenere la rete elettrica nazionale costantemente in equilibrio fra domanda e offerta di energia. Quando la domanda supera l'offerta, una delle modalità per riequilibrare il flusso di energia è far ricorso al servizio di interrompibilità elettrica. L'effettiva interruzione avviene a fronte della ricezione di un comando di distacco inviato da Terna, tramite linee telefoniche dedicate e ridondanti, ai clienti interrompibili. Per la ricezione dei segnali e, più generalmente, per la raccolta delle informazioni tecniche, lato interrompibile sono predisposti ed installati apparati opportunamente certificati.

Lo scopo del lavoro di tesi è mettere in luce le attività critiche non ottimizzate e quelle non a valore aggiunto e, a valle di ciò, individuare una soluzione per il miglioramento dei principali processi di business coinvolti nell'ambito dell'interrompibilità.

La soluzione individuata a seguito di un'attenta attività di analisi, è l'introduzione di un'applicazione mobile a supporto del lavoro dei certificatori che si occupano quotidianamente delle certificazioni degli apparati tecnici.

L'applicazione mobile è stata progettata e, quindi, sviluppata in HTML5 + CSS3 + Javascript, con il supporto del framework PhoneGap per orientamento al multiplatforma. Lato back-end, mediante il linguaggio Java, è stato sviluppato un web service per recepire, tramite utilizzo di protocollo SOAP, ed elaborare le richieste del client, tipicamente in formato XML.

Infine, mediante le librerie Apache FOP e Apache POI, è stato possibile realizzare dei report contenenti le informazioni utili sulla certificazione, inviati dal certificatore, per mezzo dell'applicazione mobile.

Il lavoro di tesi è frutto di un tirocinio di 6 mesi, svolto nell'ambito del programma formativo **Junior Consulting** presso il **CONSEL - Consorzio ELIS** tra novembre 2012 – maggio 2013. Questo programma si propone di accrescere le conoscenze acquisite durante il periodo universitario, nonché trasferire competenze specifiche necessarie ad operare nel mercato del lavoro. La principale attività svolta durante i mesi del tirocinio consiste in un'esperienza di consulenza aziendale che vede come cliente una delle varie aziende partner del CONSEL: si lavora su un progetto reale e per un periodo limitato di tempo, allo scopo di rispondere alle esigenze dell'azienda committente, rispettandone i requisiti e le tempistiche. Ogni progetto ha uno o più referenti aziendali che seguono gli sviluppi del lavoro e valutano i risultati intermedi prodotti dal team e il rilascio finale. Gli studenti partecipanti hanno, in questo modo, la possibilità di affacciarsi nel mondo del lavoro, conoscerne le dinamiche e capire ciò che dovranno affrontare (si spera) una volta raggiunto il tanto atteso traguardo della laurea.

1. L'AZIENDA

1.1 IL GRUPPO SOCIETARIO

Terna è un grande operatore di reti per la trasmissione dell'energia elettrica guidato da Flavio Cattaneo, Amministratore Delegato, e presieduto da Luigi Roth.

Con oltre 63.500 km di linee, Terna è il gestore e principale proprietario della Rete di Trasmissione Nazionale di energia elettrica ad Alta e ad Altissima Tensione, il primo operatore indipendente in Europa e sesto al mondo per chilometri di linee elettriche gestiti. L'assetto societario di Gruppo è costituito dalla holding Terna Spa, da cui dipendono due società operative interamente controllate: Terna Rete Italia e Terna Plus.



Figura 1.1 - Operatori su un traliccio gestito da Terna

Attraverso la società controllata, Terna Plus, è responsabile dello sviluppo dei nuovi business, sia in Italia che all'estero, e punta a cogliere le opportunità di business derivanti dalle discontinuità tecnologiche e dai trend caratterizzanti il settore dell'energia, quali: crescita delle fonti rinnovabili, evoluzione verso le Smart Grid, efficienza energetica.

Attraverso la società controllata **Terna Rete Italia** gestisce in sicurezza la Rete di Trasmissione Nazionale in Alta tensione e ha la responsabilità, 365 giorni l'anno, 24 ore su 24, della trasmissione e del dispacciamento dell'energia e quindi della gestione in sicurezza dell'equilibrio tra la domanda e l'offerta di energia elettrica in Italia. Fornisce, inoltre, servizi in regime di concessione e ne garantisce la sicurezza, la qualità e l'economicità nel tempo. Assicura parità di condizioni di accesso a tutti gli utenti delle reti.

Il ruolo da protagonista svolto rende Terna responsabile del servizio verso l'intera collettività nazionale anche se gli utenti finali del servizio elettrico non sono clienti diretti dell'Azienda, ma delle società di distribuzione e vendita dell'energia elettrica. Il senso di responsabilità per un servizio di interesse generale è parte fondamentale della cultura lavorativa dell'azienda stessa.

La Missione di Terna, oltre a sviluppare attività di mercato, nuove opportunità di business e creare valore per gli azionisti, è garantire la sicurezza, la qualità e l'economicità nel tempo della trasmissione di energia e mantenere, dunque, la rete elettrica nazionale costantemente in equilibrio. [1]

1.2 IL SISTEMA ELETTRICO NAZIONALE

Il sistema elettrico nazionale è articolato in tre fasi: **produzione**, **trasmissione** e **distribuzione** di energia elettrica.



Figura 1.2 - Rete Elettrica Nazionale

L'energia elettrica come la conosciamo non esiste in natura e bisogna, quindi, produrla. Produrre energia vuol dire trasformare in "elettricità" l'energia ricavata da fonti primarie. Questa trasformazione avviene nelle centrali elettriche.

La *trasmissione* di energia elettrica è il passaggio intermedio tra la *produzione* e la *distribuzione* agli utilizzatori dell'energia elettrica. Essa viene attuata con l'appoggio ad una infrastruttura di rete qual è la rete di trasmissione elettrica a grande distanza ed alta tensione (AT: 220 – 380 KV).

La *distribuzione* elettrica è l'ultima fase nel processo di consegna dell'elettricità all'utente finale dopo produzione e trasmissione e si realizza attraverso un'infrastruttura di rete tipica qual è la rete di distribuzione elettrica capillare fino agli utenti o utilizzatori finali. Generalmente tale rete comprende linee elettriche a MT (tra i 10 e i 20 KV) e linee a BT (inferiore a 1000 V, normalmente 230/400 V).

La rete elettrica di trasmissione si interfaccia con quella di distribuzione tramite cabine elettriche primarie di trasformazione da alta o altissima tensione (AAT) a media tensione. Lo scopo è quello di portare l'energia dalle centrali elettriche di produzione, dove l'energia prodotta è in media tensione, fino ai luoghi di utilizzo, città e zone industriali, che possono essere distanti decine o centinaia di chilometri.

La rete di trasmissione ha anche la funzione di interconnettere i centri di produzione non solo nazionali, ma anche transnazionali al fine di ottimizzare la produzione secondo le necessità di utilizzazione.

Un esempio per semplificare: le linee di trasmissione di Terna sono come le autostrade, che collegano una città all'altra in un Paese, come da "casello a casello". Le linee di distribuzione, invece, sono le strade che collegano un punto all'altro all'interno delle città. Per questa attività è necessario un sistema di trasmissione, molto complesso e articolato, composto da numerosi elementi, i più importanti sono: le linee elettriche, i tralicci, le stazioni elettriche e di trasformazione che costituiscono la Rete di trasmissione Nazionale: la spina dorsale del sistema elettrico italiano. Questo sistema va coordinato e gestito da specialisti attraverso un'attività fondamentale per l'intero Paese: il **dispacciamento**.

Come ben sappiamo, l'energia elettrica non si può immagazzinare. E' quindi necessario produrre, istante per istante, la quantità di energia richiesta dall'insieme dei consumatori (famiglie e aziende) e gestirne la trasmissione in modo che l'offerta e la domanda siano

sempre in equilibrio, garantendo così la continuità della fornitura del servizio e sicurezza per il territorio. La gestione di questi flussi di energia sulla rete si chiama, appunto, dispacciamento. Tale attività, svolta da Terna, richiede il monitoraggio dei flussi elettrici e l'applicazione delle disposizioni necessarie per l'esercizio coordinato degli elementi del sistema, cioè gli impianti di produzione, la rete di trasmissione e i servizi ausiliari. La gestione in tempo reale del nostro sistema, interconnesso con quello europeo, viene svolta attraverso un sistema di controllo altamente tecnologico, che fa capo al Centro Nazionale di Controllo (CNC), il cuore del sistema elettrico italiano dove, in un "bunker" dell'energia con oltre 100 schermi di controllo e un wallscreen di 40 metri quadrati, Terna monitora, per 24 al giorno e 365 giorni l'anno, 293 linee, tra cui 9 interconnessioni con l'estero, 3 cavi sottomarini (tra i quali è doveroso menzionare il SA.PE.I.¹) e 281 linee nazionali a 380 kV.

Il Centro nazionale di controllo ha il compito di assicurare il funzionamento del sistema elettrico nelle condizioni di massima sicurezza, per garantire la continuità e la qualità del servizio.

¹ Sa.Pe.I. è il cavo sottomarino che connette la Sardegna con la Penisola Italiana. E' stato ribattezzato "cavo dei record" essendo: il più lungo del Mediterraneo, con i suoi 435 Km, il più lungo al mondo a 1000 MW, il più profondo al mondo a 1640 m sotto il livello del mare, il più sostenibile, in quanto rispetta il territorio sottomarino sul quale si snoda.

2. I PROCESSI DI STUDIO

2.1 L'INTERROMPIBILITA'

Quando la domanda di energia supera l'offerta, una delle modalità per riequilibrare il flusso di energia è far ricorso al servizio di interrompibilità elettrica. L'erogazione e l'assegnazione del servizio di interrompibilità avviene mediante la stipulazione di un contratto con i clienti industriali che si rendono disponibili ad essere distaccati senza preavviso.

L'assegnatario del servizio, il cliente interrompibile, rende disponibile per l'interruzione uno o più carichi, all'interno del proprio o dei propri stabilimenti. Per ogni MW di potenza interrompibile messa a disposizione, Terna remunera l'assegnatario sulla base dei consumi mensili dei carichi interrompibili, corrispondendo un importo composto da una quota fissa e da una quota variabile, la quale è funzione del numero di distacchi subiti.

Questo servizio è di fondamentale importanza per Terna, dato che le consente di disporre di una quantità di potenza interrompibile che può essere utilizzata:

- nel caso di *interrompibilità istantanea* prevalentemente per ricostituire, con rapidità, riserva e bilanciamento dell'energia in tempo reale;
- nel caso di *interrompibilità in emergenza*, per delimitare alle situazioni di effettivo rischio per il sistema elettrico nazionale con il ricorso all'attivazione di procedure di alleggerimento del carico.

L'effettiva interruzione avviene a fronte della ricezione di un comando di distacco inviato da Terna tramite linee telefoniche dedicate e ridondanti. Per la comunicazione dei segnali, e più generalmente delle informazioni tecniche, lato Terna e lato interrompibile sono predisposti ed installati due apparati, rispettivamente: il Banco Manovra Interrompibili (BMI) e l'Unità Periferica Distacco Carichi (UPDC).

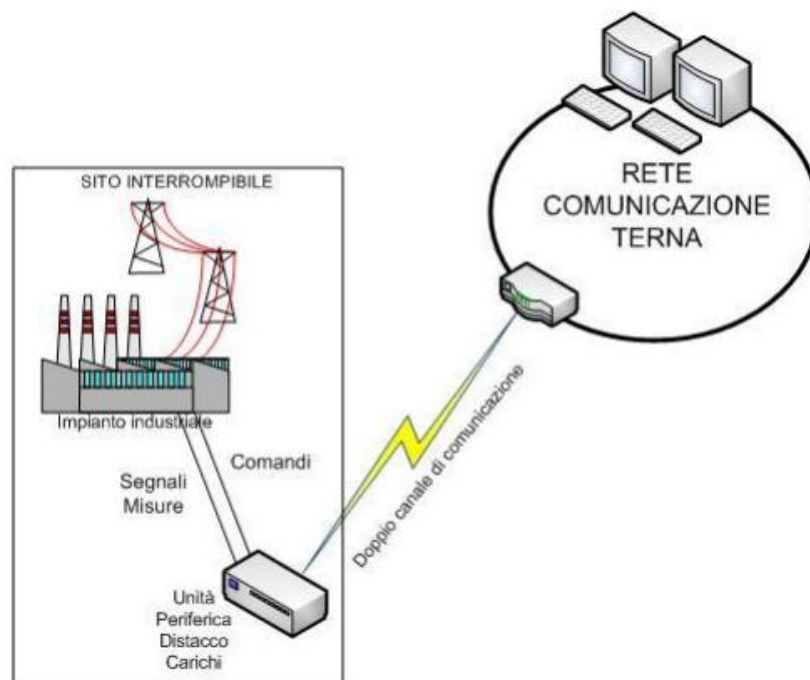


Figura 2.1 - Architettura del sistema BMI

2.1.1 BMI

Il Banco Manovra Interrompibili consente a Terna di distaccare il carico contrattualizzato con l'interrompibile attraverso una serie di comandi di apertura simultanei verso gli organi di manovra.

2.1.2 UPDC

L'Unità periferica Distacco Carichi (UPDC) è una macchina di tele-operazioni che permette agli utenti industriali in Media/Alta Tensione di essere "interrompibili", e deve essere realizzata, installata e connessa al BMI secondo le prescrizioni tecniche definite e documentate nel dettaglio da Terna². E' il BMI, infatti, che ne controlla a distanza le funzionalità.

² TERNA DRRPX02038 – "Unità Periferica Distacco Carichi Guida alla realizzazione"; TERNA DRRPX02039 – "Unità Periferica Distacco Carichi Profilo del protocollo IEC 870-5-104; TERNA DREPX02005 - "Prescrizioni Tecniche – Procedura di verifica delle installazioni Unità Periferica di Distacco Carichi "; TERNA DRRPX03050 - "Prescrizione tecniche integrative per la connessione al Banco Manovra Interrompibili"

L'UPDC ha la funzione di acquisire la misura del carico, nonché altre informazioni ausiliarie, e di attuare il comando di distacco del carico contrattualizzato, a seguito dell'ordine proveniente dal BMI di Terna.

L'UPDC è caratterizzato da quattro parti principali:

- **RTU:** ha il compito di gestire la comunicazione su protocollo IEC 60870-5-104 da/verso il BMI di Terna. Implementa un'interfaccia **Telnet** per la configurazione in locale dei suoi parametri e un **server FTP** per il download dei files di configurazione. Gestisce sia le **trame TCP** sia **UDP multicast** verso il router e il protocollo IGMP. Mantiene il firmware su due memorie flash separate, consentendone l'aggiornamento senza interrompere la normale funzionalità dell'apparato. L'unità dispone di 256KB di memoria non volatile per la memorizzazione dei parametri, delle liste dinamiche e dei log degli eventi. Attraverso un bus di comunicazione RS485, l'RTU è in grado di monitorare l'attività delle ICC e di segnalare localmente i malfunzionamenti mediante segnalazione visiva e relè di allarme. Ogni RTU è in grado di gestire più schede di I/O per un massimo di quattro carichi interrompibili. Non contiene elementi tampone che necessitano di manutenzione o di sostituzioni periodiche.
- **FC DI/DO/AI:** Questa parte comprende tutto l'I/O digitale ed analogico necessario alla gestione completa del carico interrompibile. L'unità è controllata da microprocessore ed è dotata di propria diagnostica con uscita a relè. La scheda dispone di segnalazioni visive a otto led in grado di emettere otto colori ciascuno e grazie ai quali è possibile avere un controllo immediato sullo stato della scheda senza la necessità di consultare un display alfanumerico. Ciascuna scheda dispone di un I/O di test in modo che, ad ogni comando di "*prova UPDC*" inviato da Terna, sia verificata la funzionalità di ogni singola scheda.

- **GPS:** A ogni RTU è associato un ricevitore GPS che garantisce operatività continuativa in tutte le condizioni atmosferiche, di facile configurazione e di semplice installazione. Inoltre fornisce un accurato segnale impulsivo “one pulse-per-second” (PPS) necessario alla sincronizzazione della UPDC con i server di Terna.
- **ROUTER:** L’UPDC e i suoi componenti sono connessi ad un **router** che si interfaccia direttamente alla rete di comunicazione per lo scambio di informazioni con il BMI: riceve i comandi ed inoltra i dati raccolti e i segnali che riportano lo stato delle operazioni e delle misure provenienti dal campo. Il router svolge, quindi, la funzione di “switch ethernet” e deve necessariamente essere gestito da Terna (il modello di router/switch deve essere, quindi, compatibile con l’architettura della rete BMI): non è possibile installare uno switch esterno al router. Gli apparati RTU devono essere connessi al router/switch direttamente, uno per ogni porta ethernet, ed identificati tramite MAC address. Tutti i dispositivi devono essere alloggiati in appositi quadri predisposti per la sigillatura. Il sistema di monitoraggio va connesso ad una particolare porta ethernet dello switch integrato nel router, configurata in “mirroring”; da questa porta non è consentito interagire con il router/switch.

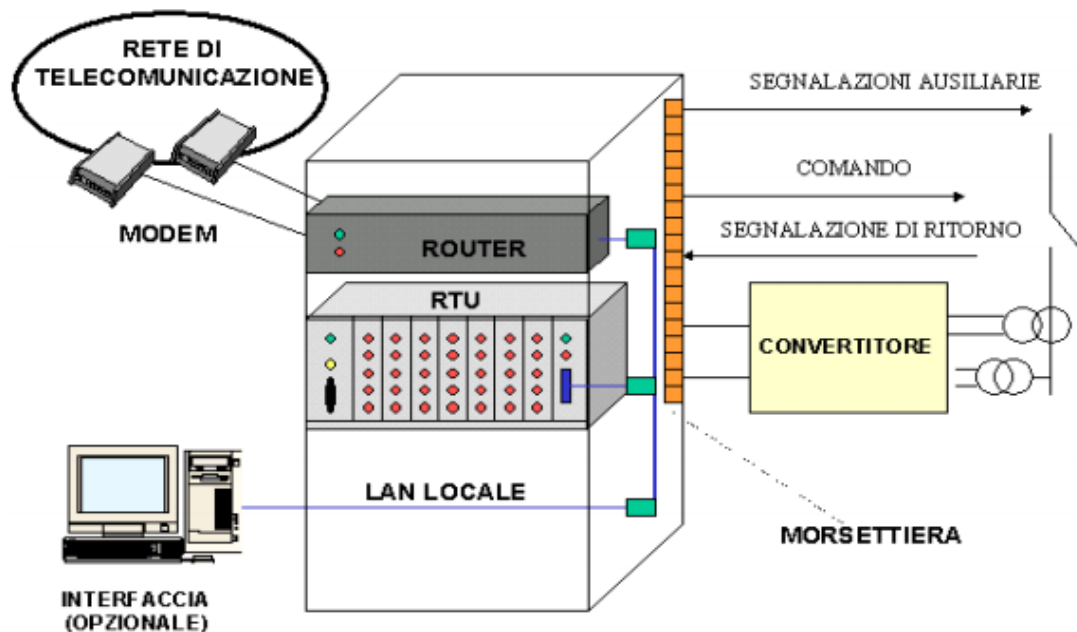


Figura 2.2 - Architettura del Sistema UPDC

La comunicazione con il BMI avviene attraverso un **canale di comunicazione** principale di tipo **PVC (Permanent Virtual Circuit)** su Frame Relay, supportato da un ulteriore canale di comunicazione backup ISDN, il cui utilizzo si rende necessario in caso di malfunzionamenti o avaria del canale principale.

2.2 LA CERTIFICAZIONE

Per assicurare il corretto funzionamento del servizio di interrompibilità, l'UPDC e i dispositivi correlati devono essere sottoposti ad attività di verifica da parte di un istituto accreditato³ (IA), o da un istituto equivalente a livello europeo, per le prove su componenti elettromeccanici in alta tensione, che provvederà anche alla loro sigillatura. Oltre ad un primo intervento di certificazione, che avviene a seguito della prima installazione degli apparati presso la sede del cliente Interrompibile, possono verificarsi eventi che richiedono ricertificazione obbligatoria:

- interventi sugli apparati predisposti alla sigillatura o già sottoposti a sigillatura (apparati RTU, morsettiere di confine, relè di comando e di sommatoria degli stati interruttori, convertitori di misura e di sommatoria delle stesse, apparati di comunicazione quali router, switch, convertitori fibra ottica);
- installazione di una nuova RTU o modifica delle configurazioni delle RTU già presenti;
- interventi a seguito di un distacco fallito.

Due sono gli eventi che potrebbero rendere necessaria la ricertificazione e, quindi, l'intervento di un certificatore:

- 1. Interventi Programmati**
- 2. Malfunzionamenti**

³ ai sensi della Norma UNI CEI EN ISO/IEC 17020:2005

2.2.1 INTERVENTI PROGRAMMATI

In caso di interventi programmati sugli apparati predisposti all'interrompibilità, il titolare di risorse interrompibili deve necessariamente segnalare a Terna tutte le informazioni relative all'intervento, almeno 10 giorni prima dell'intervento stesso, utilizzando una funzione apposita sulla Procedura BMI, messa a disposizione da Terna⁴: "Gestione indisponibilità programmata".

La procedura BMI, contestualmente all'inserimento delle informazioni sulla manutenzione, verifica che i dati inseriti siano corretti, altrimenti ne segnala eventuali incongruenze, associa automaticamente un codice all'intervento programmato, un Referente Terna (al quale il cliente interrompibile potrà rivolgersi in caso di modifiche relative all'intervento o per eventuali necessità) ed, in base alla causale specificata, segnala al cliente se l'intervento programmato necessita di certificazione.

The screenshot shows the 'Indisponibilità programmata' (Planned Unavailability) form in the BMI Terna system. The form is titled 'Dettaglio indisponibilità programmata : Inserimento'. It includes the following fields and options:

- Impianto:** ragione sociale dell'impianto
- Codice UPDC:** codice dell'UPDC
- * Data inizio Manutenzione:** 01 / 07 / 2012
- * Data fine Manutenzione:** 01 / 07 / 2012
- Ora:** 09 / 00
- Ora:** 16 / 30
- Durata:**
- * Causale:** Manutenzione convertitori A.O.
- Note:** (empty text area)
- Codice Intervento:**
- Stato Intervento:**
- Referente Terna:** TSP/SDP - SDP_BMI@terna.it - _SDP_BMI@terna.it

At the bottom, there is a note: 'E' richiesta documentazione di certificazione' and three buttons: 'Conferma', 'Annulla', and 'Esci'. A footer note states: 'I campi contrassegnati da * sono obbligatori'.

Figura 2.3 - Procedura BMI Terna

In quest'ultimo caso, il cliente interrompibile dovrà inviare *documentazione accessoria* a Terna, almeno 10 giorni prima dell'inizio dell'intervento, comunicare con il certificatore

⁴ Al link: <https://procedure.terna.it>, selezionando la voce "Banco Manovra Interrompibili".

per stabilire la data d'intervento, ed inserire nella procedura BMI un ulteriore **intervento programmato**: *“Esecuzione prove per certificazione UPDC da parte di ente autorizzato”*, contestualmente all'inserimento dell'intervento programmato tramite la funzione *“Gestione indisponibilità programmata”*.

La documentazione accessoria inviata dovrà contenere: descrizione variazione, schema unifilare aggiornato con indicazione dei carichi interrompibili, schemi dei circuiti di ricezione e smistamento dei comandi, segnali e misure, schemi di cablaggio dell'apparato UPDC, configurazione aggiornata UPDC.

La certificazione deve essere consecutiva all'intervento ed effettuata entro 10 giorni dalla fine dell'intervento stesso.

A seguito dell'intervento di certificazione, il certificatore compilerà un Rapporto Tecnico dettagliato delle verifiche effettuate ed un attestato di certificazione, li invierà al cliente interrompibile, il quale, a sua volta, dopo averne verificato la veridicità di quanto dichiarato dal certificatore, provvederà ad inviare tutta la documentazione ricevuta a Terna.

2.2.2 MALFUNZIONAMENTI

Nel caso si verifichi un malfunzionamento improvviso (un generico malfunzionamento relativo all'apparato UPDC, segnalazioni invalide, anomalie nel colloquio con il BMI di Terna, guasti nel canale di comunicazione) tempestivamente il cliente interrompibile deve notificarlo a Terna, tramite procedura BMI, utilizzando la funzione apposita: *“Gestione malfunzionamento UPDC”*.

Il cliente interrompibile dovrà comunicare a Terna tutte le informazioni necessarie per descrivere al meglio il malfunzionamento: impianto, data di presunta fine (se conosciuta), note esplicative sul malfunzionamento.

La procedura BMI, contestualmente all'inserimento delle informazioni sul malfunzionamento, verifica che i dati inseriti siano corretti, altrimenti ne segnala eventuali incongruenze, associa automaticamente un codice al malfunzionamento ed un referente Terna (al quale il cliente interrompibile potrà rivolgersi per eventuali necessità).

Se il tipo di malfunzionamento richiede certificazione, il cliente interrompibile dovrà necessariamente inserire nella procedura BMI l'intervento programmato: "Esecuzione prove per certificazione UPDC da parte di ente autorizzato".

Anche in questo caso, la certificazione deve essere conseguente all'intervento ed effettuata entro 10 giorni dalla fine dell'intervento stesso. E' necessario, dunque, che il cliente interrompibile concordi con l'Istituto Accreditato una data di certificazione che sia immediatamente successiva all'intervento.

Appena ripristinato il corretto funzionamento di tutti i sistemi, il cliente interrompibile dovrà comunicarlo a Terna specificando la fine del malfunzionamento, la causa e il tipo di intervento effettuato ed inviare a Terna, non appena disponibile, la documentazione di certificazione ricevuta dal certificatore.

2.3 LA CERTIFICAZIONE SEMPLIFICATA

Se l'intervento (manutenzione programmata o malfunzionamento) non comporta alcuna alterazione dei cablaggi o della configurazione dell'apparato UPDC, il cliente interrompibile può evitare di ripetere le procedure di test previste dalla certificazione e, quindi, evitare l'inserimento sulla Procedura BMI dell'intervento programmato "Esecuzione prove per certificazione UPDC da parte di ente autorizzato", previa comunicazione esplicita a Terna (via e-mail) del codice intervento programmato o malfunzionamento UPDC, identità dell'Istituto Accreditato che deve aver presenziato a tutte le attività (de-sigillatura, intervento e ri-sigillatura).

Al termine dell'intervento di certificazione semplificata, il cliente interrompibile dovrà inviare a Terna la documentazione rilasciata dall'Istituto Accreditato, in cui viene dichiarata la conformità degli apparati UPDC rispetto a quanto certificato in precedenza.

3. AMBITO DEL PROGETTO

Come è stato già anticipato, uno degli obiettivi di Terna è quello di ridurre, in tempo reale, gli assorbimenti dei clienti interrompibili allacciati in AT al fine di mantenere la rete elettrica nazionale costantemente in equilibrio fra domanda e offerta di energia; a tale scopo Terna stipula, mediante aste periodiche, dei contratti con i clienti industriali che si rendono disponibili ad essere distaccati senza preavviso.

Al fine di fornire tale servizio, ogni cliente interrompibile deve sottoporsi a un **processo di certificazione** degli apparati tecnici, installati presso i propri impianti, atti a consentire l'interrompibilità stessa; la certificazione degli apparati viene eseguita da un ente certificatore terzo.

Obiettivo del progetto è analizzare il processo di certificazione, al fine di individuare quei **flussi informativi destrutturati** che vanno a inficiare l'efficienza del processo stesso e, quindi, fornire una soluzione adeguata per migliorarlo.

3.1 SETTING DEL PROBLEMA

Come è stato già anticipato, oltre ad un primo intervento di certificazione, che avviene a seguito della prima installazione degli apparati presso la sede del cliente interrompibile, possono verificarsi eventi che richiedono la certificazione obbligatoria:

- manutenzione ordinaria degli apparati;
- installazione di una nuova RTU o modifica delle configurazioni delle RTU già presenti;
- riparazione degli apparati (a seguito ad esempio di un distacco fallito).

In caso di interventi che non alterino i cablaggi o la configurazione dell'apparato UPDC (quali ad esempio il riavvio delle apparecchiature), il titolare di risorse interrompibili può evitare la ripetizione delle procedure di test previste dalla certificazione, a condizione che il certificatore presenti tutte le attività (de-sigillatura, intervento e ri-sigillatura) e garantisca, tramite apposito attestato, la conformità dell'UPDC rispetto a quanto precedentemente certificato: in questo caso si parla di certificazione light.

Nel caso si verifichi uno degli eventi di cui sopra, gli apparati installati presso il cliente interrompibile devono essere ricertificati.

Il problema principale è legato alla struttura del processo attuale, molto complesso e caratterizzato da numerose attività non a valore aggiunto e in cui il tempo assume ruolo centrale. I clienti interrompibili, dal momento in cui segnalano un intervento che richiede ricertificazione, al momento in cui Terna riceve tutta la documentazione e può verificare l'esito delle verifiche, risultano non interrompibili. Poiché essi vengono remunerati per ogni MW messo a disposizione, più è lungo il periodo di intervento-ricertificazione, più le **perdite economiche** aumentano. D'altra parte Terna avrà a disposizione un tot di MW da interrompere notevolmente inferiore rispetto al caso in cui tutti i clienti risultino interrompibili: ciò rende necessaria una ri-pianificazione in tempo reale dei carichi da interrompere, il che potrebbe, in casi estremi, comportare una momentanea destabilizzazione del sistema (evento che non dovrebbe mai verificarsi).

Essendo, quindi, un processo di estrema importanza sia per l'azienda Terna, sia per i clienti interrompibili, la necessità di semplificarlo ed ottimizzarlo ha assunto negli ultimi mesi priorità assoluta per Terna.

4. ANALISI DEI PROCESSI DI BUSINESS

4.1 BUSINESS PROCESS MANAGEMENT

Tutte le organizzazioni svolgono le proprie attività organizzandole per processi. Secondo la definizione dalla Norma UNI EN ISO 9000:2000 si considera **Processo**: “l’insieme di attività correlate o interagenti che trasformano elementi in entrata in elementi in uscita fornendo valore aggiunto”.

Volendo partire da tale definizione per descrivere i processi gestionali (business process) attraverso i quali l’organizzazione persegue i propri obiettivi, si può definire un Processo di Business come quell’insieme di attività o fasi interrelate svolte all’interno dell’organizzazione che creano valore trasformando delle risorse (input del processo) in un prodotto (output del processo) destinato ad un soggetto interno o esterno all’azienda (cliente). Spesso si è portati ad identificare il cliente solo con chi riconosce un corrispettivo per il risultato fornito e ad individuare solo i processi di produzione oppure di vendita, tralasciando tutte le altre attività. Il cliente è invece il destinatario del risultato e in quanto tale può identificarsi con un cliente vero e proprio, un fornitore, un individuo all’interno di un’organizzazione, un reparto dell’azienda, un altro processo, e così via.

I processi costituiscono le fondamenta dell’impresa e sono il mezzo grazie al quale essa porta avanti il proprio business. In generale i processi possono essere suddivisi in:

- Processi Strategici:
 - *processi competitivi*: processi con cui l’organizzazione compete sul mercato e con i quali mira a superare la concorrenza (es. fidelizzazione del cliente, elaborazione strategie, individuazione delle esigenze del cliente..);
 - *processi di innovazione e trasformazione*: processi che forniscono le capacità richieste per il futuro (es. innovazione prodotti/servizi, investimenti strategici..);
- Processi Operativi:

- processi che realizzano i prodotti/servizi dell'organizzazione (es. gestione marketing e vendite, produzione/erogazione servizi, gestione della logistica..);
- Processi di Supporto:
 - Processi che aggiungono efficienza ed efficacia ai processi operativi (es. gestione risorse umane, gestione finanza e amministrazione, gestione servizi IT).

Nell'analisi dei processi aziendali è importante raccogliere le seguenti informazioni: definire lo scopo del processo, definire il responsabile del processo, stabilirne inizio e fine (delimitazione del processo), identificare il cliente, specificarne l'output, definirne gli input, verificare i vincoli, le leggi e i regolamenti, stabilire i controlli da effettuare, identificare le infrastrutture necessarie per produrre l'output, identificare le risorse umane necessarie e i livelli di competenza richiesti, identificare i sottoprocessi ed evidenziare se l'output del processo costituisce l'input per un altro processo.

L'obiettivo del BPM è quello di gestire il ciclo vita di un processo in tutte le sue fasi:

- I. **Modellazione:** gli obiettivi di questa fase sono quelli di documentare e disegnare il processo, selezionando eventuali modelli, best practice e standard. Inoltre devono essere chiaramente espressi gli obiettivi di business (KGI, Key Goal Indicator) e i misuratori di performance (KPI, Key Performance Indicator);
- II. **Simulazione:** consiste nella valutazione di qual è il miglior modello da implementare sulla base dei dati di monitoraggio di precedenti esecuzioni di processo e dei KPI. Vengono simulati vari processi alternativi e scelto quello che presenta le migliori performance;
- III. **Implementazione:** è la traduzione di ogni step del modello scelto in procedure operative;
- IV. **Esecuzione:** in questa fase il processo è finalmente pronto per essere messo in produzione e viene eseguito nelle sue fasi operative;
- V. **Monitoraggio:** consiste nella definizione delle grandezze relative al processo misurato, nella misurazione dei KPI e nella valutazione di cosa funziona e cosa no;

VI. **Ottimizzazione:** è questo il momento in cui occorre identificare le aree di miglioramento, quantificando i benefici ottenuti.

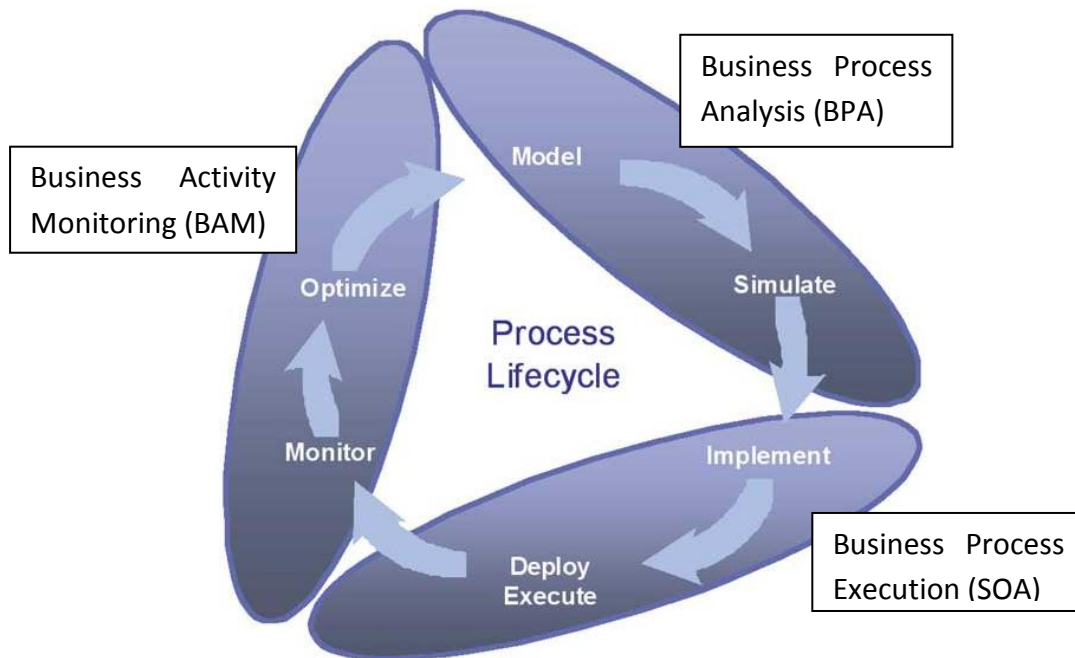


Figura 4.1 - Ciclo vita di un Processo

I processi si possono migliorare, ma salvo casi estremi non devono essere riprogettati ex novo: distruggere un processo significa perdere tutta la conoscenza insita nel processo stesso e quindi distruggere la capacità di fare business dell'impresa.

4.2 I VANTAGGI DEL BPM

Il business è costituito da diversi fattori: azioni, procedure, sistemi informativi, processi frammentati. L'azienda che non conosce i propri processi non conosce il proprio business e non può quindi valutare l'impatto di un eventuale **cambiamento**. La mancanza di conoscenza crea inevitabilmente duplicazioni di funzioni, di processi, di attività nelle diverse divisioni, con l'accollo dei relativi costi. Individuare e gestire tutti i processi dell'azienda, renderli governabili, sviluppare ed applicare nuove soluzioni a realtà

consolidate nel rispetto della storia aziendale sono gli obiettivi del Business Process Management (BPM).

Il Business Process Management (BPM) fornisce i mezzi per avere una visione globale e puntuale dell'organizzazione. Attraverso il BPM l'impresa consegue i benefici di una maggiore efficienza, tracciabilità e **miglioramento qualitativo** delle attività svolte e i processi divengono asset immateriali, che ne accrescono costantemente il **valore** e permettono di generare un reale vantaggio competitivo. Il BPM infatti è un insieme di strumenti fondamentali per la definizione, la modellazione, l'esecuzione, l'integrazione, la gestione e l'ottimizzazione dei processi gestionali, per creare un sistema orientato a rendere efficiente ed efficace il business dell'organizzazione.

Essere eccellenti in un business consiste nel raggiungere il massimo di efficienza ed efficacia nello svolgimento dei processi che portano valore all'organizzazione. Questo si traduce nella **riduzione dei costi** e del time-to-market, in quanto le risorse vengono utilizzate al meglio [2].

4.3 MODELLIZZAZIONE E DOCUMENTAZIONE DEI PROCESSI DI BUSINESS

Rappresentare i processi migliora la condivisione delle informazioni, permette di eseguire rapidamente interventi che portano ottimizzazioni evidenti ed immediate e libera quindi delle risorse. I processi rappresentano ciò che il business realizza: i **processi sono il business**. Analizzare e gestire i processi significa perciò gestire al meglio il proprio business.

Il primo passo in una corretta gestione dei processi è, senza dubbio, la loro identificazione e documentazione. Per completare questa fase esistono diverse tecniche, sia diagrammatiche che testuali; alcune di esse sono diventate ormai degli standard riconosciuti, come il BPMN e lo XPD, altre, invece, hanno una più facile immediatezza e sono più indicate, ad esempio, per relazionarsi con il management.

- I **diagrammi di flusso a swimlane**, nell'ambito del Workflow Management, sono ottimi tool grafici utilizzati per mappare il flusso di un processo ed identificare

potenziali aree problematiche che necessitano miglioramento, al fine di rendere il processo più **efficiente**. Devono necessariamente includere: *attori coinvolti* (che possono essere individui, dipartimenti, divisioni, gruppi, macchine, entità e così via) che sono i responsabili della linea di lavoro a cui sono associati (ogni linea è associata ad un attore), le *fasi* che identificano fasi successive del processo stesso, il *processo* ovvero il processo attuale e il flusso che deve essere tracciato, i *simboli* con cui si rappresenta graficamente cosa succede in ogni step del processo.

- **Business Process Modeling System (BPMS)** è uno strumento, o meglio in framework di riferimento a supporto del BPM, che consente un'efficace implementazione dei processi di business, una rappresentazione dei modelli, delle persone, dei gruppi di lavoro coinvolti, una verifica della raggiungibilità degli obiettivi prefissati e dei relativi costi. Grazie ad un BPMS non solo si possono realizzare strumenti per la rappresentazione e la documentazione dei processi, ma anche dirigerne e monitorarne l'esecuzione, i dati, le informazioni, lo stato dei processi, l'avanzamento dei lavori, le aree migliorabili, le attività assegnate agli individui o ai gruppi di lavoro, ecc. Nell'ambito della Reingegnerizzazione dei Processi di Business, consente di gestire la documentazione, adattamento, modellizzazione e ottimizzazione dei processi di business e di individuare le potenzialità di riorganizzazione.
- La **Business Process Modeling Notation (BPMN)** ha come obiettivo fornire uno **standard** di rappresentazione efficace facile da utilizzare e da comprendere da parte degli utenti business interessati al problema della modellazione, progettazione ed eventuale informatizzazione dei processi aziendali: analisti di processo che costruiscono le bozze iniziali dei processi organizzativi in esame o da progettare, programmatori e sviluppatori delle applicazioni informatiche per la gestione di tali processi, e infine manager e dirigenti responsabili della gestione e del monitoraggio dei processi stessi. La BPMN è sostanzialmente una derivazione del formalismo dei flow chart ma con alcune aggiunte e modificazioni che permettono di superarne alcuni limiti nella modellazione dei processi aziendali. Permette di costruire dei diagrammi di processo (BPD – Business Process Diagram)

che rappresentano in pratica dei grafi o reti costituiti da *oggetti* rappresentati dalle attività di processo, collegati da *flussi di controllo* che definiscono la relazione logica, le dipendenze e l'ordine di esecuzione delle attività stesse.

4.4 ANALISI DEL PROCESSO DI CERTIFICAZIONE AS-IS

Dopo una prima analisi superficiale del processo di Interrompibilità, l'attenzione è ricaduta sul processo di Certificazione, che si colloca, per l'appunto, all'interno del processo di Interrompibilità stesso. Si tratta di un processo cruciale in quanto, il mancato corretto funzionamento degli apparati predisposti all'Interrompibilità, inficerebbe non di poco, l'intero processo a monte, inducendo notevoli perdite economiche⁵ e un concreto rischio per il sistema elettrico nazionale.

A seguito dell'analisi della documentazione è stato possibile, in un primo momento, individuare le macro-attività del processo di certificazione, che hanno permesso di avere una visione complessiva e schematizzata del processo nella sua totalità.

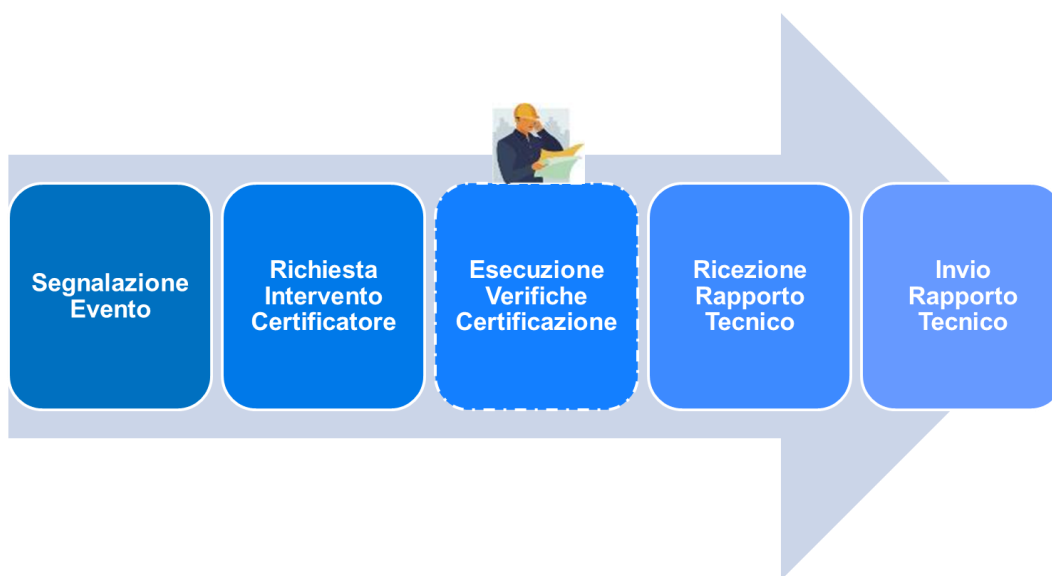


Figura 4.2 - Macro-attività del processo certificazione

⁵ Se le apparecchiature UPDC non ricevono correttamente il segnale di interruzione e, quindi, non possono attuare il distacco stesso, il contratto stipulato inizialmente tra le parti, viene meno (al terzo distacco fallito). Inoltre, dati vincoli temporali molto stringenti per l'attuazione del segnale di distacco, un distacco fallito potrebbe compromettere la corretta gestione del dispacciamento a discapito della sicurezza nazionale.

Ad eccezione della macro attività di “Esecuzione Verifiche Certificazione”, che è svolta dal certificatore, le altre macro attività sono svolte dal cliente interrompibile.

In una seconda fase è stato possibile coinvolgere direttamente gli **attori del processo** per ottenere informazioni più dettagliate in merito al processo stesso. E' stato così possibile giungere ad una prima descrizione verbale delle attività che compongono il processo.

A seguito di un evento che richiede certificazione, l'*interrompibile* deve:

1. Accedere via Web alla Procedura BMI per notificare l'evento a Terna: a partire da questo momento il cliente si sta temporaneamente dichiarando indisponibile ad essere distaccato;
2. Contattare personalmente il certificatore per concordare la data dell'intervento;
3. Accedere via Web alla Procedura BMI ed inserire un evento di certificazione per il giorno concordato;
4. Inviare al certificatore tutta la documentazione relativa all'impianto da certificare.

A seguire, il *certificatore*:

5. Effettua un'analisi dettagliata della documentazione;
6. Si reca presso il cliente ed esegue tutte le attività di verifica necessarie (il cosiddetto 'intervento');
7. Terminato l'intervento rilascia un attestato al cliente interrompibile;
8. Compila un rapporto tecnico che invia al cliente interrompibile;

L'*interrompibile*:

9. Ricevuto l'attestato, provvederà ad inviarlo a Terna tramite email.

L'*addetto Terna*, a seguito della ricezione dell'attestato e del rapporto tecnico:

10. Provvederà ad analizzare il rapporto tecnico ricevuto per individuare eventuali esiti di verifiche negativi.

In tal caso, infatti, il cliente interrompibile non potrà essere ritenuto tale. Ciò rende necessaria l'attivazione di procedure per l'intervento e la risoluzione degli eventuali problemi sorti in fase di certificazione oltre, soprattutto, ad inficiare la

possibilità per Terna di interrompere, in caso di emergenza, l'erogazione di potenza per i carichi contrattualizzati dal particolare cliente interrompibile.

11. Compila manualmente un file excel che contiene informazioni sugli esiti delle verifiche ed eventuali valori tecnici a supporto.

Dalla descrizione verbale del processo di certificazione è stato possibile realizzare un primo schema del processo utilizzando un diagramma di attività a swimlane semplificato (Figura 8).

Dopo la realizzazione del diagramma di attività a swimlane, ho realizzato uno schema del processo in notazione BPMN per avere una visione più dettagliata dell'intero processo (Figure 9-10).

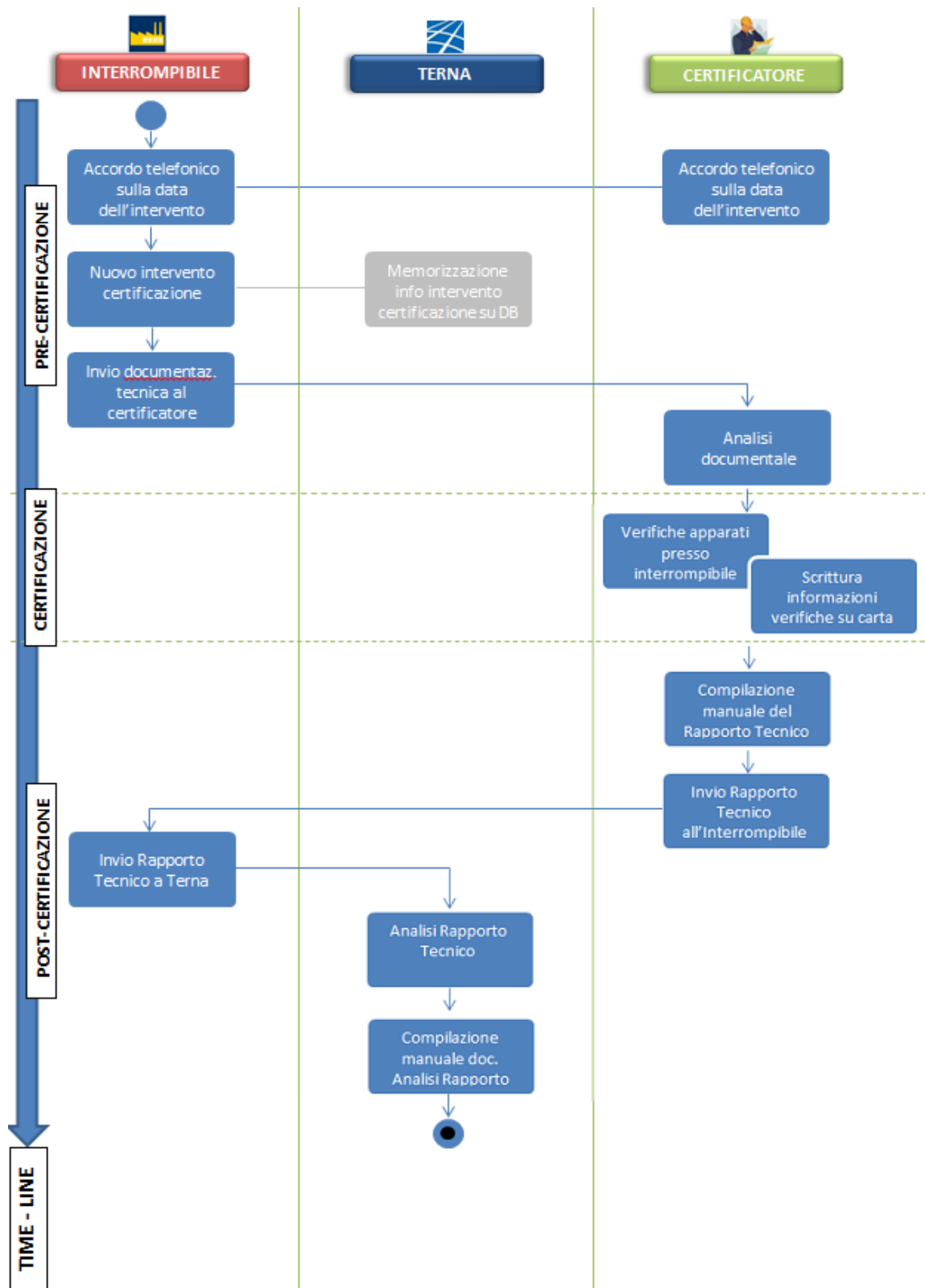


Figura 4.3 - Processo certificazione AS-IS - Diagramma di flusso a swimlane

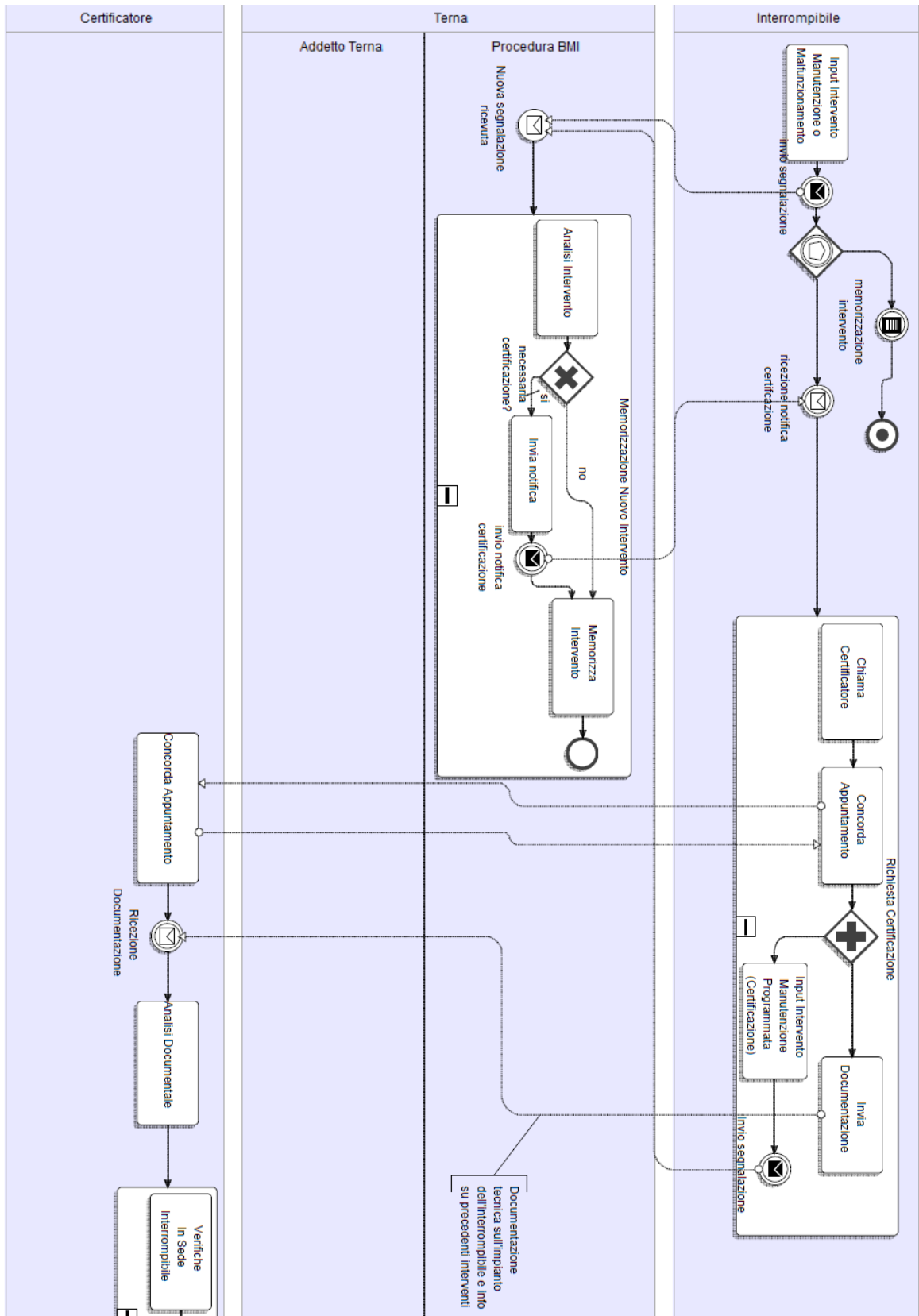


Figura 4.4 - Processo Certificazione BPMN AS-IS part1

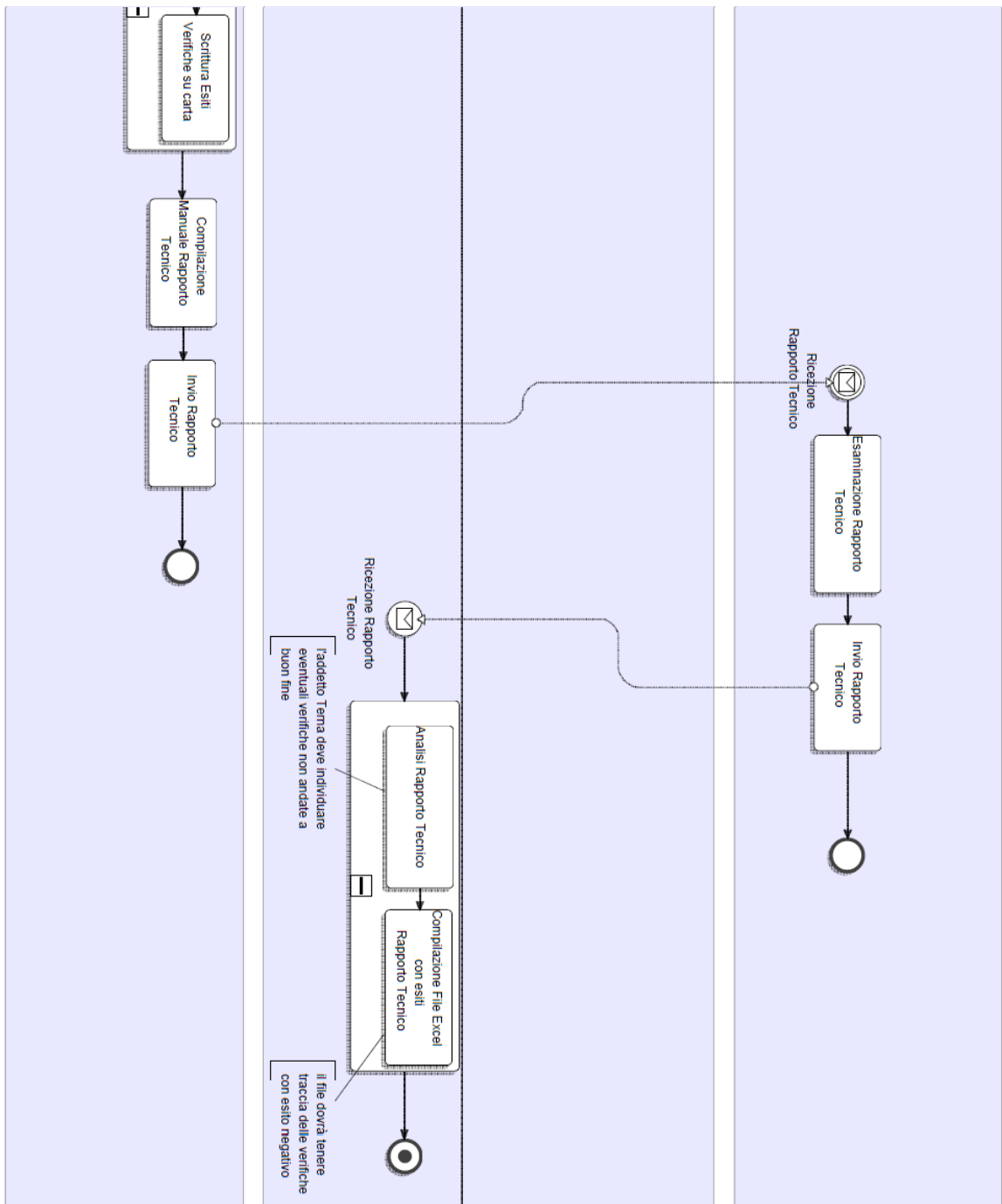


Figura 4.5 - Processo Certificazione BPMN AS-IS part2

4.5 CRITICITA' E PUNTI DI MIGLIORAMENTO DEL PROCESSO DI CERTIFICAZIONE

- Da una prima visione del processo, è evidente la presenza di attività non a valore aggiunto. In modo particolare è possibile identificare un'attività che è ripetuta da

attori diversi: *Invio Rapporto Tecnico*. Prima che l'addetto Terna possa analizzare il rapporto tecnico prodotto, occorre che lo stesso sia stato prodotto dal certificatore, poi inviato all'interrompibile e da quest'ultimo a Terna.

Si tratta di un passaggio di documentazioni obbligatorio per clausole contrattuali (contratto che viene firmato tra il cliente interrompibile e Terna) e quindi avente valenza legale, ma che palesemente dovrebbe essere eliminato, in quanto aggiunge inefficienza al processo stesso.

Dal momento che il contratto viene definito da Terna, la modifica di tale clausola contrattuale a monte e una reingegnerizzazione del processo a valle migliorerebbero inevitabilmente il processo.

- Adottando un approccio analitico di tipo time-driven, in cui si considera come driver fondamentale dell'esecuzione di un processo e, quindi, delle attività, il tempo, si possono individuare diverse attività, lato Terna e lato certificatore, che impiegano risorse per un tempo non trascurabile. Queste attività, che sono per lo più attività di scrittura, allo stato attuale sono svolte **manualmente** dai relativi attori: *Scrittura Informazioni Verifiche su Carta, Scrittura manuale Rapporto Tecnico, Scrittura Manuale Analisi Rapporto Tecnico.*

Occorrerebbe **automatizzare** determinate attività, che non possono essere eliminate, in maniera tale da ridurre il tempo di esecuzione e liberare le risorse.

- E' evidente la mancanza di un **canale di comunicazione diretto** tra il certificatore e Terna durante l'esecuzione dell'intervento di certificazione: tale mancanza rende invisibile a Terna tutto ciò che avviene tra l'inizio dell'iter di certificazione e la fine dello stesso, impedendo un aggiornamento in tempo reale sull'esito della certificazione stessa con la conseguente impossibilità di effettuare una stima attendibile sullo stato degli impianti e sull'effettiva durata del periodo di indisponibilità dei carichi del cliente interrompibile.

Per ovviare a questo problema, sarebbe necessario e utile introdurre un "nuovo" canale di comunicazione che colleghi direttamente Terna e il certificatore.

- Ultimo problema individuato riguarda l'aspetto di gestione delle informazioni sull'intervento di certificazione da parte del certificatore, in particolare:
 - data di certificazione;
 - sede dell'interrompibile;
 - causale e attività da svolgere ai fini della certificazione;
 - documentazione da consegnare.

Trattasi di informazioni utili ma non gestite in maniera integrata: è infatti il certificatore a dover provvedere ad una gestione personale dell'intervento, sia per quanto riguarda l'aspetto organizzativo sia quello operativo, rendendo carente di efficienza l'intero processo di certificazione, soprattutto in ottica di una gestione efficiente del tempo. Ciò risulta più chiaro se si considera che gli enti certificatori attualmente sono solo 2 e i certificatori, intesi come operatori che effettuano le attività di verifiche, sono in totale 3 o 4 in tutta Italia. Se, ad esempio, venissero posticipati gli interventi di certificazione dal certificatore, per via di una cattiva gestione degli appuntamenti, ciò creerebbe difficoltà e porterebbe svantaggi (con notevoli perdite economiche) sia per il cliente interrompibile che per Terna.

Risulta evidente che, trovando una soluzione che favorisca la gestione delle informazioni sugli interventi di certificazione per il certificatore, tutto il processo ne trarrebbe vantaggio.

4.6 VINCOLI ESTERNI

D'altra parte occorre considerare che allo stato attuale esistono in Terna, come già accennato in precedenza, una serie di vincoli sia operativi sia legislativi, che limitano il raggio d'azione della soluzione pensata:

- Il contratto di interrompibilità obbliga il cliente interrompibile all'invio a Terna del rapporto tecnico ricevuto dal certificatore.

Questo vincolo può essere superato solamente se Terna decide di modificare il contratto attualmente vigente, abolendo il vincolo esistente. E' una soluzione fattibile, ma necessita di un periodo di tempo non propriamente breve.

- La creazione di un'agenda di interventi per il certificatore comporterebbe la necessità di associare, ad ogni intervento di certificazione, il certificatore scelto dal cliente interrompibile. In questo modo, un' integrazione della soluzione con i sistemi informatici esistenti in Terna permetterebbe la ricezione automatica di informazioni utili da parte del certificatore, informazioni che allo stato attuale sono memorizzate in basi di dati popolate con i dati provenienti dalla procedura BMI.

Tale modifica è fattibile, ma comporta l'inserimento di un ulteriore campo nel form di compilazione dell'intervento di certificazione da parte del cliente interrompibile, oltre alla modifica dei database esistenti, per l'aggiunta di ulteriori informazioni, e dei servizi che recepiscono i dati e popolano il db. Dal momento che Terna esternalizza la gestione della procedura BMI e dell'IT in genere, anche questo vincolo potrà essere superato ma solo in un periodo di tempo non propriamente breve.

Tenendo in considerazione questi vincoli, la soluzione rilasciata è un **prototipo**, perfettamente funzionante e con tutte le caratteristiche ipotizzate. Ciò sarà di supporto a Terna per verificare e misurare il reale impatto della soluzione sul processo di certificazione. L'eventuale messa in esercizio del sistema è quindi demandata a una fase successiva.

5. SOLUZIONE: OBIETTIVO DEL PROGETTO

La soluzione individuata permette di colmare tutti (o quasi) i gap esistenti mediante la **tecnologia**. Sarà progettato e sviluppato un sistema software caratterizzato da una parte front end: un'**applicazione per dispositivo mobile** da fornire al certificatore e una parte **back end**. Il back end, situato presso le strutture di Terna, ospiterà sia l'application server, che avrà il compito di eseguire le funzionalità a supporto dell'applicazione client, sia il DBMS che memorizzerà i dati di interesse.

Al certificatore viene offerto un ambiente integrato per:

- Una gestione efficiente del proprio lavoro: ricevere, su un'agenda realizzata all'interno dell'applicazione, gli appuntamenti di certificazione concordati con i clienti interrompibili e informazioni utili sugli impianti dei clienti stessi (indirizzo, contatti e ulteriori dati inclusi nella documentazione tecnica)
- Ottenere l'elenco aggiornato della attività da svolgere in fase di certificazione qualora Terna dovesse aggiornarlo.
- Inviare a Terna, in tempo reale, le informazioni e i feedback sull'esito di ogni intervento di certificazione.

Il back end consentirà di:

- Implementare web services per gestire le richieste dei client e fornire servizi adeguati.
- Generare una coppia di file (pdf ed excel) contenenti gli esiti delle verifiche e ulteriori dettagli.

Il file pdf sarà inviato al certificatore e andrà integrare il rapporto tecnico ed, eventualmente, sarà messo a disposizione sulla procedura BMI per il cliente interrompibile, in maniera tale che quest'ultimo sia a conoscenza delle informazioni inserite.

Il file excel sarà messo a disposizione dell'addetto Terna per le analisi necessarie.

Terna potrà:

- Ricevere automaticamente le informazioni e i feedback sui risultati di ogni certificazione.
- Immagazzinare suddette informazioni nei propri sistemi informatici ed utilizzarle per future analisi statistiche.
- Avere, eventualmente, uno storico dettagliato degli interventi che sono stati effettuati in passato.

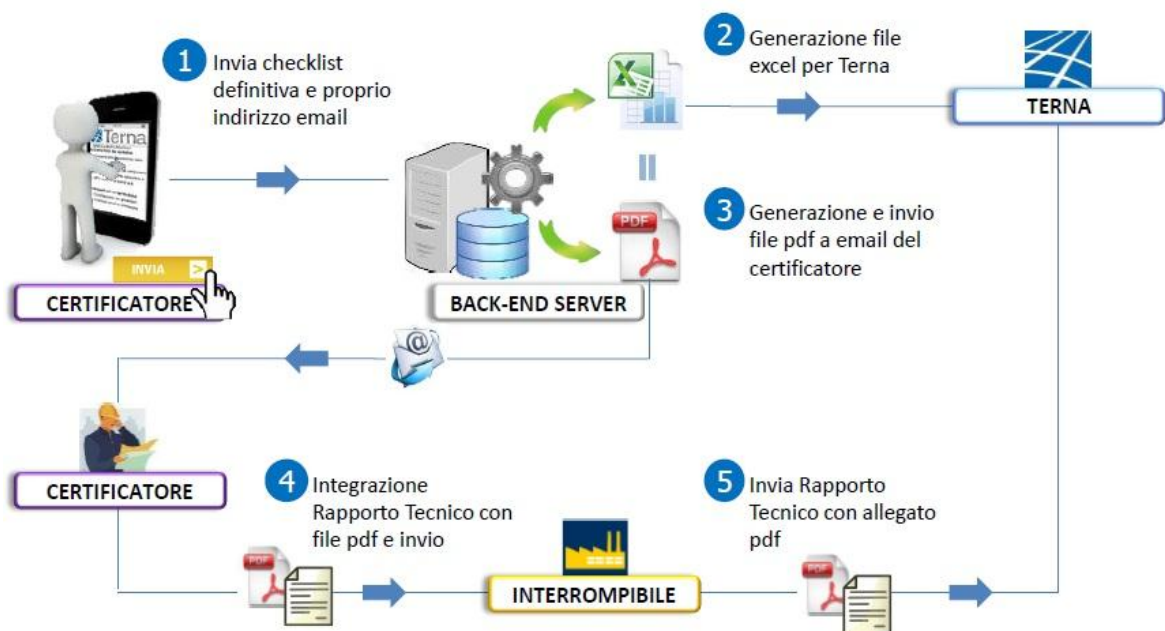


Figura 5.1 - Soluzione proposta

Tale soluzione permette effettivamente di risolvere tutte le criticità emerse in fase di analisi: consente di andare incontro alle necessità di Terna e di supportare il certificatore durante le procedure operative di certificazione, oltre ad evitare le azioni dirette di invio e ricezione del rapporto tecnico da parte del cliente interrompibile: tutti gli stakeholders del processo ne traggono vantaggio e il processo di certificazione stesso viene snellito e reso più efficiente.

5.1 I VANTAGGI DI UNA SOLUZIONE SOFTWARE

Il web è, allo stato attuale, uno dei mezzi di comunicazione preferito dalle aziende, inoltre costituisce la prima scelta per clienti, business partners, investitori, e altre parti

interessate. Esso è diventato pervasivo nelle vite di tutte le persone, sia a livello professionale che personale. L'espansione e le continue innovazioni delle tecnologie connesse al web offrono alle aziende nuove opportunità per supportare ogni tipo di attività, in quanto orizzontali rispetto alle diverse categorie dei processi di business: può essere introdotto all'interno di processi legati al Customer Relationship Management (CRM), ai processi produttivi, ai processi manageriali e così via. Nonostante il web sia stato introdotto da tempo nelle aziende per adempiere a specifici scopi, ad esempio il marketing, ci sono diverse opportunità di utilizzo delle diverse tecnologie connesse ad esso; il loro utilizzo può apportare un impatto considerevole in termini di miglioramento dei key performance indicator (KPI) di molti processi aziendali, fino a divenire un attore importante nel cambiamento strutturale di un'azienda. Particolarmente efficace può essere l'utilizzo del web attraverso i dispositivi mobili, come smartphone e tablet, grazie alle innovazioni apportate in termini di user experience e features fornite. Questi tipi di dispositivi offrono l'accesso a una serie di servizi innovativi, peculiari dei diversi sistemi operativi mobili, attraverso un'interazione intuitiva con l'utente; l'uso del paradigma orientato alle applicazioni mobili, che consente di avere a disposizione una serie di applicazioni che forniscono pochi servizi verticali per rispondere a specifici requisiti, incrementa la loro importanza e utilità all'interno delle attività aziendali. L'incremento nell'uso del paradigma orientato alle applicazioni mobili deve portare le aziende a prendere in considerazione nelle proprie strategie di sviluppare applicazioni da distribuire attraverso i market ufficiali delle diverse piattaforme.

La soluzione individuata, ovvero l'applicazione mobile, va perfettamente incontro alle necessità di un certificatore che giorno dopo giorno deve spostarsi in diverse location sul territorio italiano. Permettergli di avere costantemente sotto controllo il proprio lavoro, in qualunque momento, mediante un supporto mobile, da tenere sempre con se, facilita il lavoro svolto; ciò è un vantaggio non trascurabile.

5.2 HTML5 VS CODICE NATIVO

In un primo momento lo sviluppo dell'applicazione software era stato pensato esclusivamente per Android (anche su richiesta dell'azienda stessa), data la sua natura di sistema operativo mobile open source e flessibile e in quanto attualmente detiene la fetta più grande del mercato dei sistemi operativi mobili⁶:

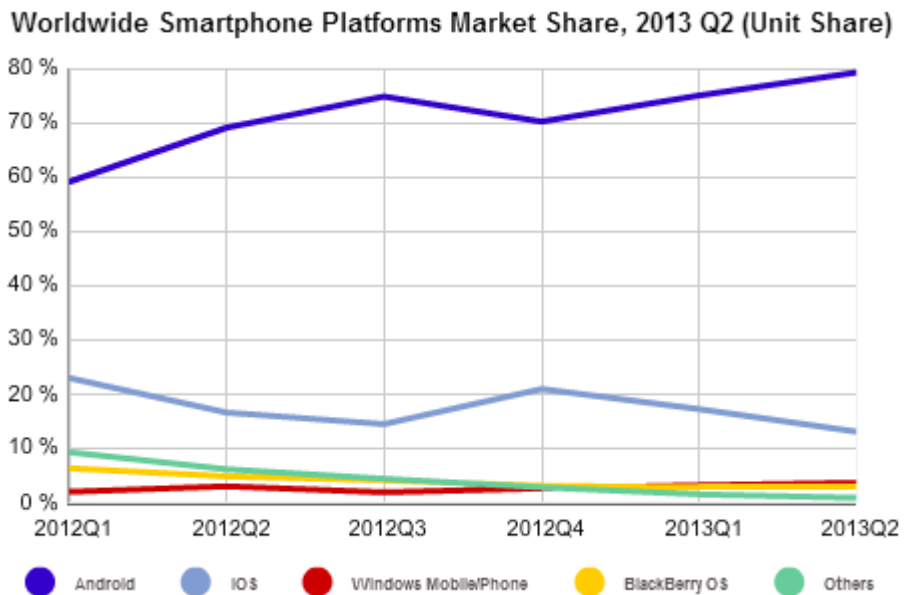


Figura 5.2 - Market Share Q2 2013

In un secondo momento l'attenzione è ricaduta su **HTML5** e **CSS3**, con orientamento, quindi, allo sviluppo multiplatforma: oggi i dispositivi in uso nel mondo hanno una maggiore pluralità di sistemi operativi, ma l'HTML5 ha compatibilità con tutti.

Questa scelta è stata attuata, dunque, non solo per andare incontro alle esigenze dei certificatori, che possono avere differenti dispositivi mobili, con differenti sistemi operativi, ma anche per favorire l'azienda stessa, in quanto i costi di sviluppo e manutenzione del prodotto, con questa soluzione, si riducono notevolmente.

A questo punto è doveroso fare una distinzione tra i concetti di **applicazioni native**, **applicazioni web** e **applicazioni ibride**.

⁶ IDC Worldwide Mobile Phone Tracker, August 7, 2013

- 1) Parlando di applicazioni native, s'intende solitamente una versione per Android e una iOS (Blackberry sta rapidamente perdendo quote di mercato e Windows Mobile è ancora una grande incognita). Le applicazioni native hanno il vantaggio di essere più veloci (mediamente del 10-15%) ma i costi e tempi di sviluppo delle stesse sono sicuramente maggiori, in quanto occorre farne una versione differente per ogni sistema operativo. Generalmente i cicli di sviluppo e testing sono molto più lunghi e, tipicamente, un utente che deve aggiornare l'app. all'ultima versione disponibile, deve effettuare il login allo store (Google Play ed Apple Store, tanto per citarne due) e scaricare l'ultima versione.

Le applicazioni native forniscono una migliore usabilità, migliori caratteristiche e, in generale, un'esperienza migliore. Ci sono caratteristiche che è possibile avere soltanto su le app native: multi touch, API grafiche veloci, animazioni fluide, componenti "built-in" (fotocamera, geolocalizzazione, rubrica), facilità d'uso, documentazione.

- 2) Un'applicazione web si fa portavoce dell'approccio "Write Once, Run Anywhere": è ospitata su un web server e può essere eseguita all'interno di qualunque browser e su qualunque dispositivo. Un aspetto importante dell'approccio dell'HTML5 "write-once-run-anywhere" è legato proprio alla distribuzione e al supporto, che risultano essere molto più semplici rispetto alle app. native: occorre "fixare" un bug o aggiungere una caratteristica? Sono tutte operazioni che vengono eseguite una volta sola e i cui risultati sono resi disponibili a tutti gli utenti. Per accedere ad una web app, però, è necessaria una connessione dati. Altro lato negativo è legato all'impossibilità di accesso ad alcune delle caratteristiche native del dispositivo stesso.
- 3) Un'applicazione ibrida (come nel caso in esame) è sviluppata con tecnologie web, quali HTML5 e JavaScript, ma viene eseguita all'interno di un app. container nativo, che fornisce direttamente l'accesso alle features native; inoltre non vi è necessità alcuna di avere il dispositivo connesso alla rete dati. Combina, dunque, il meglio (o il peggio) sia delle app. native che di quelle web. Esistono framework, inoltre, come Adobe PhoneGap, IBM Worklight e Titanium Appcelerator che supportano il

programmatore: a partire da un linguaggio “X” più diffuso e conosciuto, sono in grado di generare app. native avvalendosi di traduttori e delle SDK proprietarie di iOS, Android, BB, Symbian, etc. [3].

L’HTML5 risulta la soluzione più economica, perché con un’unica attività di sviluppo⁷, si realizzano applicazioni buone per tutti i sistemi operativi, sia desktop che mobile. L’unico aspetto negativo è che soffre di una minore performance rispetto ai linguaggi nativi.

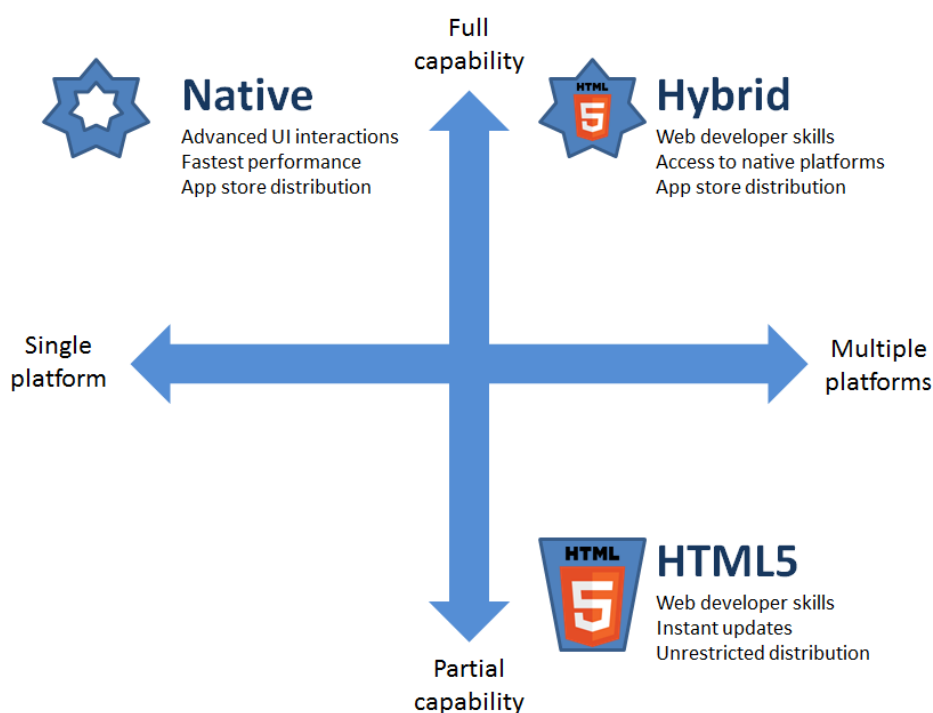


Figura 5.3 - App. Native vs App. Ibride

Lo sviluppo multiplatforma in ambito mobile consiste, dunque, nella produzione di software progettato per supportare dispositivi diversi sia per caratteristiche hardware che sistemi operativi. Questo processo si avvale di tecnologie e strumenti specifici e consente di scrivere programmi che si adattano automaticamente o con sforzo minimo al contesto in cui vengono eseguiti.

HTML5 tramite un insieme di API innovative ed un markup potenziato permette di creare applicazioni sul web avanzate senza ricorrere a software di terze parti, usati tramite plug-

⁷ Escludendo generalmente un 20% del codice, o anche meno (in relazione all’applicazione sviluppata), che fa da bridge per l’accesso ai sensori del particolare device o ad altre funzionalità del sistema operativo altrimenti inaccessibili.

in nei browser (esempio Flash o JavaFx). HTML5 può essere utilizzato anche nello sviluppo multiplatforma di applicazioni che risiedono interamente sui dispositivi mobili e desktop. Questa possibilità è fornita dallo sviluppo di diverse specifiche innovative che permettono di utilizzare le funzionalità del sistema operativo, come audio e video in maniera nativa, oppure l'accesso ai sensori dei dispositivi mobile tramite Javascript.

A valle di ciò è naturale chiedersi: qual è, quindi, il migliore approccio utilizzabile? In realtà non esiste una risposta universale, ma tutto dipende da ciò che l'applicazione deve rendere disponibile all'utente finale, dal "servizio" che deve offrire. Per rispondere correttamente, occorre tenere in considerazione i principali aspetti di un'applicazione mobile e dei risultati che vogliamo ottenere dalla stessa:

- Performance: che tipo di performance è richiesta dalle funzionalità dell'applicazione? Ad esempio, si richiede all'app. di immagazzinare dati di utilità in cache o su database locali? Ciò è supportato solo da app. native o ibride.
- Connettività e disponibilità: che tipo di connettività richiede l'applicazione? Richiede accesso al web per tutto il tempo di esecuzione, in modo da recuperare sempre i dati aggiornati dal server? O l'accesso al web non è necessario? App. native e ibride possono operare offline, mentre le web app richiedono necessariamente connettività.
- Complessità: Quanto è complessa l'applicazione? E' necessaria un'app. veloce che acceda ad un database o ad un web service per recuperare dati da mostrare all'utente? In questo caso una web app. dovrebbe essere sufficiente. Comunque, ci sono casi in cui il supporto dell'app. a funzionalità critiche è necessario, ecco perché in questi casi è necessario utilizzare applicazioni ibride o native.
- Ricchezza nell'interattività dell'utente: che tipo di esperienza utente è richiesta per l'app.? Le app. mobile basate su web dovrebbero essere sufficienti per operazioni che richiedono semplici interazioni da parte dell'utente. Comunque, applicazioni ibride/native sono tipicamente richieste per fornire all'utente esperienze altamente interattive. In casi di app. di gioco, le applicazioni native potrebbero essere necessarie per offrire una migliore user experience.

- Requisiti multi-piattaforma: oggi come oggi ognuno porta i propri dispositivi a lavoro (BYOD - Bring Your Own Device) e spesso ci si aspetta che vengano utilizzati per eseguire parti del lavoro che spetta loro. Provare a supportare ogni piattaforma in maniera nativa richiede risorse sempre maggiori, oltre a costi di manutenzione, derivati dai frequenti aggiornamenti delle versioni per ciascuna piattaforma. Laddove è necessario il supporto multipiattaforma, le app. mobili o ibride probabilmente sono avvantaggiate. Utilizzare app. native, e provare a supportare differenti sistemi operativi può essere proibitivo in termini di costi, in base alle risorse richieste e alle skill di sviluppo, richieste agli sviluppatori.
- Accesso ai servizi del dispositivo: se l'applicazione richiede accesso ai servizi locali, quali fotocamera, accelerometri, geo localizzazione, etc. , allora la scelta è limitata alle app. native o ibride. [4]

A valle di questa analisi risulta chiaro come la scelta di utilizzare un'applicazione ibrida in HTML5, in questo caso preso in esame, sia risultata la scelta che meglio rispondeva alle esigenze esistenti.

5.3 PHONEGAP

Abbiamo visto quali sono principalmente i pro e i contro legati alle differenti tipologie di sviluppo di un'applicazione mobile. Ma i principali dubbi restano: quale deve essere la piattaforma target dell'applicazione? In quale linguaggio deve essere scritta?

PhoneGap⁸ ci viene incontro. L'obiettivo di questo framework è proprio quello di eliminare tutte le complicazioni, legate a particolari scelte di sviluppo, e i problemi di sviluppare applicazioni per una specifica piattaforma, usando il supporto del linguaggio nativo per ogni sistema operativo, e lasciando al programmatore solo il compito di

⁸ Spesso ci si riferisce a PhoneGap anche con il nome di Cordova, progetto open source Apache, nato nel 2011 che ha sostituito l'iniziale progetto PhoneGap. Si può specificare, per chiarezza, che le attuali librerie PhoneGap sono chiamate Cordova.jar. I due progetti sono esattamente la stessa cosa.

sviluppare applicazioni cross-platform, usando HTML, CSS e JavaScript (tecnologie web esistenti e consolidate che sono familiari alla maggior parte degli sviluppatori e non).

PhoneGap, dunque, viene utilizzato nello sviluppo delle applicazioni mobili per differenti motivazioni:

- Se l'applicazione mobile era già stata sviluppata usando tecnologie web e la si vuole rendere deployabile attraverso gli application stores (Google Play, Apple Store, etc.)
- Se si vuole sviluppare un'applicazione mobile sfruttando personali conoscenze in ambito web, ma per la quale occorre far leva su features lato dispositivo che non sono supportate dal browser mobile
- Se si vuole implementare un prototipo di un'applicazione mobile e non si ha tempo di imparare Java o altri linguaggi orientati agli oggetti

Molte sono le applicazioni commerciali che oggi fanno uso di PhoneGap. E' principalmente utilizzato per applicazioni utilizzabile lato Consumer, come giochi, applicazioni social media, etc., ma sempre più aziende si stanno convertendo all'uso di PhoneGap per lo sviluppo delle applicazioni aventi come target il Business stesso.

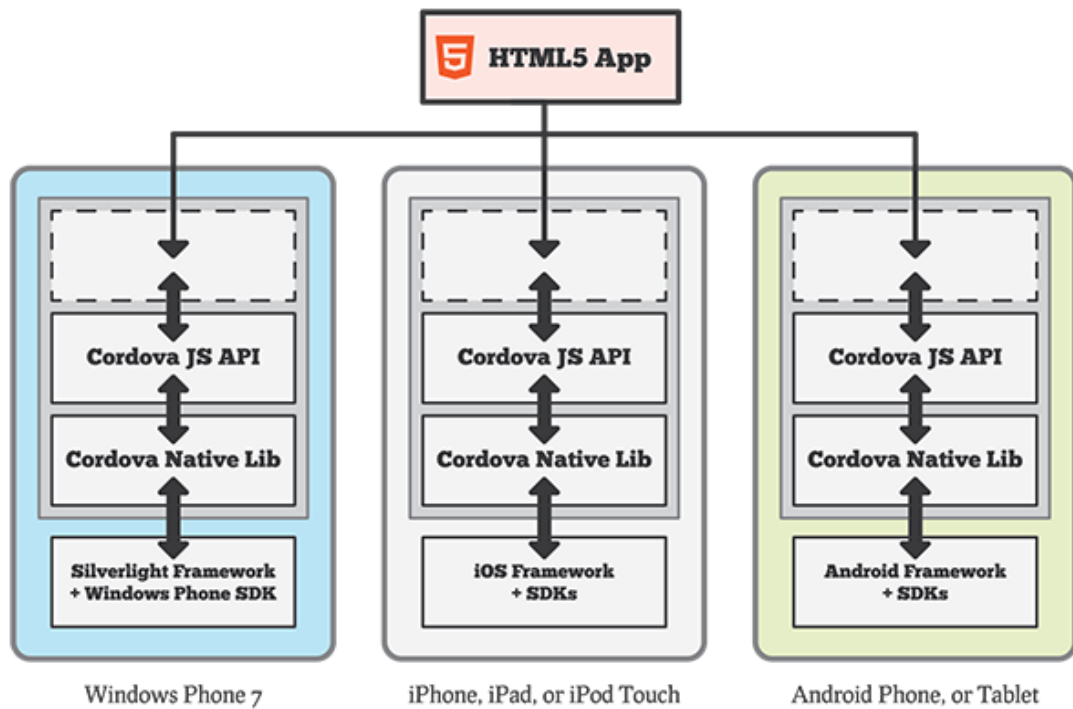


Figura 5.4 - Applicazioni Mobili in HTML5 e PhoneGap

Come già anticipato, un programmatore sviluppa un'applicazione web per il dispositivo mobile e in seguito a ciò, tool speciali forniti da PhoneGap "impacchettano" l'applicazione web in un'applicazione nativa per ciascuna piattaforma mobile.

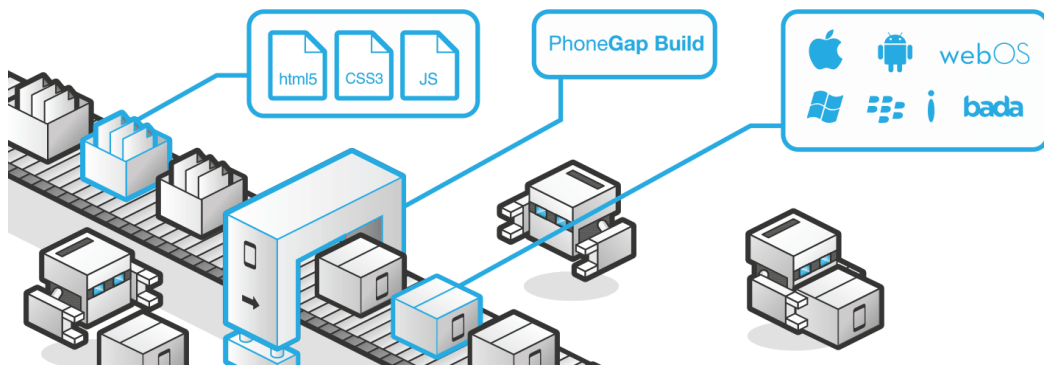


Figura 5.5 - PhoneGap Packaging Process

All'interno dell'applicazione nativa, l'interfaccia utente consiste semplicemente di un'unica schermata, una vista web, che occupa tutto lo spazio sullo schermo del dispositivo. Quando l'applicazione viene lanciata, essa carica la pagina di startup dell'app. web (tipicamente la index.html) e poi passa il controllo alla vista web per consentire all'utente di interagire con l'app. web. Quando l'utente interagisce con il contenuto dell'applicazione, codice JavaScript può caricare altro contenuto all'interno del codice dei

file sorgente inclusi nell'app. o può richiedere contenuto dal web o da un application server.

Per quanto riguarda l'accesso alle features native del dispositivo, quando l'applicazione ha necessità di accedervi, chiama le API PhoneGap utilizzando JavaScript, ed in seguito a ciò uno speciale layer all'interno dell'applicazione trasforma la chiamata all' API PhoneGap nella particolare API nativa per la specifica caratteristica richiesta. [5]

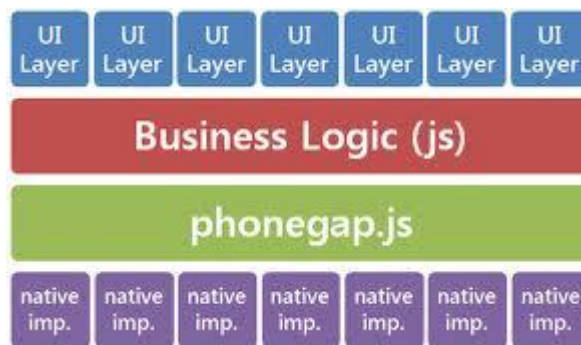


Figura 5.6 - Struttura Applicazione con PhoneGap

Per completezza occorre specificare che PhoneGap non è l'unico framework esistente sul mercato. Altro framework ampiamente utilizzato è Titanium Appcelerator, sempre open source e consente, allo stesso modo di PhoneGap, lo sviluppo di applicazioni ibride. Titanium lavora molto similmente a PhoneGap: l'applicazione nativa che gira sul dispositivo è soltanto un container che esegue codice JavaScript. Le interfacce utente dell'applicazione e la logica applicativa sono codificate interamente in Javascript.

5.4 SQLITE

SQLite è un package software di pubblico dominio, in linguaggio C, che fornisce un RDBMS (Relational DataBase Management System). I database relazionali vengono utilizzati per memorizzare record definiti dall'utente in grandi tabelle. Oltre all'immagazzinamento e alla gestione dei dati, un motore di database può processare comandi di query complesse che combinano i dati da più tabelle per generare report e sintesi di dati.

SQLite è definito dalle seguenti caratteristiche:

- Servless: non richiede un processo server separato o un sistema per operare. Le librerie SQLite accedono direttamente ai file memorizzati.
- Zero Configuration: non occorre configurare nulla. Creare un'istanza di un database SQLite è semplice come aprire un file.
- Cross-Platform: l'istanza intera del database risiede su un singolo file cross-platform, che non richiede amministrazione.
- Self-Contained: un'unica libreria contiene l'intero sistema di database, che si integra direttamente in un'applicazione host, a differenza della maggior parte dei prodotti RDBMS, che richiedono un'architettura client/server.
- Small Runtime Footprint: il settaggio di default del database occupa meno di 1 MB di codice e richiede solo pochi MB di memoria. Con alcuni miglioramenti, si può ridurre notevolmente sia la dimensione della libreria che l'uso di memoria.
- Transazionale: le transazioni sono completamente ACID, consentono accessi sicuri da processi e thread multipli.
- Full-Featured: SQLite supporta la maggior parte delle caratteristiche di un query language, contenute nello standard SQL92 (SQL2).
- Altamente Affidabile.

Ma presenta dei limiti legati in parte alla finalità di semplice database da incorporare in altre applicazioni:

- non offre le stored procedure.
- non prevede la gestione dei permessi d'accesso, demandata al software con cui si interagisce con il database e/o al meccanismo dei permessi del file system;
- non ha una vera gestione della concorrenza: le applicazioni che lo utilizzano, se necessario, devono implementarla;
- per garantire la coerenza del file del database sono usati i lock del file system, e quindi vi possono essere problemi qualora quest'ultimo non li implementi correttamente, ad esempio con file system di rete;
- non offre alcuna cache per le query (e non ne ha la possibilità, non esistendo un processo server centrale);

- non ha protocolli di rete, non essendo utilizzabile come programma a sé; è possibile utilizzare un database remoto, ma solo tramite file system di rete del sistema operativo, con prestazioni difficilmente accettabili;
- non supporta alcuni importanti costrutti SQL quali RIGHT JOIN e FULL OUTER JOIN;
- non supporta le sotto-query variabili.
- non supporta la scrittura diretta nelle viste (occorre usare trigger "INSTEAD OF");
- non consente di modificare, cancellare o rinominare le colonne di una tabella: il comando ALTER TABLE è infatti limitato alla modifica del nome della tabella e all'aggiunta di colonne in coda alla stessa. Molti front-end di terze parti aggirano comunque il limite rigenerando in modo trasparente la tabella (perdendo però trigger e indici ad essa correlati);
- non supporta trigger di tipo "FOR EACH STATEMENT" (solo "FOR EACH ROW", eventualmente combinato con "INSTEAD OF");

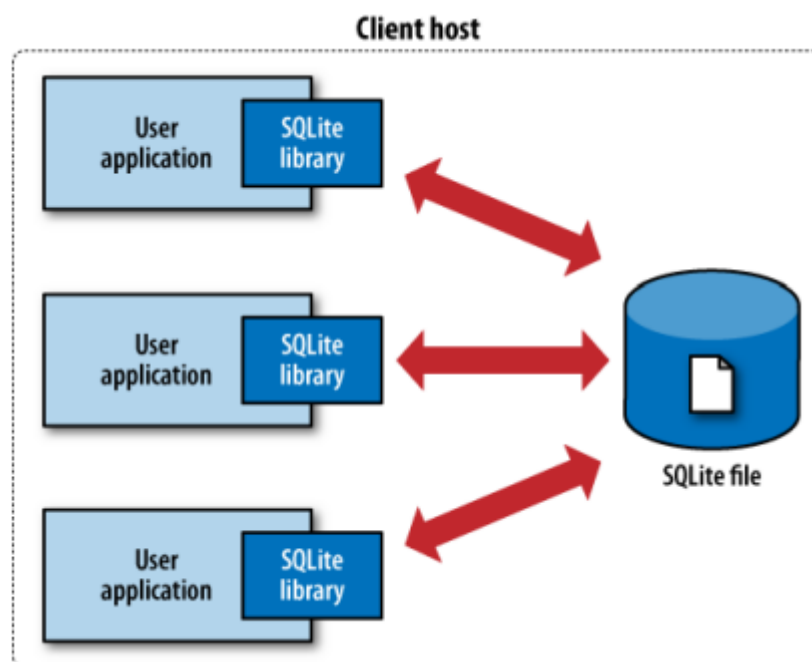


Figura 5.7 - Architettura SQLite

Il fatto che il server sia eliminato in questa architettura ha imposto alcune restrizioni: SQLite è progettato per indirizzare informazioni/dati memorizzati localmente. Ciò implica che non risulta adatto a situazioni in cui più client debbano accedere ad un database centralizzato.

Attualmente è impiegato in alcuni dei principali browser per memorizzare cronologia di navigazione, bookmarks, e in applicazioni che richiedono la memorizzazione di altre informazioni⁹.

Grazie alla sua semplicità, SQLite risulta uno “strumento” adatto ad essere utilizzato su ogni tipo ambiente, inclusi smartphone, console di gioco, e altri, laddove un tradizionale database system non sarebbe potuto esistere. [6]

D’altro canto è stato utilizzato proprio nel progetto qui in esame, allo scopo di memorizzare le informazioni, riguardanti lo stato di una certificazione, inserite dai certificatori tramite l’applicazione mobile. Informazioni che, prima di un invio definitivo al server, necessitano di essere memorizzate localmente e recuperate ogni qual volta l’utente lo richieda, per semplici revisioni/visualizzazioni o modifiche.

5.5 SOAP VS REST

L’architettura funzionale comprende oltre al dispositivo mobile (più nello specifico l’applicazione mobile installata su di esso), una parte back end situata presso le strutture di Terna.

Il back end comprende:

- le strutture dati e la logica dell’applicazione
- la base di dati ospitante tutte le informazioni necessarie per il corretto funzionamento dell’applicazione

Per la comunicazione tra i componenti viene proposta un’architettura basata su **web services**, in modo da avere un sistema modulare, facilmente modificabile ed espandibile in futuro. A tale scopo, quindi, la struttura di back end avrà il compito di:

- consentire a Terna di gestire l’elenco della checklist di attività da effettuare in fase di certificazione

⁹ Firefox, Mozilla Thinderbird, Skype, etc.

- recepire ed elaborare i dati e le informazioni ricevuti dall'applicazione mobile
- mantenere un database contenente tutti gli interventi di certificazione effettuati
- realizzare l'integrazione dell'agenda del certificatore con il calendario delle attività di manutenzione programmata gestito dagli strumenti informatici di Terna.

In questo scenario è immaginabile un flusso di dati bidirezionale: il certificatore avrà la possibilità di scrivere determinate informazioni che verranno poi trasmesse a Terna, quali ad esempio l'intera checklist delle attività e dei task associati, svolti durante l'iter di certificazione, mentre dovrà ricevere gli aggiornamenti sugli interventi da effettuare giornalmente.

La sincronizzazione dispositivo mobile – database si è supposto avvenga con frequenza relativamente bassa, ad esempio giornaliera.

Per lo scambio delle informazioni, è stato condotto uno studio iniziale sulle principali tecnologie per la comunicazione e lo scambio dati a supporto dei servizi web: il protocollo **SOAP** e lo stile architetturale **REST**.

Da questo semplice confronto è stato fin da subito evidente come lo stile REST, unitamente al formato dati **JSON**, fosse nettamente più versatile, leggero e scalabile oltre a rendere lo sviluppo del codice più immediato e semplice. La peculiarità di REST è caratterizzata proprio dal fatto che i dati trasmessi sulla rete, a differenza di SOAP, non devono essere esclusivamente in formato XML. Da qui è nato il formato dati JSON, che permette di rappresentare i dati come una sequenza di informazioni del tipo *nome:valore*, eliminando così la necessità di inviare sulla rete tag xml e header vari, diminuendo in questo modo nettamente l'overhead legato alla dimensione dei dati spediti sul canale di comunicazione: i pacchetti dati si riducono e la comunicazione risulta più rapida. D'altro canto l'XML è un linguaggio di markup potente, in quanto consente di descrivere documenti strutturati, oltre a favorire l'esportazione di dati tra i diversi DBMS. [7]

I risultati del confronto sono stati riportati, sintetizzati, in tabella:

SOAP	REST
PRO	
Standard prevalente per i Web services	Vicino alla filosofia progettuale alla base del Web
Progettato per gestire ambienti di calcolo distribuito	Indipendente da linguaggio e piattaforma
Gestione errori	Scalabilità nell'interazione tra i componenti
Estensibilità	Deployment indipendente dei componenti
Essendo basato su XML, ha natura testuale e quindi la lettura e la comprensione è facilitata per l'uomo	Generalità delle interfacce
	Le regole di sicurezza in REST possono essere impostate tramite gli standard HTTP: un firewall (o un amministratore di rete) possono comprendere l'intento del messaggio analizzando il comando nella richiesta HTTP.
	Ogni messaggio include sufficienti informazioni che descrive come va processato
	Non richiede il parsing XML e non necessita di header del messaggio da e verso il server: usa quindi meno banda rispetto a SOAP
CONTRO	
Concettualmente più complesso e verboso di REST	Mancanza di supporto standard per gestione la sicurezza e l'affidabilità dei messaggi: i servizi con requisiti più sofisticati sono più complicati da

	sviluppare
Più "difficile" da sviluppare, richiede tools appropriati	

Ho deciso di non soffermarmi troppo sulla distinzione tra SOAP e REST, in quanto la scelta è ricaduta, per richiesta da parte di Terna, su SOAP: gli attuali servizi web esposti dai loro server sono basati esclusivamente su questo protocollo di comunicazione.

5.6 IL PROTOCOLLO SOAP

SOAP (Simple Object Access Protocol) è un protocollo universale per la trasmissione di dati di RPC, o di chiamate a Servizi Web (implementazioni specifiche di RPC) . Una RPC è una chiamata del codice residente su un altro server (passandogli eventualmente dei parametri) e la ricezione della risposta, anch'essa con un'opportuna codifica.

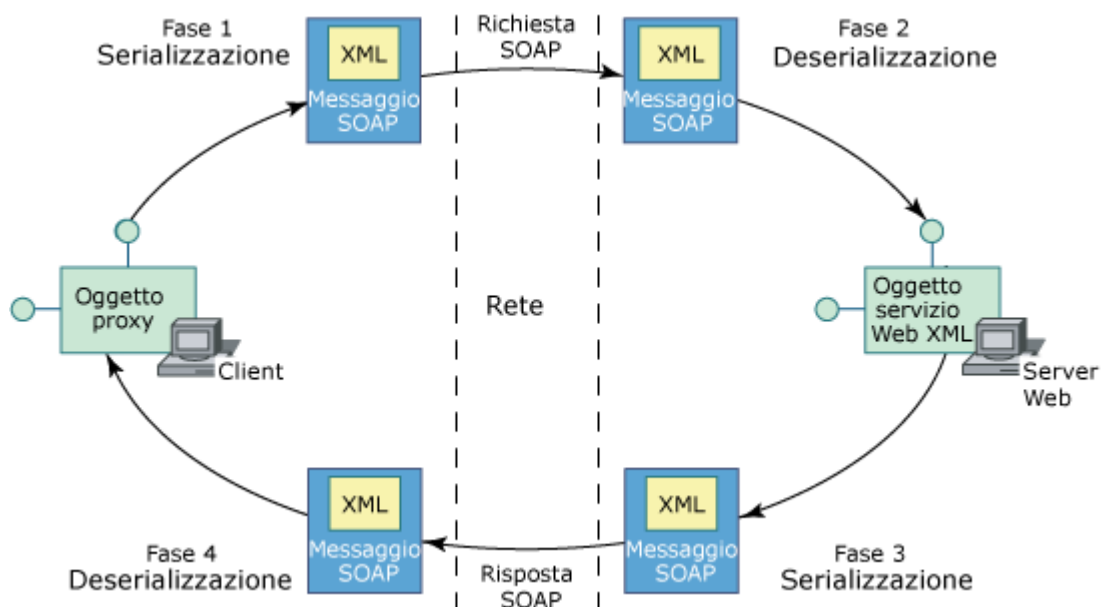


Figura 5.8 - Funzionamento protocollo SOAP

Esso si usa quando due framework diversi devono comunicare tra di loro e, generalmente parlando, quando si devono scambiare informazioni in ambiente distribuito. Non si basa su tecnologie proprietarie e la sua applicazione è completamente libera. Ovviamente SOAP non è l'unico standard di codifica esistente per trasmettere dati di RPC ma, in questo documento trattiamo solamente SOAP, appunto, e REST.

Le seguenti sono alcune delle caratteristiche salienti di SOAP:

- È basato su XML;
SOAP è basato su XML, ha quindi natura testuale (a differenza dei suoi colleghi CORBA, DCOM, .NET etc...). Questo fa sì che il debugging sia semplificato perché XML è leggibile anche da esseri umani. Purtroppo, questa sua natura testuale rappresenta anche il suo principale punto debole in quanto lo rende molto meno performante rispetto alle sue controparti binarie (CORBA e .NET Remoting in particolare)
- Gestisce informazione strutturata;
- Gestisce informazione tipata;
- È pensato per usare HTTP come protocollo "di trasporto";
I framework SOAP come Apache SOAP 2.2 e Apache AXIS rendono quasi del tutto trasparente questo approccio. Questa trasparenza è estremamente importante per il programmatore dato che semplifica notevolmente le RPC.
- Non definisce alcuna semantica per applicazioni o scambio di messaggi, ma fornisce un mezzo per definirla.
Grazie ad un intenso uso dei Namespace XML, SOAP permette agli autori dei messaggi di dichiararne la semantica usando grammatiche XML definite per lo scopo in particolari namespace.

5.4.1 STRUTTURA DI UN MESSAGGIO SOAP

SOAP è pensato per usare HTTP come trasporto. A questo proposito è estremamente interessante comprendere come è strutturato un messaggio SOAP.

Un messaggio SOAP è composto da:

- Un elemento radice, **envelope**, obbligatorio. Il namespace di SOAP viene dichiarato all' interno di questo elemento.
- Un elemento **header** opzionale. Il suo scopo è quello di trasportare informazioni non facenti parte del messaggio, destinate agli "attori", cioè alle varie parti che il messaggio attraverserà per arrivare al suo destinatario finale.
- Un elemento **body** obbligatorio. Questo elemento contiene il messaggio vero e proprio.



Figura 5.9 - Struttura messaggio SOAP

5.4.2 HEADER SOAP

L'header contiene informazioni sulla sicurezza, il routing, i formati, il destinatario finale

- L'header deve essere il primo figlio di *envelope*
- Tutti gli elementi figli di *header* vanno considerati singole header entries;

Come per il *body*, non esiste un formato per le header entries: la grammatica va importata da altri namespaces.

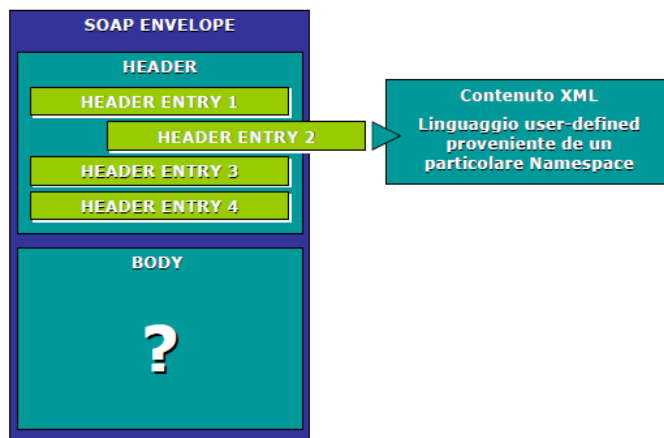


Figura 5.10 - Header SOAP

5.4.3 BODY SOAP

Il body contiene le informazioni per il destinatario finale del messaggio.

- Il contenuto del *body* è definito dall'implementatore del messaggio tramite l'uso di un particolare namespace.
- Il destinatario dovrà elaborare i dati ed obbedire alla loro semantica, specificata nel namespace di appartenenza.

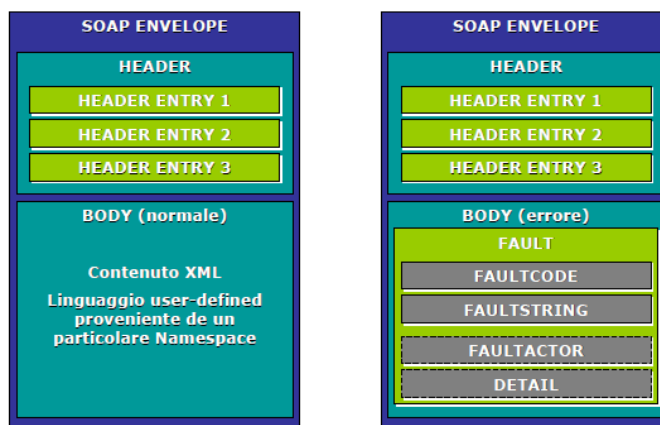


Figura 5.11 - Body SOAP

5.4.4 SOAP ENCODING

Il concetto di encoding riguarda le informazioni applicative contenute nel tag Body: definisce come i dati vengono rappresentati in XML. Ad esempio, è possibile capire direttamente dal flusso XML se un determinato tag è di tipo stringa o numerico. Consideriamo l'esempio

`<x1>12</x1>`

Il valore 12 potrebbe essere numerico, ma il tag potrebbe anche potenzialmente essere di tipo alfanumerico. Con l'encoding SOAP avremmo:

```
<x1 xsd:type="String">12</x1>
```

Sappiamo cioè che è una stringa e questo semplicemente dando un'occhiata al namespace.

5.4.5 ESECUZIONE DI RPC CON SOAP

SOAP è un protocollo usato per la trasmissione di dati di RPC e ciò è trasparente per il programmatore. Vediamo adesso come avviene.

La codifica in una RPC con SOAP segue alcune regole convenzionali. Esse riguardano:

- L'indicazione della **risorsa** (oggetto) alla quale è indirizzata la chiamata;
- L'indicazione del **metodo** da invocare;
- L'eventuale trasmissione della **signature** del metodo, per una sua più corretta identificazione. Con signature si intende la trasmissione dei parametri del metodo e del valore di ritorno dello stesso.

È molto importante notare che l'indicazione della risorsa (oggetto) alla quale è indirizzata la chiamata avviene in maniera dipendente dal protocollo usato per trasportare il messaggio. Usando HTTP, sarà la URI (Uniform Resource Identifier) richiesta al server a specificare l'oggetto del quale si vuole invocare un determinato metodo. Per chiarire il concetto, si considera un esempio che spiega come avviene la chiamata, da parte del client, al metodo pubblico che si trova sul server.

Esempio: si consideri il seguente metodo sayHelloTo situato nella classe HelloWorld.java. Intendo invocare tale metodo con SOAP/RPC:

```
public class HelloWorld{
    public String sayHelloTo(String name, int hh){
        if(hh>18)
            return "Buona Sera" + name;
        else
            return "Buon Giorno" + name;
        }
    }
```


La chiamata al metodo viene codificata come una struttura XML nel body del payload SOAP. Per fare ciò si passa attraverso 3 diverse fasi.

Fase 1: Codifica XML della signature del metodo.

- Si usa la codifica standard di SOAP/RPC per descrivere lo schema degli elementi che incapsulano la chiamata al metodo.
- L'elemento sayHelloTo conterrà una sequenza di elementi name e hh, corrispondenti agli argomenti del metodo e con gli stessi tipi (stringa e intero).

```
<xs:element name="sayHelloTo">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="hh" type="xs:integer"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Fase 2: Creazione del messaggio SOAP.

Il messaggio contenente la RPC viene creato seguendo le regole sintattiche di SOAP ed associato allo schema appena creato.

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:sayHelloTo xmlns:m="http://miosito.com/helloworld">
      <name>Mionome</nome>
      <hh>16</hh>
    </m:sayHelloTo>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Fase 3: Creazione della request HTTP.

- Il nome dell'oggetto da chiamare (HelloWorld) è specificato all'interno dell'URI richiesta, mentre l'header SOAPAction riporta anche il nome del metodo da

chiamare. Il payload della richiesta POST contiene il messaggio preparato nella fase precedente.

```
POST /HelloWorld http/1.1
Content-Type: text/xml; charset="utf-8"
Content-Length: 321
SOAPAction: "http://miosito.com/HelloWorld#sayHelloTo"
<SOAP-ENV:Envelope
.....
</SOAP-ENV:Envelope>
```

In Figura 5.12 - Messaggio SOAP di richiesta ad un web service di prova” è possibile vedere il contenuto di un pacchetto scambiato tra client e server, visualizzato mediante il programma Wireshark. Il pacchetto rappresenta il contenuto di una richiesta da client ad un web service, per ottenere la differenza tra l’anno di nascita dell’ utente, passato come parametro alla richiesta, e l’anno corrente:

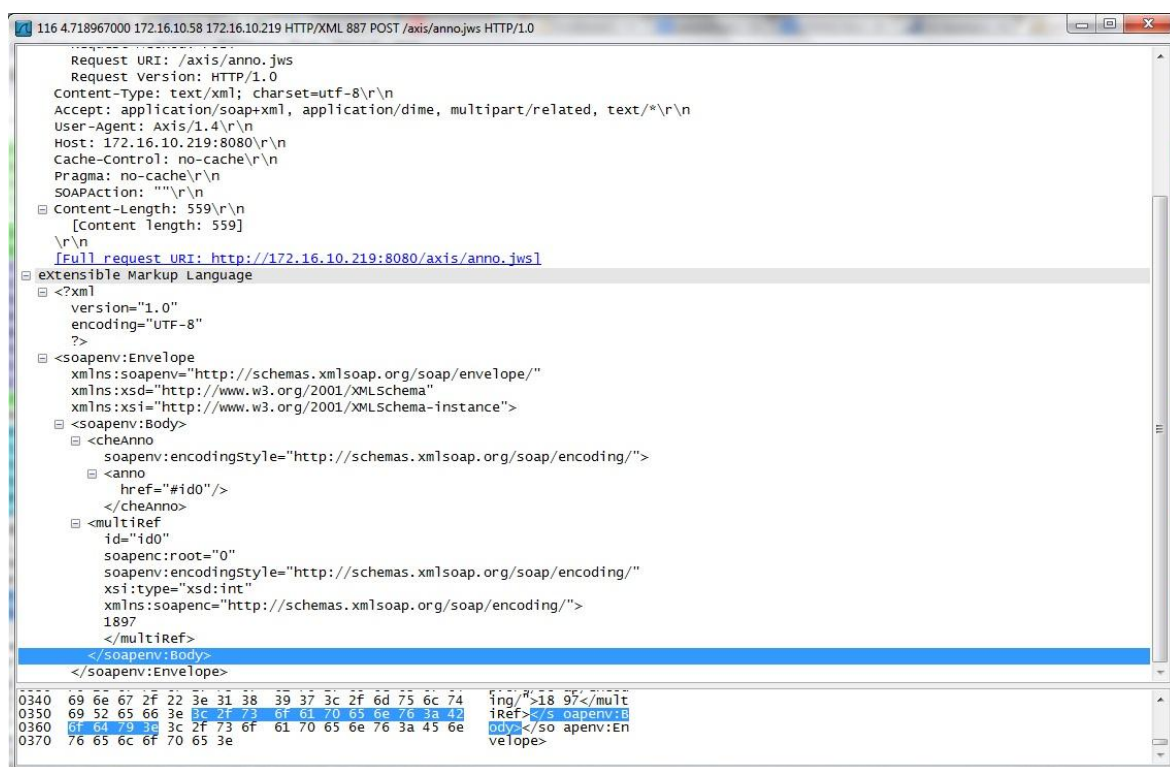


Figura 5.12 - Messaggio SOAP di richiesta ad un web service di prova

5.4.6 IL FILE WSDL E UDDI

Componenti importanti nell'implementare un'applicazione client – server sono proprio i file **WSDL** e **UDDI**. Il file WSDL non è altro che il descrittore dell'interfaccia di un web service.

Il linguaggio WSDL (Web Service Description Language) descrive quindi il servizio e come deve essere usato. L'UDDI (acronimo di Universal Description Discovery and Integration) è un registry (ovvero una base dati ordinata e indicizzata), basato su XML e indipendente dalla piattaforma hardware, che permette alle aziende la pubblicazione dei propri dati e dei servizi offerti su internet. L'UDDI è uno degli standard alla base del funzionamento dei Web Service: è stato progettato per essere interrogato da messaggi in SOAP e per fornire il collegamento ai documenti WSDL che descrivono i vincoli protocollari e i formati dei messaggi necessari per l'interazione con i Web Service elencati nella propria directory.

Di seguito viene schematizzato il funzionamento alla base del protocollo SOAP.

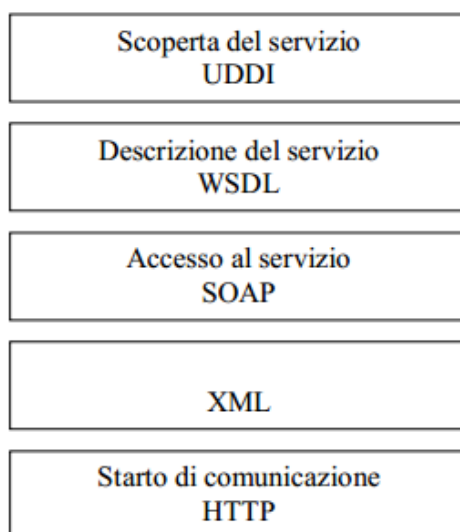


Figura 5.13 - Schematizzazione funzionamento SOAP

5.7 IL PROCESSO DI CERTIFICAZIONE TO-BE

In seguito a quanto detto nei paragrafi precedenti, ecco come si configura il processo di certificazione dopo l'introduzione della soluzione individuata, nel breve termine:

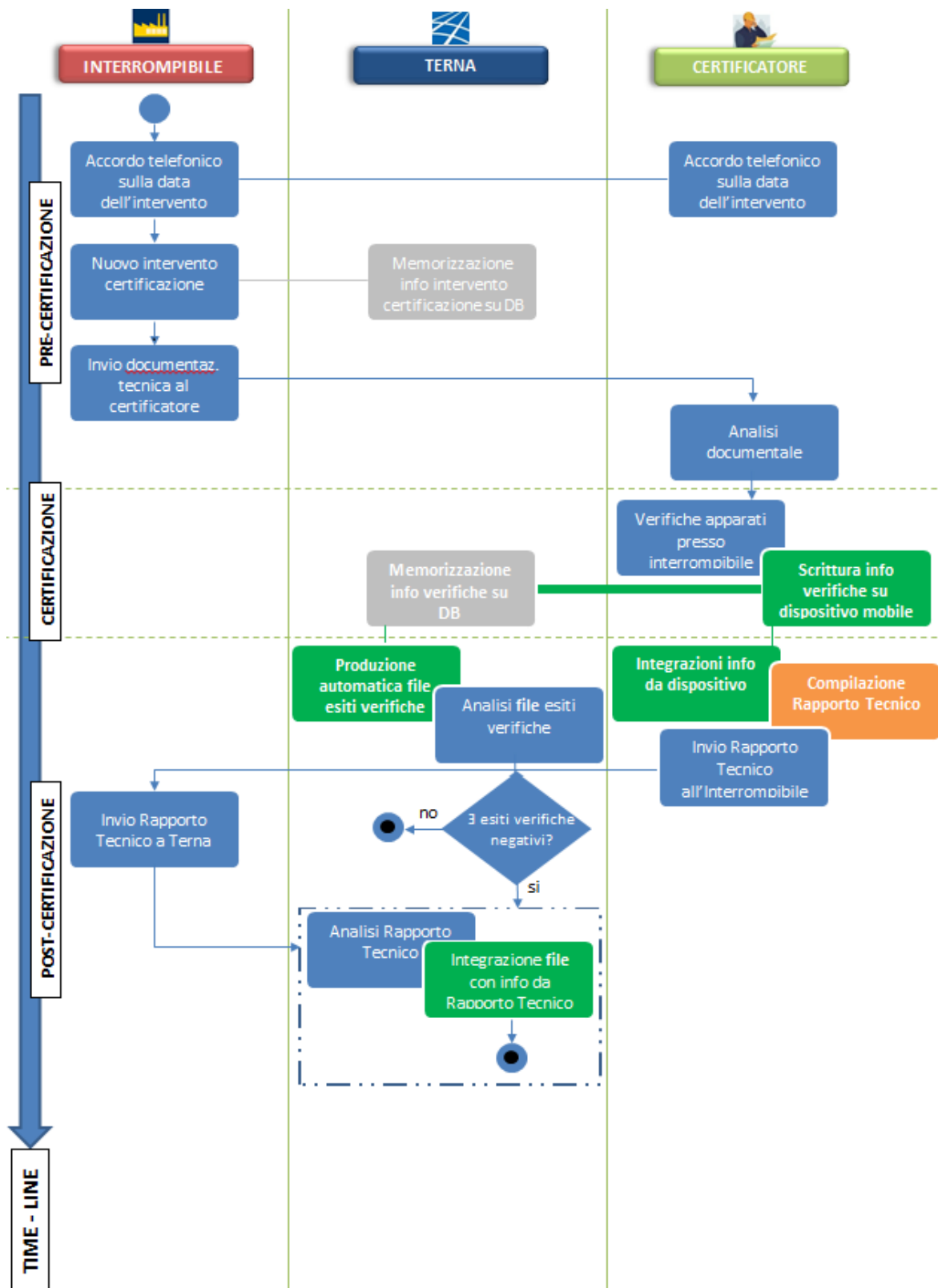


Figura 5.14 - Processo certificazione TO-BE - Diagramma di flusso a swimlane

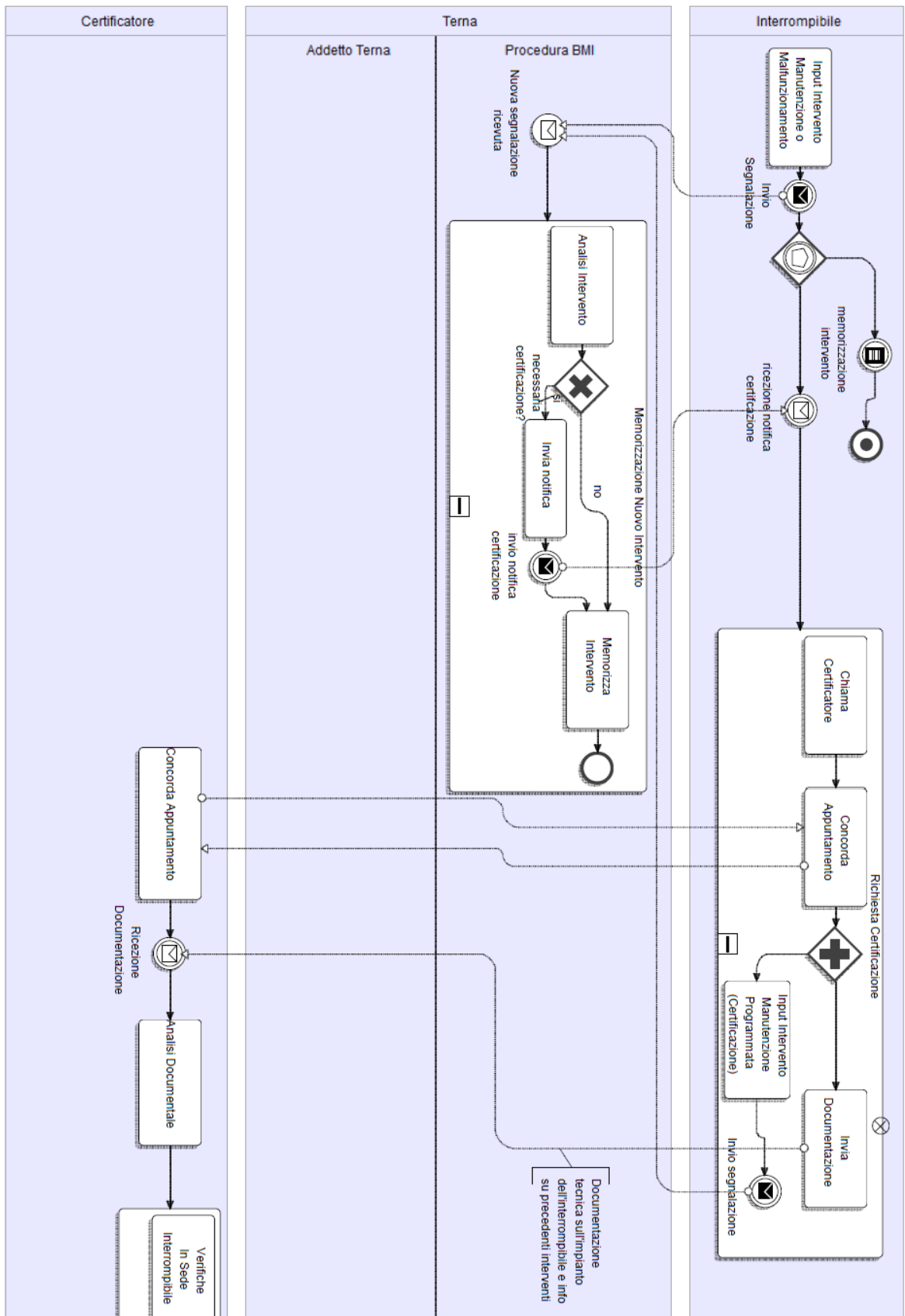


Figura 5.15 - Processo Certificazione BPMN TO-BE part1

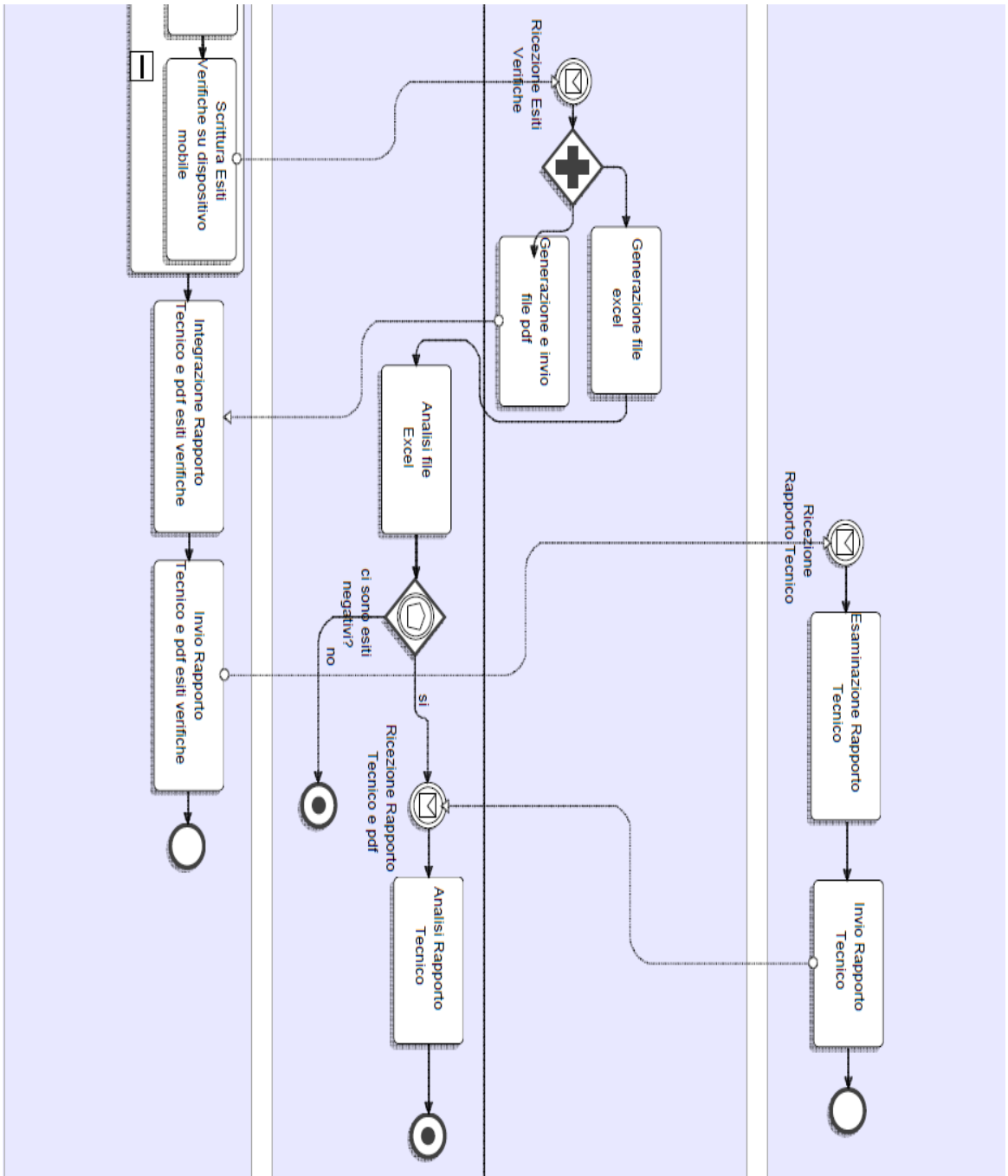


Figura 5.16 - Processo Certificazione BPMN TO-BE part2

6. PROGETTAZIONE DELLA SOLUZIONE

6.1 REQUISITI FUNZIONALI LATO CLIENT

Per quanto riguarda l'applicazione per dispositivo mobile, essa consentirà al certificatore di:

- visualizzare un'**agenda semplificata** degli interventi di certificazione, integrata con il calendario delle attività di manutenzione programmata gestito dagli strumenti informatici di Terna, all'interno della procedura BMI.
- gestire la **checklist** delle attività da svolgere in fase di certificazione: egli potrà segnare gli esiti di ciascun task di ogni attività di verifica, indicando se l'esito è stato positivo (OK) o negativo (KO). In tal modo potrà avere sempre sotto controllo il lavoro svolto e da svolgere.
- gestire un **set di attributi** associato a ciascun task di ogni attività di verifica relativo all'intervento di certificazione. Per ogni attributo di ciascun task, inoltre, vi sarà la possibilità di scrivere note che appariranno poi nello storico degli interventi effettuati ed aiuteranno il certificatore a ricordare i problemi incontrati e risolti o peculiarità di un certo task effettuato.
- inviare le informazioni memorizzate, inizialmente su dispositivo, a Terna.

Nel seguito si fornisce una descrizione dei requisiti funzionali individuati. L'utente di seguito menzionato è il certificatore. Si è ritenuto utile suddividere i requisiti fra quelli relativi al Client e quelli relativi al Server.

6.1.1 AUTENTICAZIONE CERTIFICATORE

Codice	AC1
Nome	Login
Descrizione	L'utente effettua il login all'applicazione autenticandosi con Username e Password.
Priorità	Media

Use Case associato	UC01: Login
---------------------------	-------------

Codice	AC2
Nome	Logout
Descrizione	L'utente effettua il logout chiudendo l'applicazione stessa.
Priorità	Media
Use Case associato	UC02: Logout

6.1.2 GESTIONE CALENDARIO

Codice	GC1
Nome	Visualizzazione calendario
Descrizione	L'utente visualizza un calendario mensile che rappresenta la sua agenda degli interventi di certificazione presso i clienti di Terna. Tale agenda è aggiornata a partire dai dati contenuti nei server Terna.
Priorità	Alta
Use Case associato	UC03: VisualizzaCalendario

Codice	GC2
Nome	Navigazione calendario
Descrizione	L'utente naviga il calendario su base mensile spostandosi in avanti o indietro per avere una visione per mese.
Priorità	Alta
Use Case associato	-

6.1.3 GESTIONE INTERVENTI

Codice	GI1
Nome	Visualizzazione checklist
Descrizione	L'utente clicca sull'intervento e visualizza la checklist, ovvero la lista delle attività inerenti all'intervento che deve essere effettuato presso un impianto di un determinato cliente interrompibile.
Priorità	Alta
Use Case associato	UC04: VisualizzaElencoAttività

Codice	GI2
Nome	Visualizzazione task
Descrizione	L'utente visualizza i task associati ad ogni attività della checklist.
Priorità	Alta
Use Case associato	UC05 - VisualizzaElencoTask

Codice	GI3
Nome	Aggiornamento dell'agenda
Descrizione	L'applicazione permette un aggiornamento dell'agenda dell'utente a seguito di esplicita richiesta dell'utente.
Priorità	Media
Use Case associato	UC06 - AggiornaAgenda

Codice	GI4
Nome	Aggiornamento della checklist
Descrizione	L'applicazione permette un aggiornamento della checklist degli

	interventi, quindi delle verifiche e dei task, a seguito di esplicita richiesta dell'utente.
Priorità	Media
Use Case associato	UC07 - AggiornaChecklist
Codice	GI5
Nome	Visualizzazione riepilogo dati inseriti
Descrizione	L'utente può visualizzare un riepilogo di tutti i dati inseriti fino al momento di invio della checklist compilata ai sistemi informativi di Terna
Priorità	Alta
Use Case associato	UC08 - VisualizzaRiepilogoDatiInseriti

Codice	GI6
Nome	Invio stato certificazione
Descrizione	L'utente può inviare a Terna lo stato delle certificazioni effettuate, ovvero le intere checklist degli interventi.
Priorità	Alta
Use Case associato	UC09 - InviaChecklist

6.1.4 SCRITTURA ESITI

Codice	SE1
Nome	Scrittura esito
Descrizione	L'utente può scrivere l'esito di un task effettuato e determinati valori degli attributi associati al task.
Priorità	Media

Use associato	Case	UC10 - InserisciDettagliTask
----------------------	-------------	------------------------------

Codice	SE2
Nome	Scrittura note
Descrizione	L'utente può scrivere delle note associate a ogni task o attività della checklist.
Priorità	Media
Use Case associato	UC11 - InserisciNota

Codice	SE3
Nome	Salvataggio attività di verifica
Descrizione	L'utente richiede all'applicazione il salvataggio dei dati e delle eventuali note inserite per l'attività di verifica che sta eseguendo. L'applicazione provvederà ad eseguirne il salvataggio nel database locale.
Priorità	Media
Use Case associato	UC12 - SalvaDatiInseriti

6.2 REQUISITI FUNZIONALI LATO SERVER

Lato server, il backend che si andrà a realizzare dovrà essere in grado di:

- Notificare al client eventuali aggiornamenti della procedura di certificazione. Tali aggiornamenti possono riguardare sia l'aggiunta di ulteriori attività da eseguire, sia l'aggiunta di ulteriori task associati alle attività già esistenti o appena inserite.

- Produrre automaticamente un file excel con gli esiti di ogni attività e di ogni task svolto durante l'intervento di certificazione, pronto per essere usato dai funzionari di Terna.
- Produrre automaticamente un file pdf contenente un riepilogo delle attività effettuate dal certificatore con i rispettivi risultati: tale file integrerà il rapporto cartaceo che il certificatore deve inviare a Terna in seguito alla terminazione dell'intervento.
- Inviare un'email al certificatore contenente in allegato il file pdf di cui sopra, in modo da consentire allo stesso l'integrazione con il rapporto tecnico.

6.2.1 RICEZIONE ESITI

Codice	RE1
Nome	Ricezione esiti dell'intervento di certificazione
Descrizione	Alla ricezione delle informazioni inerenti l'intervento di certificazione, viene creato un file MS Excel contenente il riepilogo di tali informazioni.
Priorità	Media
Use Case associato	UC13 - RicezioneEsiti

6.3 REQUISITI NON FUNZIONALI

Nel seguito si fornisce una descrizione dei requisiti non funzionali individuati.

- L'applicazione, tramite la funzionalità di login, deve garantire totale protezione da accessi non autorizzati.
- La funzionalità di logout deve garantire un logout sicuro mediante la distruzione dei dati di sessione, senza possibilità di rientrare nell'applicazione, se non avendo previamente effettuato di nuovo login.

- Le interfacce grafiche dovranno richiamare i colori tipici dell'azienda (Terna Rete Italia): blu, bianco e grigio.
- Il numero di drill down¹⁰ sarà limitato per garantire un utilizzo dell'applicazione semplice ed immediato.

Per quanto riguarda i requisiti non funzionali relativi alla sicurezza, in cui è inclusa l'autenticazione dell'utente, essi sono stati previsti ma non implementati. Dopo un approfondito confronto con i referenti aziendali sono venuti alla luce molteplici vincoli di policy-privacy che sono attualmente in fase di modifica e cambiamento. Da ciò si è ritenuto opportuno sospendere l'implementazione e, dunque, il successivo sviluppo di queste funzionalità. Queste ultime saranno implementate prima di effettuare il deploy effettivo dei web services sui sistemi Terna.

6.4 DIAGRAMMA DEI CASI D'USO

In UML, gli Use Case Diagram sono diagrammi dedicati alla descrizione delle funzioni o servizi offerti da un sistema, così come sono percepiti e utilizzati dagli attori che interagiscono col sistema stesso.

¹⁰ Per drill-down si intende il numero di livelli di navigazione in profondità nell'applicativo (Client).

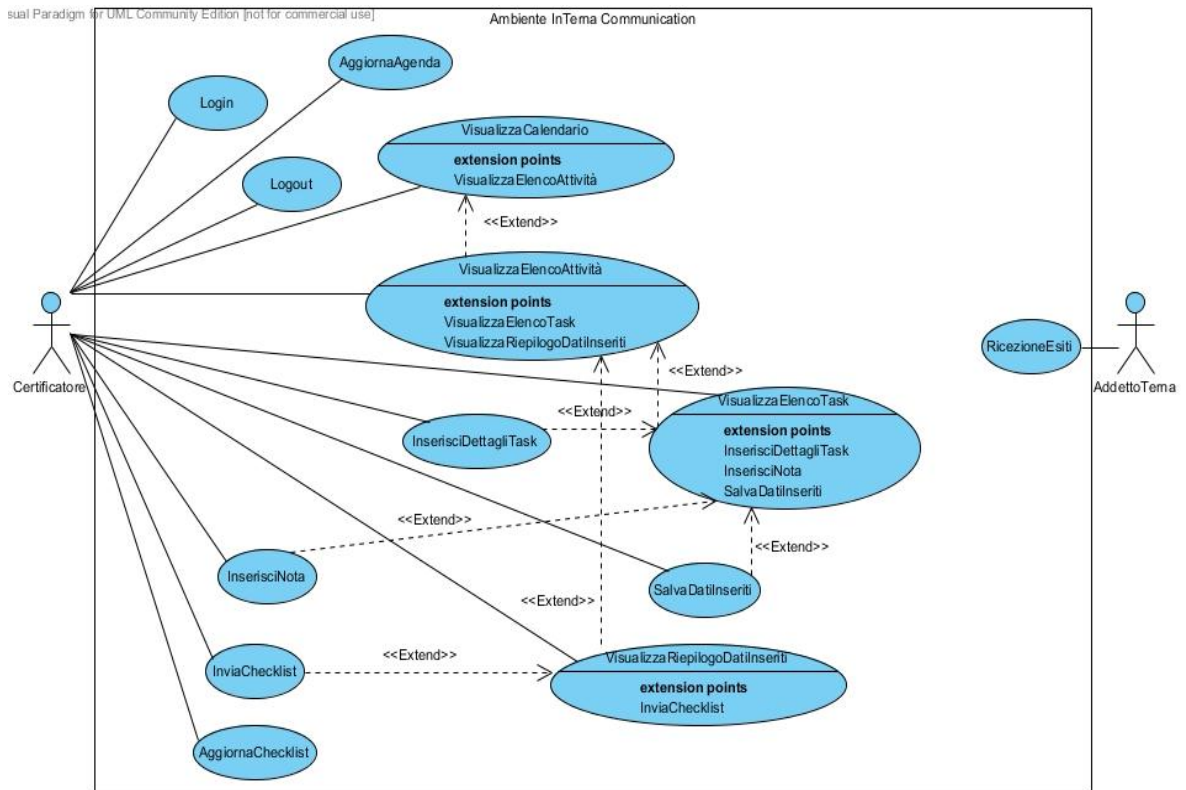


Figura 6.1 - Diagramma dei casi d'uso

6.5 CASI D'USO LATO CLIENT

Di seguito una breve descrizione dei casi d'uso individuati. Preliminarmente si forniscono alcune definizioni utili alla comprensione dei casi d'uso stessi:

- **Attività:** verifica da effettuare.
- L'elenco delle attività viene definito **checklist**.
- **Task:** sotto-attività appartenente a una specifica attività.

Caso d'uso: Login	
ID:	UC01
Descrizione	Permette il riconoscimento dell'utente tramite l'invio delle credenziali d'accesso
Precondizioni	1. Nessuna

<p>Sequenza eventi</p> <ol style="list-style-type: none"> 1. L'utente inserisce <i>username</i> e <i>password</i> personali e clicca il bottone di "Login" 2. Finché la combinazione <i>username/password</i> è inesistente: <ol style="list-style-type: none"> 2.1. l'applicazione notifica l'errore all'utente
<p>Postcondizioni</p> <p>L'utente risulta loggato e potrà quindi inviare la checklist delle attività svolte e/o richiedere l'aggiornamento della checklist stessa o dell'agenda.</p>

Caso d'uso: Logout
ID: UC02
<p>Descrizione</p> <p>Permette di disconnettere l'utente dall'applicazione e distruggere contestualmente i dati di sessione</p>
<p>Precondizioni</p> <ol style="list-style-type: none"> 1. L'utente si è precedentemente autenticato (<i>Login</i>)
<p>Sequenza eventi</p> <ol style="list-style-type: none"> 1. Quando l'utente chiude l'applicazione il Logout avviene automaticamente 2. Vengono distrutti i dati di sessione
<p>Postcondizioni</p> <ol style="list-style-type: none"> 1. L'utente non risulta più loggato.

Caso d'uso: VisualizzaCalendario
ID: UC03
<p>Descrizione</p> <p>L'applicazione recupera i dati relativi agli appuntamenti dal database locale e mostra all'utente il Calendario con il mese corrente in cui i giorni relativi agli appuntamenti sono evidenziati di un colore differente.</p>
<p>Precondizioni</p> <ol style="list-style-type: none"> 1. Nessuna
<p>Sequenza eventi</p>

1. L'applicazione mostra il calendario del mese con i giorni relativi agli appuntamenti evidenziati di un colore differente rispetto agli altri
Postcondizioni
Nessuna

Caso d'uso: VisualizzaElencoAttività
ID: UC04
Descrizione L'utente può visualizzare l'elenco delle attività da effettuare
Precondizioni 1. L'applicazione mostra il calendario con gli appuntamenti da effettuare in determinati giorni.
Sequenza eventi 1. L'utente seleziona un appuntamento tra quelli mostrati 2. L'applicazione mostra la lista delle attività che il certificatore dovrà eseguire per quel determinato impianto
Postcondizioni Nessuna

Caso d'uso: VisualizzaElencoTask
ID: UC05
Descrizione L'utente può visualizzare l'elenco dei task relativi ad una particolare attività
Precondizioni 1. L'applicazione mostra la checklist delle attività
Sequenza eventi 1. L'utente seleziona una particolare attività della checklist 2. L'applicazione mostra l'elenco dei task per quella specifica attività e, per ogni task, ne mostra gli attributi che dovranno essere compilati dal certificatore.

<p>Postcondizioni</p> <p>Nessuna</p>

Caso d'uso: AggiornaAgenda
ID: UC06
<p>Descrizione</p> <p>L'utente può effettuare un aggiornamento degli appuntamenti in agenda, ricevendo le informazioni sui nuovi interventi inseriti nei sistemi terna</p>
<p>Precondizioni</p> <p>1. L'utente si è precedentemente autenticato (<i>Login</i>)</p>
<p>Sequenza eventi</p> <ol style="list-style-type: none"> 1. L'utente seleziona il pulsante "Aggiorna Agenda" 2. L'applicazione richiede al back-end Terna una lista aggiornata degli interventi 3. L'applicazione memorizza i nuovi interventi su database locale 4. L'applicazione mostra l'agenda aggiornata con i nuovi interventi
<p>Postcondizioni</p> <p>1. Le informazioni relative ai nuovi interventi vengono salvati in maniera persistente sul database locale e l'agenda risulta aggiornata</p>

Caso d'uso: AggiornaChecklist
ID: UC07
<p>Descrizione</p> <p>L'utente può effettuare un aggiornamento della checklist. Nel caso in cui vengano aggiunti e/o modificati verifiche o task sui back-end di Terna, la checklist su dispositivo deve risultare aggiornata e consistente con le modifiche apportate.</p>
<p>Precondizioni</p> <p>1. L'utente si è precedentemente autenticato (<i>Login</i>)</p>
<p>Sequenza eventi</p> <ol style="list-style-type: none"> 1. L'utente seleziona il pulsante "Aggiorna Checklist" 2. L'applicazione richiede al back-end Terna la checklist aggiornata

<p>3. Se riceve una checklist aggiornata</p> <p>3.1. La checklist aggiornata viene memorizzata su database locale (inseriti opportuni record nelle tabelle)</p>
<p>Postcondizioni</p> <p>1. La checklist su dispositivo risulta aggiornata e il database locale risulta consistente con il database presente sui sistemi informativi di Terna</p>

Caso d'uso: InserisciDettagliTask
ID: UC07
<p>Descrizione</p> <p>L'utente può inserire i valori associati a ciascun task da eseguire.</p>
<p>Precondizioni</p> <p>1. L'applicazione mostra l'elenco dei task associati all'attività</p>
<p>Sequenza eventi</p> <p>1. L'utente inserisce un particolare dettaglio di un task, che può essere un esito, un utenza/carico, una proprietà.</p> <p>2. L'applicazione contestualmente mostra il dettaglio o i dettagli del task appena inseriti</p>
<p>Postcondizioni</p> <p>Nessuna</p>

Caso d'uso: InserisciNota
ID: UC08
<p>Descrizione</p> <p>L'utente può inserire un'annotazione ogni task/attributo dell'attività selezionata.</p>
<p>Precondizioni</p> <p>1. L'utente visualizza la lista dei task di una particolare attività</p>
<p>Sequenza eventi</p> <p>1. L'utente inserisce il testo della nota nella relativa casella</p>
<p>Postcondizioni</p>

Nessuna

Caso d'uso: SalvaDatiInseriti

ID: UC09

Descrizione

L'utente può inserire salvare i dati inseriti fino a quel momento per poterli visualizzare anche in un secondo momento

Precondizioni

1. L'utente visualizza la lista dei task di una particolare attività

Sequenza eventi

1. L'utente seleziona il bottone "Salva"

Postcondizioni

1. I dati relativi ad una particolare attività vengono salvati in maniera persistente

Caso d'uso: VisualizzaRiepilogoDatiInseriti
--

ID: UC10

Descrizione

L'utente può visualizzare i dettagli relativi ad ogni verifica della checklist (note ed informazioni varie) prima di inviarla definitivamente a Terna

Precondizioni

1. L'utente visualizza l'elenco delle verifiche da svolgere per un determinato intervento

Sequenza eventi

1. L'utente seleziona "Visualizza riepilogo dei dati inseriti"
2. L'applicazione mostra tutti i dati inseriti fino a quel momento

Postcondizioni

Nessuna

Caso d'uso: InviaChecklist
ID: UC11
<p>Descrizione</p> <p>L'utente può inviare l'intera checklist relativa all'intervento di certificazione che ha eseguito</p>
<p>Precondizioni</p> <ol style="list-style-type: none"> 1. L'applicazione visualizza il riepilogo dei dati inseriti fino a quel momento 2. L'utente si è precedentemente autenticato (<i>Login</i>)
<p>Sequenza eventi</p> <ol style="list-style-type: none"> 1. L'utente seleziona il bottone "Invia checklist" 2. L'applicazione richiede di inserire indirizzo e-mail dell'utente. 3. Se l'invio delle informazioni va a buon fine: <ol style="list-style-type: none"> 3.1. L'applicazione notifica all'utente la riuscita dell'operazione 4. Altrimenti: <ol style="list-style-type: none"> 4.1. L'applicazione notifica all'utente il fallimento dell'operazione
<p>Postcondizioni</p> <p>Nessuna</p>

6.6 CASI D'USO LATO SERVER

Caso d'uso: RicezioneEsiti
ID: UC12
<p>Descrizione</p> <p>Permette all'addetto Terna di ottenere la creazione del file Excel relativo alla certificazione di interesse.</p>
<p>Precondizioni</p> <p>La certificazione di interesse deve essere già avvenuta. L'opzione non è disponibile per certificazioni in programma ma non ancora eseguite.</p>
Sequenza eventi

- | |
|--|
| <ol style="list-style-type: none">1. Una procedura remota produce il file Excel2. L'addetto Terna accede al sistema informativo3. L'addetto Terna visualizza file Excel per l'intervento di interesse. |
|--|

Postcondizioni

Nessuna

6.7 DIAGRAMMI DI SEQUENZA

I diagrammi di sequenza permettono di mostrare l'interazione organizzata di oggetti, evidenziandone la sequenza temporale. In particolare consente di mostrare lo scambio di messaggi che avviene tra gli elementi che prendono parte ad un'interazione. [8]

Qui di seguito sono riportati i diagrammi di sequenza relativi ai principali casi d'uso esaminati.

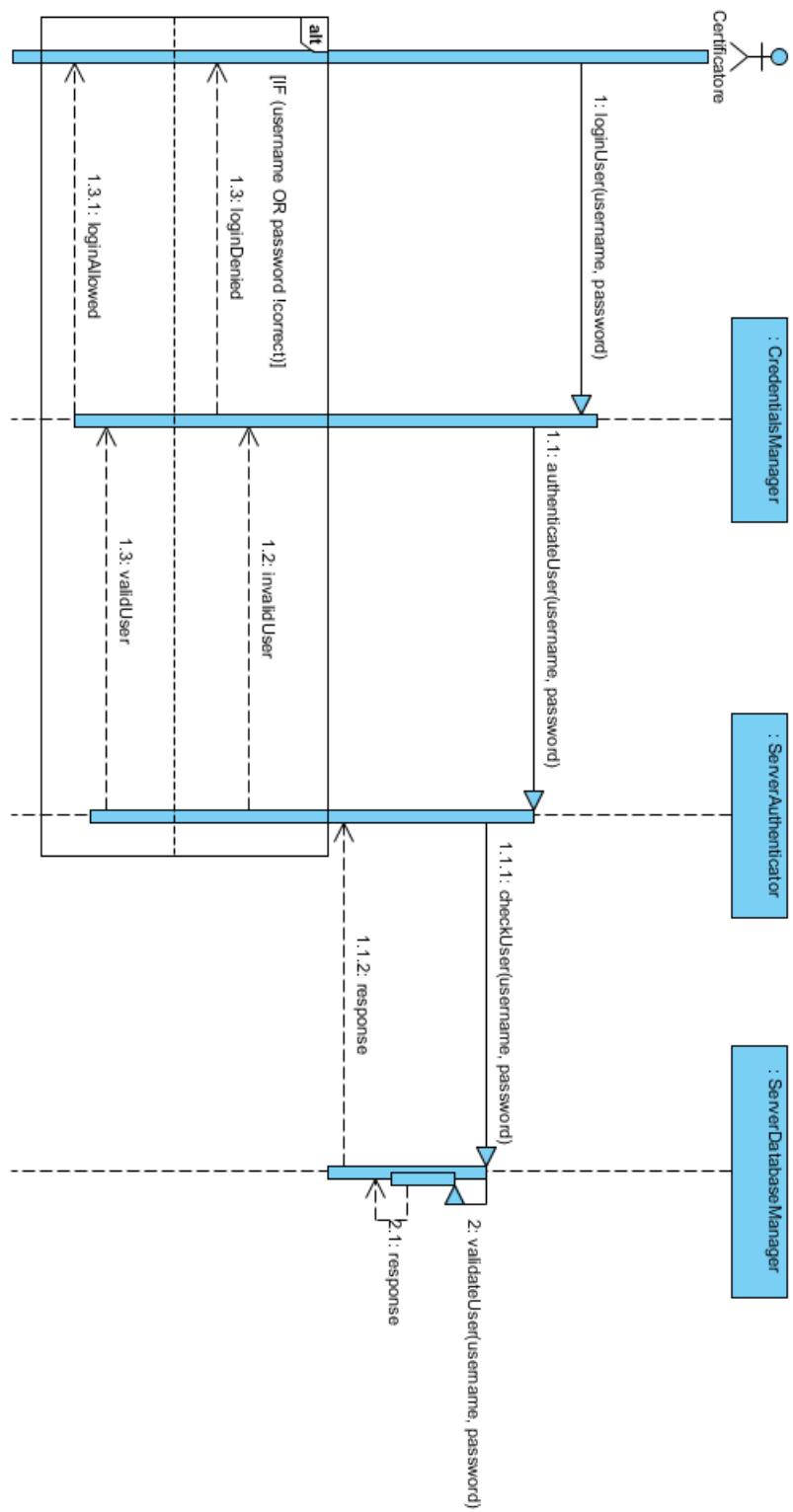


Figura 6.2 - Diagramma di Sequenza: Login

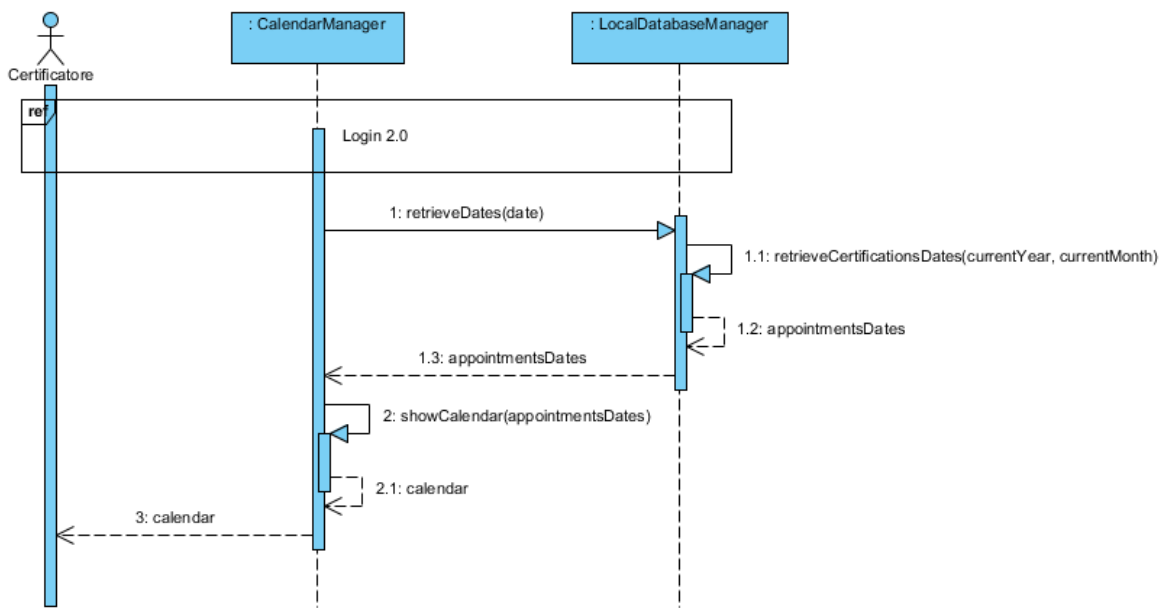


Figura 6.3 - Diagramma di Sequenza VisualizzaCalendario

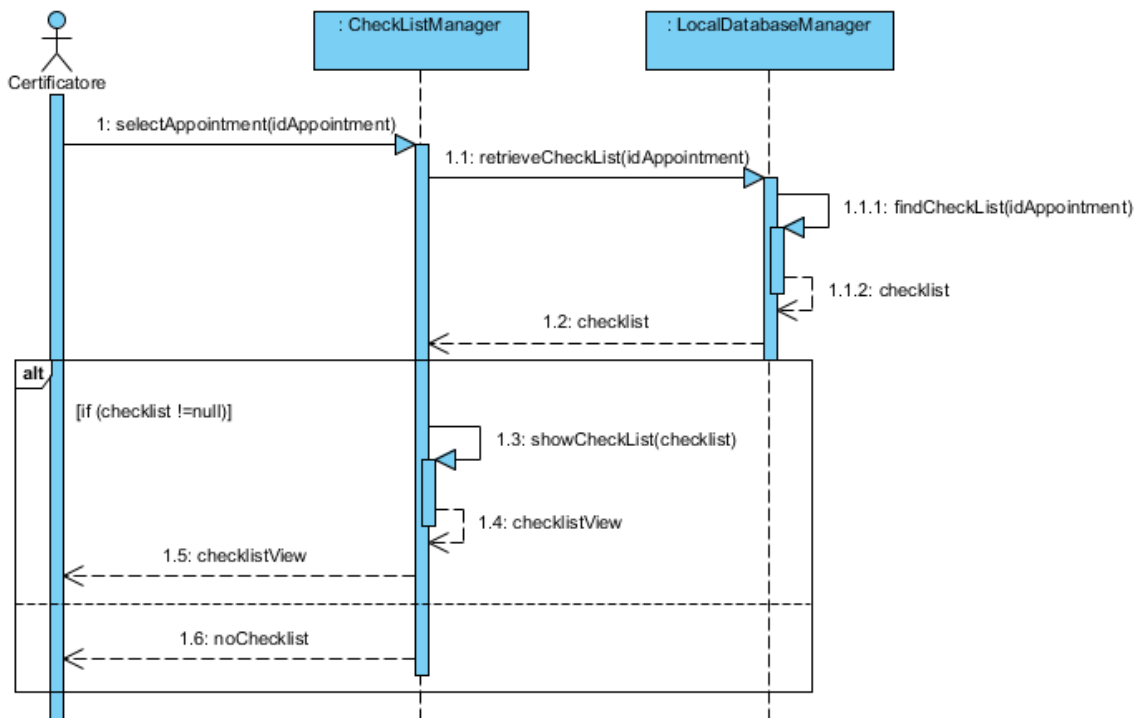


Figura 6.4 - Diagramma di Sequenza VisualizzaElencoAttività

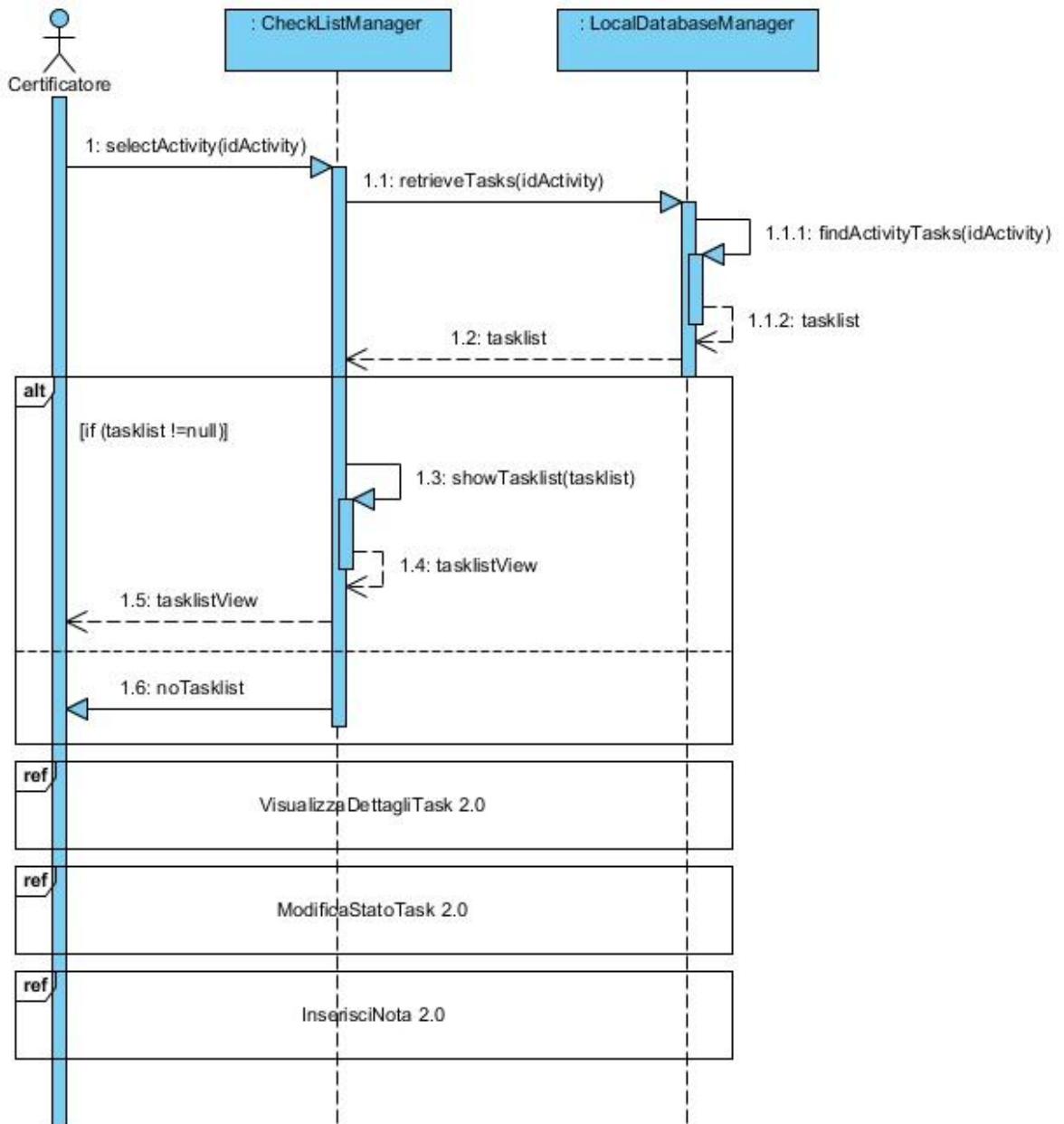


Figura 6.5 - Diagramma di sequenza VisualizzaElencoTask

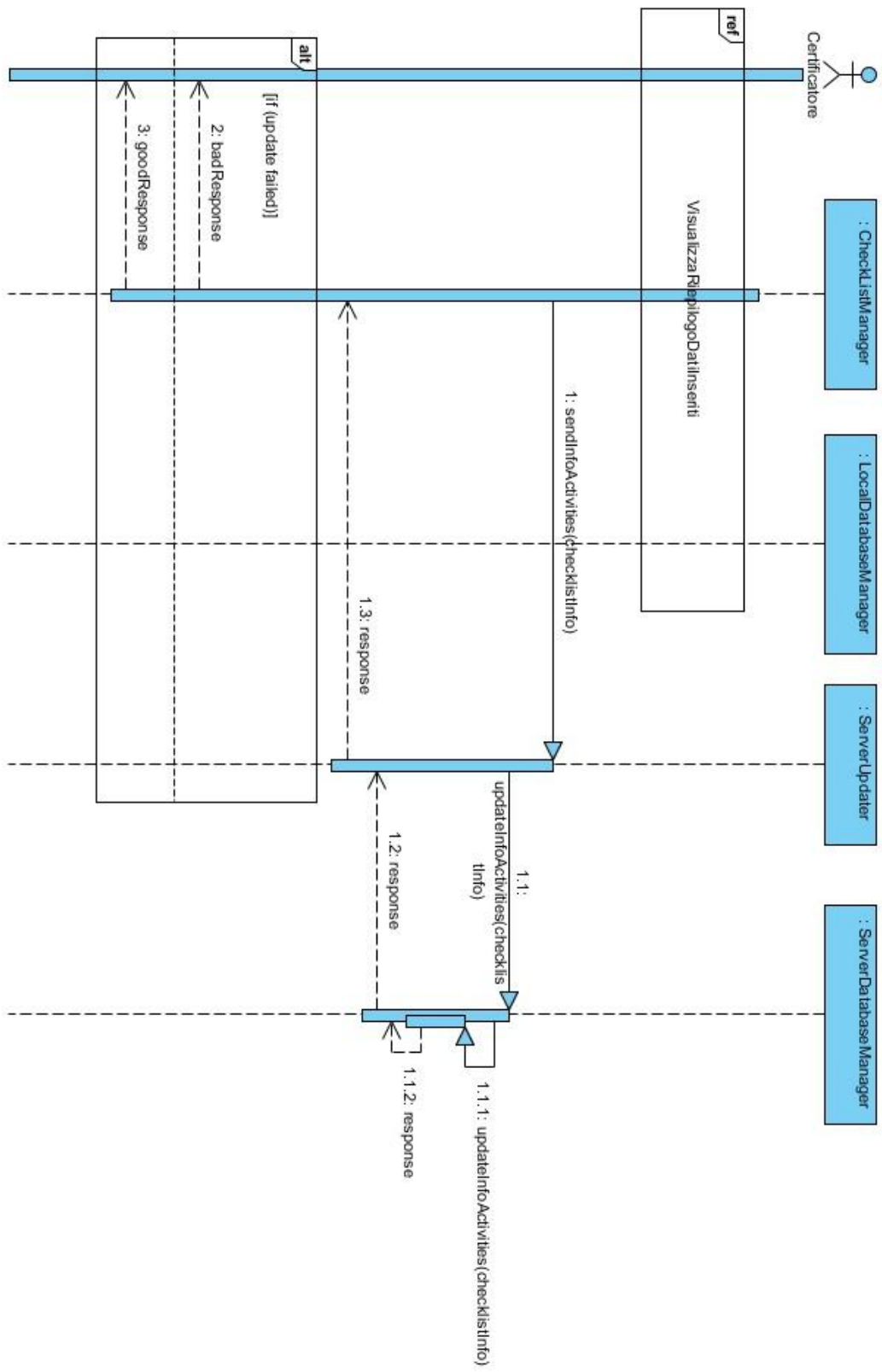


Figura 6.6 - Diagramma di Sequenza InviaChecklist

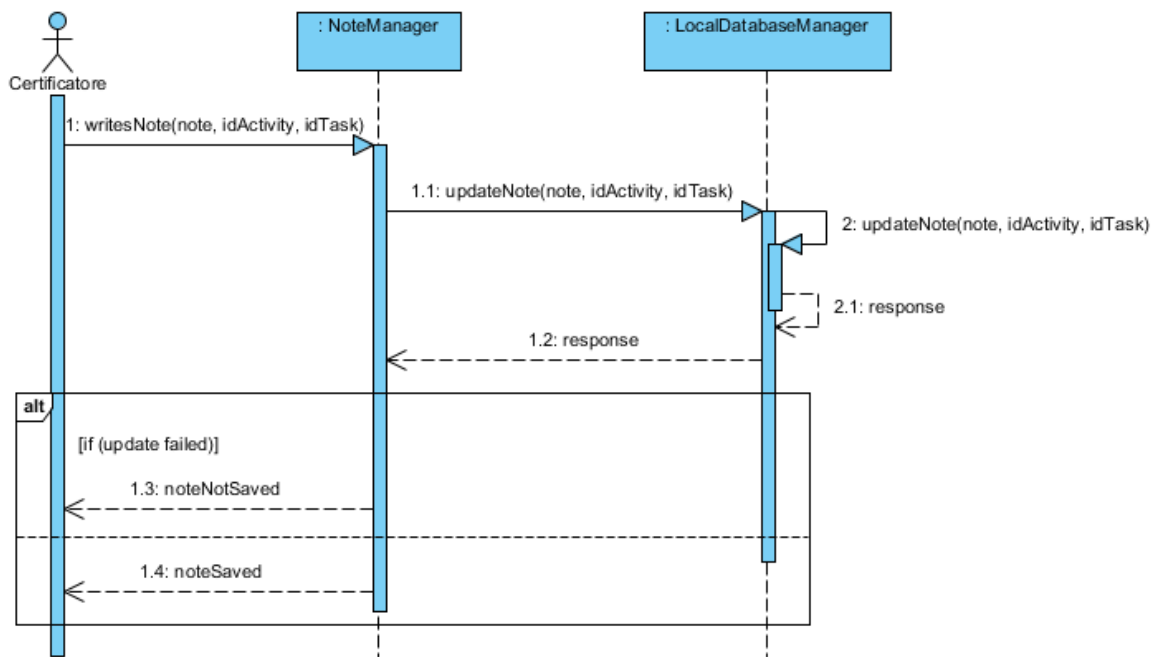


Figura 6.7 - Diagramma di sequenza InserisciNota

In Figura 6.8 - Diagramma Classi lato Client” è possibile visualizzare il diagramma delle classi lato client.

Quasi specularmente, in Figura 6.9 - Diagramma Classi lato Server” è riportato il diagramma delle classi lato server.

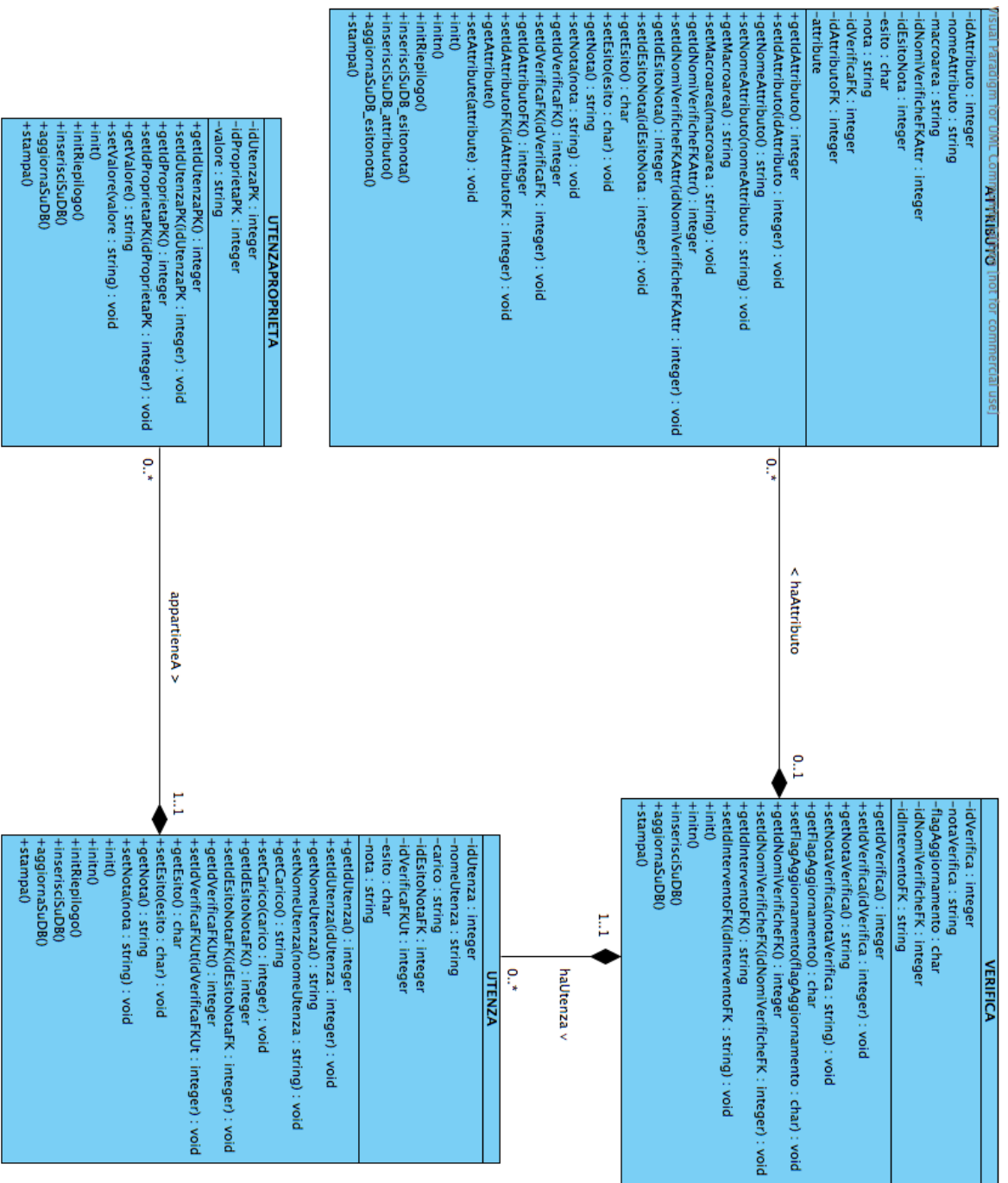


Figura 6.8 - Diagramma Classi lato Client

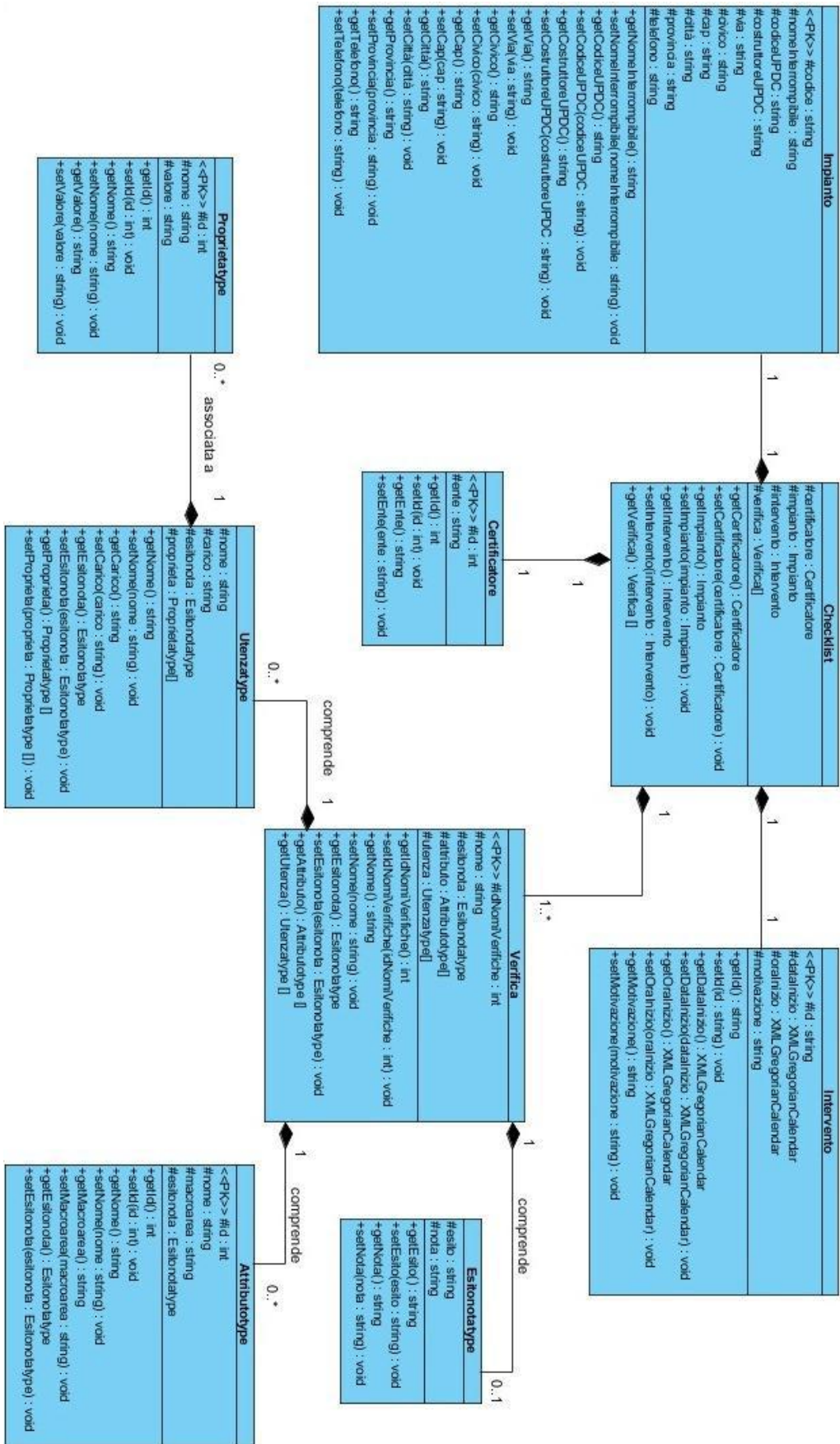


Figura 6.9 - Diagramma Classi lato Server

6.8 PROGETTAZIONE DEL DATABASE

6.8.1 GLOSSARIO DEL DATABASE

Per poter comprendere le scelte progettuali relative al database, viene riportato di seguito il glossario dei termini utilizzati.

Nome	Entità/Relazione	Descrizione
IMPIANTO	ENTITA'	Identifica l'impianto di proprietà del cliente interrompibile presso il quale il certificatore svolge l'intervento di certificazione. Il suo identificatore principale è il codice dell'impianto stesso così come presente sul sistema BMI di Terna Rete Italia.
INTERVENTO	ENTITA'	Identifica l'intervento di certificazione che il certificatore deve sostenere. Il suo identificatore principale è l'id dell'intervento stesso così come presente sul sistema BMI di Terna Rete Italia.
CERTIFICATORE	ENTITA'	Identifica l'individuo che svolge l'intervento di certificazione. Il suo identificatore principale è l'id del certificatore stesso così come riportato sul sistema BMI di Terna Rete Italia.
NOMIVERIFICHE	ENTITA'	Identifica tutte le attività di verifica che vengono svolte dal certificatore in sede di certificazione dell'UPDC. Attualmente le attività di verifica che devono essere sostenute sono dieci e già esplicitate da Terna Rete Italia.
VERIFICA	ENTITA'	Identifica la verifica che viene svolta dal certificatore associata ad ogni attività di verifica esplicitata da Terna Rete Italia. Ad esempio ogni attività di verifica "7.1 Verifica documentale" avrà una verifica associata che può

		essere modificata solo dal certificatore e non da Terna Rete Italia. L'identificatore principale è un id incrementale assegnato dal DBMS al momento dell'inserimento del record sulla base di dati.
ESITONOTA	ENTITA'	Identifica l'esito e la nota associata ad un attributo o ad una utenza. La necessità di tenere l'esito e l'eventuale nota separati dall'attributo stesso o dall'utenza nasce dalla natura stessa di Attributo. Infatti, tale entità può essere modificata solo da Terna Rete Italia quindi il certificatore non può scrivere all'interno di tale tabella. L'identificatore principale è un id incrementale assegnato dal DBMS al momento dell'inserimento del record sulla base di dati.
ATTRIBUTO	ENTITA'	Identifica gli attributi associati ad una determinata attività di verifica. Essi sono precaricati sulla base di dati da Terna Rete Italia. L'identificatore principale è un id incrementale assegnato dal DBMS al momento dell'inserimento del record sulla base di dati.
UTENZA	ENTITA'	Identifica le utenze associate ad una determinata verifica. Esse vengono aggiunte dal certificatore in base all'intervento di certificazione che sta eseguendo. Non sono mai precaricate né note in anticipo. L'identificatore principale è un id incrementale assegnato dal DBMS al momento dell'inserimento del record sulla base

		di dati.
PROPRIETA	ENTITA'	Identifica le proprietà associate ad una determinata attività di verifica e ad una determinata utenza. Esse sono precaricate sulla base di dati da Terna Rete Italia e sono quindi note in anticipo. Tutte le proprietà afferenti ad una certa attività di verifica sono esplicitate per tutte le utenze che il certificatore andrà ad inserire relativamente all'attività di verifica in questione. L'identificatore principale è un id incrementale assegnato dal DBMS al momento dell'inserimento del record sulla base di dati.
effettuatoIn	RELAZIONE	Questa relazione lega l'entità INTERVENTO a IMPIANTO e rappresenta l'impianto presso il quale il certificatore eseguirà l'intervento in questione.
associato	RELAZIONE	Questa relazione lega l'entità CERTIFICATORE a INTERVENTO e rappresenta quale dei tanti certificatori approvati da Terna Rete Italia andrà ad eseguire l'intervento.
comprende	RELAZIONE	Questa relazione lega l'entità INTERVENTO a VERIFICA e rappresenta tutte le verifiche eseguite dal certificatore (e da esso aggiunte e/o modificate) associate ad un dato

		intervento.
riferisceAtt	RELAZIONE	Questa relazione lega l'entità ATTRIBUTO a NOMIVERIFICHE ed indica quali sono gli attributi associati ad una determinata attività di verifica.
riferisce	RELAZIONE	Questa relazione lega l'entità VERIFICA a NOMIVERIFICHE ed indica a quale attività di verifica si riferisce la verifica che viene inserita dal certificatore.
comprendeEN	RELAZIONE	Questa relazione lega l'entità VERIFICA a ESITONOTA ed indica a quale verifica afferiscono l'esito e l'eventuale nota inseriti dal certificatore.
appartiene	RELAZIONE	Questa relazione lega l'entità UTENZA a VERIFICA ed indica a quale verifica si riferisce una determinata utenza. È stata inserita per facilitare il recupero delle informazioni sulle utenze durante la creazione del file xml lato server.
haAttributo	RELAZIONE	Questa relazione lega l'entità ESITONOTA ad ATTRIBUTO e rappresenta l'esito e l'eventuale nota afferente ad un attributo di una determinata verifica.
haUtenza	RELAZIONE	Questa relazione lega l'entità ESITONOTA ad UTENZA e rappresenta l'esito e l'eventuale nota afferente ad una utenza di una determinata verifica.
utenzaProprieta	RELAZIONE	Questa relazione lega l'entità UTENZA

		a PROPRIETA e rappresenta il valore che il certificatore assegna ad una proprietà associata ad una tale utenza.
NomiVerificheProprieta	RELAZIONE	Questa relazione lega l'entità NOMIVERIFICHE a PROPRIETA ed individua tutte le proprietà possedute da una determinata attività di verifica. Per esempio, nel caso della verifica 7.2.7, le proprietà associate sono due: "Tipo di interrompibilità" e "Timetag". Questa conoscenza non è nota a priori, viene bensì ricavata dinamicamente osservando tale relazione.

Per soddisfare il bisogno di Terna Rete Italia nel tenere traccia delle attività di certificazione svolte dal certificatore è stato deciso di progettare il database in base a due requisiti basilari:

- persistenza dei dati;
- velocità di salvataggio e di utilizzo dei dati;

e su alcuni requisiti specifici richiesti dal cliente:

- realizzazione di un database estensibile che offra la possibilità di aggiungere nuove attività di verifica all'elenco di attività che devono essere svolte durante l'iter di certificazione;
- realizzazione di un database estensibile che offra la possibilità di aggiungere nuovi attributi ad una determinata attività di verifica già presente nel database;
- realizzazione di un database estensibile che offra la possibilità di aggiungere nuove proprietà ad una determinata attività di verifica già presente nel database.

Il soddisfacimento dei primi due requisiti fondamentali è assicurata dalla definizione stessa delle basi di dati, mentre il soddisfacimento degli ultimi tre requisiti specifici richiesti dal cliente non è stato immediato. E' stato necessario individuare una soluzione ad-hoc, modellizzando opportunamente uno schema di database.

La soluzione proposta consiste nel vedere sia le attività di verifica che gli attributi e le proprietà come record di tabelle già esistenti. Così facendo l'aggiornamento della struttura si riduce ad un semplice inserimento di record all'interno della base di dati. Azione molto semplice se consideriamo che, per sua definizione, un database offre l'esecuzione di operazioni CRUD (Create, Read, Update, Delete).

Per realizzare gli inserimenti nel database lato server si è deciso di optare per la realizzazione di **stored procedures** che hanno consentito di inglobare la logica di inserimento nel database piuttosto che nel codice Java del web service, cosa che fornisce un livello di sicurezza maggiore. Per quanto riguarda il client, dal momento che SQLite non permette di immagazzinare stored procedures, il codice SQL è inglobato esplicitamente nei file .js.

Un'ultima osservazione riguarda il tipo di dati immagazzinati in entrambi i database: essendo SQLite un DBMS che offre solo tipi di dati base, sono stati utilizzati solamente i tipi *varchar*, *char* e *integer* (quest'ultimo sia per rappresentare i numeri interi, sia per rappresentare i booleani) oltre a *date* e *time* per rappresentare ora e data dell'intervento.

In "Figura 6.10 - Struttura fisica del database lato client" è riportata la struttura fisica del database progettato, lato client.

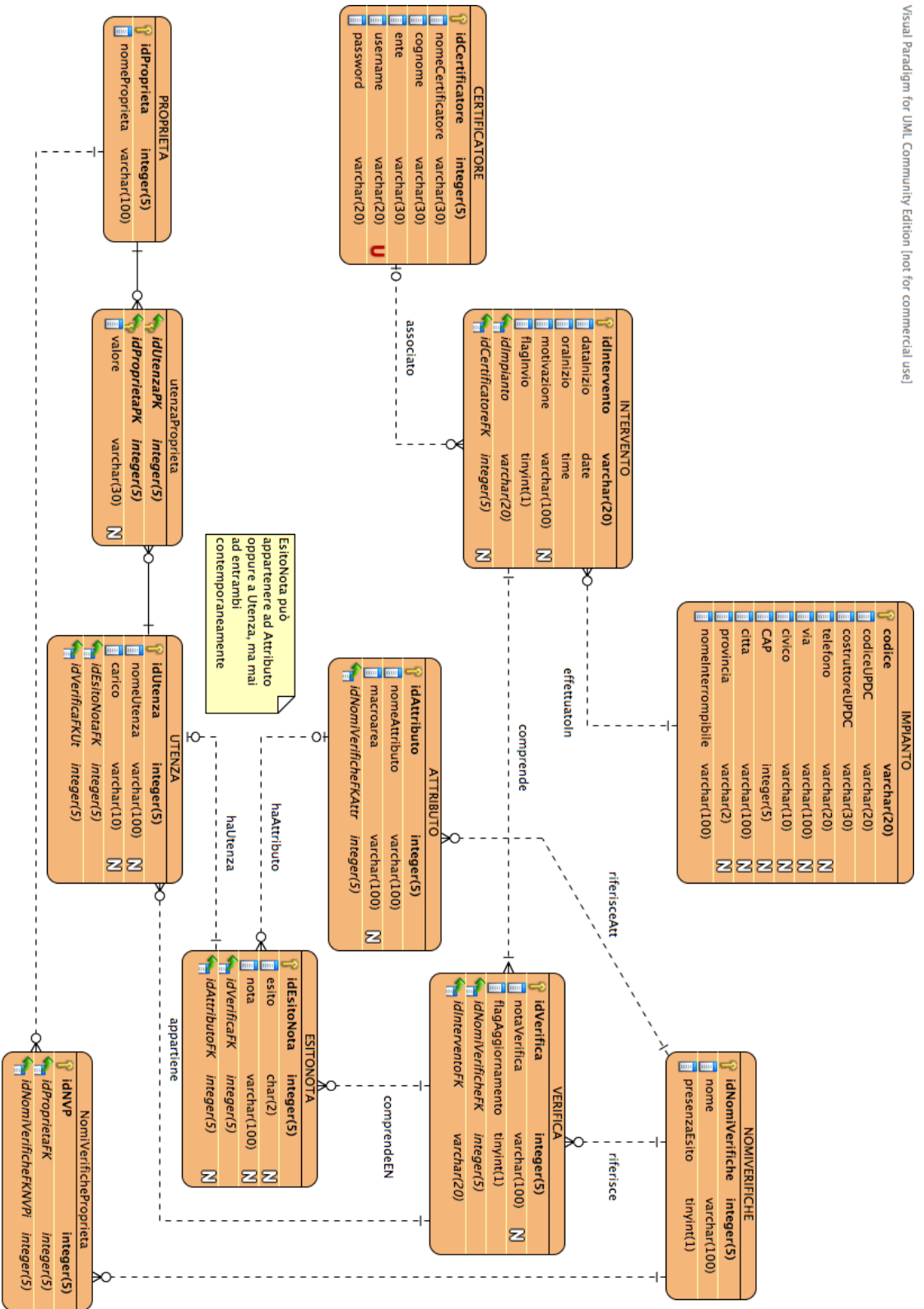


Figura 6.10 - Struttura fisica del database lato client

6.8.2 ATTRIBUTI, VERIFICHE E PROPRIETA'

Come già specificato, all'interno della base di dati progettata vi è la necessità di garantire l'aggiunta di attributi, verifiche e proprietà a discrezione del cliente.

Per assicurare ciò, si è deciso di utilizzare tre tabelle dedicate ognuna ad una voce e renderle modificabili solo dal cliente e mai dall'utilizzatore finale dell'applicazione. Questo garantisce integrità dei dati e pieno controllo degli stessi solo al cliente che può così modificarli in totale sicurezza.

La necessità di tenere separati in due tabelle diverse gli attributi e le proprietà nasce dallo studio delle relazioni a cui appartengono. Osservando la Figura 6.10 - Struttura fisica del database lato client è possibile notare che Attributo si relaziona in modo diretto con NomiVerifiche e si relaziona con Verifica attraverso la partecipazione alla relazione con EsitoNota.

Proprietà si relaziona in modo altrettanto diretto con NomiVerifiche ma necessita di partecipare a svariate relazioni prima di relazionarsi con Verifica. In particolare Proprietà necessita di relazionarsi sia con Utenza che con EsitoNota. Da qui nasce la decisione di tenere distinte le due entità che, chiaramente, non possono essere modellate usandone una sola in quanto partecipano a relazioni diverse.

6.8.3 PROGETTAZIONE CONCETTUALE

L'obiettivo della progettazione concettuale è di rappresentare i dati della realtà di interesse mediante un modello formale, ad alto livello ed indipendente dal DBMS. Il suo input è la descrizione della realtà di interesse (raccolta dei requisiti), mentre il suo output è lo schema concettuale.

Una buona progettazione concettuale rispetta i requisiti di qualità se è corretta e completa nel rappresentare la realtà di interesse.

A seguito di un'esauriente raccolta di requisiti, lo schema concettuale risultante, rappresentato dal Modello Entità – Relazione, è riportato di seguito, in Figura 6.11 - Schema concettuale E-R".

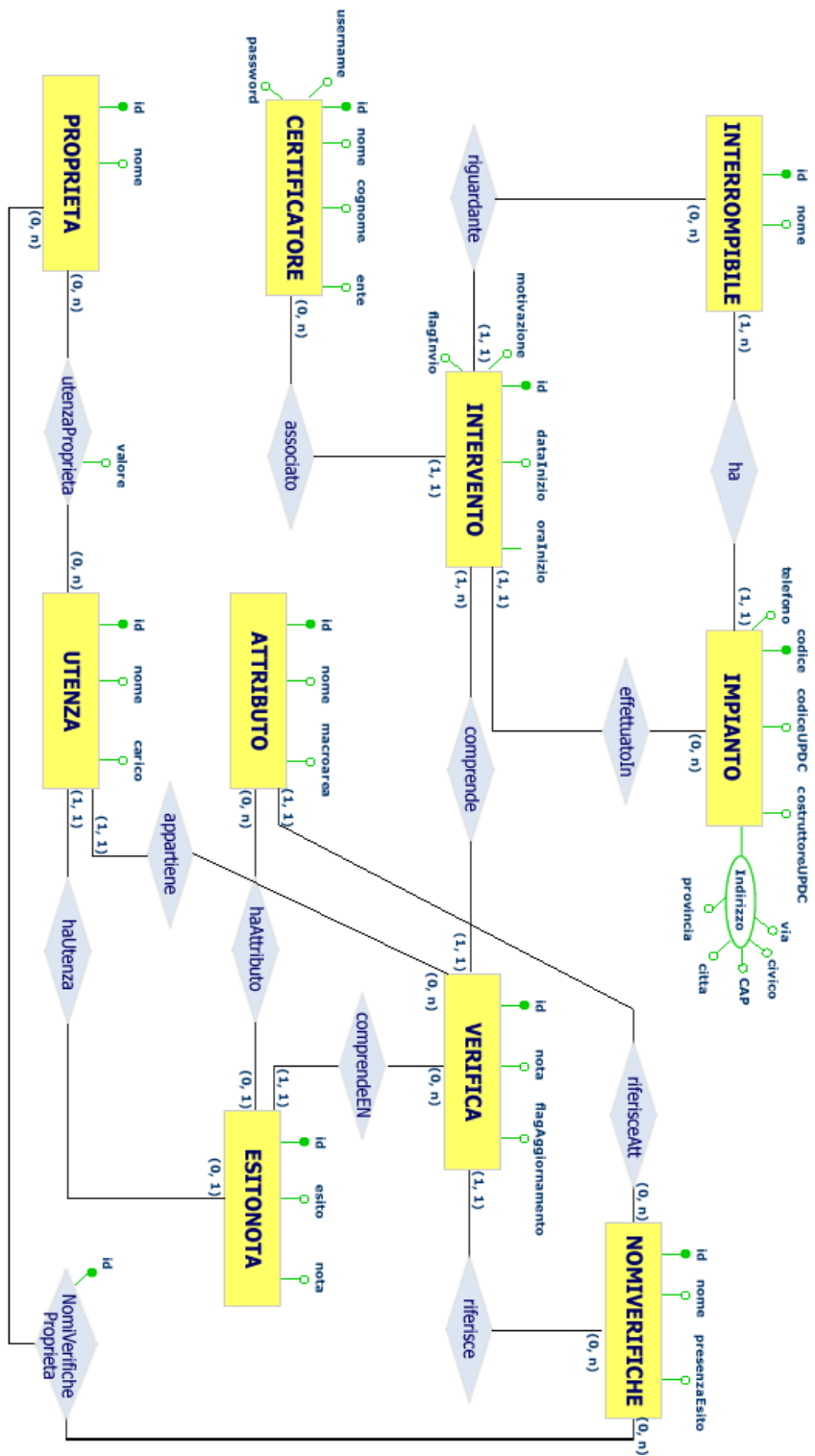


Figura 6.11 - Schema concettuale E-R

In seguito alla stesura del Modello ER è stata effettuata una ristrutturazione dello stesso per poter rappresentare la realtà meno ad alto livello ma, al contempo, più predisposta

alla progettazione logica. L'obiettivo della ristrutturazione è proprio quello di semplificare la successiva fase di traduzione nel modello relazionale (modello logico) eliminando i costrutti non facilmente traducibili.

Le attività di ristrutturazione del modello ER sono:

1. Analisi di eventuali ridondanze
2. Eliminazione degli attributi multivalore
3. Eliminazione degli attributi composti
4. Eliminazione delle ISA e delle generalizzazioni
5. Scelta degli identificatori principali di entità e relazioni
6. Specifica degli ulteriori vincoli esterni

Lo schema definito non necessita di particolari ristrutturazioni, a parte l'eliminazione di un attributo composto (indirizzo). Esso ha cardinalità (1, 1) pertanto passerà dall'essere un attributo composto all'essere diviso nei 5 attributi che lo compongono, direttamente associati all'entità Impianto:



Figura 6.12 - Attributo composto "Indirizzo"

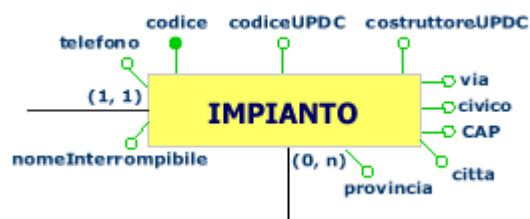


Figura 6.13 - Attributi associati all'Entità "Impianto"

6.8.4 PROGETTAZIONE FISICA

La progettazione fisica è parte della fase complessiva di realizzazione della base di dati e si deve pertanto coordinare con tali aspetti. Essa è atta a rappresentare i dati della realtà di interesse attraverso le strutture dati di uno specifico DBMS. L'output prodotto da questa fase è lo schema fisico (tipicamente codice SQL atto ad essere eseguito sul DBMS selezionato). L'input è lo schema logico prodotto nella fase precedente e il DBMS scelto.

Per una più semplice comprensione del database che si andrà a realizzare, è stato utilizzato il tool *Visual Paradigm Community Edition 10.1* in modo da rappresentarne graficamente la struttura fisica. Esso è già stato riportato in Figura 6.10 - "Struttura fisica del database lato client".

Il database presente sul server presenta la stessa identica struttura ma, per semplicità, sono stati modificati alcuni campi utilizzando un nome più semplice. Per completezza d'informazione, si riporta di seguito anche la struttura fisica dello schema di base di dati presente sul server.

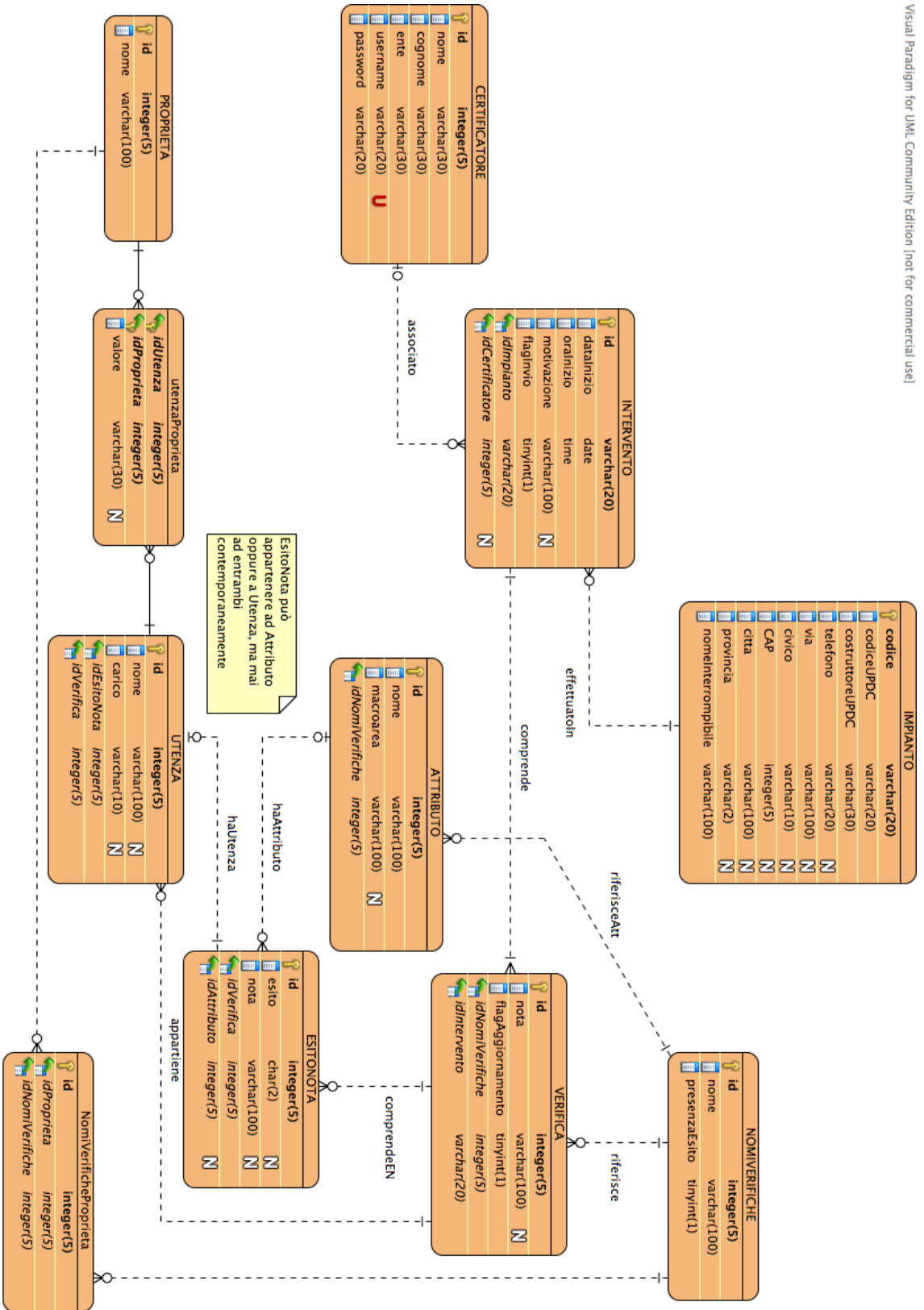


Figura 6.14 - Struttura fisica del database lato server

6.9 PROGETTAZIONE INTERFACCE GRAFICHE

Per la progettazione delle interfacce è stato utilizzato il programma Balsamiq Mockups, che ha consentito di produrre delle viste utente, in maniera da avere un'idea di come sarebbe stato il prodotto finale prima ancora dello sviluppo.

Il mockup in generale è un modello che permette di valutare e di apportare eventuali modifiche all'aspetto del prodotto finale prima di produrne un prototipo. Per un'applicazione mobile, in parole povere, il mock-up è l'anteprima - solitamente cartacea o grafica (nel senso di immagine) - che si presenta al cliente in fase di definizione del progetto. [9]

Realizzare un mockup è il modo più rapido e semplice di avere un'idea visuale di ciò che si andrà a sviluppare effettivamente: è come disegnare a matita, ma con il vantaggio dell'utilizzo del mezzo digitale. E' possibile, dunque, condividere il lavoro fatto e modificarlo costantemente, senza dover ridisegnare o ri-sviluppare il tutto nuovamente.

Qui di seguito i mockup realizzati¹¹:



Figura 6.15 - Mockup visualizzazione agenda

Figura 6.16 - Mockup visualizzazione checklist

¹¹ I dati riportati nei mockup sono puramente a scopo informativo. Non vi è alcuna corrispondenza realistica



Figura 6.17 - Mockup visualizzazione riepilogo checklist

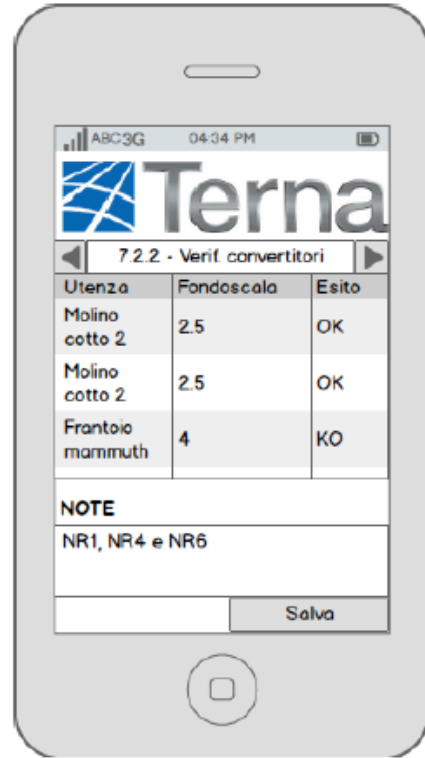


Figura 6.18 - Mockup visualizzazione attività di verifica

7. SVILUPPO DELLA SOLUZIONE

7.1 SVILUPPO APPLICAZIONE CLIENT

Il primo passo implementativo lato client è stato caratterizzato dall'implementazione delle viste utente.

I test di funzionamento sono stati effettuati sia su smartphone Android che su tablet.

Di seguito sono riportate tutte le viste utente, realizzate mediante screenshot da tablet.

7.1.1 VISUALIZZA CALENDARIO E APPUNTAMENTI



Figura 7.1 - Vista Calendario e Appuntamenti

Mediante questa vista, l'utente dell'applicazione può avere sotto controllo la propria agenda personale degli interventi da realizzare. Cliccando su un intervento, viene aperta la vista con l'elenco delle attività da effettuare (le attività della checklist).

7.1.2 VISUALIZZA ELENCO ATTIVITA'



Figura 7.2 - Vista Elenco Attività

Se l'utente clicca su una delle attività, l'attività si espande mostrando i singoli task (o meglio, i singoli interventi) che l'utente deve effettuare. In ogni momento è possibile espandere o ridurre ciascuna attività.

7.1.3 VISUALIZZA ELENCO TASK

MACROAREA	DOCUMENTI DA ESAMINARE	ESITO	NOTA
A) Conformita alle normative	Dichiarazione di conformita della UPDC	OK	+
B) Architettura	UPDC: descrizione dei componenti	KO	+
B) Architettura	RTU: caratteristiche costruttive e funzionali	OK	+
B) Architettura	RTU: schemi funzionali e di dettaglio	OK	+
B) Architettura	ROUTER: caratteristiche costruttive e funzionali	OK	+
B) Architettura	CONVERTITORE: caratteristiche costruttive	OK	+
B) Architettura	CONVERTITORE: classe di precisione	OK	+
C) Alimentazione	UPDC: schema di alimentazione	OK	+
D) Esecuzione meccanica	Armadio RTU/router: esecuzione (grado IP ecc.)	OK	+
D) Esecuzione meccanica	Schema della morsettiera	OK	+
D) Esecuzione meccanica	Caratteristiche dei cavi utilizzati	OK	+
E) Classe TV e TA	Classe di precisione	OK	+
E) Classe TV e TA	Errore di rapporto	OK	+
E) Classe TV e TA	Errore di angolo	OK	+
F) Sigillatura	Predisposizione alla sigillatura dei circuiti di smistamento, apertura e carichi	OK	+
G) Schema impianto	Schema unifilare impianto, indicazione carichi interrompibili	OK	+
H) Connessione UPDC: interruttori	Schema di dettaglio	OK	+
I) Caratteristiche	Tempo di apertura interruttore	OK	+

Figura 7.3 - Vista Elenco Task per Attività

Una volta che l'utente ha cliccato sull'attività specifica, può andare ad inserire le informazioni raccolte durante un intervento, memorizzandone in modo particolare l'esito ed inserendo, opzionalmente, una nota. I dati vengono salvati localmente quando l'utente clicca su "Salva Tutti i Dati".

7.1.4 VISUALIZZA RIEPILOGO CHECKLIST E INVIO

7.2.2 Verifica convertitori			
NOME UTENZA/CARICO	FONDOSCALA [MW]	ESITO	NOTA
Molino Cotto2	2.5	KO	-
Molino Cotto3	4	OK	-

7.2.3 Verifica dei comandi			
NOME UTENZA/CARICO	TIPO DI INTERROMPIBILITA	ESITO	NOTA
Molino Cotto2	-	KO	-
Molino Cotto3	-	OK	-
11	TR	KO	-
19	EM	OK	-

7.2.4 Verifica segnale interruttore			
NOME UTENZA/CARICO	TIPO DI INTERROMPIBILITA	ESITO	NOTA
Molino Cotto2	-	OK	-
Molino Cotto3	-	OK	-
11	TR	KO	-
19	EM	KO	ciao ciao

7.2.5 Verifica misura analogica			
NOME UTENZA/CARICO	ESITO	NOTA	
Molino Cotto2	OK	-	
Molino Cotto3	OK	-	

7.2.6 Verifica dialogo con BMI			
NOME UTENZA/CARICO	ESITO	NOTA	
Molino Cotto2	OK	-	
Molino Cotto3	OK	-	

7.2.7 Verifica sincronizzazione apparato			
NOME UTENZA/CARICO	ESITO	NOTA	
Molino Cotto2	OK	-	
Molino Cotto3	OK	-	

7.2.8 Verifica allarmistica			
NOME UTENZA/CARICO	ESITO	NOTA	
Molino Cotto2	OK	-	
Molino Cotto3	OK	-	

Figura 7.4 - Vista Visualizza Riepilogo Checklist

Una volta che sono stati compilati i diversi campi delle attività, l'utente, cliccando sul bottone "Riepilogo Dati e Invio" può visualizzare un riepilogo delle informazioni inserite, per controllare di aver compilato tutti i campi e in maniera corretta, prima di inviare definitivamente i dati al server.

Qui di seguito il corpo HTML della vista Visualizza Riepilogo Checklist.

```

<body onload="initialize()">
<div id="header">
  <div id="Logo"></div>
  <div id="dataInizio"></div><br></div>
  <div id="clienteIntervento"></div>
  <br></div>
</div>
<br></hr>
</hr>
</script>
  function jAlertFunction() {
    $('body').jAlert("L'applicazione consente invii multipli del rapporto compilato"+
      "<i>ma solo il primo</i> verrà  accettato dal server"
    )
  }
</script>
<div class="LoadingBox" style="display:none">
  <div id="waitingText"> Attendere una risposta dal server...</div>
  <div id="LoaderImage"></div>
  <br></div>
  <br></div>
  <br></div>
  <div id='titolo'></div>
  </br>
  </br>
  <div onclick="jAlertFunction()" class="stylish"></div>
  <div id="divInvio" style="display: block">
    <button type="button" id="bottoneInvio" onclick="insertEmail()">INVIA RAPPORTO</button>
  </div>
  </br>
  </br>
  <div id="containerDiv" style="display: block"></div>
</body>
</html>

```

7.2 CREAZIONE FILE XML

La generazione del file xml che contiene la checklist che il certificatore, tramite applicazione mobile dovrà inviare al server, viene effettuata automaticamente per mezzo di codice javascript scritto ad-hoc, a partire dal database locale.

Qui di seguito parte del codice sviluppato per la creazione del file xml.

```

function validateEmail(email)
{
    var re = /^[^<>()[\]\.\,\;\:\s@\\"]+(\.[^<>()[\]\.\,\;\:\s@\\"]+)*|(\\".+\\")@((\[[0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\])|((?!-)[a-z\d-]{2,6}){1,4})$/i;
    return re.test(email);
}

/*****da qui parte la lettura da DB e creazione xml*****/
function initializeCreateXml()
{
    $("#containerDiv").hide("slow");
    $("#divInvio").hide("slow");
    $("#titolo").hide("slow");
    $(".loadingBox").show("slow");
    console.log("stampo sta query:"+window.location.search.substr(1));
    var query = window.location.search.substr(1);
    var par = query.split('&');
    var c=0;
    var val = new Array();

    while(par.length > c)
    {
        val[c] = par[c].split('=');
        c++;
    }

    XMLstring=XMLstring+"<checkboxlist xmlns='http://www.example.org/NewXMLSchema'>";

    if(val[0][0] == 'cliente' && val[1][0] == 'data' && val[2][0] == 'idIntervento')
    { /*eseguo questo controllo per non fare crashare l'app nel caso non avessi passato parametri
    decodeURI toglie %20 dagli spazi presenti nell'indirizzo*/
        idIntervento = val[2][1];
        cliente= val[0][1];
        data= val[1][1];

        /*Stampo data e cliente*/
    }
    getInfoCertificatore();
}

```

Qui di seguito è riportato parte del file xml generato.


```

<?xml version="1.0" encoding="ISO-8859-1"?>

<checklist xmlns="http://www.example.org/NewXMLSchema">
  <certificatore id="1">
    <nome>Certificatore1</nome>
    <cognome>Test</cognome>
    <ente>Cesi</ente>
  </certificatore>

  <impianto codice="1">
    <codiceUPDC>AQ0001</codiceUPDC>
    <costruttoreUPDC>TecnoWatt</costruttoreUPDC>
    <telefono>+393333333333</telefono>
    <via>Vai</via>
    <civico>80</civico>
    <cap>67100</cap>
    <citta>L'Aquila</citta>
    <provincia>AQ</provincia>
    <nomeInterrompibile>Acciaieria 27</nomeInterrompibile>
  </impianto>

  <intervento id="AH023J">
    <dataInizio>2013-12-01</dataInizio>
    <oraInizio>13:00:00</oraInizio>
    <motivazione>Malfunzionamento</motivazione>
    <!-- needed idImpianto & idCertificatore -->
  </intervento>

  <!-- Verifica 7.1 Verifica Documentale -->

  <verifica idNomiVerifiche="1" nome="7.1 Verifica Documentale">
    <esitonota>
      <esito>OK</esito>
      <nota>PuffaNota</nota>
    </esitonota>
  </verifica>

```

E' stato necessario definire un file XSD, ovvero uno XML Schema Definition. L' **XML Schema** è un linguaggio di descrizione del contenuto di un file XML il cui scopo è delineare quali elementi sono permessi, quali tipi di dati sono ad essi associati e quale relazione gerarchica hanno fra loro gli elementi contenuti in un file XML. Ciò permette principalmente la convalida del file XML, ovvero la verifica che i suoi elementi siano in accordo con la descrizione in linguaggio XML Schema.

L'XSD è un'istanza di schema XML scritto in linguaggio XML Schema. Una XSD definisce il tipo di un documento XML in termini di vincoli: quali elementi ed attributi possono apparire, in quale relazione reciproca, quale tipo di dati può contenere, ed altro. Può essere usata anche con un programma di validazione, al fine di accertare a quale tipo

appartiene un determinato documento XML. La differenza principale fra XSD ed altri linguaggi di descrizione del contenuto XML sta nel fatto che XSD è stato sviluppato con l'intento che la determinazione della validità di un documento possa produrre una collezione di informazioni aderenti a specifici tipi di dati. A differenza delle DTD, gli XML Schema permettono la validazione del contenuto di un elemento o di un attributo rispetto a un determinato tipo di dato. Per esempio, un attributo potrebbe essere vincolato alla memorizzazione esclusiva di una data valida o di un numero decimale.

L' XSD fornisce un insieme di 19 tipi di dati primitivi (boolean, string, decimal, double, float, anyURI, QName, hexBinary, base64Binary, duration, date, time, dateTime, gYear, gYearMonth, gMonth, gMonthDay, gDay, e NOTATION) e consente la costruzione di nuovi tipi di dato partendo da questi tipi primitivi attraverso tre possibili meccanismi: restrizione (riduzione dell'insieme dei valori permessi), lista (estensione ad una sequenza di valori) ed unione (possibilità di scelta di un valore da differenti tipi). Le specifiche stesse del linguaggio indicano venticinque tipi derivati, ed ulteriori derivazioni di tipi possono essere definite dagli utenti all'interno dei propri schemi.

Di seguito è riportato parte del file xsd sviluppato.

```

<?xml version="1.0" encoding="UTF-8"?>

<schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace=
"http://www.example.org/NewXMLSchema" xmlns:tns="http://www.example.org/NewXMLSchema"
elementFormDefault="qualified">

<simpleType name="capytype">
  <restriction base="string">
    <pattern value="[0-9]{5}" />
  </restriction>
</simpleType>

<complexType name="esitonotatype">
  <all>
    <element name="esito" type="string" minOccurs="0" />
    <element name="nota" type="string" minOccurs="0" />
  </all>
</complexType>

<complexType name="attributotype">
  <all>
    <element name="macroarea" minOccurs="0" <!-- facoltativo -->
      <complexType>
        <attribute name="nome" type="string" />
      </complexType>
    </element>
    <element name="esitonota" minOccurs="1" type="tns:esitonotatype" /> <!-- obbligatorio
-->
  </all>
  <attribute name="id" type="positiveInteger" />
  <attribute name="nome" type="string" />
</complexType>

<complexType name="proprietatype">
  <all>
    <element name="nome" type="string" minOccurs="1" />
    <element name="valore" type="string" minOccurs="0" /> <!-- nella verifica 7.2.3 alla
singola utenza non è associato il tipo di interrompibilita' -->
  </all>
  <attribute name="id" type="positiveInteger" />
</complexType>

```

7.3 SVILUPPO WEB SERVICES

Come già accennato in precedenza, lo sviluppo dei web services lato server è stato effettuato avvalendosi del linguaggio Java e i web services sono stati deployati su Tomcat.

Il principale supporto nello sviluppo dei Web Services è stato fornito dalla libreria **JAX-WS**, inclusa nel progetto java **METRO 2.0**¹².

JAX-WS è un nuovo modello di programmazione che semplifica lo sviluppo delle applicazioni attraverso il supporto di uno standard, modello *annotation-based*, per

¹² <http://metro.java.net/guide/>

sviluppare Web Service application e client. Le specifiche di JAX-WS si allineano strategicamente con la tendenza attuale del settore verso un modello di messaggistica più document-centric e sostituisce il modello di chiamata di procedura remota (object-centric model) come definito da JAX-RPC. Con un servizio document-centric, la logica di fatto viene tenuta separata dal contenuto del documento. Il servizio riceve un documento XML, che contiene solo dati. I metodi specifici non vengono richiamati, ma il web service Web applica la sua logica di business al documento, il cui contenuto determina il flusso di lavoro di elaborazione. Il web service di un'agenzia di viaggi è un buon esempio di servizio document-centric. Il servizio riceve una richiesta: un documento XML che contiene i dettagli dell'itinerario di viaggio che si desidera effettuare (date, luoghi da visitare, alloggio preferito, e così via), il servizio elabora la richiesta in base al contenuto del documento. JAX-WS è il modello di programmazione strategica per lo sviluppo di servizi Web. L'implementazione dello standard di programmazione della JAX-WS fornisce importanti miglioramenti come: migliore indipendenza della piattaforma per le applicazioni Java, annotazioni, richiamare i servizi Web in modo asincrono, client dinamici e statici, Supporto per il Message Transmission Optimized Mechanism (MTOM, metodo di trasmissione di dati binari da e verso web service), il supporto per SOAP.

Il progetto fornisce il nucleo del progetto Metro (stack di web service di alto rendimento, espandibile, facile da usare, che raccoglie le funzionalità che racchiudono tutte le esigenze per la creazione di servizi web), all'interno della comunità GlassFish.

I web services, fin dalla loro comparsa, si sono proposti come la soluzione principe nella realizzazione di architetture interoperabili, interconnesse e integrabili tra loro. La vera innovazione è stata quella di fare uso di standard aperti utilizzando XML e HTTP (modello SOAP), insieme alla presenza del Web Services Description Language (WSDL) e introduzione di un sistema di ricerca/pubblicazione (UDDI). Sebbene da un lato questo progetto ricevesse una spinta in avanti nello sviluppo, dall'altro vi furono forti elementi di impedimento: i vari produttori e soggetti, che partecipavano allo sviluppo di componenti e implementazioni per web service, hanno ostacolato questo processo impedendo agli organismi preposti alla definizione di standard e specifiche di rimanere al passo.

Questo portò alla creazione di specifiche e implementazioni proprietarie, contrarie e contraddittorie agli obiettivi iniziali della realizzazione di web services. Nel 2003 i vari produttori, ormai consapevoli di come fosse privo di senso avere web services scritti in linguaggi diversi, si sono riuniti per cercare una soluzione a questo problema. Nasce così la WS-I Organization (Web Services Interoperability Organization) volta, come dice il nome, ad assicurare l'interoperabilità dei servizi web. Nel giro di poco tempo tutti i più importanti protagonisti del settore aderirono, al fine di creare un nuovo standard che garantisse i principi originari dei web services. Il consorzio operò alla ricerca e selezione del sottoinsieme minimo di standard e specifiche comuni a tutte le implementazioni, che furono poi raggruppati e raccolti in profili a seconda degli obiettivi che si volevano raggiungere, opera tuttora in corso. In ambito WS-I, Sun ha rilasciato recentemente una nuova implementazione chiamata WSIT, che consiste nel Java API for XML Web Services.

Alcune delle annotazioni più utilizzate sono:

- **@WebService** Marca una classe o un'interfaccia Java rispettivamente come una implementazione o una definizione di un Web Service. A questa possono essere specificati alcuni attributi come:
 - *name*: nome del wsdl: portType (nome della classe Java priva di package)
 - *portName*: nome del wsdl: portName (nome del WebService concatenato a Port)
 - *serviceName*: nome del servizio (wsdl:service) (nome della classe concatenato alla Stringa "Service")
 - *endpointInterface*: (es. `endpointInterface = "package.MioSOAP.MioWS"`) specifica il nome completo della SEI ("Service endpoint Interface", interfaccia Java che dichiara i metodi che un client può invocare sul servizio)
 - *endpointInterface*: (classe java, comprensiva del package, concatenata alla Stringa "ServicePortType")
 - *targetNamespace*: namespace di riferimento per il WSDL e gli XML generati (nome del package java)
 - *wsdlLocation*: l'indirizzo documento WSDL del Web Service. L'indirizzo può essere relativo o assoluto.

- **@WebMethod** Posta sopra la firma del metodo, dice all'interprete Java che si intende pubblicare questo metodo come un Web Service. Il corrispettivo metodo dovrà essere dichiarato "public" e dovrà restituire qualcosa (nel nostro caso restituisce un double). Anche questa ha alcuni attributi come:
 - *action*: definisce l'azione per l'operazione. (= al nome del metodo)
 - *exclude*: specifica se il metodo va escluso dalla pubblicazione del Web Service. (es. "false") (Booleano)
 - *operationName*: specifica il nome di wsdl: operation per il metodo (nome del metodo)
- **@SOAPBinding** Specifica la mappatura del Web Service nel protocollo di messaggi SOAP. In particolare si vede nel costrutto tra parentesi lo stile dei messaggi scelto. In caso i metodi del Web Service utilizzino tipi semplici come String o int è possibile utilizzare l'annotazione "@SOAPBinding(style=Style.RPC)".

In figura Figura 7.5 - Web Service: GestoreDati" è riportato il codice del Web Service sviluppato. Il Web Service richiama dei metodi di business per gestire opportunamente tutte le operazioni di: salvataggio dei dati ricevuti su memoria persistente, creazione file pdf e file excel, invio del pdf generato per email al certificatore che ha inviato la richiesta.

```

package com.terna.services;

import com.terna.gestorefile.CreaFileXml;
import com.terna.gestorefile.CreaPdf;
import com.terna.gestorefile.InvioEmailTest;
import com.terna.gestorefile.XmlToObj;
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;

/**
 *
 * @author Domenica
 */
@WebService(serviceName = "GestoreDati")
public class GestoreDati {

    @WebMethod(operationName = "elaboraDati")
    public void elaboraDati(@WebParam(name = "xmlfile") String xmlfile,
        @WebParam(name = "email") String emailAddr) {

        System.out.println("chamata ricevuta!");
        System.out.println(xmlfile);
        CreaFileXml nuovo=new CreaFileXml(xmlfile);
        String xmlstring=nuovo.creaxml();
        CreaPdf pdfnuovo=new CreaPdf();
        pdfnuovo.generaPdf();
        XmlToObj xmltoobj=new XmlToObj();
        xmltoobj.writeXML();
        /*inserire la mail del destinatario.. la mail deve
        * provenire dal client*/
        InvioEmailTest email=new InvioEmailTest(emailAddr);

        email.sendEmail();
        //return "Qui di seguito la checklist in xml: "+xmlstring;
    }
}

```

Figura 7.5 - Web Service: GestoreDati

Per completezza, in Figura 7.6 - WSDL associato al Web Service GestoreDati” è riportato per intero il file WSDL in XML, generato automaticamente mediante la libreria JAX-WS, mediante inserimento nel campo della barra indirizzi di un browser, la stringa formata da *ContextPath/GestoreDati?wsdl*.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<!--
  Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.2-hudson-740-.
-->
<!--
  Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.2-hudson-740-.
-->
<definitions xmlns:wsu="http://docs.oasis-open.org/ws/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsp="http://www.w3.org/ns/ws-policy"
  xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://services.terna.com/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://services.terna.com/" name="GestoreDati">
  <types>
    <xsd:schema>
      <xsd:import namespace="http://services.terna.com/" schemaLocation="http://localhost:8090/InternaServer/GestoreDati?xsd=1"/>
      </xsd:schema>
    </types>
    <message name="elaboraDati">
      <part name="parameters" element="tns:elaboraDati"/>
    </message>
    <message name="elaboraDatiResponse">
      <part name="parameters" element="tns:elaboraDatiResponse"/>
    </message>
    <portType name="GestoreDati">
      <operation name="elaboraDati">
        <input wsam:Action="http://services.terna.com/GestoreDati/elaboraDatiRequest" message="tns:elaboraDati"/>
        <output wsam:Action="http://services.terna.com/GestoreDati/elaboraDatiResponse" message="tns:elaboraDatiResponse"/>
      </operation>
    </portType>
    <binding name="GestoreDatiPortBinding" type="tns:GestoreDati">
      <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
      <operation name="elaboraDati">
        <soap:operation soapAction=""/>
        <input>
          <soap:body use="literal"/>
        </input>
        <output>
          <soap:body use="literal"/>
        </output>
      </operation>
    </binding>
    <service name="GestoreDati">
      <port name="GestoreDatiPort" binding="tns:GestoreDatiPortBinding">
        <soap:address location="http://localhost:8090/InternaServer/GestoreDati"/>
      </port>
    </service>
  </definitions>
```

Figura 7.6 - WSDL associato al Web Service GestoreDati

7.4 CREAZIONE FILE PDF

A partire dal file XML, memorizzato in locale dal server, contenente le informazioni sulla certificazione e i controlli effettuati dal certificatore e da quest'ultimo inviato, viene scritto un file PDF.

Per la scrittura di tale file sono state utilizzate le librerie **Apache FOP 1.1**¹³. Come prima cosa è stato necessario produrre un file **.xsl** (**XSL**: eXtensible Stylesheet Language, linguaggio di descrizione dei fogli di stile per i documenti in formato XML): grazie ad esso è stato definito il layout dell'intero file pdf utilizzando, in particolare, il linguaggio **XSL-FO**, che è parte dell'XSL. XSL-Formatting Objects è un linguaggio, basato su XML, per la specifica di una semantica di formattazione dei contenuti di un documento XML. All'interno di un documento XSL-FO è possibile definire le regole di impaginazione delle informazioni che vogliamo visualizzare, andando ad esprimere la posizione dei singoli contenuti nella pagina.

Grazie a XSL-FO possiamo produrre, ad esempio, dei documenti PDF, RTF o PS, rappresentanti il contenuto di un documento XML. Il documento XSL-FO viene analizzato da un processore XSL-FO che si occupa di produrre il documento di output, in base alle regole specificate nel documento di input.

All'interno dell'elemento `<layout-master-set>` vanno inserite le regole di impaginazione per la pagina master, ovvero la pagina che costituisce il modello per tutte le altre pagine del documento. Il secondo elemento `<declarations>` contiene eventuali dichiarazioni globali al file XSL-FO; mentre l'elemento `<page-sequence>` contiene la definizione delle informazioni contenute nelle pagine che costituiranno il documento finale. [10] Altri elementi utili che occorre mettere in luce sono:

- `<title>`: permette di specificare il titolo del documento;
- `<static-content>`: indica le informazioni che verranno ripetute in ogni pagina;
- `<flow>`: elemento che contiene il contenuto delle pagine.

¹³ <http://xmlgraphics.apache.org/fop/>

Per l'invio delle email sono state utilizzate le API **JavaMail 1.4.7**¹⁴

(Qui di seguito è riportato parte del codice sviluppato in XSL-FO)

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:fo-
"http://www.w3.org/1999/XSL/Format" exclude-result-prefixes="fo" xmlns:my-
"http://www.example.org/NewXMLSchema">

  <xsl:key name="attributi-per-macroarea" match="my:attributo" use="my:macroarea/@nome" />

  <xsl:template match="my:checklist">

    <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
      <fo:layout-master-set>
        <fo:simple-page-master master-name="A4">
          <fo:region-body margin="1in" margin-top="4cm"/>
          <fo:region-before extent="1cm"/>
        </fo:simple-page-master>
      </fo:layout-master-set>

      <fo:page-sequence master-reference="A4">
        <fo:static-content flow-name="xsl-region-before">
          <fo:block font-size="24pt" text-align="center" font-weight="bold" margin-top-
"1cm"><xsl:value-of select="my:certificatore/my:ente" /></fo:block>
          <fo:block-container>
            <fo:block >
              <fo:inline font-weight="bold" font-size="14pt" text-align="left"
padding-left="10%"> Allegato Rapporto </fo:inline>
              <fo:inline padding-left="53%" font-size="11pt" margin-right="0cm">
Pag. <fo:page-number ></fo:page-number></fo:inline>
            </fo:block>
            <fo:block border-bottom-color="black" border-bottom-style="solid"
margin-left="2cm" margin-right="2cm" />
          </fo:block-container >
        </fo:static-content>

        <fo:flow flow-name="xsl-region-body">

          <fo:block margin-top="1cm">
            <fo:inline font-weight="bold"> Cliente </fo:inline>
            <fo:inline padding-left="10%" font-size="11pt" ><xsl:value-of select-
"my:impianto/my:nomeInterrompibile" />
            <fo:block margin-left="20%" font-size="11pt">Via <xsl:value-of select-
"my:impianto/my:via" />, <xsl:value-of select="my:impianto/my:civico" />
            </fo:block>
            <fo:block margin-left="20%" font-size="11pt"> <xsl:value-of select-
"my:impianto/my:cap" />, <xsl:value-of select="my:impianto/my:citta" />(<
<xsl:value-of select="my:impianto/my:provincia" /></fo:block>
```

¹⁴ www.oracle.com/technetwork/java/javamail/

7.5 CREAZIONE FILE EXCEL

All'interno della classe `ObjToExcel`, appositamente creata, avviene la scrittura vera e propria del file `.xlsx`. Per la scrittura di tale file sono state utilizzate le librerie **Apache POI**¹⁵.

Il progetto Apache POI ha come obiettivo quello di creare delle API Java per poter manipolare i diversi formati su cui sono basati i documenti Office di Microsoft, in particolare l'API è in grado di manipolare file di tipo OOXML (Office Open XML) e di tipo OLE 2 (OLE 2 Compound Document). Lo standard OLE 2 comprende i file di Office con estensione XLS, DOC e PPT mentre lo standard OOXML comprende quelli con estensione XLSX, DOCX e PPTX. Le API per OLE 2 si appoggiano su API di livello più basso chiamate POIFS (OLE2 Filesystem) che serve per manipolare il contenuto del file, e HPSF (OLE2 Document Properties) che si occupa di manipolare le proprietà del file, quali il titolo o l'autore. Le API per OOXML si appoggiano invece sulla libreria `openxmlj4`. In ogni caso generalmente non vengono usate queste librerie di basso livello ma si usano i "componenti".

Per ogni applicazione Office esiste un suo componente, che fornisce una API di livello più alto di astrazione per manipolare i file in formato OLE 2 e OOXML. Per Excel i moduli sono HSSF (per OLE 2) e XSSF (per OOXML) ed esiste un modulo per manipolare entrambi i formati che si chiama SS, per Word sono HWPF e XWPD mentre per PowerPoint sono HSLF e XSLF. [11]

Il progetto Apache POI è alla base di altri progetti Apache quali Lucene¹⁶ e Tika¹⁷.

In Figura 7.7 - "Struttura file excel" è riportata la struttura di riferimento che è stata utilizzata per la generazione del file excel.

¹⁵ <http://poi.apache.org>

¹⁶ <http://lucene.apache.org/>

¹⁷ <http://tika.apache.org/>

	A	B	C	D
1	Macro area	Documenti da esaminare	Tipo valore	Commento
2	A) Conformità alle normative			
3		Dichiarazione di conformità della UPDC (*)	Flag o no Flag - indicante se è stato esaminato o meno	
4	B) Architettura			
5		UPDC: descrizione dei componenti	Flag o no Flag - indicante se è stato esaminato o meno	
6		RTU: caratteristiche costruttive	Flag o no Flag - indicante se è stato esaminato o meno	
7		RTU: schemi funzionali e di dettaglio	Flag o no Flag - indicante se è stato esaminato o meno	Cisco 2821 SN.FHK1439F2HN
8		RTU: caratteristiche funzionali	Flag o no Flag - indicante se è stato esaminato o meno	
9		ROUTER: caratteristiche costruttive e funzionali	Flag o no Flag - indicante se è stato esaminato o meno	
10		CONVERTITORE: caratteristiche costruttive	Flag o no Flag - indicante se è stato esaminato o meno	
11		CONVERTITORE: classe di precisione	Flag o no Flag - indicante se è stato esaminato o meno	
12	C) Alimentazione			
13		UPDC: schema di alimentazione	Flag o no Flag - indicante se è stato esaminato o meno	
14	D) Esecuzione meccanica			
15		Armadio RTU/router: esecuzione (grado IP ecc.)	Flag o no Flag - indicante se è stato esaminato o meno	
16		Schema della morsettiera	Flag o no Flag - indicante se è stato esaminato o meno	
17		Caratteristiche dei cavi utilizzati	Flag o no Flag - indicante se è stato esaminato o meno	
18	E) Classe TV e TA			
19		Classe di precisione	Flag o no Flag - indicante se è stato esaminato o meno	
20		errore di rapporto	Flag o no Flag - indicante se è stato esaminato o meno	
21		errore d'angolo	Flag o no Flag - indicante se è stato esaminato o meno	
22	F) Sigillatura			
23		Prediso. alla sigillatura dei circuiti di smistamento, apertura e carichi	Flag o no Flag - indicante se è stato esaminato o meno	

Figura 7.7 - Struttura file excel

7.6 OPERAZIONI SU DATABASE

Per effettuare operazioni su database, in particolar modo per scrivere le informazioni ricevute tramite il body del messaggio SOAP in formato XML, sono state utilizzate le API JAXB.

JAXB (Java Architecture for XML Binding) è un framework a supporto dell'utilizzo dell'XML in Java. Esso mette a disposizione un set di API per semplificare l'accesso e la realizzazione di documenti XML attraverso applicazioni scritte direttamente in Java. Ciò che consente di fare JAXB è accedere e processare l'XML senza dover conoscere il contenuto del documento stesso e senza dover lavorare "pensando" in XML. Le operazioni principali per poter elaborare un documento XML sono le operazioni di Bind e di Unmarshall. Con l'operazione di Bind si semplifica l'operazione di accesso a un documento XML partendo dal corrispondente schema XML. Con l'Unmarshall invece si ottengono degli oggetti Java che rappresentano il contenuto e l'organizzazione del documento XML. [12]

Il meccanismo di mapping tra XML Schema e Java viene implementato mediante l'uso delle annotazioni definite nel package *javax.xml.bind.annotation*.

Un'implementazione JAXB è costituita dai seguenti componenti architetturali:

- Schema compiler: associa uno schema di origine ad un insieme di elementi derivati dallo schema. Il legame è descritto da un linguaggio vincolante basato su XML.
- Schema generator: mappa un insieme di elementi esistenti in uno schema. La mappatura è descritta tramite annotazioni.
- Binding runtime framework: fornisce le operazioni di unmarshalling e marshalling per l'accesso, la manipolazione, e la validazione dei contenuti XML utilizzando alternativamente lo schema o gli elementi esistenti. [13]

Gli oggetti ottenuti mediante Unmarshalling sono stati utilizzati direttamente lato server, senza modifiche alcune, in quanto è stato constatato che riescono a modellare bene la realtà di dominio.

È stato utilizzato MySQL come DBMS e sono state garantite opportune operazioni combinate di *commit()* e *rollback()*, con gestione di eventuali eccezioni, per garantire che le proprietà ACID delle transazioni fossero rispettate e per evitare di lasciare il database in uno stato inconsistente.

8. CONCLUSIONI

Il prototipo di applicazione così realizzato può, a tutti gli effetti, essere trasformato in un'applicazione vera e propria da distribuire agli utilizzatori finali, ma occorre fare delle precisazioni che inducono ad ipotesi di modifiche del software sviluppato fin'ora e, quindi, ad aggiornamenti futuri.

Il web service per il recepimento delle informazioni inviate tramite l'applicazione è stato sviluppato esclusivamente in Java e messo in ascolto sul server Tomcat:

- Da un riscontro, non immediato, con i referenti aziendali è risultato che, per necessità aziendali, il web service debba essere modificato e tradotto in **C#**, oltre ad essere eseguito su piattaforma **.NET**.

L'applicazione necessita di essere testata in caso di aggiornamento delle attività/task. Così com'è stata progettata e sviluppata, in caso di aggiornamento delle operazioni di certificazione lato back-end, l'applicazione supporta bene queste modifiche, ma:

- In che modo eventuali modifiche della checklist di certificazione predefinita debbano essere trasmesse (se mediante download della app. dallo store e aggiornamento della stessa, o tramite richiesta esplicita attraverso l'applicazione stessa), è ancora da stabilire. Avendo previsto entrambe le modalità come possibili, si è reso il database locale, come è già stato specificato, notevolmente flessibile e atto a recepire qualsiasi modifica senza bisogno di aggiornare/installare l'applicazione ex-novo.
- Ulteriori vincoli relativi alla sicurezza dei dati trasmessi (oltre all'autenticazione) restano da definirsi.
- L'interfaccia grafica potrebbe essere resa più accattivante e ancor più user-friendly, sfruttando appieno le potenzialità offerte dal binomio HTML5-CSS3.

BIBLIOGRAFIA

- [1] T. S.p.A.. [Online]. Available: <http://www.terna.it/>.
- [2] A. Sinibaldi, La gestione dei processi in azienda. Introduzione al Business Process Management., Milano: Franco Angeli, 2009.
- [3] M. K. a. E. Oksman, «developforce,» [Online]. Available: <http://wiki.developforce.com/>.
- [4] M. Kimihira, «Developing Mobile Applications: Web, Native, or Hybrid?,» 10 2012. [Online]. Available: <https://blogs.oracle.com>.
- [5] J. Wargo, «PhoneGap Essentials: Building Cross-Platform Mobile Apps,» 2012.
- [6] J. A. Kreibich, O'Reilly Media, 2010.
- [7] Wikipedia, «XML,» [Online]. Available: <http://it.wikipedia.org/>.
- [8] L. V. Tagliati, «UML e ingegneria del software: dalla teoria alla pratica.,» Milano, Tecniche Nuove, 2003.
- [9] P. Galli, «Usabilità, consigli pratici per migliorare e gestire il proprio sito web,» 2008. [Online]. Available: <http://usabilita.blogosfere.it/2008/12/e-fatti-un-mockup-del-sito.html>.
- [10] A. Chiarelli, 2006. [Online]. Available: <http://www.html.it/pag/31762/xsl-fo/>.
- [11] angeloonline, 2010. [Online]. Available: <http://www.appuntisoftware.it/apache-poi-manipolare-documenti-microsoft-in-java/>.
- [12] M. Parente, 2013. [Online]. Available: <http://www.html.it/articoli/jaxb-lxml-in-java-binding/>.

- [13] «Docs Oracle,» [Online]. Available:
<http://docs.oracle.com/javase/tutorial/jaxb/intro/arch.html>.
- [14] E. T. S.p.A, «UPDC - Rev.0 IT -,» Trezzano Sul Naviglio (Milano), 03/2011.
- [15] M. Lacey, 2012. [Online]. Available:
<http://mobile.dzone.com/articles/phonegapcordova-misconception>.
- [16] «Apache PhoneGap,» 2013. [Online]. Available: <http://phonegap.com/>.