POLITECNICO DI TORINO

Corso di Laurea Magistrale in

Ingegneria Informatica (Computer Engineering)

Tesi di Laurea Magistrale

Progettazione e sviluppo di un'applicazione web per la gestione del personale della pubblica amministrazione utilizzando il framework UniGUI

Relatore
Prof. Giovanni Malnati

Candidato
Andrea Franco
Matricola 256973

A.A. 2020/2021 - Dicembre 2021

Sommario

Il seguente lavoro di tesi è stato redatto a seguito dell'esperienza maturata presso l'azienda Mondo Edp S.r.l. di Cuneo, fornitore di software per la rilevazione presenze, la gestione economica e giuridica dei dipendenti della pubblica amministrazione, con installazioni su tutto il territorio nazionale.

Si articola in cinque capitoli e descrive i risultati ottenuti nel corso del lavoro di progettazione e sviluppo di IrisAPP: un'applicazione web ottimizzata per l'uso su dispositivi mobili, realizzata utilizzando il linguaggio di programmazione Delphi già in uso in azienda per la creazione dei software commercializzati.

L'obiettivo dell'applicazione è fornire al dipendente della pubblica amministrazione alcune funzionalità utili per la gestione della rilevazione presenze, come la timbratura virtuale (senza rilevatore fisico), la richiesta e l'autorizzazione delle assenze (come ferie e permessi), la richiesta e l'autorizzazione dei cambiamenti di orario, la ricezione di messaggi inviati dall'ufficio del personale, il download del cartellino mensile e del cedolino paga. La web app si integra con gli altri applicativi che compongono il sistema informativo Iris, formato dai software gestionali già utilizzati dal back office dalle aziende clienti.

Nella fase di sviluppo è stato utilizzato il nuovo framework UniGUI, per la prima volta in uso in azienda, che consente di realizzare single page application riducendo i tempi ed i costi di sviluppo, mediante una forte riduzione della quantità di codice necessario per la realizzazione della parte front-end dell'applicazione.

Una prima fase di analisi dei requisiti ha portato alla definizione delle funzionalità richieste, tenendo conto dei vincoli e delle opportunità offerte dall'integrazione con i software del sistema informativo Iris. La seconda fase, che ha compreso la valutazione del framework e lo sviluppo dell'applicazione, ha confermato la bontà del nuovo strumento di lavoro utilizzato ed ha portato alla realizzazione del prodotto richiesto. La fase di testing finale ha confermato i risultati attesi ed ha consentito il rilascio di un software che conta, a novembre 2021, una quindicina di installazioni presso aziende sanitarie, università ed enti locali in Italia.

Indice

1. Introduzione

In questo primo capitolo viene descritto il contesto in cui si colloca il lavoro svolto, fornendo una presentazione dell'azienda presso cui ho svolto la tesi ed una descrizione delle caratteristiche dei prodotti software che compongono il sistema informativo Iris, di cui l'applicazione sviluppata fa parte. La web app presenta infatti un elevato livello di integrazione con i software aziendali utilizzati per la gestione del personale dipendente delle aziende clienti.

1.1 L'azienda Mondo Edp S.r.l.

Mondo Edp S.r.l. è una software house specializzata nello sviluppo e nella fornitura di soluzioni software per la gestione delle risorse umane nella pubblica amministrazione e nella sanità, la sua offerta, che comprende anche i servizi di assistenza, manutenzione, formazione ed avviamento è rivolta a:

- Aziende Sanitarie Locali
- Aziende Ospedaliere
- Pubblica amministrazione locale (Comuni, Provincie, Regioni)

Nata nel 1995 dall'esperienza decennale dei suoi fondatori in software per la gestione del personale, è conosciuta su tutto il territorio nazionale per la completezza ed affidabilità dei suoi prodotti software e per la qualità del servizio offerto sia agli utenti che ai partners che desiderano proporre i prodotti Mondo Edp alla loro clientela.

La maggior specializzazione, che vede Mondo Edp protagonista a livello nazionale, ha da sempre riguardato e riguarda lo sviluppo di software applicativo per la gestione del personale; nel settore della gestione delle presenze-assenze, in cui la Mondo Edp è leader nazionale, vanta un'esperienza di oltre 25 anni e centinaia di installazioni, in Sanità e Pubblica Amministrazione.

Il target ideale di utenza degli applicativi di Mondo Edp è costituito da enti di medie-grandi dimensioni la cui complessità normativo-organizzativa necessita di soluzioni adeguate e di servizi di elevato profilo professionale. Dal 1999 la Mondo Edp è titolare di certificazione di Qualità ISO 9001 e dal 2018 è azienda certificata ISO/IEC 27001.

Le soluzioni software IrisCloud/WEB sono qualificate AGID nel Cloud Marketplace https://cloud.italia.it/marketplace/service/685.

I suoi prodotti software, identificati dal marchio registrato "Iris" sono costantemente aggiornati sia sotto il profilo funzionale che sotto quello normativo e tecnologico. La fiducia riscontrata costituisce una spinta allo sviluppo di nuove funzionalità e aggiornamento delle metodologie e tecnologie di sviluppo.

1.2 Il sistema informativo Iris

Il sistema informativo Iris si compone di una serie di prodotti software dalla struttura modulare destinati alle aziende clienti per la gestione del personale dipendente, disponibili sia in versione applicazione desktop (denominata "IrisWIN") sia in versione applicazione web (denominata "IrisCloud"):

- **Rilevazione presenze**

  Permette l'acquisizione automatica delle timbrature dal sistema degli orologi, interfacciandosi con tutti i modelli in modo parametrico, la gestione dei giustificativi di presenza e di assenza e la conseguente emissione del cartellino mensile. I dati raccolti ed elaborati possono essere poi condivisi con la gestione economica sia utilizzando l'integrazione nativa con il modulo gestione economica di Iris, sia integrandosi con applicativi per la gestione degli stipendi di altri fornitori. Consente inoltre la pianificazione dei turni di servizio e di reperibilità, la gestione di incentivi, buoni pasto, accessi alla mensa aziendale, malattie e visite fiscali.

- **Gestione economica**

  Permette la gestione della complessa materia riferita alle retribuzioni delle aziende sanitarie ed ospedaliere, consentendo l'elaborazione dei cedolini mensili, ma anche i ricalcoli ed i conguagli in maniera automatizzata attraverso l'acquisizione dei dati provenienti sia dalla rilevazione presenze fornita dalla stessa Mondo EDP, sia da altri fornitori di software con funzionalità analoghe. Il quadro della gestione economica si completa con l'elaborazione di: modello CU, modello 770, F24 EP mensili, conto annuale, denuncia annuale INAIL e gestione dei rimborsi del modello 730.

- **Stato giuridico**

  Permette la gestione delle principali informazioni di cui l'amministrazione necessita per un completo inquadramento nelle proprie strutture del personale dipendente: oltre a tutti i dati anagrafici e di inquadramento giuridico, le assunzioni, le cessazioni, l'attribuzione delle qualifiche e del reparto di lavoro, i titoli acquisiti, gli aggiornamenti professionali, le variazioni dello stato di servizio, le eventuali sanzioni disciplinari, le annotazioni di merito e relativi provvedimenti avuti, le assenze di rilevanza giuridica. Ad esso sono collegati sia la rilevazione presenze che la gestione economica, utilizzando le informazioni raccolte e fornendo a loro volta informazioni di interesse reciproco.

- **IrisWEB**

  Ai moduli descritti in precedenza si affianca il portale per il dipendente IrisWEB, l'applicazione web che rappresenta un'interfaccia tra il dipendente e l'ufficio del personale, agevolando l'interazione tra le due parti e riducendo il carico di lavoro per il back office aziendale. Molto importante è la funzionalità di timbratura virtuale, che consente di limitare l'utilizzo degli orologi di timbratura tradizionali, e l'implementazione degli iter autorizzativi, che consentono di gestire l'interazione tra il dipendente ed il relativo responsabile attraverso meccanismi basati su richieste ed autorizzazioni.

Oltre agli applicativi principali descritti in precedenza sono stati sviluppati numerosi software di supporto, utilizzati, ad esempio, per la gestione degli aggiornamenti dell'intera suite applicativa distribuita e per consentire l'interoperabilità con altri sistemi informativi presenti sia all'interno che all'esterno delle aziende clienti.

Tutti gli applicativi elencati consentono per ogni installazione una gestione dei dipendenti detta "multi-azienda": è infatti possibile creare sulla stessa istanza del database comune di Iris molteplici aziende, rappresentate da una rigida suddivisione logica dei dipendenti gestiti e di conseguenza degli operatori/utenti che accedono ai programmi, questi potranno operare solamente sui dati associati all'azienda selezionata al momento dell'autenticazione e su cui sono autorizzati a lavorare.

Essendo il quadro normativo di riferimento in continua evoluzione, gli applicativi distribuiti dalla Mondo Edp vengono costantemente aggiornati con il frequente rilascio di nuove versioni, l'aggiornamento delle singole installazioni presso i clienti può essere eseguito sia dal personale interno della software house, sia in autonomia dal personale dei servizi informativi delle aziende clienti utilizzando procedure standardizzate e l'applicativo di supporto dedicato all'operazione. Le nuove versioni sono accessibili ai clienti tramite il sito web: mediante l'accesso ad un'area riservata è consentito il download dei pacchetti di aggiornamento personalizzati in base ai moduli acquistati dalla singola azienda.

2. Analisi e descrizione dell'applicazione

L'obiettivo dell'applicazione IrisAPP è quello di fornire un'interfaccia fruibile sui dispositivi mobili anche attraverso la rete internet, per consentire al dipendente della pubblica amministrazione l'accesso alle funzionalità del sistema informativo Iris già in uso presso le aziende clienti, affiancando l'utilizzo dell'applicazione IrisWEB dalle funzionalità duali ma destinata prevalentemente all'uso sulle postazioni desktop e su rete locale.

Si tratta di un'applicazione web fortemente integrata con i software del sistema informativo Iris, con cui condivide il database. Inoltre, la configurazione stessa delle funzionalità offerte, inclusa la gestione degli utenti, viene gestita tramite gli applicativi IrisWIN ed IrisCloud, utilizzati dall'ufficio del personale per la gestione dei dipendenti dell'azienda.

La gestione degli accessi degli utenti richiede la compatibilità con diverse modalità di autenticazione per consentire l'interoperabilità con i sistemi delle aziende clienti, tipicamente di medie-grandi dimensioni, che già utilizzano sistemi basati su protocolli di autenticazione standard.

Un altro requisito importante è legato alla distribuzione ed all'aggiornamento dell'applicazione. Il concetto di versione del programma non è legato alla singola applicazione distribuita ma all'intero sistema informativo Iris. Tutti gli applicativi che lo compongono devono infatti essere aggiornati alla stessa versione, per garantire la compatibilità delle operazioni effettuate dai vari programmi sulla base dati comune. Il rilascio di nuove versioni è frequente e segue la continua evoluzione delle normative nazionali e regionali, richiedendo quindi la necessità di integrare l'aggiornamento della nuova applicazione nelle procedure esistenti. Essendo gli altri applicativi web eseguiti con il web server Microsoft IIS, anche IrisAPP viene rilasciata come applicazione ISAPI ed aggiornata utilizzando la stessa procedura predefinita.

Le funzionalità offerte possono essere suddivise in due categorie che determinano due tipologie di profili per gli utenti utilizzatori, un profilo "dipendente" che può accedere solamente ai dati ad esso associati e che assume il ruolo di richiedente nel contesto degli iter autorizzativi, ed un profilo "responsabile" che può accedere ai dati dei collaboratori gestiti ed assume il ruolo di autorizzatore negli iter autorizzativi.

Nel seguito di questo secondo capitolo viene fornita una descrizione delle funzionalità implementate e vengono approfonditi gli argomenti descritti brevemente in precedenza.

2.1 Descrizione dell'applicazione

Le funzionalità offerte dall'applicazione possono essere suddivise concettualmente in tre tipologie:

- Funzionalità di base: consentono la gestione di autenticazione e autorizzazione degli utenti nell'accesso alle funzionalità, definiscono il modello di navigazione tra le funzioni disponibili e permettono la gestione dei dati associati all'utente, definendo la struttura dell'interfaccia grafica su cui si basa il programma.
- Funzionalità di interfaccia: consentono al dipendente di interfacciarsi con l'ufficio del personale per eseguire alcune operazioni di base, come l'accesso ad informazioni riguardanti sé stesso o i collaboratori gestiti, il download di documenti resi disponibili dall'ufficio, l'inserimento delle timbrature giornaliere e la lettura dei messaggi di avviso ricevuti, snellendo il carico di lavoro sul back office aziendale.
- Iter autorizzativi: implementano gli iter richiesti dalla gestione del personale aziendale, facilitando ed automatizzando l'interazione tra il collaboratore ed il relativo responsabile, sotto la supervisione dell'ufficio del personale.

Viene fornita una descrizione delle funzionalità implementate e dei dati gestiti dall'applicazione, corredata da alcune schermate significative dell'applicazione sviluppata.

Tutte le informazioni visibili nelle schermate utilizzate contengono dati di fantasia e sono state ottenute a partire da un database di test.

2.1.1 Funzionalità di base

Login

La pagina di login (Figura 2.1) consente all'utente di autenticarsi per accede all'applicazione, i dati richiesti sono:

- Azienda: rappresenta l'azienda utilizzata nel contesto della gestione multi-azienda del sistema informativo Iris, il campo può opzionalmente essere precompilato con un valore di default e nascosto nel caso l'inserimento non sia richiesto al dipendente.
- Utente: rappresenta il nome utente che il dipendente può utilizzare per accedere all'applicazione, è assegnato dall'ufficio del personale ed è associato al dipendente presente nell'anagrafica del personale gestito nel sistema informativo Iris.
- Password: rappresenta la password assegnata dall'ufficio del personale al dipendente (o quella successivamente modificata), può essere utilizzata nelle diverse modalità di autenticazione previste.
- Profilo: rappresenta uno dei profili assegnati all'utente dall'ufficio del personale, diversi profili possono avere diversa visibilità sui dati dei dipendenti, consentendo la distinzione tra profili di tipo "dipendente" e di tipo "responsabile".

Figura 2.1 – La pagina di login dell'applicazione

L'utente può attivare l'opzione "Ricorda credenziali" che consente la memorizzazione dei dati di login nei cookies (cifrati con algoritmo AES-128-CBC con chiave privata conosciuta soltanto lato server e codificati base64). Nei successivi tentativi di accesso all'applicazione non verrà mostrata la pagina di login ma verranno verificate direttamente le credenziali salvate. È inoltre disponibile la funzionalità di recupero della password: cliccando sul link "Recupera password", inserendo soltanto l'azienda di riferimento ed il nome utente nei rispettivi campi, viene inviata una e-mail per il recupero sull'indirizzo associato all'utente utilizzando il server SMTP dell'azienda cliente. Opzionalmente è possibile inserire un testo libero sotto il logo IrisAPP, a scelta del cliente (tipicamente è utilizzato per la ragione sociale).

La gestione degli utenti e le modalità di autenticazione disponibili saranno approfondite in seguito.

Pagina principale

La pagina principale consente all'utente autenticato l'accesso alle funzionalità disponibili per il profilo in uso. Il layout della pagina varia in base al dispositivo utilizzato, sugli smartphone (Figura 2.2) viene visualizzato il menu principale che occupa tutto lo schermo in larghezza e la selezione di una delle funzionalità disponibili apre una vista sovrapposta alla pagina principale. Sui tablet

(Figura 2.3) viene visualizzato il menu principale sul lato sinistro e la funzionalità selezionata viene aperta in modo da occupare lo spazio restante sul lato destro della pagina, lasciando sempre visibile il menu. Nel secondo caso, al primo accesso alla pagina principale vengono automaticamente aperte le funzionalità "Dati giornalieri" (per i profili di tipo dipendente) o "Elenco dipendenti" (per i profili di tipo responsabile). In entrambe le modalità è limitato l'accesso ad una sola funzionalità per volta.

Nella parte alta della pagina è visibile la versione in uso di IrisAPP, la ragione sociale dell'azienda cliente, un pulsante sul lato sinistro che consente di effettuare il logout dall'applicazione (senza la cancellazione delle credenziali eventualmente salvate nei cookies) ed un pulsante sul lato destro che consente di accedere alla funzione di gestione dell'utente.

Fig. 2.2 – La pagina principale su smartphone

Il menu principale sottostante contiene la lista delle funzionalità disponibili per il profilo selezionato, l'elenco viene personalizzato in base alle abilitazioni specifiche del profilo in uso. Accanto alla singola funzionalità può essere visualizzata un'icona su sfondo rosso che consente di segnalare la presenza di notifiche rilevanti (ad esempio il numero di messaggi da leggere, il numero di richieste da approvare o la presenza di cedolini da visualizzare); il valore numerico visualizzato viene

Figura 2.3 – La pagina principale su tablet

aggiornato al primo accesso dopo l'autenticazione ed in corrispondenza della chiusura della funzionalità in uso, in modo da aggiornare periodicamente il conteggio in modo non bloccante senza compromettere l'usabilità dell'applicazione (l'aggiornamento richiede un'elaborazione lato server che richiede alcuni secondi).

Il cliente può personalizzare il comportamento della pagina nel momento dell'accesso dell'utente scegliendo tra queste possibilità:

- apertura automatica di una funzionalità tra quelle disponibili.
- apertura della scheda di gestione utente con obbligo di cambio password (ogni altra azione è inibita fino al termine dell'operazione).
- apertura della funzionalità Messaggistica in caso di presenza di messaggi da leggere definiti come "obbligatori" dal mittente.

Gestione utente

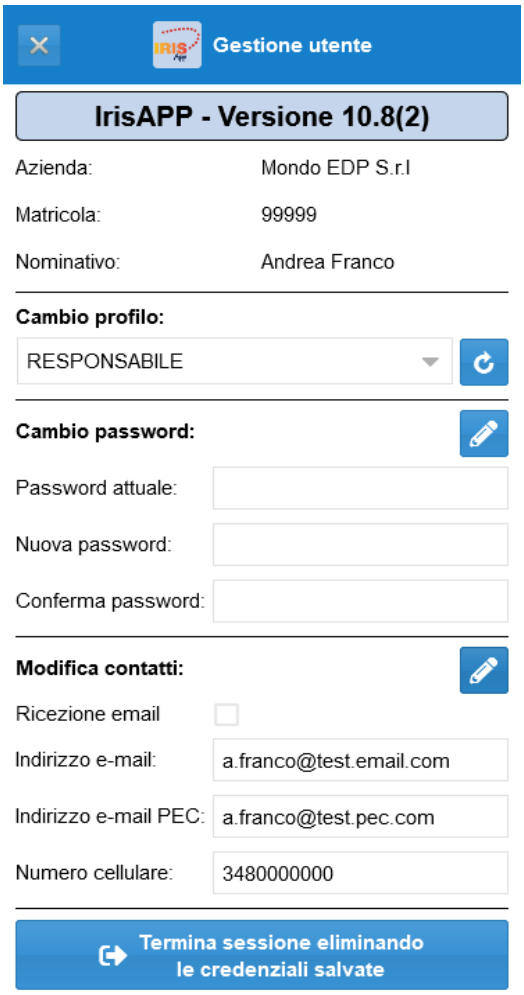


Figura 2.4 – La scheda Gestione utente

La scheda Gestione utente (Figura 2.4) consente di visualizzare alcuni dati associati all'utente autenticato e di eseguire le seguenti operazioni:

- Cambio profilo: permette la selezione di un nuovo profilo tra quelli disponibili e l'avvio del cambio di profilo, questa operazione implica il logout dall'applicazione ed il successivo login automatico con il profilo selezionato.
- Cambio password: permette di cambiare la password richiesta per l'accesso all'applicazione, richiede l'inserimento della password attuale ed il doppio inserimento della nuova password, la visibilità di questa sezione dipende dalla configurazione richiesta dal cliente e dalla modalità di autenticazione utilizzata.
- Modifica contatti: permette di modificare i contatti dell'utente (e-mail, PEC, numero di telefono) presenti nel sistema informativo Iris, la visibilità di questa sezione dipende dalla configurazione richiesta dal cliente.
- Termina sessione: se l'utente ha scelto il salvataggio delle credenziali nei cookies in fase di login, nella parte bassa della pagina è disponibile un pulsante da utilizzare per terminare la sessione corrente, rimuovendo al contempo le credenziali cifrate salvate nei cookies.

2.1.2 Funzionalità di interfaccia

Dati giornalieri

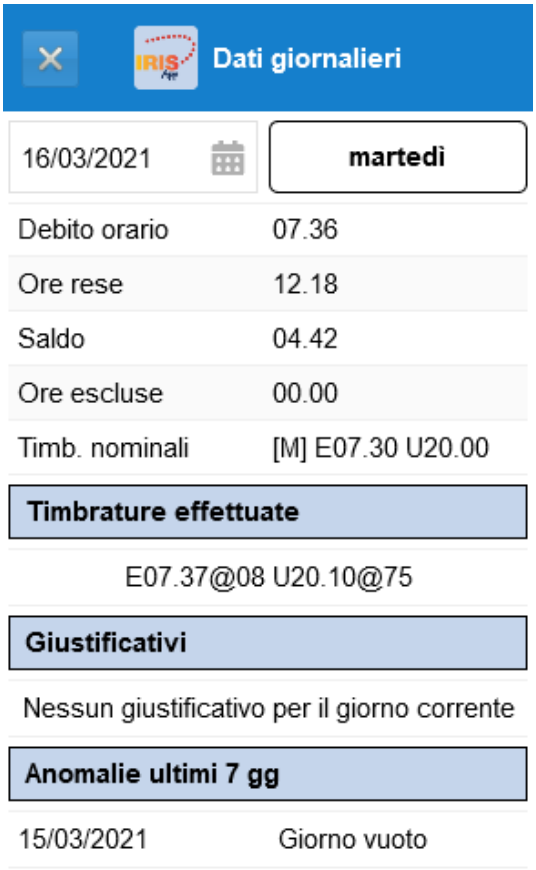


Figura 2.5 – La funzionalità Dati giornalieri

La funzionalità Dati Giornalieri (Figura 2.5) fornisce al dipendente una panoramica sui dati associati al giorno selezionato (di default il giorno corrente) visualizzando:

- Debito orario: il numero di ore giornaliere previste dall'orario del dipendente.
- Ore rese: il numero di ore effettivamente rese nella giornata selezionata.
- Saldo attuale: il saldo (positivo o negativo) delle ore rese rispetto al debito orario.
- Ore escluse: il numero di ore escluse dal conteggio precedente.
- Timbrature nominali: l'orario valido di timbratura del dipendente.
- Timbrature effettuate: l'elenco delle timbrature effettive del giorno selezionato.
- Giustificativi: l'elenco degli eventuali giustificativi di presenza o assenza validi nel giorno selezionato.
- Anomalie ultimi 7 gg: l'elenco delle eventuali anomalie nelle timbrature rilevate nei sette giorni precedenti al giorno selezionato.

La funzionalità è visualizzata di default dopo il login del profilo dipendente in modalità tablet.

Elenco dipendenti

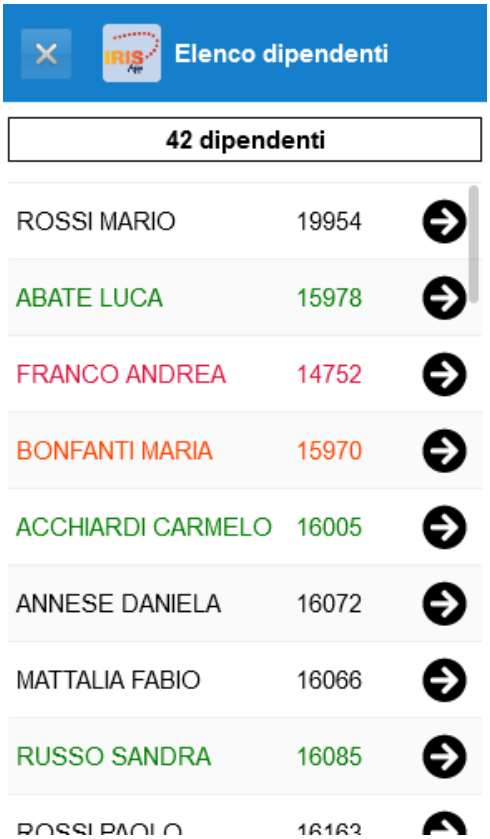


Figura 2.6 – La funzionalità Elenco dipendenti

Figura 2.7 – La scheda di dettaglio del dipendente

La funzionalità Elenco Dipendenti (Figura 2.6) consente al responsabile di visualizzare l'elenco dei dipendenti gestiti (cognome, nome e matricola) ed il loro stato attuale utilizzando diversi colori che evidenziano la presenza o l'assenza del personale:

- Verde: il dipendente è in servizio.
- Nero: il dipendente non è in servizio.
- Rosso: il dipendente non è in servizio ma ha un giustificativo di assenza in corso (ad esempio è in ferie).
- Arancione: il dipendente è in reperibilità.

L'elenco dei collaboratori visualizzato dipende dalla configurazione del profilo in uso definita dall'ufficio del personale, il loro stato viene aggiornato in base ai dati disponibili nel sistema informativo Iris al momento dell'accesso alla funzionalità.

Selezionando il singolo dipendente è possibile accedere ad una scheda di dettaglio (Figura 2.7) che consente al responsabile di visualizzare l'eventuale timbratura in corso, il giustificativo di assenza valido nella giornata, oltre ad un elenco di dati relativi al collaboratore (personalizzabile dal cliente) come ad esempio il suo ruolo, la squadra di appartenenza ed il numero di telefono.

La funzionalità è visualizzata di default dopo il login del profilo responsabile in modalità tablet.

Timbratrice virtuale mobile

La funzionalità Timbratura virtuale mobile (Figura 2.8) implementa un orologio di timbratura virtuale e consente al dipendente di inserire una nuova timbratura di entrata o di uscita. Nella parte alta della pagina è visualizzato l'orologio di riferimento con data ed ora campionati lato server all'accesso alla funzionalità. L'orario viene poi incrementato ogni secondo lato client e riaggiornato periodicamente con il valore del server per evitare disallineamenti. I pulsanti "Entrata" e "Uscita" sono utilizzati per l'inserimento della rispettiva timbratura, il cui orario di riferimento è comunque identificato lato server al momento della richiesta. Opzionalmente è consentita la selezione di una causale di timbratura tra quelle abilitate per il dipendente (per identificare ad esempio una timbratura legata ad uno straordinario). Dopo la pressione dei pulsanti viene visualizzato un messaggio di conferma che riporta l'orario effettivo di timbratura ed eventualmente le anomalie riscontrate (ad esempio l'inserimento di una timbratura che non rispetta la sequenza ingresso-uscita). Nella parte centrale della pagina è mostrato il rilevatore agganciato, nella parte bassa l'elenco delle timbrature del giorno.

Figura 2.8 – La funzionalità Timbratura virtuale mobile

Figura 1.9 – La mappa con un rilevatore virtuale geolocalizzato

Il cliente può utilizzare due modalità di timbratura, con o senza geolocalizzazione. La timbratura senza geolocalizzazione viene agganciata ad un rilevatore virtuale definito convenzionalmente ed è sempre consentita indipendentemente dalla posizione del dipendente. La timbratura con geolocalizzazione prevede la definizione di un numero arbitrario di rilevatori virtuali geolocalizzati da parte dell'ufficio del personale, ad ogni rilevatore sono associate le sue coordinate geografiche (latitudine e longitudine) ed un raggio di validità espresso in metri, fino ad

un massimo di 1 km. La timbratura sarà consentita se il dispositivo viene localizzato all'interno del cerchio avente come centro la posizione geografica di un rilevatore e come raggio il raggio di validità del rilevatore. Al momento dell'accesso alla funzionalità viene avviata la localizzazione del dispositivo e l'operazione viene ripetuta ogni 15 secondi; al termine di ogni localizzazione viene verificata la condizione definita in precedenza per l'abilitazione dei pulsanti di timbratura. Utilizzando il pulsante a destra del rilevatore è possibile visualizzare una mappa (Figura 2.9) con la posizione attuale del dispositivo, la posizione dei rilevatori vicini ed una rappresentazione grafica del raggio di validità, viene inoltre evidenziato il rilevatore eventualmente agganciato. Il cliente può decidere di consentire la timbratura anche in assenza di un rilevatore geolocalizzato agganciato, definendo un rilevatore convenzionale da utilizzare.

Nel rispetto della privacy dell'utente, la localizzazione del dispositivo è attiva soltanto durante l'utilizzo della funzionalità. Inoltre, la posizione viene utilizzata solamente per verificare la condizione di validità della timbratura descritta in precedenza, senza memorizzare alcuna informazione sulla posizione, ad esclusione dell'associazione tra l'utente ed il codice identificativo del rilevatore agganciato al momento dell'inserimento della timbratura.

Prima dell'inserimento della timbratura è possibile verificare delle condizioni bloccanti personalizzate su richiesta del cliente, come la presenza di uno specifico modello di scheda informativa compilata per il giorno di timbratura, utilizzato dai clienti, ad esempio, negli ultimi mesi come autocertificazione di negatività al Covid19.

Stampa cedolino

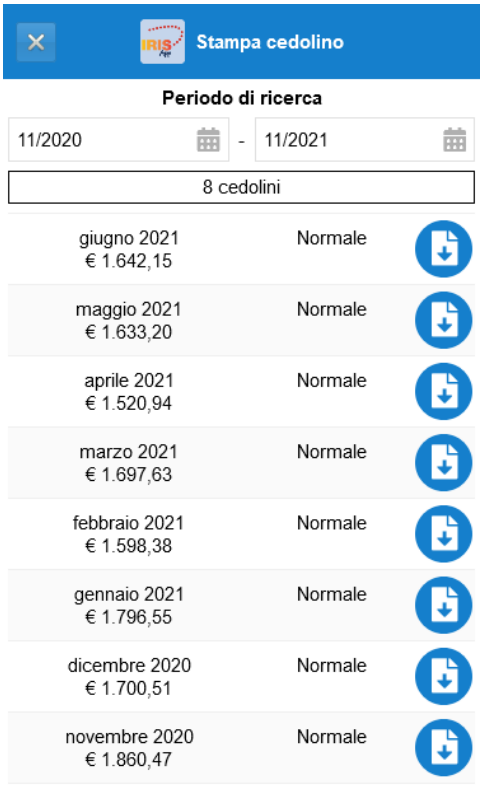


Figura 2.10 – La funzionalità Stampa cedolino

La funzionalità Stampa cedolino (Figura 2.10) consente al dipendente il download del cedolino paga in formato pdf. Nella parte alta della pagina è possibile impostare un filtro sul periodo di ricerca come intervallo di mesi visualizzati (di default è mostrato l'ultimo anno). Nella parte bassa viene visualizzata la lista dei cedolini disponibili nell'intervallo selezionato, per ogni cedolino viene mostrato il tipo (Normale, Tredicesima o Extra27), le date di riferimento ed il netto pagato. I dati contenuti nel documento vengono elaborati dall'ufficio stipendi dell'azienda cliente utilizzando l'applicativo Gestione Economica e diventano disponibili alla data stabilita dall'ufficio competente. Al primo download del cedolino da parte del dipendente può essere richiesta la conferma per la registrazione della data di lettura. Il numero di cedolini da leggere viene visualizzato sotto forma di notifica nella pagina principale. Il documento in formato pdf viene prodotto ad ogni richiesta di download con i dati consolidati presenti nel sistema informativo Iris e viene visualizzato in una nuova scheda del browser.

Stampa cartellino

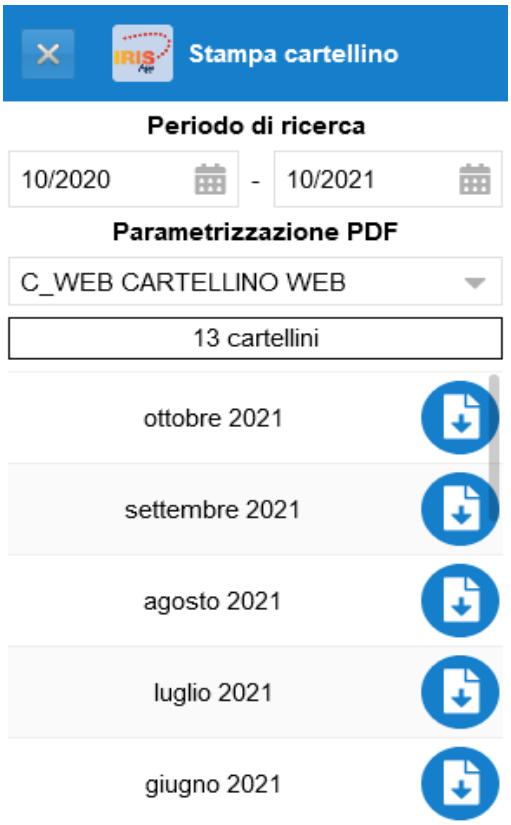


Figura 2.11 – La funzionalità Stampa cartellino

La funzionalità Stampa cartellino (Figura 2.11) consente al dipendente il download del cartellino mensile in formato pdf. Nella parte alta della pagina è possibile impostare un filtro sul periodo di ricerca come intervallo di mesi visualizzati (di default è mostrato l'ultimo anno). È inoltre possibile selezionare la parametrizzazione PDF (ovvero il layout) del documento da produrre tra quelle disponibili per l'utente autenticato. Nella parte bassa viene mostrata la lista dei documenti disponibili nell'intervallo temporale selezionato. I dati presenti nel cartellino vengono elaborati dall'ufficio del personale dell'azienda cliente utilizzando l'applicativo Rilevazione Presenze. Il cartellino mensile in formato pdf viene prodotto ad ogni richiesta di download con i dati aggiornati presenti nel sistema informativo Iris, selezionando la riga del mese richiesto viene avviata l'elaborazione lato server ed il documento viene visualizzato in una nuova scheda del browser.

Messaggistica

La funzionalità Messaggistica (Figura 2.12) consente al dipendente di visualizzare i messaggi di avviso ricevuti ed inviati. Il sistema informativo Iris consente all'ufficio del personale ed ai singoli responsabili l'invio massivo di messaggi di avviso in un formato simile alle e-mail (oggetto, testo ed eventuali allegati) tramite gli applicativi IrisWEB e Rilevazione Presenze. I destinatari possono essere tutti i dipendenti dell'azienda, sottoinsiemi definiti dei dipendenti o, nel caso dei responsabili, i collaboratori gestiti dal responsabile stesso. Il mittente può opzionalmente richiedere la lettura obbligatoria da parte dei destinatari. I messaggi gestiti possono assumere tre stati:

- Inviato: subito dopo l'invio del messaggio.
- Ricevuto: impostato dopo il primo accesso alla scheda di dettaglio del messaggio da parte del destinatario.
- Letto: impostato dopo la conferma esplicita di lettura da parte del destinatario nella scheda di dettaglio del messaggio.

La funzionalità consente di visualizzare i messaggi del dipendente suddivisi nelle schede "Messaggi ricevuti" e "Messaggi inviati". In entrambe le schede è presente un filtro a scomparsa

utile per filtrare i messaggi per stato, mittente e per ricercare una parola chiave nell'oggetto o nel testo del messaggio. La lista dei messaggi ricevuti è suddivisa in tre sezioni: vengono visualizzati prima i messaggi obbligatori da leggere, poi i messaggi non obbligatori da leggere e per ultimi i messaggi letti, ordinati per data di ricezione.

Figura 2.12 La scheda messaggi ricevuti della funzionalità Messaggistica

Figura 2.13 – La scheda di dettaglio di un messaggio

Selezionando il singolo messaggio ricevuto è possibile accede alla scheda di dettaglio (Figura 2.13) dove sono riportati il mittente, le date di invio, ricezione e lettura (con la possibilità di confermare la lettura del messaggio), l'oggetto, il testo e gli eventuali allegati scaricabili.

Per i messaggi inviati è inoltre possibile accedere ad un'ulteriore scheda che riporta la lista dei destinatari (Figura 2.14), con la possibilità di visualizzare lo stato di ricezione del messaggio per tutti gli utenti coinvolti, filtrabile per stato.

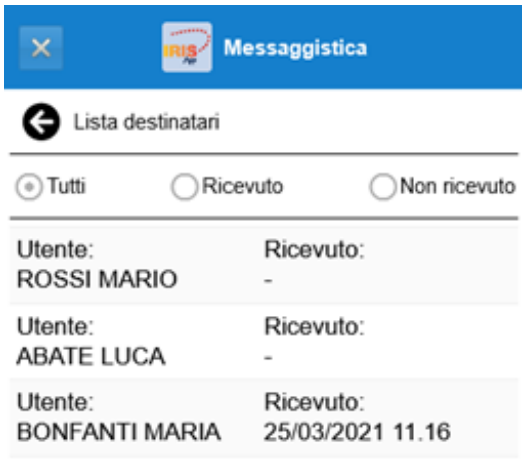


Figura 2.14 – La scheda dei destinatari del messaggio inviato

Gestione deleghe

La funzionalità Gestione deleghe (Figura 2.15) consente all'utente responsabile di delegare l'accesso alle funzionalità associate al proprio profilo ad un altro dipendente per un arco temporale definito. È utilizzata per assegnare il profilo di un responsabile assente ad un altro dipendente, consentendo anche in sua assenza la gestione delle richieste degli iter autorizzativi provenienti dai collaboratori gestiti.

La delega si concretizza con la creazione di un nuovo profilo sul dipendente delegato con un periodo di validità limitato, quest'ultimo potrà accedervi tramite la funzionalità di cambio profilo o indicandolo esplicitamente al momento del login. Durante l'intervallo di validità della delega l'utente delegante avrà comunque la possibilità di autorizzare le richieste.

Figura 2.15 – La funzionalità Gestione deleghe

Figura 2.16 – La scheda di inserimento di una nuova delega

Nella parte alta della pagina viene visualizzato il numero di deleghe attive ed il pulsante (+) che consente di accedere alla scheda di inserimento di una nuova delega, mentre nella parte bassa è presente la lista delle deleghe attive, ordinate per data. Selezionando la singola delega si accede alla scheda di modifica (analoga alla scheda di inserimento) in cui è possibile editare le caratteristiche di una delega esistente.

La scheda di inserimento di una nuova delega (Figura 2.16) consente la creazione della delega associata al profilo in uso specificando l'utente delegato (selezionabile tra gli utenti disponibili utilizzando un'apposita scheda), il nome da assegnare al nuovo profilo creato e l'intervallo temporale di validità della delega. Inoltre, è possibile indicare se l'utente delegato deve essere escluso dalla lista dei dipendenti gestiti per la durata delle delega, in modo non consentire l'autorizzazione delle proprie richieste al delegato stesso.

2.1.3 Iter autorizzativi

Gli iter autorizzativi sono una serie di funzionalità utilizzate dal personale per la gestione di richieste nell'ambito della rilevazione presenze, che verranno poi autorizzate o negate dal proprio responsabile. Ogni iter autorizzativo è formato quindi da una funzionalità denominata "richiesta", utilizzata dal dipendente per la consultazione, l'inserimento o la modifica delle richieste ed una corrispondente funzionalità denominata "autorizzazione", utilizzata dal responsabile per autorizzare o negare le richieste dei collaboratori gestiti. È possibile definire più livelli di autorizzazione, una richiesta autorizzata dal responsabile di primo livello passerà poi all'autorizzazione del secondo e successivamente agli eventuali livelli successivi. In base alla configurazione specifica dell'iter, definita dall'ufficio del personale utilizzando gli altri applicativi del sistema informativo Iris, le richieste potranno essere autorizzate in modo automatico in base al soddisfacimento di determinati requisiti, senza necessità di intervento da parte del responsabile.

Il dettaglio della configurazione degli iter autorizzativi gestita dall'ufficio del personale sarà approfondito in seguito.

La struttura di base dell'interfaccia grafica delle funzionalità è simile sia per i diversi iter autorizzativi, sia per le corrispondenti versioni richiesta ed autorizzazione ed è composta da una serie di schede:

- Scheda principale (Figura 2.17): visualizzata con l'accesso alla funzionalità, contiene per la versione "richiesta" la lista delle richieste effettuate e per la versione "autorizzazione" la lista delle richieste da gestire. In entrambi i casi le richieste sono suddivise in "Da autorizzare", "Autorizzate" e "Negate", le ultime due tipologie possono essere filtrate per data utilizzando i campi presenti nella parte alta della scheda. Nella versione "richiesta" è presente il pulsante (+) che consente al dipendente di accedere alla scheda di inserimento di una nuova richiesta. La versione "autorizzazione", solo per le richieste da autorizzare, presenta nella parte bassa due pulsanti a scomparsa che consento di autorizzare o negare tutte le richieste visualizzate, suddivise per utente richiedente. In entrambe le versioni selezionando la singola richiesta si accede alla scheda di dettaglio.
- Scheda di dettaglio (Figura 2.18): riporta un elenco di dati associati alla richiesta specifici per l'iter in esame. I pulsanti nella parte bassa della scheda consentono, nella versione "richiesta", di modificare o richiedere la revoca della richiesta visualizzata, nella versione "autorizzazione", consentono invece al responsabile di autorizzare o negare la richiesta. Nel caso la richiesta venga negata è obbligatorio l'inserimento di un testo nella rispettiva scheda note, ad esempio con la motivazione del responsabile. In entrambe le versioni sono

presenti i collegamenti che consentono l'accesso alle schede allegati e note della richiesta in esame.

Figura 2.17 – Un esempio di scheda principale

Figura 2.18 – Un esempio di scheda di dettaglio

- Scheda note (Figura 2.19): sia il dipendente richiedente, sia gli autorizzatori sui vari livelli gestiti, hanno la possibilità di inserire delle annotazioni libere e visibili a tutti gli utenti convolti, anche dopo l'inserimento della richiesta o dopo l'autorizzazione del proprio livello. La scheda è utile, inoltre, per visualizzare lo stato della richiesta, riportando l'esito delle autorizzazioni su tutti i livelli previsti, oltre al nominativo del responsabile con competenza sul singolo livello.
- Scheda allegati (Figura 2.20): la scheda allegati consente, nella versione "richiesta", l'inserimento e il download di eventuali documenti allegati dopo la creazione della richiesta (se previsto dalla configurazione dell'iter), nella versione "autorizzazione" è consentito solamente il download dei documenti allegati dal collaboratore. In base al tipo di dispositivo utilizzato possono essere caricati documenti e foto salvati in memoria o acquisiti in tempo reale utilizzando direttamente la fotocamera del dispositivo.
  Nella configurazione dell'iter è possibile definire il numero massimo e la dimensione massima degli allegati caricati.

Vengono in seguito descritti gli iter autorizzativi implementati nell'applicazione IrisAPP, mostrando principalmente le differenze rispetto alla struttura di base dell'interfaccia grafica

comune già descritta. Questa si differenzia maggiormente nelle versioni "richiesta", includendo una scheda di inserimento o modifica della richiesta, mentre le versioni "autorizzazione" non aggiungono funzionalità significative e quelle già descritte nella struttura comune.

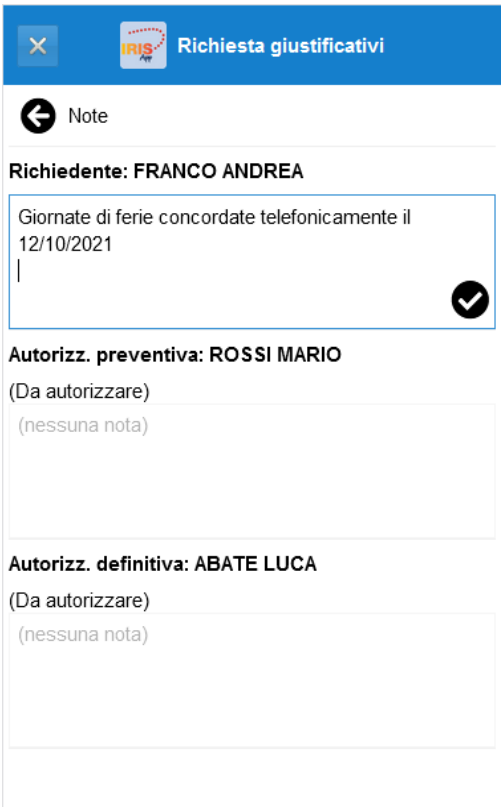


Figura 2.19 – Un esempio di scheda note

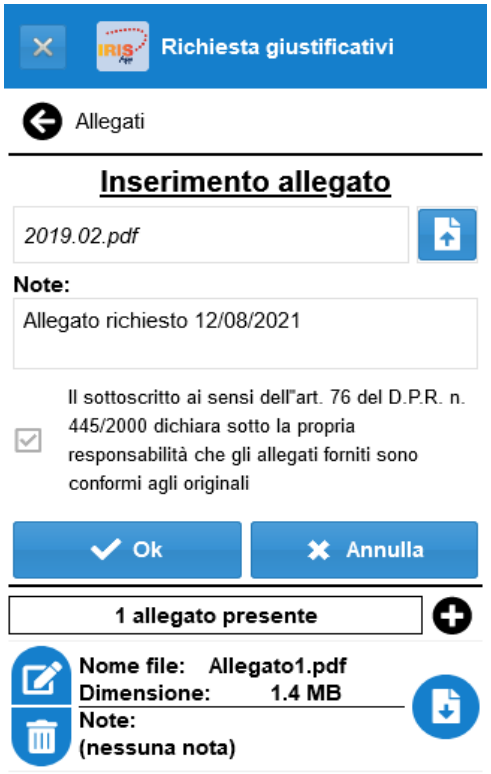


Figura 2.20 – Un esempio di scheda allegati

Richiesta/Autorizzazione giustificativi

Le due funzionalità consentono l'inserimento e la conseguente autorizzazione di una richiesta di giustificativo che potrà essere di assenza (come ferie e permessi) o di presenza (come uno straordinario).

La scheda di inserimento della versione "richiesta" (Figura 2.21) prevede la compilazione di una serie di campi che variano in base alla tipologia di giustificativo selezionato:

- Accorpamento causali: consente di selezionare un accorpamento di causali che riduce il numero di elementi visibili nel campo successivo.
- Causale: la causale di presenza o di assenza a cui la richiesta in inserimento afferisce, identificata da un codice e da una descrizione.
- Tipo fruizione: il tipo di fruizione previsto per la causale selezionata, potrà essere a giornata intera, a mezza giornata o ad ore.

- Periodo: in base al tipo fruizione selezionato viene richiesto l'inserimento dell'intervallo temporale di validità del giustificativo.
- Note richiesta: è possibile inserire delle note libere associate alla richiesta visibili dagli autorizzatori.
- Motivazione: è possibile selezionare una motivazione tra quelle presenti da un elenco precompilato.

Nella scheda di dettaglio della versione "richiesta" è consentita l'eliminazione di una richiesta non ancora approvata o la revoca della richiesta già approvata dal responsabile, quest'ultima si concretizza con l'inserimento automatico di una richiesta di revoca che andrà a sua volta autorizzata. È consentito l'inserimento di allegati alla richiesta che in base alla configurazione del giustificativo potranno anche essere contrassegnati come obbligatori (ad esempio per una richiesta di permesso per esame universitario che richiede il certificato di presenza rilasciato dall'università).

In entrambe le versioni la scheda di dettaglio (Figura 2.18, riga "Competenze") permette inoltre di accedere ad una scheda ulteriore che riporta le competenze totali, fruite e residue per il giustificativo della richiesta in esame.

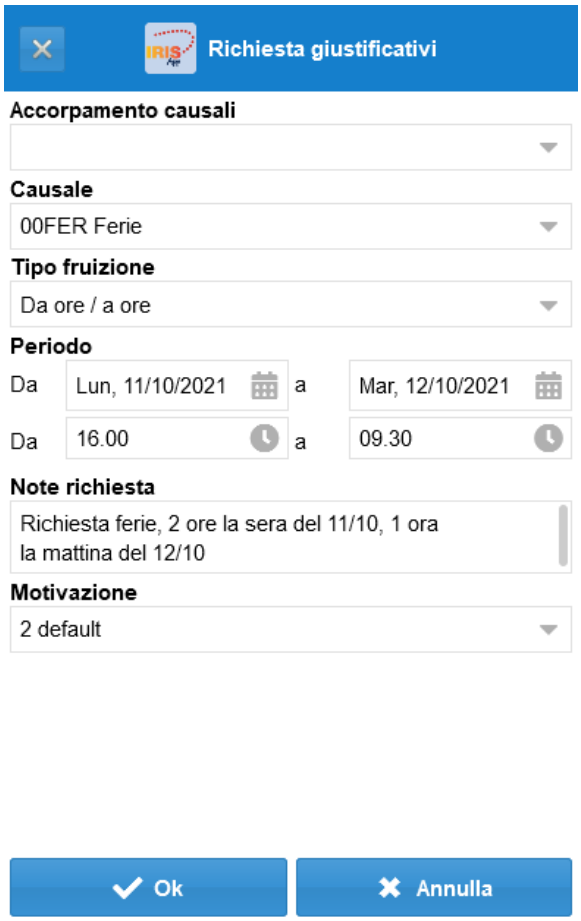


Figura 2.21 – La scheda di inserimento di una richiesta di giustificativo

Richiesta/Autorizzazione timbratura

Le due funzionalità consentono l'inserimento e la conseguente autorizzazione di una richiesta di timbratura, che potrà essere una richiesta di inserimento di una nuova timbratura (ad esempio per colmare una dimenticanza del dipendente) o una richiesta di modifica di una timbratura esistente.

Dalla scheda principale il pulsante (+) consente l'accesso ad una prima scheda (Figura 2.22) in cui vengono mostrate tutte le timbrature del giorno (di default il giorno corrente). Selezionando una timbratura si accede alla scheda di modifica, mentre con il pulsante "Nuova timbratura" si accede alla scheda di inserimento (Figura 2.23). Sia la scheda di inserimento che di modifica (dal layout identico) consentono l'inserimento dei seguenti campi:

- Data e ora della timbratura da inserire/modificare.
- Verso di timbratura (entrata/uscita).
- Causale: l'eventuale causale di timbratura associata alla richiesta.
- Rilevatore: l'eventuale rilevatore fisico o virtuale associato alla timbratura.
- Motivazione: è possibile selezionare una motivazione tra quelle presenti in un elenco precompilato.
- Note richiesta: è possibile inserire delle note libere associate alla richiesta visibili dagli autorizzatori.

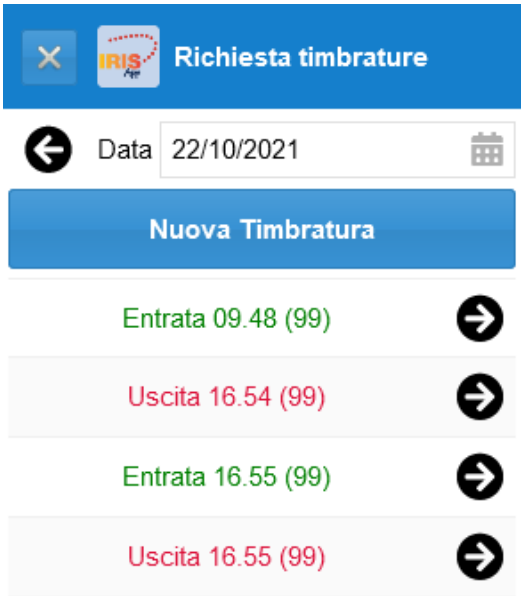


Figura 2.22 – Le timbrature modificabili del giorno selezionato

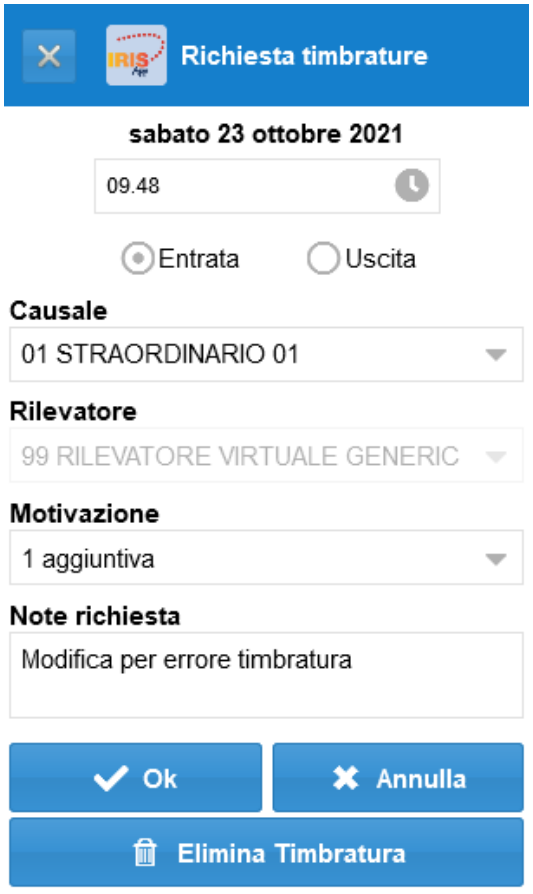


Figura 2.23 – La scheda di inserimento di una richiesta di timbratura

Richiesta/Autorizzazione cambio orario

Le due funzionalità consentono l'inserimento e la conseguente autorizzazione di una richiesta di cambio orario.

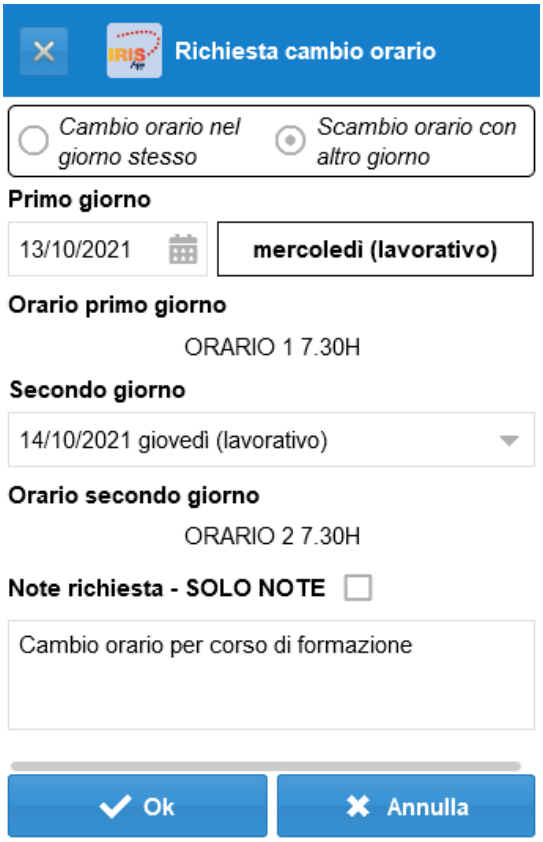


Figura 2.24 – La scheda di inserimento di una richiesta di cambio orario

Nella scheda di inserimento (Figura 2.24) sono disponibili due modalità:

- Cambio orario nel giorno stesso: consente di richiedere un nuovo piano orario per il medesimo giorno indicato.
- Scambio orario con altro giorno: consente di richiedere lo spostamento in uno dei tre giorni successivi mantenendo lo stesso piano orario.

In entrambe le modalità è possibile inserire la richiesta con soltanto il campo note compilato visibile dal responsabile in fase di autorizzazione.

La scheda di dettaglio visibile nella versione "autorizzazione" evidenzia eventuali incongruenze nella richiesta avanzata dal dipendente, come la selezione di un orario non più aggiornato a seguito di modifiche intercorse successivamente all'inserimento della richiesta da parte del dipendente.

Compilazione/Validazione scheda informativa

Le due funzionalità consentono la gestione delle schede informative di cui è richiesta la compilazione da parte del dipendente e la conseguente validazione da parte del responsabile (in questo specifico caso di iter autorizzativo sono adottati i termini compilazione/validazione invece di richiesta/autorizzazione).

La scheda informativa rappresenta un documento (ad esempio un'autocertificazione richiesta al personale) composto a partire da un modello testuale predisposto dall'ufficio del personale utilizzando il modulo Rilevazione Presenze. L'ufficio definisce il layout e prevede al suo interno una serie di campi su cui è richiesta la compilazione obbligatoria o opzionale da parte del dipendente. I campi editabili possono contenere valori numerici, date, liste di valori predefiniti o testo libero in base alle necessità del cliente. Il valore assegnato ai campi compilati dal richiedente viene associato alla richiesta di validazione e può determinare l'accettabilità dell'inserimento della richiesta.

Le schede di inserimento (Figura 2.25) e di modifica (dal layout identico) richiedono la compilazione di alcuni campi:

- Stato: la richiesta può essere salvata come temporanea, non ancora visibile al responsabile ma in attesa di conferma successiva, o come definitiva.
- Validità: consente di specificare la validità temporale della scheda informativa compilata.
- Modello: consente di selezionare il modello di scheda da compilare, sotto il campo di selezione viene visualizzato il titolo del modello selezionato.
- Note al documento: consente di inserire delle annotazioni libere nella scheda compilata.

Il pulsante "Compila scheda" consente l'accesso alla scheda di compilazione del modello selezionato (Figura 2.26). Se tutti i campi obbligatori del modello (contrassegnati dal simbolo (*)) sono compilati, è possibile confermare la richiesta e tornare alla scheda di inserimento/modifica. Accedendo in modifica su una richiesta di scheda informativa salvata è possibile modificarne lo stato ed i valori inseriti nei campi editabili della scheda. L'ufficio del personale, configurando i modelli disponibili, può definire delle limitazioni sul numero di schede che possono essere inserite per modello (ad esempio una al giorno/settimana/mese).

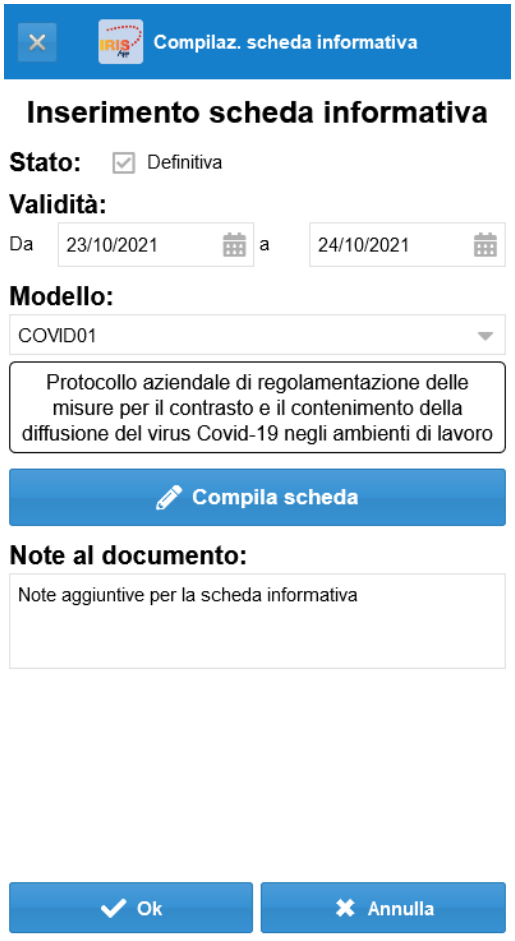


Figura 2.25 – La scheda di inserimento di una richiesta di scheda informativa

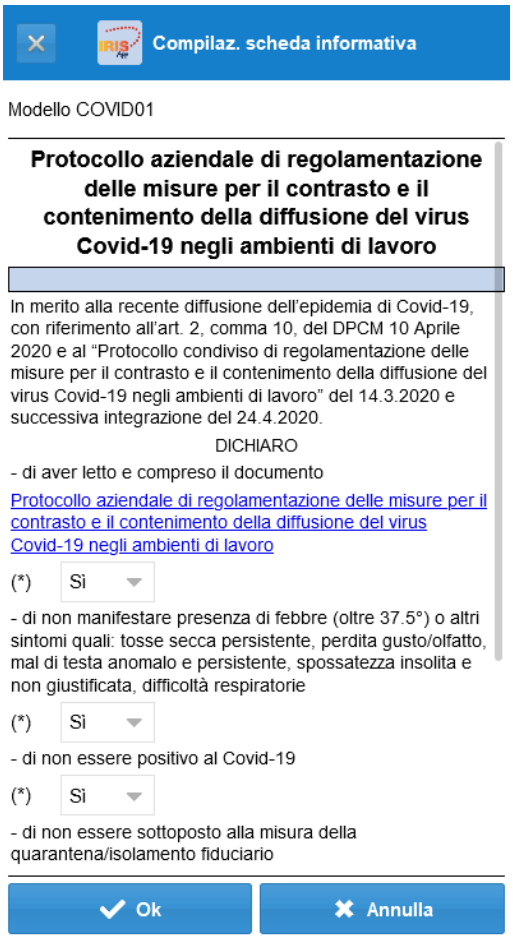


Figura 2.26 – La scheda di compilazione della scheda informativa

2.2 Interazione con gli altri applicativi aziendali

L'applicazione sviluppata presenta una forte interazione con gli altri applicativi del sistema informativo Iris, condividendo lo stesso database Oracle accede infatti agli stessi dati manipolati dal gestionale. La configurazione dell'applicazione può essere gestita dall'azienda cliente tramite i due applicativi IrisWIN e IrisCloud (con il supporto del servizio di assistenza della Mondo Edp). In particolare, è possibile la gestione degli account utente, che consentono di implementare le funzionalità di autenticazione e autorizzazione, ed è possibile la gestione della configurazione degli iter autorizzativi. Queste configurazioni sono in parte condivise con l'applicativo IrisWEB che offre funzionalità simili ma rivolte ad un utilizzo su pc desktop.

2.2.1 Gestione degli utenti

Per ogni dipendente registrato nel gestionale (identificato da una matricola), è possibile definire uno o più utenti web utilizzati per l'accesso ad entrambe le web-app IrisAPP ed IrisWEB. Ogni utente web è caratterizzato dalla matricola associata al dipendente, un nome utente, una password (presente soltanto in caso di login interno), alcuni contatti (come numero di cellulare, e-mail e PEC) ed una lista di profili disponibili. Ogni profilo definisce una configurazione degli applicativi web specifica per l'utente utilizzato. I profili possono avere un nome arbitrario ma possono essere suddivisi concettualmente in due categorie:

- Dipendente: ha visibilità solamente sui dati associati alla propria matricola e può utilizzare le funzionalità di richiesta degli iter autorizzativi.
- Responsabile: ha visibilità sui dati associati all'insieme di dipendenti gestiti e può utilizzare le funzionalità di autorizzazione degli iter autorizzativi.

Ad ogni profilo creato è poi associata una configurazione definita specificando una serie di parametri:

- Permessi: consente di creare delle configurazioni specifiche sulle funzionalità degli applicativi del sistema informativo Iris, definendo una serie di parametri che si riferiscono alle modalità operative di alcune importanti funzioni, al fine di limitare le possibilità di intervento degli utenti sulle stesse.
- Filtro anagrafe: definisce la regola (in formato sql) da applicare per l'identificazione dei dipendenti gestiti dal profilo responsabile, non valorizzato per il profilo dipendente.
- Filtro funzioni: definisce l'abilitazione delle funzionalità a cui l'utente web può accedere, per ogni funzionalità può assumere tre valori, R: sola lettura, S: abilitato, N: non abilitato.
- Iter autorizzativo: definisce la configurazione degli iter autorizzativi da adottare (meglio descritta in seguito).
- Filtro dizionario: permette di specificare dei filtri sull'accesso alle tabelle del dizionario dati, ad esempio, consente di limitare le causali di assenza visibili o le parametrizzazioni del cartellino mensile utilizzabili per la stampa.

Per ogni parametro descritto possono essere preventivamente definite una serie di configurazioni salvate ed abbinate ad un identificativo da associare ad un numero arbitrario di profili, in modo da ridurre il tempo richiesto e le possibilità di errore nella creazione dei profili.

2.2.2 Configurazione degli iter autorizzativi

La configurazione degli iter autorizzativi prevede la definizione di regole a livello aziendale e di regole personalizzate a livello di dipendente, mediante il parametro "Iter autorizzativo" definito nella configurazione del profilo descritta in precedenza.

Le regole aziendali permettono di specificare, per ogni iter, la revocabilità delle richieste da parte del dipendente, una lista di strutture, l'oggetto ed il corpo della mail inviata in automatico al richiedente e/o all'autorizzatore. L'invio della mail può essere utilizzato per segnalare le operazioni significative eseguite sulla richiesta dal personale coinvolto.

Per ogni struttura, identificata da un codice e una descrizione, è possibile definire:

- Condizione di riconoscimento: in formato sql, permette di identificare automaticamente a quale struttura dell'iter autorizzativo deve essere ricondotta la richiesta in inserimento.
- Condizione di autorizzazione automatica: in formato sql, è utilizzata per valutare l'eventuale autorizzazione automatica della richiesta al momento dell'inserimento.
- Condizione di validità: in formato sql, viene valutata al momento dell'inserimento della richiesta e può bloccare l'inserimento della stessa.
- Presenza, obbligatorietà e modificabilità degli allegati.
- Modificabilità delle note: indica il livello oltre il quale, se esiste l'autorizzazione, il dipendente non può più modificare le note della richiesta.
- Livelli di autorizzazione: a partire dall'inserimento di una richiesta l'iter richiederà l'autorizzazione su tutti i livelli definiti per la struttura identificata dalla condizione di riconoscimento. Per ogni livello, identificato da un numero progressivo e da una descrizione, è possibile definire l'obbligatorietà del livello ai fini dell'iter, una condizione di dettaglio per l'autorizzazione automatica (in formato sql), la visibilità degli allegati all'autorizzatore corrente, l'obbligatorietà degli allegati ai fini dell'autorizzazione del singolo livello ed uno script sql personalizzato da eseguire dopo l'azione dell'autorizzatore sul livello.

Le regole personalizzate permettono di definire, tramite la configurazione del profilo, a quali iter autorizzativi ha accesso il dipendente, in quali strutture possono rientrare le sue richieste ed a quali livelli ha accesso (l'accesso al singolo livello potrà essere negato, in sola lettura o legato dal filtro funzioni assegnato al profilo che limita l'accesso alla singola funzionalità).

Tipicamente un profilo dipendente non ha accesso ai livelli di autorizzazione, ma gli è consentito solamente l'inserimento della richiesta tramite l'abilitazione delle funzionalità del filtro funzioni, i profili autorizzatore hanno invece accesso ai livelli di autorizzazione delle richieste.

2.3 Autenticazione dell'utente

L'applicazione sviluppata richiede diverse modalità di autenticazione dell'utente per adattarsi alle necessità ed ai sistemi preesistenti delle aziende clienti. Le modalità di autenticazione disponibili si possono suddividere in due categorie:

- Autenticazione interna: utilizza le credenziali gestite dal sistema informativo Iris attraverso la definizione di account utente legati al dipendente registrato nel gestionale ed ai profili disponibili, la gestione è solitamente affidata all'ufficio del personale o ai servizi informativi del cliente tramite gli applicativi IrisWIN o IrisCloud.
- Autenticazione esterna: implementa dei protocolli di autenticazione che richiedono l'interazione con dei sistemi esterni al sistema informativo Iris. In questo caso vengono comunque creati degli utenti tramite IrisWIN o IrisCloud che vengono poi utilizzati per l'acceso all'applicazione previa autorizzazione proveniente dagli authentication server esterni.

In particolare, le modalità di autenticazione esterna disponibili sono le seguenti:

2.3.1 Autenticazione su dominio Active Directory

Active Directory è la tecnologia di riferimento utilizzata in aziende di varia dimensione per la gestione di un domino, ovvero di un raggruppamento di computer che eseguono il sistema operativo Microsoft Windows. Si basa sulla presenza di un server definito "controller di dominio", che gestisce tutti gli aspetti legati la sicurezza della rete definita dai computer appartenenti al raggruppamento, centralizzando l'amministrazione e limitando l'accesso alle risorse disponibili. I computer appartenenti al dominio posso essere sulla stessa rete LAN o essere dislocati anche molto distanti tra loro attraverso una rete WAN o collegati da una VPN.

Active directory è caratterizzato dalla presenza di una struttura gerarchica basata su oggetti di cui viene gestita l'organizzazione e la sicurezza. Gli oggetti gestiti possono essere utenti, gruppi di utenti, risorse (computer o periferiche) o servizi offerti e possono a loro volta essere definiti come contenitori di oggetti. Gli oggetti presenti in un dominio possono poi essere suddivisi in unità organizzative che consentono di definire una suddivisione gerarchica (solitamente basata sulla posizione geografica degli oggetti contenuti) che permette di utilizzare politiche di gestione dei gruppi di utenti (denominante Group Policy Objects) comuni a tutti gli oggetti contenuti in un'unità organizzativa. A loro volta i domini possono essere ulteriormente organizzati in una struttura basata sui concetti di albero (raggruppamenti di domini che condividono uno spazio dei nomi contiguo) e foreste (raggruppamenti di alberi), gestiti utilizzando diversi livelli di controller di dominio, fino ad arrivare al livello più alto denominato Primary Domain Controller.

Per l'accesso alle funzionalità offerte da Active Directory (tra cui la possibilità di autenticare un utente appartenente ad uno specifico dominio) è necessario utilizzare le Active Directory Service Interfaces (ADSI) che consentono di accedere alle risorse utilizzando la piattaforma SDK (software development kit) basata su programmazione COM (Component Object Model) e messa a disposizione degli sviluppatori per la creazione di client ADSI eseguibili in ambiente Windows

(che non deve necessariamente essere una versione windows server su cui deve invece essere installato Active Directory).

L'interfaccia ADSI è accessibile al programmatore utilizzando la libreria dinamica "activeds.dll", che consente l'autenticazione dell'utente richiedendo l'accesso all'oggetto utente corrispondente. L'operazione richiede di specificare il dominio, lo username e la password attraverso l'operazione di associazione, che permette l'identificazione dell'oggetto utilizzando la sua stringa di associazione specifica per il provider di servizi ADSI supportato da Active Directory. L'applicazione IrisAPP supporta i provider di servizi denominati LDAP e WINNT, che utilizzano rispettivamente le stringhe di associazione nel formato "LDAP://nome_dominio" e "WinNT://nome_domino".

2.3.2 Autenticazione NTLM nell'architettura SSPI

NTML (NT LAN Manager) è un protocollo di sicurezza che consente l'autenticazione degli utenti in ambiente Microsoft Windows, anche in assenza di un dominio Active Directory, ma in presenza di un workgroup tra computer sulla stessa rete. Si tratta di un protocollo a sfida simmetrica, che richiede lo scambio di tre messaggi tra client e server senza inviare esplicitamente la password. Il primo messaggio è inviato dal client al server ed è utilizzato per avviare l'operazione indicando le proprie capacità di connessione, il secondo messaggio è inviato dal server al client e contiene la sfida richiesta per l'autenticazione dell'utente ed il terzo è inviato dal client al server e contiene la risposta alla sfida simmetrica proposta.

Il protocollo descritto si colloca in un'architettura più ampia denominata Security Support Provider Interface (SSPI – Figura 2.27) che costituisce la base per l'autenticazione degli utenti di Microsoft Windows. Utilizzando un'architettura a livelli consente ad un'applicazione client di utilizzare le funzionalità di autenticazione fornite dai Security Support Provider (SSP) attraverso l'interfaccia fornita dal SSPI layer. Nel caso specifico di NTML il corrispondente SSP è accessibile tramite la libreria dinamica "Secur32.dll".

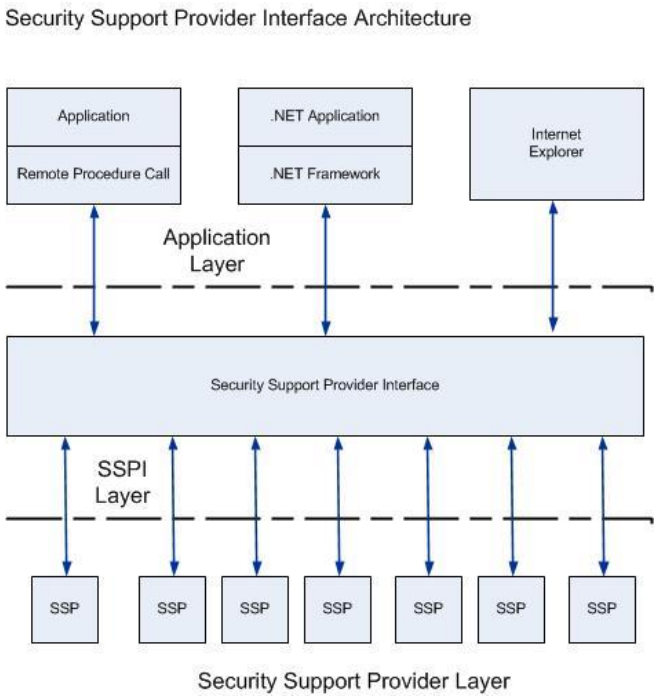


Figura 2.27 – L'architettura SSPI che consente alle applicazioni l'accesso al Security Support Provider (SSP)

2.3.3 Autenticazione OAuth 2.0

OAuth 2.0 è un protocollo di autenticazione descritto nello standard IETF RFC 6749 che permette l'autenticazione di una parte senza la necessità di conservare le password sul server che ospita la risorsa a cui l'entità vuole accedere, basandosi invece su un authorization server esterno considerato sicuro.

Il protocollo individua quattro ruoli nel processo di autenticazione:

- Resource owner: è l'entità che richiede l'accesso alla risorsa protetta, che può essere sia una persona fisica, sia un software.
- Resource server: è il server che ospita la risorsa protetta e che può consentire l'accesso alla stessa a fronte della presentazione di un access token valido.
- Client: è l'applicazione che avvia l'autenticazione con l'authorization server e richiede l'accesso alla risorsa protetta al resource server per conto del resource owner.
- Authorization server: è il server che autentica il resource owner con i dati ricevuti dal client e rilascia l'access token necessario per accedere alla risorsa protetta.

Questi 4 ruoli (che non sono sempre necessariamente distinti) si scambiano tre tipologie di informazioni nel corso del protocollo:

- Authorization grant: rappresenta il segreto condiviso utilizzato dall'authorization server per autenticare il resource owner.
- Access token: è una stringa di caratteri rilasciata dall'authorization server al client (con scadenza definita) ed utilizzata dal resource server per identificare il resource owner che richiede l'accesso alla risorsa protetta. Il contenuto dell'access token non è conosciuto dal client che funge solamente da tramite per le parti coinvolte.
- Refresh token: è una stringa di caratteri rilasciata dall'authorization server al client (con scadenza definita maggiore rispetto all'access token) ed utilizzata da quest'ultimo per rinegoziare un nuovo access token senza che sia necessario l'intervento del resource owner.

La comunicazione tra le parti coinvolte è condotta utilizzando il protocollo http e richiede lo scambio di una serie di messaggi (Figura 2.28):

1) Il client richiede al resource owner un authorization grant (tipicamente username e password) necessario per l'accesso alla risorsa protetta.
2) Il resource owner fornisce le informazioni richieste al client stesso (se considerato fidato) o direttamente all'authorization server.
3) Il client invia all'authorization server l'authorization grant ricevuto che viene utilizzato da quest'ultimo per autenticare il resource owner.
4) Se l'autenticazione ha esito positivo, l'authorization server risponde al client con l'access token generato ed eventualmente il refresh token se utilizzato.

5) Il client invia l'access token al resource server che ne verifica la validità e consente l'accesso alla risorsa protetta al resource owner tramite il client.

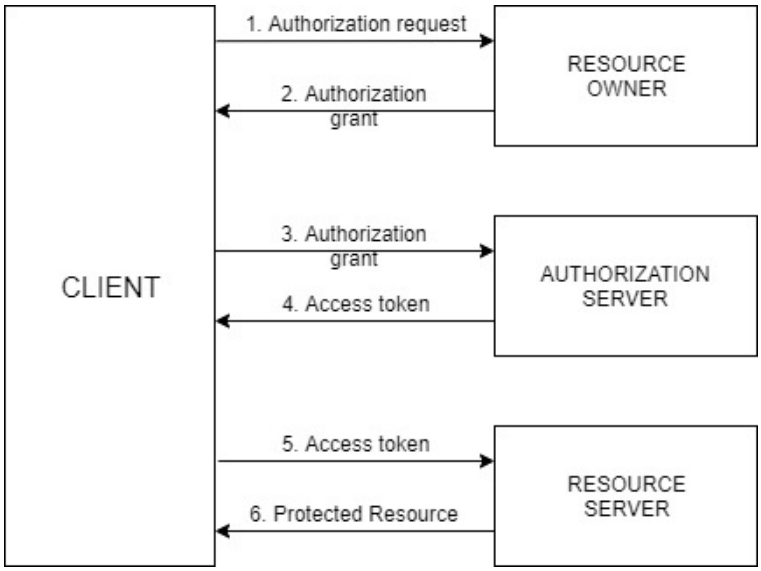


Figura 2.28 – Ruoli coinvolti e messaggi scambiati dal protocollo OAuth 2.0

Prima di eseguire il protocollo di autenticazione precedentemente descritto, il resource owner ed il client si devono registrare sull'authorization server. Il primo per ottenere le credenziali associate ai dati dell'utente (authorization grant) ed il secondo per ottenere le credenziali utilizzate dall'authorization server per autenticare il client al momento della richiesta dell'access token.

Nel contesto specifico dell'applicazione IrisAPP, l'azienda cliente che ha richiesto questa modalità di autenticazione ha imposto alcuni vincoli sull'implementazione del protocollo di autenticazione che hanno portato allo sviluppo di una soluzione strutturata nel seguente modo:

- Il resource owner è rappresentato dall'utente che tramite il browser sul dispositivo mobile vuole accedere all'applicazione IrisAPP.
- Il client è rappresentato dalla web-app stessa, che tramite la form di login visualizzata nel browser richiede all'utente lo username e la password (authorization grant) richiesti dall'authorization server aziendale per l'autenticazione.
- La verifica della validità dell'access token viene fatta dall'authorization server, che fornisce all'applicazione un identificativo utilizzato per associare all'utente autenticato il dipendente gestito dal sistema informativo Iris (resource server).
- Non è richiesto l'utilizzo del refresh token, dopo la scadenza dell'access token è richiesto all'utente nuovamente l'inserimento delle credenziali nel form di login dell'applicazione.

Dal punto di vista dell'applicazione il processo di autenticazione può essere quindi suddiviso concettualmente in due fasi:

- FASE 1: vengono richiesti all'utente username e password che vengono poi inviate all'authorization server aziendale (previa autenticazione dell'applicazione) che risponde

con l'access token. Questo viene memorizzato nei cookies scambiati tra browser e web server.

- FASE 2: l'access token viene inviato all'authorization server che lo valida e risponde con l'identificativo associato nel sistema informativo Iris al dipendente che esegue l'accesso.

Dopo aver concluso con successo la prima fase, fino alla scadenza del token, l'utente potrà accedere all'applicazione senza utilizzare le credenziali. L'access token contenuto nei cookies viene infatti utilizzato lato server per autenticare l'utente eseguendo solamente la fase 2, verrà quindi visualizzata direttamente la pagina principale senza passare dalla pagina di login.

La soluzione sviluppata prevede comunque una struttura sufficientemente flessibile da poter essere applicata in contesti che richiedono un'implementazione del protocollo standard con vincoli diversi rispetto a quelli del cliente specifico in esame.

2.4 Il database Oracle

La persistenza dei dati nell'applicazione è gestita utilizzando un database Oracle comune a tutti gli applicativi aziendali. Nel database relazionale a disposizione erano già presenti tutte le strutture dati necessarie per l'esecuzione dell'applicazione, non è quindi stato necessario creare nuove tabelle o in generale apportare modifiche alla struttura del database preesistente.

L'architettura di un server Oracle è rappresentata da due elementi principali, il database, formato da tutti i file fisici in cui vengono memorizzati i dati, e l'istanza, formata dall'insieme dei processi e dello spazio di memoria necessari per l'accesso ai dati. Il database è costituito da strutture fisiche di memorizzazione che contengono una scala gerarchica di strutture logiche (Figura 2.29), le prime sono dipendenti dall'hardware e dal sistema operativo utilizzato, mentre le seconde sono indipendenti e forniscono quindi sempre la stessa interfaccia per i servizi offerti.

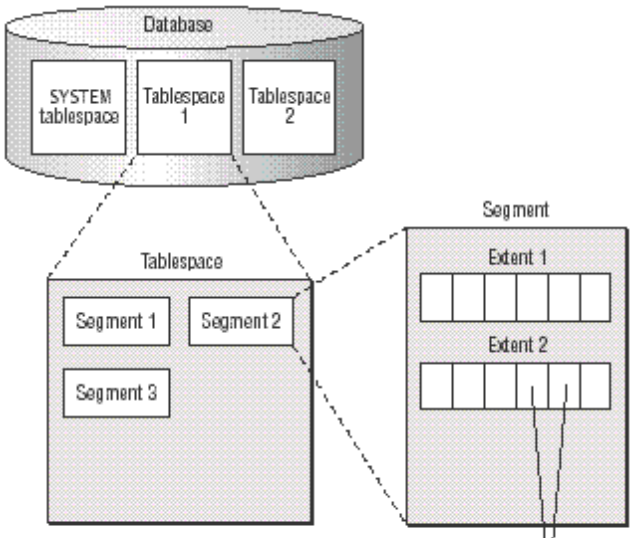


Figura 2.29 – La struttura gerarchica delle strutture logiche di un database Oracle

La struttura logica in cima alla scala gerarchica è il tablespace, la creazione di più tablespace consente una separazione logica dei dati presenti nel database, consentendo anche di diversificare la loro amministrazione. L'architettura gerarchica prevede poi che all'interno del tablespace i dati siano organizzati in segment, ogni volta che si crea una tabella o un indice a questo corrisponderà uno di questi elementi. Al loro interno troveremo in insieme di extent, formati da una sequenza di block che costituiscono la più piccola unità di memorizzazione di Oracle. Un database contiene sempre un tablespace di sistema denominato SYSTEM.

Le strutture fisiche che contengono i dati sono invece i data file, ad ogni tablespace corrisponde almeno un data file, mentre la cronologia delle modifiche apportate ai dati stessi è contenuta nei redo log file, molto importanti per la gestione dell'integrità del database che ne conterrà almeno due utilizzanti in modo circolare. Ogni database contiene almeno un control file, all'interno del quale sono registrate le informazioni riguardanti ciascun data file e redo log file disponibile.

Per quanto riguarda invece l'istanza, questa è formata dalle aree di memoria allocate al suo avvio e contiene tutti i dati, i comandi SQL, le istruzioni PL/SQL e le informazioni del dizionario dei dati necessarie per l'esecuzione. L'istanza comprende anche i numerosi processi eseguiti in background dal sistema operativo utilizzati per i servizi offerti ai client utilizzatori.

La gestione degli accessi al database avviene tramite i concetti di utente e di schema. Ad ogni utente che può connettersi al database (identificato tramite un nome utente ed una password) corrisponde uno schema, ovvero un contenitore logico di oggetti di proprietà dell'utente. Gli oggetti contenuti in uno schema possono essere di vario tipo, come ad esempio tabelle, viste, indici, trigger, sequenze, funzioni e procedure.

Per la connessione al database da parte delle applicazioni (anche in un ambiente distribuito) è necessario installare sul computer utilizzato il software Oracle Client. La sua configurazione prevede la definizione del contenuto del file tnsnames.ora, che contiene un elenco di nomi associati ai parametri di configurazione che consentono la connessione a diverse istanze di database presenti sulla rete, anche attraverso una connessione VPN, consentendo quindi connessioni dirette alle istanze delle aziende clienti.

Il database aziendale utilizzato dal sistema informativo Iris contiene oltre 700 tabelle identificate da un codice e da una descrizione specifica per i dati contenuti. Il codice varia in base all'applicativo principale che utilizza i dati al suo interno e può assumere i seguenti valori (xxx rappresenta un numero):

- Ixxx: tabelle generiche non legate ad un unico applicativo.
- Txxx: tabelle specifiche dell'applicativo Rilevazione presenze.
- Pxxx: tabelle specifiche dell'applicativo Gestione economica.
- SGxxx: tabelle specifiche dell'applicativo Stato giuridico

Alcuni esempi di tabelle per le 4 categorie sono i seguenti:

- I001_LOGDATI: contiene le informazioni registrate come log delle applicazioni del sistema informativo Iris.
- I025_CESTINO: consente di recuperare dei record cancellati per errore dall'utente nel gestionale su determinate tabelle monitorate.
- I210_REGOLE_ARCHIVIAZIONE: contiene le regole utilizzate per le funzionalità di archiviazione documentale comuni a tutti gli applicativi.
- T030_ANAGRAFICO: che contiene i dati anagrafici del dipendente gestito nell'azienda cliente.
- T265_CAUASSENZE: contiene le causali di assenza create e la loro configurazione nel contesto della rilevazione presenze.

- T282_MESSAGGI: contiene il dettaglio dei messaggi scambiati utilizzando la funzionalità di messaggistica.
- T850_ITER_RICHIESTE: contiene lo stato delle richieste associate ai vari iter autorizzativi.
- P200_VOCI: contiene le voci stipendiali utilizzate per l'elaborazione del cedolino paga e la loro configurazione nel contesto della gestione economica.
- P441_CEDOLINO: contiene i dati di intestazione dei cedolini paga elaborati dall'applicativo gestione economica.
- P430_ANAGRAFICO: contiene l'anagrafica stipendiale del dipendente, utilizzata per configurare i parametri per l'elaborazione del cedolino.
- SG101_FAMILIARI: contiene i dati relativi ai familiari del dipendente, necessari, ad esempio, per la gestione economica e la fruizione di particolari causali di assenza.
- SG230_ITER_CERTIFICAZIONI: contiene le informazioni relative all'iter autorizzativo "Scheda informativa".
- SG235_MODELLI_CERTIFICAZIONI: contiene la definizione dei modelli di scheda informativa che il dipendente può creare nell'omonimo iter autorizzativo.

Per facilitare le operazioni sui dati e sulle tabelle sono disponibili numerose funzioni e procedure create in azienda utilizzando il linguaggio PL/SQL, un linguaggio procedurale che estende SQL con la possibilità di definire variabili, utilizzare istruzioni condizionali e cicli, gestire accezioni e definire nuovi tipi per i dati manipolati.

2.5 Distribuzione dell'applicazione

L'applicazione IrisAPP viene distribuita come applicazione ISAPI nel web server Microsoft IIS e può essere installata sia su un web server dell'azienda cliente, sia come Saas (software as a service) nel web server del data center Mondo Edp.

Microsoft IIS (Internet Information Services) è un software web server per il sistema operativo Windows Server che permette l'esecuzione di applicazioni ISAPI (Internet Server Application Programming Interface). Questo strumento consente di sviluppare applicazioni web con prestazioni migliori rispetto alla tecnologia CGI, in quanto formate da codice compilato e da un unico processo utilizzato per gestire tutte le richieste ricevute, limitando quindi la quantità di risorse utilizzate. Le applicazioni ISAPI sviluppate vengono distribuite sotto forma di dll e possono essere di due tipi:

- Estensioni ISAPI: rappresentano le applicazioni vere e proprie eseguite nel web server. La dll dell'applicazione viene caricata nel processo controllato da IIS, consentendo l'accesso alla web app tramite la dll come fosse una pagina html statica.
- Filtri ISAPI: consentono di filtrare (leggendo e/o modificando) tutte le richieste in arrivo sul web server e le relative risposte in uscita. Permettono ad esempio di implementare a livello di web server funzionalità di autenticazione, registrazione di log e restrizioni di accesso.

Le estensioni ISAPI vengono caricate dinamicamente sotto forma di dynamic-link library dal processo w3wp.exe di IIS in corrispondenza della prima richiesta ricevuta e destinata all'applicazione, identificata in base all'url utilizzato. Successivamente tutte le richieste ricevute vengono gestite dallo stesso processo utilizzando threads differenti.

IIS consente la creazione di una struttura gerarchica basata sulla creazione di siti, applicazioni e directory virtuali. I siti sono contenitori di applicazioni e directory virtuali a cui è possibile accedere utilizzando un protocollo (ad esempio http o https) e un indirizzo comune (ad esempio formato da indirizzo ip e porta) che costituiscono l'url del sito. A questo livello è possibile limitare la banda utilizzata, il numero di connessioni attive e attivare strumenti di registrazione dei log. All'interno del sito possiamo trovare più applicazioni, ogni applicazione è raggiungibile aggiungendo il percorso della stessa all'url del sito che la contiene. Le diverse applicazioni eseguite nel web server possono essere separate e/o raggruppate in application pool. Ad ogni application pool corrisponde un processo di IIS separato ed una configurazione comune a tutte le applicazioni al suo interno, eventuali errori in un'applicazione non hanno impatto sugli altri application pool che continueranno l'esecuzione correttamente. L'utilizzo di application pool diversi permette anche di implementare politiche di load balancing sulla stessa applicazione, destinando le richieste in modo bilanciato su più application pool separati e limitando l'utilizzo di risorse da parte del singolo processo. Le directory virtuali (sempre contenute in un sito) sono raggiungibili aggiungendo il loro percorso all'url del sito corrispondente e consentono l'accesso ad una o più directory fisiche sul web server, esponendo quindi i file al loro interno. L'accesso ai file è possibile ad esempio utilizzando un browser, opzionalmente configurando delle credenziali e specificando una modalità di autenticazione per limitare l'accesso ai dati.

Per aumentare la sicurezza del web server è necessario configurare le ISAPI/CGI Restrictions, che richiedo di abilitare in modo esplicito quali estensioni ISAPI (e Common Gateway Interface) possono essere eseguite in IIS.

Nel caso particolare dell'applicazione sviluppata, questa viene distribuita come estensione ISAPI ed eseguita in un application pool specifico che contiene solo questa applicazione. L'url utilizzato per raggiungere l'applicazione è nella forma: https://sito/IrisAPP/WM000PIrisAPP.dll. Creando all'interno della directory fisica a cui l'applicazione fa riferimento un file index.htm, che esegue il redirect sull'url completo indicato in precedenza, è possibile rimuovere il riferimento alla dll in modo da ottenere una url più breve e facile da ricordare per l'utente.

3. Sviluppo dell'applicazione

In questo terzo capitolo viene fornita una panoramica sulla fase di sviluppo della web app, partendo da una descrizione degli strumenti utilizzati (Delphi e UniGUI), per passare poi alle scelte progettuali che hanno portato prima alla definizione dell'architettura dell'applicazione ad alto livello e poi all'implementazione delle singole funzionalità. La descrizione testuale è integrata con l'utilizzo di class diagram, realizzati sia a scopo illustrativo che come documentazione di progetto utilizzando gli strumenti di modellazione del Delphi IDE. La rappresentazione delle classi nei diagrammi è in alcuni casi fornita in versione semplificata, evidenziando gli attributi più importanti ed omettendo quelli meno significativi o gestiti in maniera automatica dal linguaggio, con l'obiettivo di produrre diagrammi graficamente comprensibili.

3.1 Il linguaggio Delphi

Il linguaggio di programmazione utilizzato per lo sviluppo dell'applicazione è Delphi, un linguaggio ad oggetti evoluzione dell'object pascal, che attraverso l'utilizzo del corrispondente Delphi IDE consente la produzione rapida di software gestionali. Fornendo al programmatore un grande numero di componenti software da utilizzare per la creazione dell'interfaccia grafica e per l'interazione con i comuni database, consente di ridurre la quantità di codice richiesto per lo sviluppo di software gestionali, caratterizzati dalla necessità di visualizzare dati attraverso una rappresentazione tabellare, dalla gestione di elaborazioni massive ed da estrazioni di dati con strumenti di reporting.

Nel caso di applicazioni distribuite con architettura client-server, Delphi fornisce strumenti adatti alla creazione di applicazioni desktop. Interfacciandosi con i database mediante componenti visuali data-aware, consente allo sviluppatore di collegare in modo agevole l'interfaccia grafica alla base dati, con la possibilità di eseguire un processamento delle informazioni estratte mediante la definizione di event handler, eseguiti al verificarsi di determinati eventi.

Per lo sviluppo di applicazioni in ambiente web, il linguaggio fornisce nativamente strumenti adatti alla creazione del back end dell'applicazione, come il framework Datasnap per la realizzazione di web service REST. È tuttavia necessario utilizzare framework sviluppati da terzi per la creazione della parte front-end dell'applicazione, come IntraWeb (già utilizzato in azienda per lo sviluppo degli applicativi IrisCloud e IrisWEB) o il più recente UniGUI utilizzato in questo progetto.

Alcune caratteristiche significative del linguaggio e del Delphi IDE, utili alla comprensione dell'architettura del progetto descritta in seguito, sono le seguenti:

Form e Frame

L'interfaccia grafica delle applicazioni sviluppate in Delphi è formata da una serie di oggetti comunemente chiamati "form", istanze di classi che ereditano da TForm, che rappresentano le singole finestre visualizzate nelle applicazioni desktop. All'interno delle form vengono posizionati i vari componenti visuali, utilizzati dall'utente per visualizzare i dati ed interagire con il programma, andando a formare una struttura ad albero. Ogni form ed ogni componente in essa contenuto è caratterizzato di una serie di proprietà che consentono di modificarne le caratteristiche

sia in fase di sviluppo (design time), sia durante l'esecuzione (run time). Inoltre, una serie di event handler innescati da eventi significativi per lo specifico componente (come il click su un pulsante) consentono di gestire l'interazione con l'utente o eseguire operazioni in risposta ad eventi specifici. Un esempio è l'evento OnCreate, eseguito dopo la creazione della form tramite il costruttore, e l'evento OnDestroy, eseguito prima della distruzione della form con il distruttore della classe. Altri componenti con caratteristiche simili alle form sono i frame, sottoclassi di TFrame, rappresentano dei contenitori di componenti visuali dal layout e dal comportamento definito dal programmatore. I frame possono essere riutilizzati ed a loro volta posizionati come componenti all'interno delle form, incapsulando funzionalità comuni a più punti dell'applicazione.

Datamodule

Così come le form ed i frame consentono la creazione dell'interfaccia grafica dell'applicazione attraverso il posizionamento dei componenti visuali al loro interno, i datamodule, classi che ereditano da TDataModule, consentono la gestione degli oggetti necessari per l'implementazione della logica applicativa e l'interazione con i database. Nei datamodule vengono tipicamente istanziati oggetti di tipo TQuery o TDataset, usati per l'interazione con la base dati, la cui gestione può poi essere incapsulata nelle funzioni pubbliche esposte dal datamodule.

Form Designer

Il Form Designer è lo strumento del Delphi IDE utilizzato per la creazione e la modifica delle form, dei frame e dei datamodule che compongono l'applicazione. Consente di definire le loro caratteristiche a design time, come il posizionamento dei componenti e la configurazione, senza la necessità di scrivere codice ma editando la loro posizione mediante il trascinamento nella rappresentazione grafica dell'elemento.

Component Palette

La Component Palette è lo strumento del Delphi IDE che consente la selezione ed il trascinamento dei componenti disponibili nell'area del Form Designer. Numerosi componenti (visuali e non) sono già disponibili raggruppati per tipologia, è inoltre possibile definire dei package di componenti personalizzati in base alle proprie esigenze. Il posizionamento di un componente nella form in modifica comporta la creazione automatica da parte dell'IDE di un corrispondente attributo della classe, il cui ciclo di vita verrà gestito in automatico dal linguaggio mediante il meccanismo di ownership.

Object TreeView

Per una migliore gestione della struttura ad albero formata dagli oggetti creati è disponibile nel Delphi IDE lo strumento Object TreeView. Consente di visualizzare una vista sincronizzata con il form designer della struttura creata, permettendo un migliore controllo sugli oggetti di piccole dimensioni che compongono l'interfaccia grafica, gestendo anche la sovrapposizione dei componenti visuali richiesta dall'applicazione.

Object Inspector

Per la modifica delle proprietà e l'assegnazione degli event handler agli eventi dei componenti è disponibile lo strumento del Delphi IDE denominato Object Inspector. Consente di modificare i valori di default delle proprietà a design time attraverso l'uso di un'interfaccia grafica che facilita

l'operazione, limitando le possibilità di errore attraverso la selezione di valori da liste predefinite e l'uso di specifiche finestre dedicate allo scopo.

Il codice delle applicazioni Delphi è organizzato in file con estensione pas chiamati "unit". Tipicamente ogni unit contiene la definizione di una classe, ma è consentita anche la definizione di variabili e funzioni con visibilità globale tipiche della programmazione procedurale. Dato che il linguaggio ha la caratteristica di non essere case-sensitive, per convenzione l'identificativo associato ai tipi inizia sempre con la lettera T (maiuscola) per classi, record ed enumerazioni, mentre inizia con la lettera I (maiuscola) per le interfacce. La struttura del codice all'interno delle unit prevede una suddivisione in quattro sezioni:

- Interface: contiene la definizione di tipi, costanti, funzioni, procedure e variabili a disposizione delle altre unit che includono la unit in esame. Contiene inoltre la clausola uses che elenca le altre unit visibili alla unit in esame.
- Implementation: contiene l'implementazione delle procedure e delle funzioni definite nella sezione interface, sia quelle definite come funzioni membro nella dichiarazione di una classe, sia quelle definite con visibilità globale. Può contenere un'ulteriore clausola uses che elenca le unit il cui contenuto è visibile nella sola sezione implementation della unit in esame.
- Initialization: è una sezione opzionale che contiene un elenco di istruzioni eseguite all'avvio del programma. L'esecuzione delle varie sezioni initialization di unit differenti avviene secondo l'ordine di apparizione delle stesse nelle clausole uses.
- Finalization: è una sezione opzionale che contiene un elenco di istruzioni eseguite prima della terminazione del programma. L'esecuzione delle varie sezioni finalization di unit differenti avviene in ordine inverso rispetto alle rispettive sezioni inialization.

Per ogni form, frame e datamodule gestito mediante il Form Designer viene creato e mantenuto dal Delphi IDE un ulteriore file con estensione dfm (Delphi Form Module, con lo stesso nome del corrispondente file pas), contenente la configurazione del componente definita a design time mediante gli strumenti precedentemente descritti. La modifica diretta del file dfm da parte del programmatore è sconsigliata in quanto il formato dei dati contenuti, sebbene human-readable, è adatto all'importazione automatica nel Delphi IDE ed eventuali errori rischiano di compromettere l'integrità della classe.

Una caratteristica del linguaggio importante da considerare nel contesto di un'applicazione server è l'assenza del garbage collector, tipico di altri linguaggi, che obbliga il programmatore ad una puntuale distruzione degli oggetti per evitare memory leak. L'allocazione e il rilascio della memoria sono quindi da gestire con attenzione in un'applicazione destinata ad essere eseguita per lunghi periodi. Un meccanismo disponibile per facilitare la gestione del ciclo di vita degli oggetti è l'ownership: quando viene creato un componente (sottoclasse del tipo TComponent) è possibile specificare come parametro del costruttore il componente owner dell'oggetto creato, consentendo così la creazione di una struttura ad albero. Alla distruzione di un nodo padre vengono automaticamente distrutti in cascata tutti i nodi figli, facilitando la gestione del rilascio della memoria allocata. Questo meccanismo è utilizzato automaticamente per tutti i componenti visuali

che compongono le form, i frame e i datamodule definiti nel Form Designer, senza necessità di intervento da parte del programmatore.

Nelle applicazioni desktop sviluppate in linguaggio Delphi la gestione delle eccezioni viene implementata mediante la creazione di blocchi try/except (dal funzionamento analogo ai blocchi try/catch di altri linguaggi). Le eccezioni non gestite esplicitamente dal programmatore non comportano però la terminazione immediata del programma, vengono infatti automaticamente gestite visualizzando a video una finestra con il relativo messaggio di errore.

3.2 Il framework UniGUI

Prima di entrare nel dettaglio della descrizione dell'implementazione, è utile capire il funzionamento del framework utilizzato nello sviluppo dell'applicazione IrisAPP.

Il framework UniGUI si pone l'obiettivo di ridurre i tempi ed i costi richiesti dalla creazione di un'applicazione web, proponendo un modello di sviluppo simile a quello previsto dalle applicazioni desktop tradizionali create con il linguaggio Delphi, basate su componenti visuali ospitati all'interno di form (pulsanti, labels, editbox, combobox, griglie ecc. presenti nella visual control library del linguaggio). Vengono inoltre utilizzati gli eventi dei componenti, tramite gli event handler definiti dallo sviluppatore innescano la navigazione e la logica applicativa.

Integrandosi con il Delphi IDE, fornisce allo sviluppatore un'architettura di base per la gestione del ciclo di vita delle sessioni web. Permette, inoltre, la creazione dell'interfaccia grafica mediante il posizionamento dei componenti visuali all'interno di una o più form, sfruttando lo stesso Form Designer delle applicazioni desktop. In questo contesto le form rappresentano la pagina visualizzata dall'utente nel browser.

In particolare, il framework sfrutta la libreria javascript Sencha ExtJS per la creazione della pagina web e l'interazione con l'utente lato client. Ogni componente ExtJS visualizzato nel browser avrà un corrispondente oggetto Delphi lato server associato alla sessione web attiva. Gli eventi innescati dall'interazione dell'utente con la pagina vengono mappati sugli event handler dell'oggetto lato server e, viceversa, le modifiche apportate alle proprietà dell'oggetto lato server vengono convertite in codice javascript eseguito lato client, con lo scopo di aggiornare il corrispondente componente ExtJS nel browser. L'interazione tra client e server avviene tramite richieste http asincrone che non richiedono il refresh della pagina web, consentendo l'implementazione di una single page application.

Questo tipo di interazione tra client e server consente di ridurre al minimo la produzione di codice javascript richiesto per il funzionamento dell'applicazione, lo sviluppatore deve solamente creare il layout della pagina e gestire gli eventi lato server scrivendo il codice in linguaggio Delphi.

Inoltre, il framework ExtJS mette a disposizione una serie di temi predefiniti per caratterizzare l'interfaccia grafica, consentendo di ridurre al minimo anche la necessità dell'utilizzo di fogli di stile (CSS) personalizzati.

Vengono di seguito descritti: l'architettura delle classi che costituiscono lo scheletro del progetto di base, i componenti visuali a disposizione del programmatore ed i meccanismi per la gestione dell'interazione tra client e server nel contesto di un'applicazione web.

3.2.1 Struttura di base del progetto

Il framework permette di creare un progetto che potrà essere compilato come standalone server (utile per il debug durante la fase di sviluppo), come windows service o come applicazione ISAPI da eseguire nel web server Microsoft IIS.

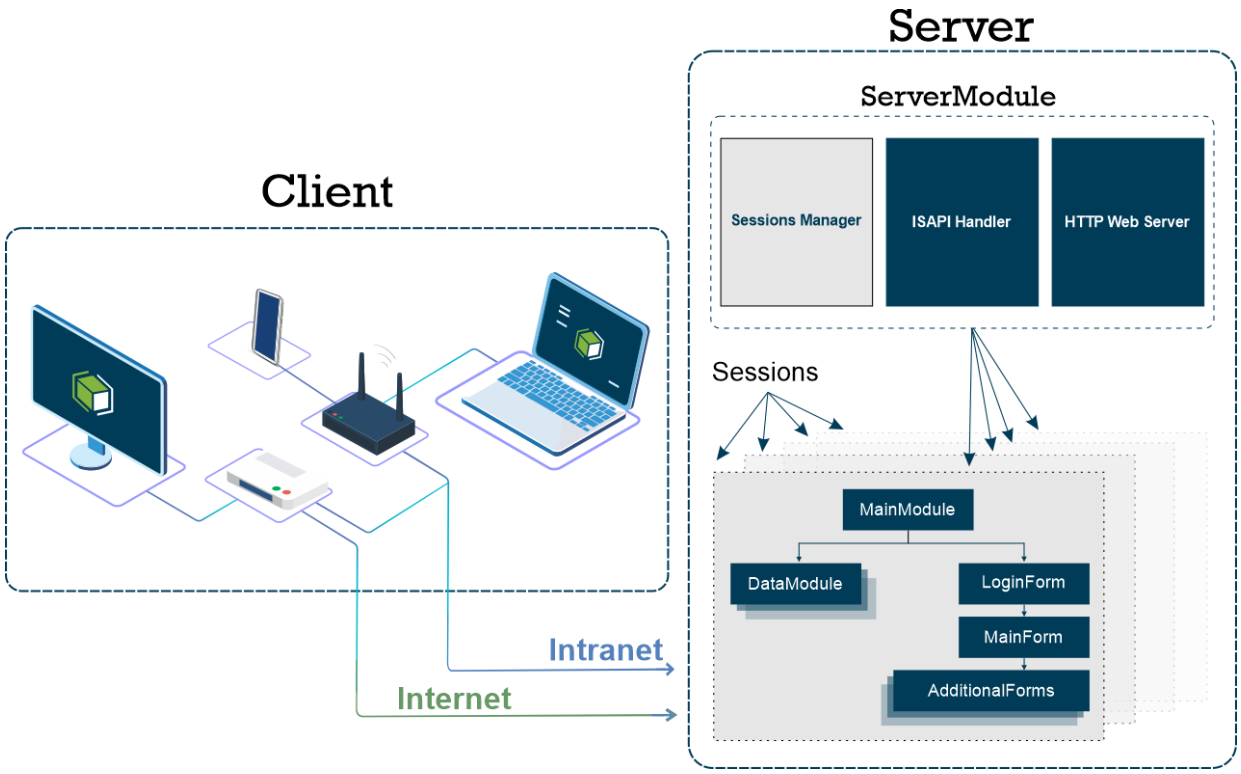


Figura 2.1 – La struttura di base di un progetto UniGUI
(tratta da http://www.unigui.com/explore/technology-overview)

La struttura di base di un progetto UniGUI prevede l'estensione di alcune classi fornite dal framework, in particolare:

- ServerModule: sottoclasse di TUniGUIServerModule, permette di definire la configurazione del server (SSL, log management, ecc.). Ogni server UniGUI contiene un'unica istanza della classe ServerModule, visibile e condivisa da tutte le sessioni web (singleton). È accessibile tramite la funzione globale UniServerInstance.
- MainModule: sottoclasse di TUniGUIMainModule, viene creata un'istanza di questa classe per ogni sessione web, consentendo la gestione degli eventi innescati all'avvio ed alla distruzione della sessione. All'interno della classe MainModule possono essere create istanze di oggetti visibili alle varie form del progetto nella singola sessione.

- LoginForm: sottoclasse di TUniLoginForm, se prevista dall'applicazione, è la prima form visualizzata dall'utente al momento della creazione della sessione web ed è utilizzata per la gestione dell'operazione di autenticazione.
- MainForm: sottoclasse di TUniMainForm, rappresenta la form principale della single page application ed è visualizzata dopo la corretta autenticazione dell'utente o, in assenza della LoginForm, come prima form dell'applicazione.

Il progetto può poi contenere un numero arbitrario di form aggiuntive che saranno sempre visualizzate nel browser come sovrapposte alla MainForm.

È possibile utilizzare le seguenti funzioni globali per ottenere informazioni e manipolare le sessioni web:

- UniApplication: ritorna un oggetto di tipo TUniGUIApplication che rappresenta l'owner di tutte le risorse della sessione. Fornisce proprietà per accedere ad informazioni associate alla sessione web, come i cookies, la dimensione dello schermo ed il tipo di dispositivo client.
- UniSession: ritorna un oggetto di tipo TUniGUISession che rappresenta la sessione web corrente. Fornisce alcuni metodi ed eventi per la gestione della sessione e permette l'accesso alle informazioni associate alla richiesta http correntemente gestita dal web server nel contesto della sessione.

Le sessioni web sono gestite dal framework stesso, con la creazione della sessione lato server al primo accesso dell'utente (alla LoginForm o alla MainForm) e la sua distruzione nei seguenti casi:

- Dopo un tempo di inattività definito (timeout).
- Alla chiusura del browser da parte dell'utente (sfruttando l'evento onbeforeunload per comunicare al server il termine di una sessione, funzionante però solo per i browser desktop).
- Utilizzando il metodo "Terminate" sull'oggetto TUniGUISession associato alla sessione corrente.

3.2.2 Componenti visuali

Il framework mette a disposizione due diverse librerie di componenti visuali che utilizzano lato client la libreria javascript Sencha ExtJS. La prima è specifica per lo sviluppo di applicazioni web per browser desktop (i cui componenti hanno il nome della classe caratterizzato dal prefisso "TUni"), la seconda invece è dedicata allo sviluppo di applicazioni web per browser mobile (i cui componenti hanno il nome della classe caratterizzato dal prefisso "TUnim").

Nel progetto dell'applicazione IrisAPP si è scelto di utilizzare la seconda libreria, più adatta a dispositivi con schermo di dimensione ridotta e con un migliore supporto per l'interazione tramite touch screen, tipica di smartphone e tablet. La libreria scelta, denominata "UniGui Mobile", contiene numerosi componenti da utilizzare per la creazione dell'interfaccia grafica accessibili tramite la Component Palette del Delphi IDE.

Molto importanti sono le classi TUnimContainerPanel e TUnimPanel, entrambe associate al componente ExtJS Ext.Container, rappresentano un contenitore logico di elementi da utilizzare per la definizione del layout della pagina. Utilizzando la proprietà AlignmentControl è possibile specificare il tipo di comportamento previsto nella rappresentazione del contenuto, sono possibili due valori:

- uniAlignmentServer: riproduce nel browser la posizione del contenuto definito dallo sviluppatore a design time nel Delphi IDE, con l'obiettivo di sfruttare le proprietà di allineamento e spaziatura dei componenti tipiche della programmazione Delphi per applicazioni desktop.
- uniAlignmentClient: utilizza le proprietà del framework ExtJS per la gestione del layout del contenitore, disponendo i componenti in maniera indipendente dalla posizione definita a design time nel Delphi IDE.

Dopo una valutazione iniziale di entrambe le modalità disponibili, è stata scelta la modalità uniAlignmentClient che consente un maggiore controllo e una maggiore precisione nella definizione del layout della pagina. Le proprietà da valorizzare sull'oggetto contenitore per la gestione del layout in questa modalità sono le seguenti:

- Layout: definisce la modalità di posizionamento dei componenti nel contenitore, può assumere i valori:
    - vbox - gli elementi sono allineati verticalmente su un'unica colonna dall'alto al basso del contenitore.
    - hbox – gli elementi sono allineati orizzontalmente su un'unica riga da sinistra verso destra del contenitore.
    - fit - il primo elemento presente nel contenitore occupa l'intero spazio disponibile, sovrapponendosi ad altri elementi se presenti.
    - float - gli elementi occupano lo spazio disponibile dall'alto al basso e da sinistra verso destra, formando nuove righe se lo spazio orizzontale non è sufficiente.
- LayoutAttribs.Align; definisce il posizionamento degli elementi nel contenitore nella direzione ortogonale a quella di allineamento per i layout vbox e hbox (può assumere i valori start, center, end, stretch).
- LayoutAttribs.Pack: definisce il posizionamento degli elementi nel contenitore nella direzione di allineamento per i layout vbox e hbox (può assumere i valori start, center, end, justify).
- LayoutConfig: permette di impostare alcune caratteristiche del contenitore (come altezza e larghezza) utilizzando il linguaggio CSS.
- Autoscroll: se abilitato, permette lo scorrimento degli elementi nel contenitore in caso di overflow.
- CreateOrder: da valorizzare sugli elementi inseriti nel contenitore, definisce il loro ordine di posizionamento.

Le stesse proprietà sono disponibili anche per la gestione del layout dei vari form e frame dell'applicazione (sottoclassi dei tipi TUnimForm e TUniFrame in questo contesto specifico).

Utilizzando più componenti contenitore nella stessa form, è possibile creare il layout della pagina adattando automaticamente la posizione e la dimensione degli elementi in base alle caratteristiche della finestra del browser sul dispositivo.

Uno degli obiettivi del framework è la riduzione della quantità di codice javascript da produrre durante lo sviluppo dell'applicazione, viene comunque fornita su tutti i componenti visuali la proprietà ClientEvents (di tipo TUniClientEvents) che consente di associare del codice javascript ai componenti utilizzati. La classe TUniClientEvents permette di specificare il codice che deve essere eseguito lato client in risposta a due tipologie di eventi javascript associati, ExtEvents e UniEvents. I primi sono eventi propri del corrispondente oggetto definito dal framework ExtJS, i secondi sono sempre eventi lato client, ma aggiunti all'oggetto dal framework UniGUI. Sfruttando questa possibilità è possibile eseguire delle operazioni sull'oggetto ExtJS che non necessitano di interazione con il server, tenendo presente però che le operazioni eseguite non vengono automaticamente riportate sul corrispondente oggetto Delphi mantenuto lato server, comportando quindi un rischio di disallineamento tra le due parti.

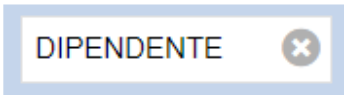
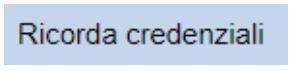
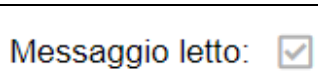
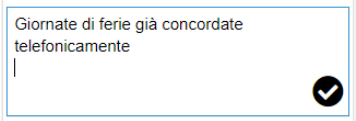
	COMPONENTE UNIGUI	COMPONENTE ExtJS
	TUnimButton	Ext.Button
	TUnimEdit	Ext.field.Text
	TUnimLabel	Ext.Label
	TUnimSelect	Ext.field.Select
	TUnimCheckbox	Ext.field.Checkbox
	TUnimMemo	Ext.field.TextArea
	TUnimDatePicker	Ext.field.Date
	TUnimTimePicker	Ext.field.Time

Figura 3.2 – Un esempio dei componenti visuali del framework UniGUI, nella prima colonna la loro rappresentazione grafica nel browser, nella seconda colonna il nome della classe Delphi utilizzata lato server e nella terza colonna il corrispondente componente del framework javascript ExtJS.

3.2.3 Interazione client-server

Come anticipato in precedenza, l'interazione tra il client dell'applicazione, eseguito nel browser come codice javascript del framework ExtJS, ed il componente server, sviluppato in linguaggio Delphi, avviene tramite richieste HTTP asincrone non bloccanti. Le richieste vengono create utilizzando i metodi specifici del framework ExtJS e possono diventare bloccanti per l'esecuzione lato client se richiesto dalla logica applicativa. Il server UniGUI sviluppato gestisce le richieste HTTP ricevute individuando la corrispondente sessione web, se la sessione è assente procede con la creazione di una nuova sessione di cui ritorna l'identificatore univoco, se non è più attiva risponde con una pagina di errore personalizzabile. Tipicamente la chiusura di una sessione avviene su richiesta esplicita dell'utente o dopo la scadenza del timeout di sessione.

La sessione web viene avviata lato server al momento della ricezione della richiesta HTTP (eventualmente HTTPS con protocollo TLS/SSL in base alla configurazione del server) a metodo GET sull'url di base dell'applicazione, ad esempio "https://irisapp.aziendacliente.com". Nel caso dei progetti mobile come quello sviluppato è previsto poi un redirect automatico sulla versione mobile del sito, raggiungibile aggiungendo il percorso "/m" all'url base. La pagina ritornata innesca poi una serie di richieste di risorse (file js e css) necessarie per l'esecuzione della single page application. Le successive richieste, generate dall'interazione dell'utente con l'applicazione, comportano l'invio di richieste HTTP a metodo POST con Content-Type "application/x-www-form-urlencoded" sul percorso "/HandleEvent" aggiunto all'url base. Nel corpo della richiesta è presente l'id della sessione web associata ed una serie di parametri utilizzati dal server per identificare l'operazione da eseguire e per trasferire i dati necessari. L'operazione richiesta è poi associata al corrispondente oggetto Delphi lato server che avvia l'esecuzione dell'event handler creato dal programmatore per produrre la modifica voluta.

Nella risposta prodotta dal server è visibile il codice javascript generato automaticamente, eseguito nel browser per manipolare i componenti ExtJS presenti nella pagina visualizzata e per gestire la navigazione nell'applicazione lato client. Eventuali eccezioni lanciate dal server nell'event handler vengono automaticamente gestite dal framework e trasformate in un messaggio di errore visualizzato in una message box. L'operazione di logout dall'applicazione comporta la distruzione della sessione web. Questa operazione avviene utilizzando sempre una richiesta sul percorso "/HandleEvent", ma comporta in risposta un redirect sull'url principale dell'applicazione o su una pagina definita appositamente.

Un'ulteriore possibilità per l'interazione del client con il server è data dalla funzione javascript "ajaxRequest" fornita dal framework UniGUI. La funzione consente al programmatore di eseguire del codice javascript che effettua una richiesta HTTP asincrona analoga a quella fatta in automatico durante l'interazione dell'utente con l'applicazione. Richiede di specificare come parametri i dati da inviare ed il nome associato lato server all'oggetto Delphi che deve gestire la richiesta. Sull'oggetto individuato viene eseguito l'event handler associato dallo sviluppatore all'evento OnAjaxRequest, dove si può accedere ai parametri ricevuti, eseguire le operazioni richieste ed integrare la risposta creata automaticamente con ulteriore codice javascript che verrà eseguito lato client.

Contestualmente alla creazione della sessione web lato server, nel percorso di installazione del server UniGUI ed all'interno di un'apposita cartella "cache", viene creata un'ulteriore cartella avente come nome l'id della sessione creata. Al suo interno è possibile creare i file resi disponibili al client per il download nel corso della sessione, come i documenti pdf generati, che sono resi accessibili all'utente utilizzando il percorso "/cache/session_id/file_richiesto" aggiunto all'url base dell'applicazione. La cartella creata viene poi automaticamente cancellata quando la sessione web termina, limitando l'occupazione di spazio su disco.

3.3 Componenti visuali personalizzati

Nella prima fase dello sviluppo dell'interfaccia grafica dell'applicazione è emersa la necessità di creare dei componenti visuali personalizzati. I componenti forniti nella libreria UniGUI Mobile presentano infatti caratteristiche di base utili per la creazione del layout della pagina, ma richiedono alcuni accorgimenti per un migliore posizionamento e distanziamento. Inoltre, la loro definizione consente di evitare la duplicazione di codice nella gestione dell'interfaccia, incapsulando caratteristiche grafiche comuni nei componenti da riutilizzare.

I componenti sviluppati possono essere suddivisi in due tipologie:

- Componenti di base: sono classi che ereditano dai componenti visuali già disponibili nella libreria del framework UniGUI o da altri componenti di base, andando ad estendere la loro definizione con l'aggiunta di proprietà specifiche ed impostando dei valori di default su quelle esistenti.
- Componenti avanzati: sono classi definite come composizione di componenti di base o come sottoclassi di altri componenti avanzati. Consentono di implementare funzionalità più complesse ma comuni a più pagine dell'applicazione.

Le nuove classi create fanno parte del presentation layer dell'applicazione e sono state inserite all'interno di un package specifico chiamato "uniGUI Medp Mobile". Una volta installato può essere utilizzato tramite la Component Palette del Delphi IDE come gli altri componenti visuali messi a disposizione dal linguaggio.

Di seguito vengono elencati i componenti creati, fornendo una descrizione del loro funzionamento integrata con alcune immagini significative del loro aspetto all'interno delle pagine dell'applicazione.

3.3.1 Componenti di base

Come anticipato, i componenti di base sono definiti estendendo le classi fornite dal framework, con l'aggiunta di proprietà significative per il loro utilizzo nell'applicazione. In particolare, la scelta di utilizzare la modalità di gestione del layout uniAlignmentClient ha evidenziato la necessità di una migliore gestione degli elementi HTML visualizzati nella pagina, mediante gli strumenti a disposizione dello sviluppatore per la gestione del box model CSS.

Per avere un maggiore controllo sui valori di margin, padding e border degli elementi rappresentati, sono state create le classi TBoxModelProperty e TBoxModel, con le seguenti caratteristiche:

- TBoxModelProperty: consente di gestire separatamente i valori assegnati agli attributi Top/Bottom/Left/Right di una delle tre proprietà CSS del box model, rappresentate sotto forma di stringa. Inoltre, emette un evento OnChange nel caso una delle quattro proprietà della classe venga modificata.
- TBoxModel: consente la gestione di margin, padding e border dell'elemento HTML, incapsulando tre proprietà di tipo TBoxModelProperty (CSSMargin, CSSPadding e CssBorder). Inoltre, permette la gestione della caratteristica border-radius mediante la corrispondente proprietà Delphi. L'aggiunta dell'evento OnChange (a sua volta legato agli eventi OnChange degli oggetti incapsulati) permette di eseguire un event handler specifico dopo aver apportato delle modifiche alle istanze della classe.

Un oggetto di tipo TBoxModel è presente come attributo con visibilità published per ogni componente di base, garantendo, grazie all'ereditarietà, la propagazione di questa caratteristica a tutti i componenti personalizzati. All'evento OnChange della classe è poi associato un event handler specifico che consente di aggiornare i componenti ExtJS lato client con le modifiche apportare agli oggetti Delphi lato server.

Un'altra caratteristica da gestire con maggior precisione negli elementi HTML è l'allineamento del testo presente al loro interno, a questo scopo sono stati definiti i due tipi enumerativi TAlignItems (che può assumere i valori AlignStart, AlignEnd e AlignCenter) e TJustifyContent (che può assumere i valori JustifyStart, JustifyEnd e JustifyCenter). Nei componenti personalizzati che richiedono questo tipo di gestione sono presenti le proprietà AlignItems e JustifyContent, i cui metodi setter gestiscono anche l'aggiornamento lato client del valore della rispettiva proprietà CSS.

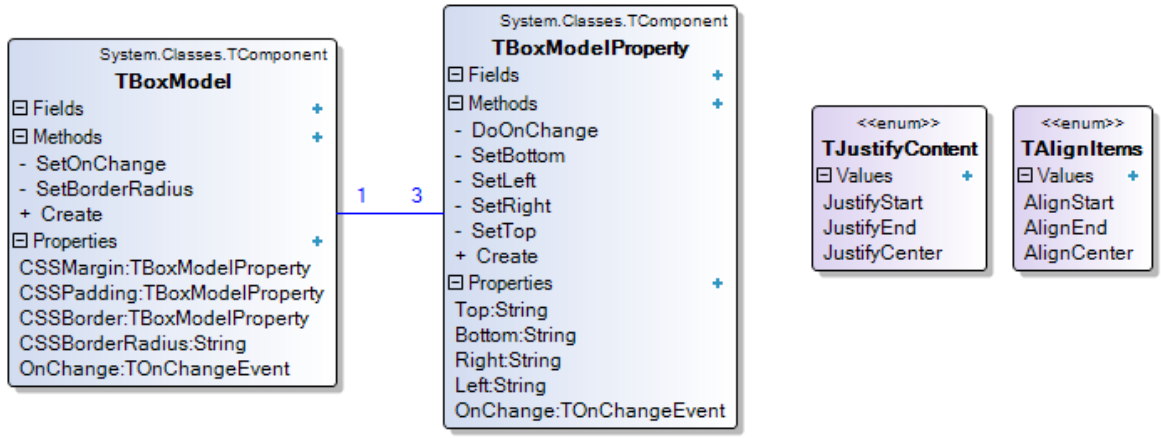


Figura 3.3 – Il class diagram con le classi TBoxModel, TBoxModelProperty e le enumerazioni TJustifyContent e TAlignItems

I componenti di base sviluppati sono i seguenti:

TMedpUnimButton
Sottoclasse di TUnimBitBtn, è utilizzato per creare i pulsanti presenti nell'applicazione al cui interno è presente un testo con opzionalmente l'aggiunta di un'icona.

Figura 3.4 – Un esempio con tre componenti di tipo TMedpUnimButton

TMedpUnimLabel
Sottoclasse di TUnimCustomLabel, è utilizzato per creare le etichette testuali presenti nell'applicazione. Sono state aggiunte alla superclasse due proprietà di tipo TAlignItems e TJustifyContent per la corretta gestione dell'allineamento orizzontale e verticale del testo.

TMedpUnimLabelIcon
Sottoclasse di TMedpUnimLabel, è utilizzato per rappresentare un'etichetta contenente un'icona della libreria Font Awesome 5. Impostando la proprietà Caption con il valore da assegnare alla classe CSS definita nella libreria si ottiene un'icona la cui dimensione è gestita tramite la proprietà font size. In aggiunta, è possibile modificare il colore dell'icona impostando il valore della nuova proprietà CSSColor.

TMedpUnimEdit
Sottoclasse di TUnimCustomEdit, è utilizzato per creare i campi per l'inserimento di un testo libero visualizzato su una singola riga. Consente inoltre di affiancare un'etichetta testuale senza la necessità di utilizzare un ulteriore oggetto di tipo TMedpUnimLabel.

TMedpUnimSelect
Sottoclasse di TUnimSelect, è utilizzato per creare un campo di input che permette la selezione da un elenco a discesa. Sono fornite tre versioni del metodo pubblico Popola che consentono di inserire i valori visualizzati nell'elenco a partire da tipi di parametro differenti, come TStringList e TList<String> a seconda delle necessità. Il metodo Add può essere utilizzato per inserire nell'elenco i singoli elementi. Inoltre, permette la gestione di valori definiti tramite un codice identificativo ed una descrizione testuale, dando allo sviluppatore la possibilità di scegliere cosa visualizzare nel componente (solo il codice, solo la descrizione o entrambi) e l'eventuale carattere separatore visualizzato nell'elenco a discesa. L'indice numerico associato all'elemento selezionato (ItemIndex) può essere utilizzato per accedere al codice e alla descrizione, gestiti tramite due liste ordinate separate accessibili tramite le proprietà ListaCodici e ListaDescrizioni. Un'ulteriore proprietà booleana permette di abilitare la gestione dell'elemento vuoto, visualizzato nella prima posizione dell'elenco a discesa.

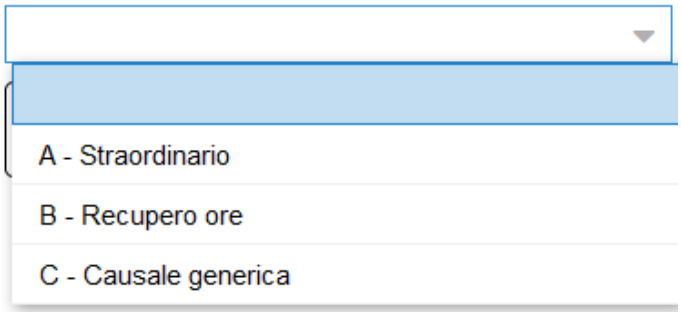


Figura 3.5 – Un esempio di componente TMedpUnimSelect con tre elementi oltre all'elemento vuoto

TMedpUnimDatePicker
Sottoclasse di TUnimCustomDatePicker, è utilizzato per creare i campi di input per la selezione di una data dal calendario. Nel metodo SetDefaultProperties viene fissato l'ordinamento day/month/year e viene disabilita la possibilità di editare direttamente la data utilizzando la tastiera del dispositivo.

TMedpUnimTimePicker
Sottoclasse di TUnimCustomTimePicker, è utilizzato per creare i campi di input per la selezione di un orario dall'orologio. Nel metodo SetDefaultProperties viene fissata la tipologia di orologio da visualizzare per la scelta e viene disabilita la possibilità di editare direttamente l'orario utilizzando la tastiera del dispositivo.

Figura 3.6 – Un esempio di componente TMedpUnimDatePicker (in alto) e TMedpUnimTimePicker (in basso)

TMedpUnimNumberEdit
Sottoclasse di TMedpUnimNumberEdit, è utilizzato per creare i campi di input per la sola selezione numerica. La tastiera visualizzata dal dispositivo permette solamente l'inserimento di numeri.

TMedpUnimRadioIcon
Sottoclasse di TUnimRadio, è utilizzato per creare i campi di input di tipo radiobutton per la selezione mutuamente esclusiva di un elemento. L'etichetta associata all'icona di selezione della superclasse è nascosta, il vero componente radiobutton viene definito in seguito nei componenti avanzati utilizzando questo elemento come base.

TMedpUnimPanelBase
Sottoclasse di TUnimContainerPanel, è l'elemento contenitore alla base della definizione di tutti i componenti avanzati e viene utilizzato per raggruppare i componenti collocati graficamente al suo interno. Consente di gestire il layout visualizzato con le stesse proprietà della superclasse (Layout, LayoutAttribs, LayoutConfig ecc. descritte in precedenza). Gli eventi OnClick e OnChange

vengono innescati dagli stessi eventi emessi dai componenti che contiene. Inoltre, la proprietà booleana DaContare può essere utilizzata per indicare se l'oggetto istanziato deve essere conteggiato quando inserito in uno dei contenitori avanzati definiti in seguito.

TMedpUnimTabPanelBase
Sottoclasse di TUnimTabPanel, è utilizzato per la gestione di diverse schede sovrapposte, ogni scheda sarà a sua volta un elemento contenitore. Nel metodo SetDefaultProperties viene nascosto l'header del componente in modo da non consentire all'utente la selezione diretta della scheda. La navigazione è consentita solamente da codice usando le funzioni Next, Prior, First e Last o settando la proprietà ActivePage del componente.

3.3.2 Componenti avanzati

I componenti avanzati consentono di implementare funzionalità più elaborate nell'interfaccia grafica dell'applicazione. Essendo definiti come composizione di elementi più semplici, utilizzano la classe TMedpUnimPanelBase come contenitore all'interno del quale vengono collocati graficamente i componenti visualizzati. Le funzionalità implementate sono comunque limitate alla visualizzazione ed alla gestione dell'interfaccia grafica, mantenendo la separazione con la logica applicativa.

La dimensione dei componenti avanzati è, in linea generale, adattata in automatico alla dimensione del contenuto. I singoli componenti visuali presenti nel contenitore sono esposti come proprietà con visibilità published, per consentire allo sviluppatore la modifica delle caratteristiche anche a design time.

I componenti avanzati creati sono i seguenti:

TMedpUnimMemo
Rappresenta un'estensione del componente TUnimMemo del framework UniGUI, che occupa tutto lo spazio all'interno del panel di base, con l'aggiunta di un'icona di conferma posizionata con coordinate relative nell'angolo in basso a destra. Viene utilizzato come componente di input per l'inserimento di un testo su più righe. Cliccando sul componente si abilita la modifica del testo, visualizzando l'icona di conferma e la tastiera di inserimento del dispositivo. Cliccando sull'icona si confermano le modifiche apportate e si esce dalla modalità di inserimento, nascondendo di conseguenza la tastiera del dispositivo e l'icona stessa.

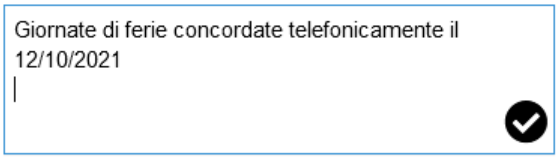


Figura 3.7 – Un esempio di componente TMedpUnimMemo

TMedpUnimCheckBox
Rappresenta un'estensione del componente TUnimCheckBox del framework UniGUI, utilizzato come campo di input per l'inserimento di valori booleani (può assumere gli stati checked e unchecked). All'interno del panel di base è presente sul lato sinistro un'etichetta di testo (dal

contenuto e dalla dimensione personalizzabile) e l'elemento TUnimCheckBox sul lato destro. La gestione dell'evento OnClick fa sì che la pressione esercitata su tutta l'area del componente inneschi il cambiamento di stato della check box, altrimenti limitato al solo elemento di tipo TUnimCheckBox.

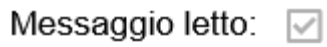


Figura 3.8 – Un esempio di componente TMedpUnimCheckBox

TMedpUnimRadioButton
Rappresenta un'alternativa al componente TUnimRadioButton fornito dal framework, utilizzato per l'input di valori booleani in un contesto di mutua esclusione definito dalla presenza di più elementi all'interno di un contenitore (vedi TMedpUnimRadioGroup definito in seguito). All'interno del panel di base è presente sul lato sinistro un'etichetta testuale e sul lato destro un componente di tipo TMedpUnimRadioIcon. La gestione dell'evento OnClick fa sì che la pressione esercitata su tutta l'area del componente inneschi il cambiamento di stato del radiobutton selezionato, innescando il comportamento inverso su tutti i radiobutton presenti nel contenitore in cui l'oggetto è istanziato, identificato tramite l'attributo Parent.

TMedpUnimRadioGroup
Incapsula una lista di oggetti di tipo TMedpUnimRadioButton istanziati dinamicamente. A seconda del layout impostato a design time (vbox o hbox) gli elementi possono essere disposti verticalmente o orizzontalmente nel panel di base. Gli oggetti vengono creati dopo il caricamento del componente a partire da una lista di stringhe, definita dallo sviluppatore tramite la proprietà Items. Per ogni elemento presente nella lista viene creato un radiobutton avente la stringa come etichetta. La selezione in mutua esclusione dei radiobutton è già gestita dai componenti stessi come descritto in precedenza.

Figura 3.9 – Un esempio di componente TMedpUnimRadioGroup con all'interno tre elementi di tipo TMedpUnimRadioButton

TMedpUnimPanel2Button
Incapsula due pulsanti affiancati nel panel di base, la configurazione di default prevede per entrambi i pulsanti l'orientamento orizzontale e l'occupazione del 50% dello spazio orizzontale disponibile.

TMedpUnimPanel2Label
Analogo alla classe TMedpUnimPanel2Button ma con due etichette testuali invece dei pulsanti.

TMedpUnimPanelIconLabel
All'interno del panel base, definito con layout orizzontale (hbox), sono presenti nella parte sinistra un'icona e nella parte destra un'etichetta testuale, entrambe dal contenuto personalizzabile.

TMedpUnimPanel4Label
Incapsula quattro etichette testuali all'interno del panel base con layout float. La configurazione di default prevede una disposizione degli elementi su due righe, ogni etichetta occupa il 50% dello spazio orizzontale disponibile.

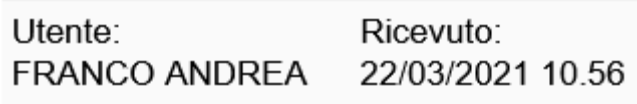


Figura 3.10 – Un esempio di componente TMedpUnimPanel4Label

TMedpUnimPanel4LabelsIcon
Incapsula un componente di tipo TMedpUnimPanel4Label disposto sul lato sinistro del panel di base e un'icona sul lato destro. La configurazione di default prevede l'utilizzo di un'icona di download della libreria Font Awesome 5.

TMedpUnimPanelSelect
Utilizzato per affiancare un'etichetta testuale ad un campo di input con selezione da elenco a discesa, incapsulando i componenti di tipo TMedpUnimLabel e TMedpUnimSelect. Nella configurazione di default del layout il primo elemento è rappresentato sul lato sinistro ed il secondo sul lato destro, seguendo un orientamento orizzontale (hbox).

TMedpUnimPanelEdit
Utilizzato per affiancare un'etichetta testuale ad un campo di input di testo su una singola riga, incapsula i componenti di tipo TMedpUnimLabel e TMedpUnimEdit. Nella configurazione di default del layout il primo elemento è rappresentato sul lato sinistro ed il secondo sul lato destro, seguendo un orientamento orizzontale (hbox).

TMedpUnimPanelMemo
Utilizzato per affiancare un'etichetta testuale ad un campo di input di testo su più righe, incapsula i componenti di tipo TMedpUnimLabel e TMedpUnimMemo. Nella configurazione di default del layout il primo elemento è rappresentato sul lato sinistro ed il secondo sul lato destro seguendo un orientamento orizzontale (hbox).

MedpUnimPanelNumberEdit
Utilizzato per affiancare un'etichetta testuale ad un campo di input che accetta soltanto numeri, incapsula i componenti di tipo TMedpUnimLabel e TMedpUnimNumberEdit. Nella configurazione di default del layout il primo elemento è rappresentato sul lato sinistro ed il secondo sul lato destro seguendo un orientamento orizzontale (hbox).

TMedpUnimPanelDatePicker
Utilizzato per affiancare un'etichetta testuale ad un campo di input per la selezione di una data, incapsula i componenti di tipo TMedpUnimLabel e TMedpUnimDatePicker. Nella configurazione di default del layout il primo elemento è rappresentato sul lato sinistro ed il secondo sul lato destro seguendo un orientamento orizzontale (hbox).

TMedpUnimPanelEditIcons
Incapsula due icone di tipo TMedpUnimLabelIcon utilizzate per rappresentare due pulsanti, il primo da utilizzare per abilitare la modifica ed il secondo per avviare la cancellazione di un'informazione inserita. Prevede di default un orientamento verticale (vbox) ed imposta la classe CSS activeClick, utilizzata opzionalmente per assegnare un'animazione eseguita sul click dell'utente.

Figura 3.11 – Un esempio di componente TMedpUnimPanelEditIcons

TMedpUnimPanelSlideUp
Rappresenta un contenitore generico su cui è possibile variare l'altezza, utilizzando un apposito pulsante posto nella parte alta del componente. In condizioni normali l'altezza del panel è pari alla dimensione del pulsante, nascondendo quindi il resto del contenuto. È definito come sottoclasse di TMedpUnimPanel e contiene un elemento di tipo TMedpUnimLabelIcon utilizzato come pulsante per innescare la transizione tra i due stati, gestita con un'animazione definita tramite proprietà CSS.

Figura 3.12 – Un esempio di componente TMedpUnimPanelSlideUp con all'interno due pulsanti

TMedpUnimPanelDisclosure
Rappresenta l'elemento base per la creazione di liste di panel, utilizzate ad esempio per elencare le richieste degli iter autorizzativi. Sottoclasse di TMedpUnimPanel, contiene nella parte sinistra un panel vuoto il cui contenuto viene popolato dinamicamente con i dati voluti e nella parte destra un'icona. Il metodo pubblico InsertBasePanel riceve come parametro un oggetto di tipo TMedpUnimPanel e consente di iniettare nel componente la rappresentazione grafica dei dati da visualizzare al suo interno. Sfruttando l'evento OnClick della superclasse è possibile eseguire un event handler specifico in corrispondenza del click da parte dell'utente. Sono definite inoltre le proprietà Key e Data, la prima di tipo Variant e la seconda di tipo TObject, entrambe consentono di assegnare al componente visuale dei dati che potranno essere utilizzati in seguito (ad esempio un identificativo del record rappresentato dall'elemento).

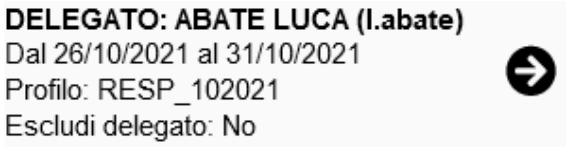


Figura 3.13 – Un esempio di componente TMedpUnimPanelDisclosure

TMedpUnimPanelDownload
Sottoclasse di TMedpUnimPanelDisclosure, imposta in automatico un'icona di download della libreria Font Awesome 5 e assegna la classe CSS activeClick, utilizzata per definire animazioni personalizzate sul componente in corrispondenza del click dell'utente.

TMedpUnimPanelDataDaA
Utilizzato per l'inserimento di un intervallo di date, presenta in disposizione orizzontale due TMedpUnimDatePicker separati da un'etichetta testuale personalizzabile. Sul lato sinistro è inoltre presente un'icona la cui pressione comporta lo svuotamento di entrambi i campi di input, resettando i dati inseriti nel componente.

Figura 3.14 – Un esempio di componente TMedpUnimPanelDataDaA

TMedpUnimPanelOraDaA
Utilizzato per l'inserimento di un intervallo di orari, contiene in disposizione orizzontale due TMedpUnimTimePicker separati da un'etichetta testuale personalizzabile.

Figura 3.15 – Un esempio di componente TMedpUnimPanelOraDaA

TMedpUnimPanelListaElem
Rappresenta una lista di panel con orientamento verticale. Il metodo pubblico Add riceve come parametro un TMedpUnimPanel e consente l'inserimento di elementi alla lista, disposti dall'alto verso il basso. È disponibile una proprietà pubblica di tipo Integer che rappresenta il conteggio degli elementi presenti nella lista. Inoltre sono presenti tre proprietà con visibilità published, che consentono di impostare a design time il colore dello sfondo degli elementi con numero pari, dispari ed il colore del bordo separatore.

TMedpUnimPanelListaDisclosure
Sottoclasse di TMedpUnimPanelListaElem, estende le funzionalità della superclasse consentendo di gestire in modo specifico l'inserimento di elementi di tipo TMedpUnimPanelDisclosure. Il medodo Add consente di specificare una serie di parametri caratteristici che andranno a configurare il componente che viene creato e inserito nella lista. È presente il metodo AddHeader che inserisce un panel di intestazione come separatore tra gli elementi della lista (creando diverse sezioni ognuna con il suo header) e due proprietà da utilizzare per impostare il colore dello sfondo ed il BoxModel da applicare a tutti elementi di questa tipologia.

Figura 3.16 – Un esempio di componente TMedpUnimPanelListaDicolsure con due header

TMedpUnimPanelListaDettaglio

Sottoclasse di TMedpUnimPanelListaDisclosure, estende le funzionalità della superclasse aggiungendo i metodi Add2Labels e AddIconLabel. Il primo consente l'inserimento di un elemento di tipo TMedpUnimPanel2Label ed il secondo di tipo TMedpUnimPanelIconLabel, in entrambi i casi il nuovo elemento creato viene configurato con i parametri passati al metodo.

Figura 3.17 – Un esempio di componente TMedpUnimPanelListaDettaglio

TMedpUnimPanelNumElem

È utilizzato per visualizzare il conteggio di un numero di elementi, ad esempio abbinato a un componente di tipo TMedpUnimPanelListaElem. Contiene un'etichetta testuale nella parte sinistra e un'icona che può essere utilizzata come pulsante nella parte destra. Il testo contenuto nell'etichetta viene aggiornato automaticamente impostando la proprietà NumElementi con un valore numerico, utilizzando le stringhe impostate sulle proprietà CaptionZero (se il numero di

elementi è zero, come “Nessuna richiesta”), CaptionSingolare (se è presente un elemento, come “1 richiesta”) e CaptionPlurale (se il numero id elementi è maggiore di uno, come “5 richieste”).

Figura 3.18 – Un esempio di componente TMedpUnimPanelNumElem

TMedpUnimPanelHeaderDett
È utilizzato come intestazione nelle pagine di dettaglio e contiene, da sinistra verso destra, un’icona utilizzata come pulsante per tornare alla scheda precedente, un’etichetta in cui inserire un testo personalizzato, due icone utilizzate come pulsanti per navigare tra gli elementi disponibili. A tutte le icone viene assegnata la classe CSS activeClick, utilizzata per definire animazioni personalizzate in corrispondenza del click dell’utente.

Figura 3.19 – Un esempio di componente TMedpUnimPanelHeaderDett

TMedpUnimPanelNota
Incapsula un componente di tipo TMedpUnimMemo per l’inserimento di una nota testuale, con l’aggiunta di un’intestazione nella parte alta del panel. Quest’ultima contiene due etichette testuali personalizzabili e un’icona utilizzata per abilitare il campo di inserimento e modifica del testo. Gestisce inoltre in modo automatico l’inserimento del testo “(nessuna nota)” nel caso il campo di input sia vuoto.

Figura 3.20 – Un esempio di componente TMedpUnimPanelNota

TMedpUnimPanelMessaggio
È utilizzato come elemento di base nella rappresentazione di una lista di messaggi. Utilizzando il costruttore è possibile configurare l’oggetto creato specificando il mittente, l’oggetto, la data di invio, la presenza di allegati e la lettura obbligatoria del messaggio. Graficamente queste informazioni comportano la creazione di una serie di etichette testuali ed icone all’interno del panel base, utilizzando elementi di tipo TMedpUnimPanel2Labels, TMedpUnimLabelIcon e TMedpUnimLabel.

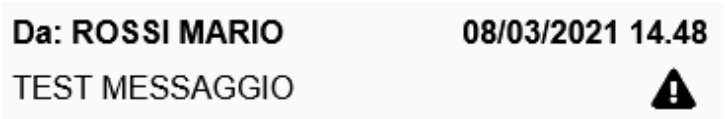


Figura 3.21 – Un esempio di componente TMedpUnimPanelMessaggio

TMedpUnimPanelAllegato

È utilizzato come elemento di base nella rappresentazione di una lista di allegati. Utilizzando il costruttore è possibile configurare l'oggetto creato specificando il nome del file, la dimensione e le note associate al documento. All'interno del panel base, da sinistra verso destra, sono presenti: un oggetto di tipo TMedpUnimPanelEditIcons utilizzato per abilitare la modifica delle caratteristiche dell'allegato o la cancellazione dello stesso, le etichette testuali per i dati associati e un'icona che può essere utilizzata come pulsante per avviare il download dell'allegato.

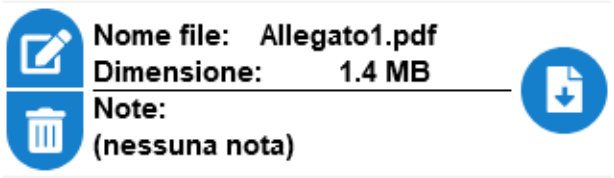


Figura 3.22 – Un esempio di componente TMedpUnimPanelAllegato

TMedpUnimMainMenu

È utilizzato per la rappresentazione del menu principale dell'applicazione, fornendo alcuni metodi e proprietà per la sua gestione. Attraverso la proprietà Title è possibile di impostare un titolo all'interno di un'etichetta nella parte alta del panel base, mentre le funzionalità disponibili vengono visualizzate all'interno di un oggetto di tipo TMedpUnimPanelListaElem. La singola funzionalità è rappresentata graficamente da un'etichetta testuale e da un'icona utilizzata per visualizzare con immediatezza il numero di notifiche. Utilizzando il metodo pubblico Add è possibile aggiungere una voce all'elenco specificando nei parametri il nome, il numero di notifiche da visualizzare, il tag numerico assegnato alla funzionalità e l'event handler da eseguire quando l'utente seleziona l'elemento. Viene inoltre fornito il metodo UpdateNotifica che consente l'aggiornamento del numero di notifiche per una specifica voce anche dopo la creazione, identificata dal tag numerico.

Figura 3.23 – Un esempio di componente TMedpUnimMainMenu

3.4 Architettura generale dell'applicazione

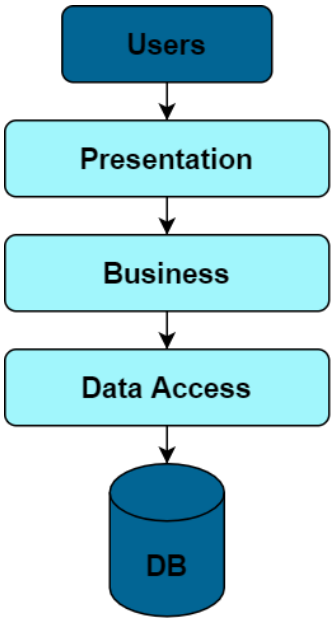


Figura 3.24 – La rappresentazione dei tre livelli dell'applicazione

Nella definizione dell'architettura generale dell'applicazione ho dovuto tenere conto di molteplici fattori: la struttura del progetto proposta dal framework UniGUI, la possibilità di utilizzare un web service aziendale per l'accesso ai dati di alcune delle funzionalità (il web service B110 descritto in seguito) e la necessità di interagire direttamente con il database Oracle per le funzionalità non coperte dal servizio. Allo stesso tempo ho però voluto mantenere una netta separazione tra il codice utilizzato per la creazione e la gestione dell'interfaccia grafica ed il codice che implementa la logica applicativa, con l'obiettivo di creare componenti riutilizzabili ed indipendenti dal framework UniGUI, facilitando la manutenzione ed il riuso del codice prodotto.

Il risultato ottenuto prevede una separazione concettuale dei componenti software in un'architettura a tre livelli, formata da:

Presentation layer

È formato da tutti i componenti visuali del framework UniGUI che costituiscono l'interfaccia grafica con cui interagisce l'utente tramite il browser, composta da un insieme di form (sottoclassi di TUnimForm) e di frame (sottoclassi di TUniFrame) che contengono a loro volta tutti gli elementi della pagina visualizzata. L'interazione con l'utente è realizzata implementando gli event handler dei componenti e la gestione della logica applicativa, così come il mantenimento dello stato dell'applicazione, è delegata agli oggetti del livello sottostante che vengono istanziati nel costruttore. I dati e le operazioni che l'utente può eseguire sono accessibili tramite le proprietà e le funzioni pubbliche delle classi del livello sottostante.

Business logic layer

È formato da componenti (sottoclassi di TObject o TDataModule) totalmente indipendenti dal framework UniGUI, con l'obiettivo di facilitare il riuso delle classi create e separare la complessità della gestione dell'interfaccia grafica dalla logica applicativa. Le istanze delle classi del business logic layer mantengono al loro interno lo stato dell'applicazione, consentendo ai componenti del livello superiore l'accesso ai dati ed alle operazioni richieste dalla logica applicativa. A loro volta, alle classi di questo livello è negato l'accesso diretto ai dati manipolati dal livello sottostante, che deve avvenire tramite le funzioni pubbliche delle classi del data access layer. Il nome delle classi appartenenti a questo livello termina con il suffisso "MW" (ad esempio TWM001FLoginMW, assegnato alla classe utilizzata per implementare l'autenticazione dell'utente).

Data access layer

È formato dai componenti (sottoclassi di TDataModule) che gestiscono l'interazione con il database e consentono al livello superiore l'accesso ai dati e la loro manipolazione. Il principio applicato è la separazione della complessità della logica applicativa dal codice richiesto per l'accesso alle informazioni, esponendo al livello superiore un serie di funzioni pubbliche che

ritornano i dati richiesti, quando possibile senza mantenere lo stato delle funzionalità a loro interno. I dati gestiti vengono ritornati dalle funzioni al livello superiore senza essere conservati nel data access layer. Anche questo livello è indipendente dal framework UniGUI e a sua volta indipendente dalle classi del business logic layer. Il nome delle classi appartenenti a questo livello termina con il suffisso "DM" (ad esempio TWM018FMessaggisticaDM, assegnato alla classe utilizzata per l'interazione con il database nella funzione Messaggistica).

In un'architettura come quella descritta, per evitare memory leak nella creazione e nella distruzione delle istanze delle classi utilizzate, ogni livello gestisce il ciclo di vita degli oggetti del livello immediatamente inferiore. In questo modo gli oggetti del business logic layer vengono sempre creati e distrutti dai componenti del presentation layer, mentre gli oggetti del data access layer vengono sempre creati e distrutti dai componenti del business logic layer. Il ciclo di vita dei componenti del presentation layer (il livello più alto) è invece gestito tramite il meccanismo di ownership di Delphi o tramite la gestione delle sessioni web propria del framework UniGUI, assegnando il MainModule come owner delle classi istanziate.

3.4.1 Il web service B110PWSMobile

Come anticipato nella descrizione dell'architettura di base, nell'applicazione viene utilizzato il web service B110PWSMobile (in seguito chiamato semplicemente B110) che consente di snellire lo sviluppo riutilizzando il web service REST già prodotto in azienda. È disponibile sia in versione eseguibile (utile nella fase di sviluppo) sia in versione applicazione ISAPI da eseguire nel web server Microsoft IIS (utilizzato invece in produzione, insieme alla web app stessa).

Il B110 è stato sviluppato utilizzando il framework Datasnap (specifico per la realizzazione di web service REST nel linguaggio Delphi) e offre un'interfaccia semplice per eseguire operazioni sui dati presenti nel database del sistema informativo Iris, nascondendo la complessità delle elaborazioni ed evitando la duplicazione di codice.

Una volta avviato il web service, il framework Datasnap offre la possibilità di generare automaticamente una classe Delphi che funge da proxy, esponendo un metodo pubblico per ogni risorsa disponibile sul server Datasnap. La classe incapsula inoltre la gestione sincrona della richiesta HTTP, le operazioni di marshalling dei parametri passati al metodo (da oggetti Delphi a stringa in formato json) e unmarshalling del contenuto della risposta ricevuta dal server (da stringa in formato json a oggetto Delphi). Le rappresentazioni delle risorse scambiate tra client e server sono definite come sottoclasse di TPersistent nella unit B110USharedTypes condivisa tra le due parti. Tutti i metodi così generati condividono i seguenti parametri:

- InfoClient: di tipo TInfoClient, contiene informazioni relative alla versione dell'applicazione chiamante del sistema informativo Iris, consente al server di negare l'accesso ai client che presentano una versione non compatibile.
- DatiLogin: di tipo TParametriLogin, contiene informazioni utilizzate dal server per autenticare l'utente che esegue le operazioni, negando l'accesso se non consentito.
- FiltroRichieste: di tipo TFiltroRichieste, presente in tutti i metodi utilizzati per l'acquisizione delle richieste degli iter autorizzativi, consente di specificare il filtro da

applicare sul risultato richiesto, che può essere di tipo temporale (data da/a) o sullo stato della richiesta (da autorizzare/autorizzata/negata).

L'oggetto ritornato da tutti i metodi generati è di tipo TRisultato e contiene informazioni sullo stato della richiesta HTTP, con l'aggiunta di eventuali descrizioni dell'errore riscontrato lato server (attraverso un'istanza della classe TInfoElaborazione) ed una serie di campi utilizzati per incapsulare i dati richiesti (sotto forma di dataset o sottoclassi di TPersistent).

I metodi utilizzati dall'applicazione IrisAPP consentono l'implementazione di alcune delle funzionalità elencate in precedenza senza ulteriori interazioni con il database. La classe proxy generata, il cui tipo è TB110FServerMethodsDMClient contiene i seguenti metodi pubblici:

- DatiGenerali: utilizzato per l'autenticazione dell'utente web che deve accedere all'applicazione. In caso di esito positivo, ritorna nel risultato un oggetto di tipo TOutputDatiGenerali contenente i dati associati all'utente autenticato, come i dati anagrafici, i profili disponibili e le abilitazioni alle funzionalità dell'applicazione per il profilo corrente.
- RichiesteGiust: ritorna un dataset contenente le richieste di giustificativo visibili all'autorizzatore richiedente, filtrate in base al parametro FiltroRichieste. Opzionalmente può essere ritornato soltanto il numero di richieste che soddisfano il filtro specificato.
- AutorizzaRichiestaGiust: utilizzato per l'inserimento dell'autorizzazione sulla richiesta di giustificativo identificata univocamente dall'id numerico passato come parametro, la richiesta può essere autorizzata o negata.
- RichiesteGiustDip: ritorna un dataset contenente le richieste di giustificativo visibili al dipendente richiedente, filtrate in base al parametro FiltroRichieste.
- InsRichiestaGiust: consente al dipendente l'inserimento di una nuova richiesta di giustificativo, passata come parametro di tipo TDatiRichiestaGiust.
- CancRichiestaGiust: consente al dipendente la cancellazione di una richiesta di giustificativo, identificata univocamente dall'id numerico passato come parametro.
- RevocaRichiestaGiust: consente al dipendente la revoca di una richiesta di giustificativo già autorizzata, identificata univocamente dall'id numerico passato come parametro.
- Cartellini: ritorna un dataset contenente la lista dei cartellini mensili disponibili per la stampa nell'intervallo di tempo individuato dai parametri DataInizio e DataFine. Opzionalmente può essere ritornato soltanto il numero di cartellini mensili che soddisfano il filtro specificato.
- Cartellino: ritorna il cartellino per il mese specificato in formato pdf (come array di byte), prodotto lato server con la parametrizzazione di stampa specificata come parametro.
- Cedolini: ritorna un dataset contenente la lista dei cedolini paga disponibili per la stampa nell'intervallo di tempo individuato dai parametri DataInizio e DataFine. Opzionalmente può essere ritornato soltanto il numero di cedolini che soddisfano il filtro specificato.
- Cedolino: ritorna il cedolino per il mese specificato in formato pdf (come array di byte).
- ImpostaDataConsegnaCedolino: consente di impostare la data di consegna del cedolino identificato univocamente da un id numerico.

- RichiesteTimb: ritorna un dataset contenente le richieste di timbratura visibili all'autorizzatore, filtrate in base al parametro FiltroRichieste. Opzionalmente può essere ritornato soltanto il numero di richieste che soddisfano il filtro specificato.
- AutorizzaRichiestaTimb: utilizzato per l'inserimento dell'autorizzazione sulla richiesta di timbratura identificata univocamente dall'id numerico passato come parametro, la richiesta può essere autorizzata o negata.
- RichiesteTimbDip: ritorna un dataset contenente le richieste di timbratura visibili al dipendente richiedente, filtrate in base al parametro FiltroRichieste.
- TimbModificabili: ritorna un dataset contente le timbrature del giorno passato come parametro per il dipendente richiedente.
- InsRichiestaTimb: consente al dipendente l'inserimento di una nuova richiesta di timbratura, passata come parametro di tipo TDatiRichiestaTimb.
- CancRichiestaTimb: consente al dipendente la cancellazione di una richiesta di timbratura, identificata univocamente dall'id numerico passato come parametro.
- NoteRichiesta: ritorna un oggetto di tipo TNoteRichiesta contenente le note associate ad una richiesta identificata univocamente dall'id numerico passato come parametro.
- SetNoteRichiesta: consente la modifica delle note di una richiesta identificata univocamente dall'id passato come parametro. Il contenuto ed il livello della nota da modificare sono specificati nel parametro di tipo TNoteLivello.
- RiepilogoAssenze: ritorna un oggetto di tipo TOutputRiepilogoAssenze contenente le competenze totali, fruite e residue per il dipendente, il giustificativo per il giorno specificato.
- Dipendenti: ritorna un dataset che contiene l'elenco dei dipendenti gestiti dal responsabile richiedente, contenente anche informazioni anagrafiche, oltre alle timbrature ed i giustificativi in corso.
- DatiGiornalieri: ritorna un oggetto di tipo TOutputDatiGiornalieri contenente informazioni sul dipendente richiedente come il debito orario, le ore rese, il saldo attuale, le timbrature del giorno e i giustificativi di assenza validi per il giorno passato come parametro, oltre alle eventuali anomalie nelle timbrature dei sette giorni precedenti.
- Timbrature: ritorna un dataset contenente la lista delle timbrature del dipendente richiedente per il giorno passato come parametro.
- InsTimbratura: consente l'inserimento di una nuova timbratura al dipendente richiedente nell'ora e giorno corrente utilizzando il parametro di tipo TDatiTimbratura.
- Dizionario: è un metodo generico che consente l'accesso alle tabelle del dizionario dati, può ad esempio essere utilizzato per acquisire le causali di assenza o i rilevatori virtuali visibili al dipendente richiedente.

La classe di tipo TB110FServerMethodsDMClient generata automaticamente viene istanziata all'interno di un oggetto di tipo TB110FClientModule, quest'ultimo consente di gestire i parametri di connessione al server DataSnap (host, url, protocollo e porta TCP) e di definire delle funzioni di utilità generale per l'interazione con il server.

Nel contesto specifico dell'architettura di base del progetto descritta in precedenza, il B110 si colloca nel data access layer, incapsulando le operazioni di accesso e manipolazione dei dati richiesti dall'applicazione, senza mantenerne una copia al suo interno. Un oggetto di tipo TB110FClientModule viene creato per ogni sessione web ed è condiviso da tutti i componenti del presentation layer della sessione creata, viene infatti passato come parametro ai vari metodi che lo utilizzano.

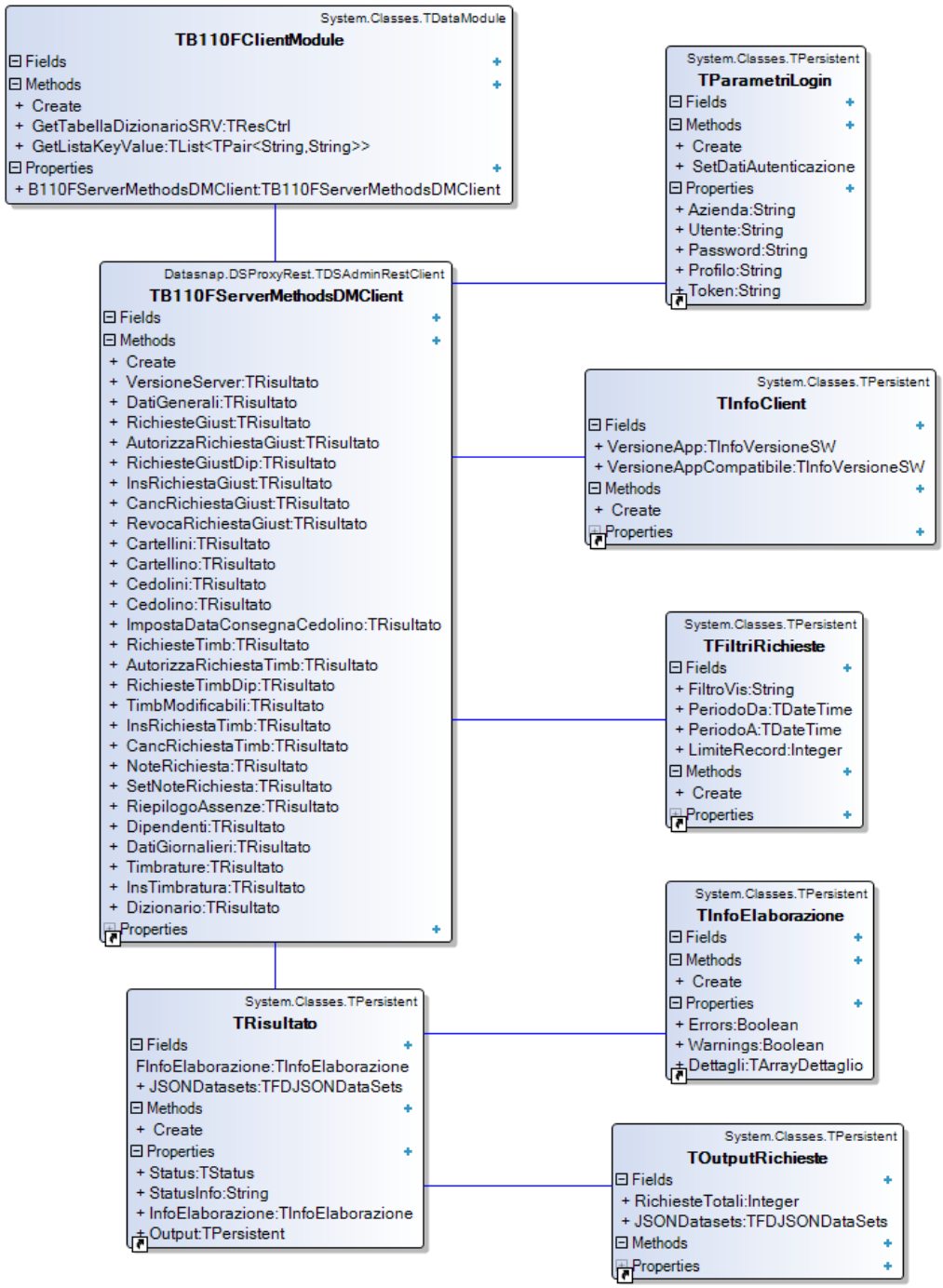


Figura 3.25 – Il class diagram con la rappresentazione delle classi coinvolte nella comunicazione con il servizio B110

3.5 Le classi del progetto

Le classi realizzate per l'implementazione del progetto ricalcano l'architettura proposta dal framework UniGUI descritta in precedenza, adattata in base alle esigenze del caso specifico ed alle scelte progettuali fatte nelle fasi iniziali dello sviluppo dell'applicazione, in particolare tenendo conto dell'architettura a tre livelli adottata.

3.5.1 Funzionalità di base

I componenti utilizzati per le funzionalità di base dell'applicazione sono i seguenti:

TWM000FServerModule
Il ServerModule dell'applicazione è rappresentato dalla classe TWM000FServerModule, che eredita da TUniGUIServerModule. Dopo la creazione dell'oggetto corrispondente (come singleton visibile globalmente dai componenti principali del framework), nell'event handler UniGUIServerModuleCreate vengono caricati i parametri di configurazione del server, acquisiti da due file ini creando un'istanza della classe TWM000FParConfig. Nelle proprietà impostate a design time troviamo ad esempio i CSS personalizzati per l'applicazione e le impostazioni del ServerLogger, che consente di registrare su file di testo delle informazioni relative a possibili eccezioni avvenute durante l'esecuzione del server.

TWM000FParConfig
Un oggetto di tipo TWM000FParConfig viene istanziato nel ServerModule all'interno dell'event handler UniGUIServerModuleCreate, innescato dopo l'avvio dell'applicazione. Contiene la configurazione comune a tutte le sessioni web definita a livello di server, accessibile tramite due file ini presenti all'interno della cartella di installazione dell'applicazione. L'istanza della classe viene creata utilizzando il costruttore che riceve come parametri i nomi dei file da caricare (tipicamente IrisAPP.ini e B110.ini) e permette di definire la configurazione del server UniGUI. I parametri utilizzati sono i seguenti:

- Azienda: l'azienda di default proposta nella form di Login nel contesto della gestione multi-azienda.
- StatoAzienda: indica lo stato del componente di input per l'inserimento dell'azienda nella form di Login, può assumere i valori "S" (visibile), "N" (non visibile) e "R" (sola lettura).
- Profilo: il profilo di default proposto nella form di Login, solitamente assume il valore "DIPENDENTE".
- StatoProfilo: indica lo stato del componente di input per l'inserimento del profilo nella form di Login, può assumere i valori "S" (visibile), "N" (non visibile) e "R" (sola lettura).
- TitoloIrisAPP: stringa di testo mostrata sotto il logo dell'applicazione nella form di Login, solitamente valorizzata con la ragione sociale dell'azienda cliente.
- MaxSessione: il numero massimo di sessioni web che possono essere gestite in contemporanea dall'applicazione.
- MaxRequests: il numero massimo di richieste HTTP che possono essere gestite in contemporanea dall'applicazione.

- TimeoutSessione: il valore in minuti del timeout oltre il quale la sessione web è considerata inattiva e viene eliminata.
- CookieCredenziali: utilizzato per abilitare il salvataggio delle credenziali nei cookies, può assumere i valori “S” e “N”.
- RecuperoPassword: utilizzato per abilitare la possibilità di richiedere il recupero della password nella form di login, può assumere i valori “S” e “N”.
- PaginaIniziale: contiene una stringa di testo utilizzata come identificativo della funzionalità che deve essere avviata in automatico dopo il login dell’utente.
- LoginEsterno: utilizzato per abilitare l’utilizzo di una modalità di login esterno, può assumere i valori “S” e “N”.
- Home: contiene l’url della pagina a cui l’utente deve essere reindirizzato dopo il logout dall’applicazione, può essere utilizzato in caso di login esterno con la pagina di autenticazione della intranet aziendale.
- HostB110/UrlB110/PortB110/ProtocolloB110: parametri di configurazione per l’utilizzo del web service B110.
- LogIndyExceptions: utilizzato per abilitare la registrazione dei log associati alla gestione delle richieste HTTP, può assumere i valori “S” e “N”.
- LogSSLIndyExceptions: utilizzato per abilitare la registrazione dei log associati al protocollo TLS/SSL nel caso si utilizzi HTTPS, può assumere i valori “S” e “N”.
- LogSessionExceptions: utilizzato per abilitare la registrazione dei log associati alla gestione delle sessioni web, può assumere i valori “S” e “N”.

TWM000FMainModule

Il MainModule dell’applicazione è rappresentato da un’istanza della classe TWM000FMainModule che eredita da TUniGUIMainModule. Le proprietà impostate a design time consentono (ad esempio) di scegliere il tema grafico adottato dai componenti ExtJS della sessione e di personalizzare i messaggi mostrati all’utente in caso di errori nella sessione (come nel caso di sessione scaduta). Essendo il MainModule istanziato per ogni sessione web creata, è utilizzato per incapsulare i dati riservati alla sessione corrente, in particolare al suo interno si trovano gli oggetti:

- InfoLogin: di tipo TWM001FLoginMW, contiene i dati relativi all’utente autenticato e consente l’autenticazione tramite il servizio B110.
- Notifiche: di tipo TWM009FNotificheMW, è utilizzato per l’acquisizione delle notifiche visualizzate nella pagina principale.
- B110ClientModule: di tipo TB110FClientModule, è l’istanza della classe utilizzata per l’accesso al servizio B110, unica per la sessione.

Molto importante per la gestione dell’autenticazione dell’utente è l’event handler UniGUIMainModuleBeforeLogin, eseguito quando l’utente richiede la creazione di una sessione web, ovvero prima di mostrare la LoginForm. Consente di saltare la fase di autenticazione esplicita dell’utente e di visualizzare direttamente la MainForm dell’applicazione, impostando il parametro booleano Handled al valore true. Questa possibilità è stata sfruttata per implementare le modalità di autenticazione automatica tramite cookies, tramite parametri nella richiesta HTTP e OAuth 2.0

(fase 2). Nel primo caso vengono decifrati i parametri di autenticazione presenti nei cookies della richiesta HTTP ed utilizzati per il login sul servizio B110. Nel secondo caso gli stessi parametri vengono acquisiti dalla querystring (in caso di richiesta HTTP a metodo GET) o dal corpo del messaggio (in caso di richiesta HTTP a metodo POST con content-type application/x-www-form-urlencoded). Infine, nel terzo caso viene letto dai cookies l'access token richiesto per l'autenticazione OAuth 2.0 (fase 2) sull'authentication server aziendale, seguita poi comunque da un'autenticazione fittizia sul servizio B110 necessaria per recuperare i parametri dell'utente.

TWM001FLoginMW
La classe TWM001FLoginMW viene utilizzata per la gestione dei dati che caratterizzano l'utente e per l'autenticazione dello stesso utilizzando il web service B110. Tramite le proprietà pubbliche sono accessibili i campi:

- ParametriLogin: di tipo TParametriLogin, contiene i dati inseriti dall'utente in fase di autenticazione (utente, password, azienda, profilo).
- DatiGenerali: di tipo TOutputDatiGenerali, rappresenta i dati ritornati dall'operazione di autenticazione sul servizio B110, al suo interno si trovano le abilitazioni per le varie funzioni dell'applicazione, i profili disponibili e i dati anagrafici dell'utente.
- SessioneIrisWin: di tipo TSessioneIrisWin, è un oggetto utilizzato per l'interazione con il database e per accede a dati di configurazione comuni a tutti gli applicativi del sistema informativo Iris, ulteriori dettagli su questa classe sono forniti in seguito.

Il metodo LoginB110 incapsula l'operazione di autenticazione dell'utente con il servizio B110, eseguendo prima la verifica sulla compatibilità della versione del web service con la web app e poi la chiamata del metodo DatiGenerali dell'interfaccia TB110FServerMethodsDMClient.

Le funzioni AggiornaSessioneIrisWin consentono invece di completare l'oggetto SessioneIrisWin in uso, dopo l'autenticazione avvenuta con successo vengono infatti aggiunte le informazioni dell'utente acquisite direttamente dal database.

TWM009FNotificheMW
È la classe che consente di acquisire il numero di notifiche da visualizzare nel menu principale per le varie funzionalità. Il metodo AggiornaNotifiche viene utilizzato per inviare le richieste al servizio B110 o per acquisire le informazioni direttamente dalla base dati, in base alle abilitazioni sulle funzioni specifiche dell'utente chiamate e la disponibilità della relativa funzione sul web service. Si tratta di un'operazione che può richiedere alcuni secondi, di conseguenza, il suo utilizzo deve essere integrato nell'applicazione in modo da non risultare troppo impattante per la navigazione da parte dell'utente.

TSessioneIrisWin
È una classe fondamentale per l'integrazione dell'applicazione con il sistema informativo Iris ed è utilizzata da tutti gli applicativi che lo compongono. Consente la gestione dell'interazione con la base dati comune e l'accesso ad una serie di parametri che caratterizzano l'azienda cliente.

Un'istanza della classe TSessioneIrisWin viene creata per ogni sessione web, al suo interno sono incapsulati una serie di campi con visibilità pubblica:

- SessioneOracle: di tipo TOracleSession, è utilizzato per rappresentare la sessione avviata con l'autenticazione dell'utente sulla base dati del sistema informativo Iris, viene utilizzato da tutti i componenti che interagiscono con il database (come TOracleDataset e TOracleQuery).
- Parametri: di tipo TParametri, contiene i parametri aziendali che consentono di personalizzare il comportamento delle applicazioni in base alle esigenze del cliente e una serie di informazioni associate all'utente autenticato. Alcuni esempi sono la modalità di autenticazione scelta dall'azienda e la sua configurazione, alcuni parametri specifici per la configurazione delle funzionalità, la matricola e il codice fiscale dell'utente che ha eseguito l'accesso al programma.
- RegistraLog/RegistraMessaggi: entrambi di tipo TRegistraLog, attraverso una serie di metodi pubblici consentono la registrazione delle informazioni di log e dei messaggi delle elaborazioni nelle tabelle apposite del database.

L'utilizzo di un'istanza di questa classe si è resa necessaria durante lo sviluppo delle funzionalità che non utilizzano il servizio B110, sono quindi presenti al suo interno informazioni accessibili anche tramite la classe TOutputDatiGenerali.

TWM001FLogin

La LoginForm dell'applicazione è rappresentata dalla classe TWM001FLogin che eredita da TUnimLoginForm. Contiene tutti i componenti visuali necessari per la gestione dell'autenticazione dell'utente ed una serie di metodi privati utilizzati nell'event handler btnAccediClick per validare le credenziali inserite, sia in caso di login esterno che in caso di login interno. L'autenticazione viene gestita utilizzando l'oggetto InfoLogin del MainModule descritto in precedenza e prevede nell'ordine le seguenti operazioni:

1) Creazione dell'oggetto SessioneIrisWin associato alla sessione web corrente.
2) Autenticazione OAuth 2.0 fase 1 e 2 (opzionale, se richiesta da parametro aziendale).
3) Verifica della compatibilità tra la versione dell'applicazione e la versione del web service B110 utilizzando il servizio VersioneServer.
4) Autenticazione interna o esterna utilizzando il servizio DatiGenerali del web service B110, le modalità di autenticazione esterna alternative alla OAuth 2.0 sono già incapsulate nella chiamata al B110, è sufficiente inviare le credenziali al servizio che in base ai parametri di configurazione le utilizza nella modalità corretta.

La chiusura della form rappresenta l'autenticazione dell'utente avvenuta con successo, avviene impostando la proprietà ModalResult al valore 1 e porta all'apertura della MainForm. In concomitanza con la chiusura, se richiesto dall'utente utilizzando l'apposito togglebutton, vengono salvate nei cookies le credenziali utilizzate per l'autenticazione, cifrate con crittografia simmetrica per impedire l'accesso in chiaro lato client.

TWM001FLoginOAuth2MW

L'implementazione dell'autenticazione OAuth 2.0 è realizzata con la classe TWM001FLoginOAuth2MW che incapsula i dati necessari per la gestione delle richieste HTTP e fornisce due metodi pubblici:

- Login: utilizzato per gestire in successione entrambe le fasi del protocollo, ovvero la creazione dell'access token dell'utente con username e password e l'autenticazione vera e propria, utilizzando il token generato presso l'authentication server aziendale.
- LoginFase2: utilizzato per la sola fase 2, in presenza del token non ancora scaduto (passato come parametro) consente la sola autenticazione presso l'authentication server aziendale.

La classe TOAuth2Token è stata creata per incapsulare le informazioni associate al token di autenticazione.

TWM001FRecuperoPasswordMW

Per l'implementazione della funzionalità di recupero password è stata creata la classe TWM001FRecuperoPasswordMW, utilizzata per inviare l'e-mail di recupero all'indirizzo di riferimento dell'utente presente nel sistema informativo Iris. Dopo aver fatto le prime verifiche sui dati forniti dall'utente, come l'esistenza di azienda e username, ed aver escluso l'utilizzo di una modalità di autenticazione esterna, gestisce l'interazione con il server SMTP aziendale acquisendo la configurazione presente nei parametri dell'applicazione. È possibile utilizzare per l'e-mail inviata un testo predefinito in formato html o plaintext, a seconda delle preferenze dell'azienda cliente.

TWM000FMain

La MainForm dell'applicazione è rappresentata dalla classe TWM000FMain che eredita da TUnimForm. Contiene diverse sezioni la cui visibilità dipende dal tipo di dispositivo utilizzato, identificato tramite la proprietà UniPlatform dell'oggetto UniSession associato alla sessione web. Utilizzando uno smartphone viene visualizzato un header ed il menu principale nel resto dello schermo (all'interno del componente pnlMainMenu). Utilizzando un tablet viene visualizzato un header, il menu all'interno di una sidebar sul lato sinistro della pagina (sempre all'interno del componete pnlMainMenu) e la funzionalità richiesta nel centro della pagina (nel componente pnlFrame). In entrambi i casi l'header, contenente il titolo, la versione dell'applicazione, il pulsante di logoff e di accesso alla form di gestione degli utenti, è gestito utilizzando le proprietà fornite dalla superclasse TUnimForm, che già prevede la presenza di un'intestazione nelle form dell'applicazione.

Il menu principale è formato dal componente personalizzato pnlMainMenu di tipo TMedpUnimMainMenu, che consente la selezione delle funzionalità disponibili in base alle abilitazioni specifiche del profilo utente utilizzato. L'apertura delle funzionalità selezionate viene gestita in modo diverso in base al tipo di dispositivo rilevato. Per gli smartphone viene utilizzata la funzione ShowFrameForm, che apre il frame della funzione all'interno di una nuova form (denominata FrameForm) sovrapposta per intero alla MainForm, mentre per i tablet viene utilizzata la funzione ShowFrame, che sostituisce il frame visualizzato all'interno del componente pnlFrame nella MainForm stessa. L'aggiornamento delle notifiche visualizzate nel menu principale viene gestito utilizzando una chiamata non bloccante alla prima apertura della MainForm e ad ogni chiusura delle funzionalità selezionate. Trattandosi di un'elaborazione che richiede un tempo misurabile in secondi, la chiamata non bloccante dell'aggiornamento delle notifiche permette di non rallentare la navigazione dell'utente nell'applicazione.

Dopo la creazione della MainForm, nell'event handler UnimFormCreate vengono valutate tutte le condizioni che possono portare all'avvio automatico di una funzionalità, come l'obbligo di sostituire la password che apre la pagina di gestione dell'utente o la presenza di messaggi obbligatori da leggere che avvia la funzione Messaggistica.

Inoltre, attraverso la MainForm è possibile accedere alla form di Gestione Utente, in caso di cambio profilo viene utilizzata la callback CambioProfiloCallBack per aggiornare il menu principale e le notifiche con i nuovi valori per il profilo selezionato.

L'operazione di logoff, gestita con l'event handler UnimFormClose, gestisce il logout e la terminazione della sessione web senza eliminare le credenziali eventualmente salvate nei cookies.

TWM000FFrameForm
Sottoclasse di TUnimForm, rappresenta la form utilizzata per ospitare il frame della funzionalità selezionata dall'utente nel caso venga identificato un accesso da smartphone. Non presenta particolari metodi ma la sua configurazione impostata a design time definisce la base del layout di pagina utilizzato per visualizzare il frame.

TWM002FInfo
La form utilizzata per la funzionalità di gestione dell'utente è rappresentata dalla classe TWM002FInfo, sottoclasse di TUnimForm. L'operazione di cambio del profilo richiede una nuova autenticazione dell'utente sul web service B110 e la successiva chiusura della form con valore di ritorno pari a 1. Le operazioni di modifica della password e dei contatti vengono delegate ad un'istanza della classe TWM001FCambioPasswordMW creata nell'event handler UnimFormCreate. La visibilità delle due sezioni utilizzate per l'inserimento dei dati dipende dall'utilizzo dell'autenticazione interna (con password gestita dallo stesso sistema informativo Iris) e dall'abilitazione specifica dell'utente alla funzionalità "Gestione Sicurezza", impostata sul filtro funzioni dell'utente web. La form viene inoltre utilizzata nel caso l'utente debba obbligatoriamente sostituire la password, in questo caso ogni operazione viene inibita finché l'utente non procede con l'operazione richiesta, disabilitando il pulsante di chiusura della form e nascondendo le sezioni di cambio profilo e modifica dei contatti.

TWM001FCambioPasswordMW
L'operazione di modifica della password utilizzata dall'utente (consentita solamente in caso di autenticazione interna) e di modifica dei contatti viene implementata utilizzando un'istanza della classe TWM001FCambioPasswordMW. Nel costruttore vengono acquisiti i contatti dell'utente direttamente dal database (numero di cellulare, e-mail e PEC), questi dati vengono poi esposti al chiamante mediante le proprietà pubbliche della classe. La modifica dei contatti e della password avviene utilizzando rispettivamente le funzioni AggiornaContatti e AggiornaPassword, preceduta da una validazione dei dati inseriti secondo le regole definite a livello aziendale.

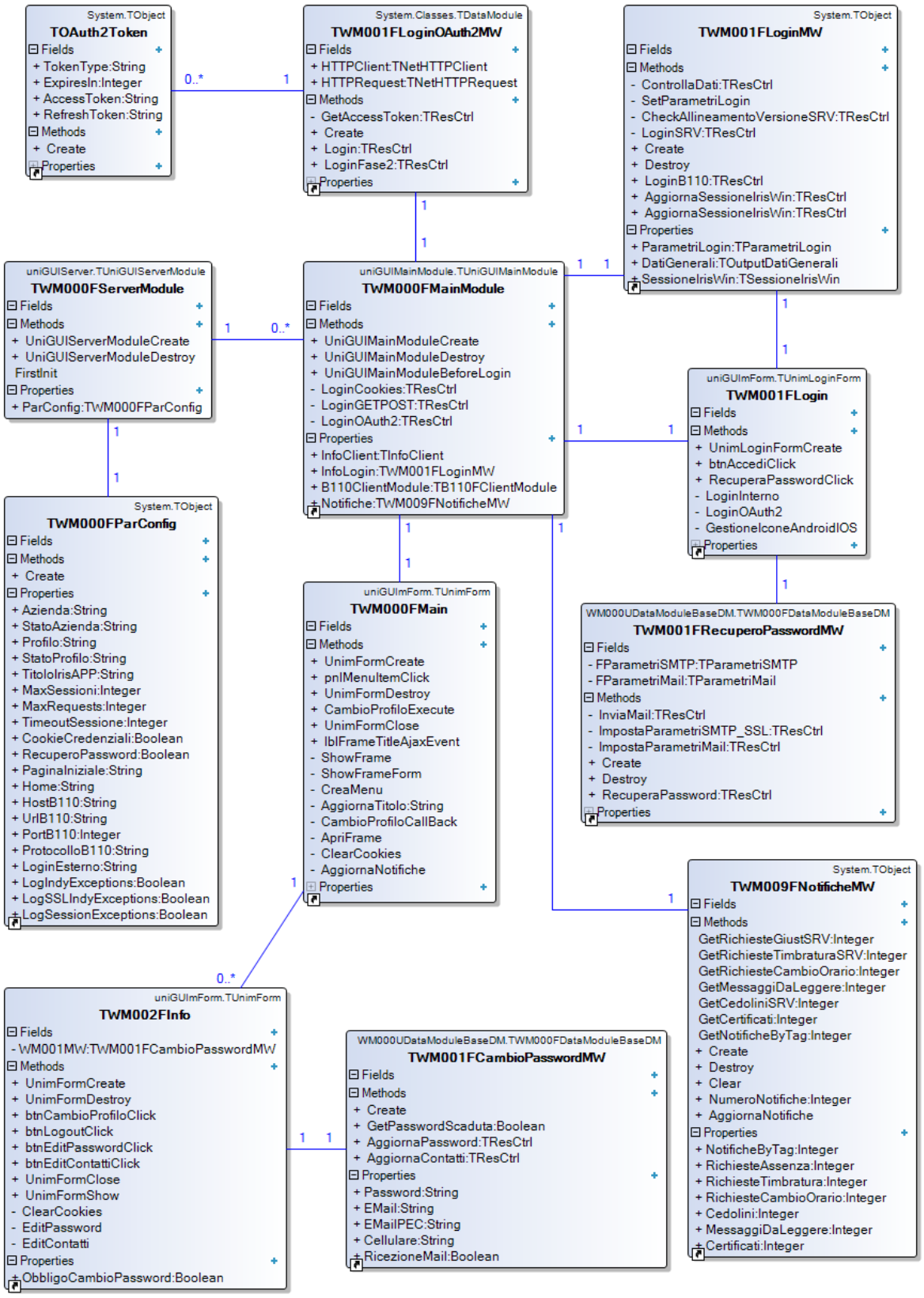


Figura 3.26 – Le classi utilizzate per la gestione delle funzionalità di base dell'applicazione

3.5.2 Funzionalità di interfaccia

Le classi realizzate per l'implementazione delle funzionalità di interfaccia seguono una struttura più facilmente riconducibile all'architettura a tre livelli descritta in precedenza.

Viene fornita prima una descrizione delle superclassi largamente utilizzate nella realizzazione delle funzionalità che compongono l'applicazione.

TWM000FFrameBase
Sottoclasse di TUniFrame, rappresenta la superclasse di tutti i frame delle funzionalità sviluppate. Utilizza le proprietà impostate a design time per definire la configurazione del layout comune a tutte le sottoclassi. Inoltre, utilizzando l'event handler UniFrameDestroy alla distruzione del frame, consente di avviare l'aggiornamento non bloccante delle notifiche presenti sul menu principale, in concomitanza con la chiusura delle funzionalità da parte dell'utente.

TWM000FDataSetMW
Sottoclasse di TObject, rappresenta la superclasse di tutti gli elementi appartenenti al business logic layer che richiedono la gestione di un dataset principale per incapsulare i dati. Permette di utilizzare dataset di tipo TOracleDataSet o TFDMemTable, in base al valore passato come parametro nel costruttore. Fornisce una serie di proprietà e metodi pubblici che permettono di navigare ed accedere ai dati presenti nell'oggetto incapsulato.

TWM000FDataModuleBaseDM
Sottoclasse di TDataModule, rappresenta la superclasse di tutti gli elementi appartenenti al data access layer, incapsulando gli oggetti SessioneIrisWin e SessioneOracle necessari per l'accesso al database. Tramite la funzione privata SetSessioneOracle, chiamata nel costruttore, imposta la proprietà OracleSession su tutti gli oggetti di tipo TOracleDataSet, TOracleQuery e TOracleScript istanziati al suo interno.

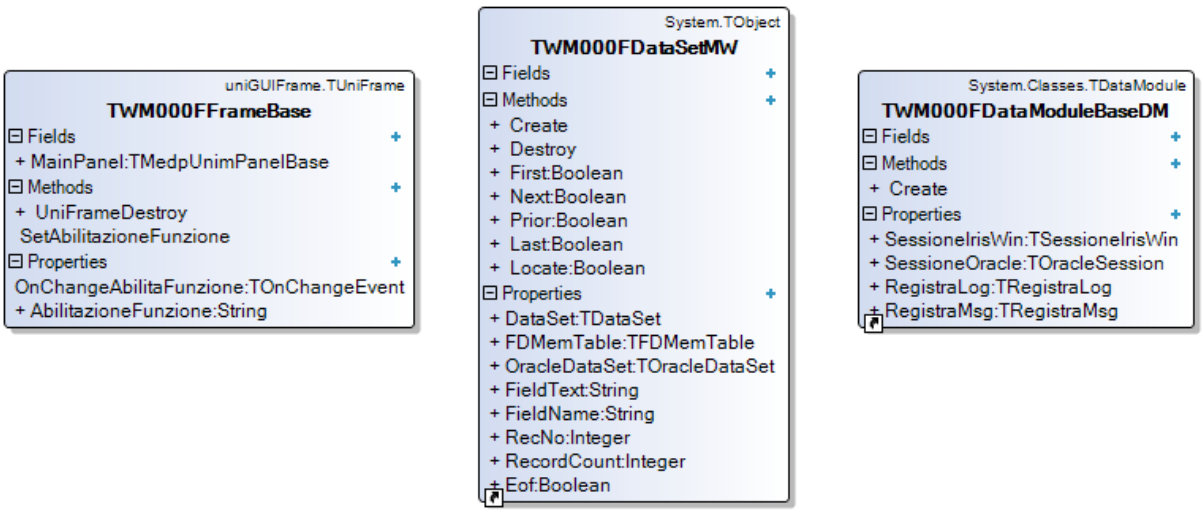


Figura 3.27 – Le superclassi utilizzate per l'implementazione delle varie funzionalità

Dati giornalieri

La funzionalità Dati giornalieri è utilizzata dal dipendente per visualizzare i dati associati al giorno selezionato, come timbrature, giustificativi o anomalie. L'implementazione della funzionalità ha richiesto la creazione di due classi specifiche, una per la gestione dell'interfaccia grafica (presentation layer) e una per incapsulare la logica applicativa (business logic layer). In questo caso è possibile utilizzare il web service B110, non è servita quindi una classe dedicata all'interazione con il database (data access layer).

TWM011FDatiGiornalieriFM

All'interno della classe TWM011FDatiGiornalieriFM (che eredita da TWM000FFrameBase) sono definiti tutti i componenti visuali che compongono l'interfaccia grafica del frame, oltre all'istanza della classe utilizzata per la gestione della logica applicativa. I metodi creati consentono l'aggiornamento dei dati visualizzati nell'interfaccia, in particolare con l'event handler edtDataChange, innescato da un'operazione di modifica della data da parte dell'utente, comporta l'invio della richiesta al web service B110 con la conseguente modifica dei dati visualizzati. La funzione CreaPanelLista viene utilizzata per creare dinamicamente i panel di tipo TMedpUnimPanel2Labels visualizzati nelle varie sezioni che compongono la pagina.

TWM011FDatiGiornalieriMW

La classe TWM011FDatiGiornalieriMW implementa la logica applicativa richiesta dalla funzionalità, incapsulando tramite il metodo pubblico AggiornaDatiGGSRV la chiamata al servizio B110. I dati acquisiti vengono poi conservati in un'apposita variabile privata il cui contenuto è accessibile tramite i vari metodi pubblici disponibili. L'oggetto di tipo TB110FClientModule in uso viene passato come parametro dal chiamante della funzione che effettua la richiesta, viene quindi riutilizzata l'istanza della classe condivisa a livello di sessione web.

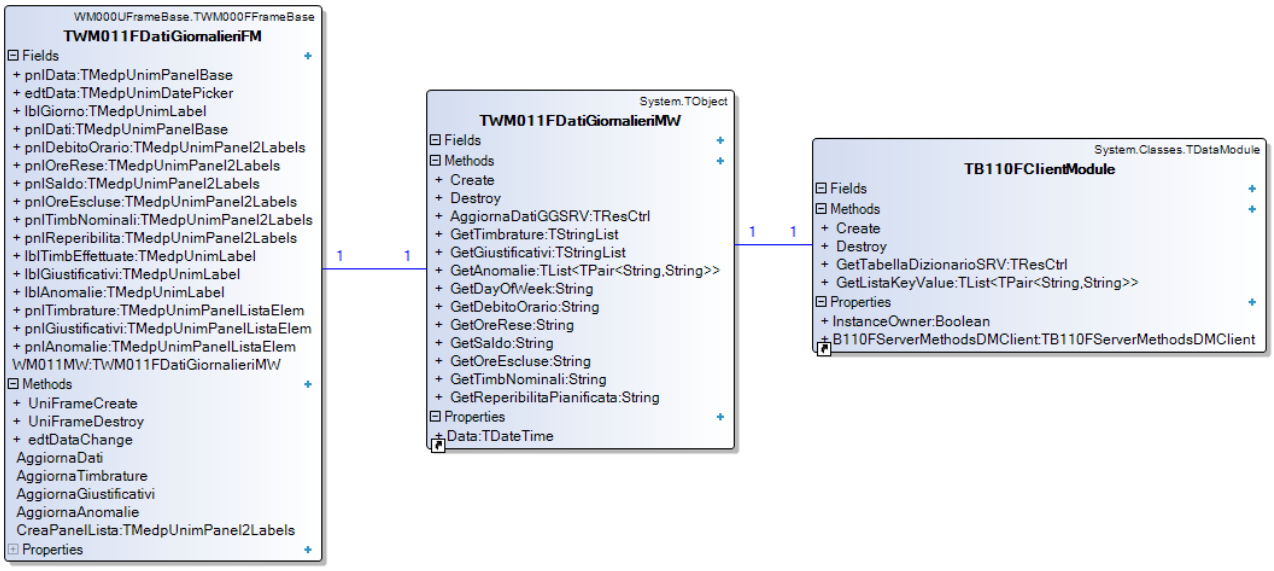


Figura 3.28 – Le classi utilizzate per l'implementazione della funzionalità Dati giornalieri

Elenco dipendenti

La funzionalità Elenco dipendenti è utilizzata dal responsabile per visualizzare i dati legati alla presenza dei collaboratori gestiti. Anche in questo caso l'implementazione della funzionalità ha richiesto la creazione delle due classi per la gestione dell'interfaccia grafica (presentation layer) e per incapsulare la logica applicativa (business logic layer). L'accesso ai dati avviene tramite il web service B110, non è stato quindi necessario creare una classe dedicata per l'interazione con il database (data access layer).

TWM012FElencoDipendentiFM

All'interno della classe TWM012FElencoDipendentiFM (che eredita da TWM000FFrameBase) sono presenti tutti i componenti visuali che compongono l'interfaccia grafica del frame e l'istanza della classe utilizzata per la gestione della logica applicativa. La struttura della vista è composta da tre schede a cui l'utente può accedere in successione, rappresentate da altrettanti tab all'interno di un componente di tipo TMedpUnimTabPanelBase. A partire dalla prima scheda (tabElencoDipendenti) è possibile accedere alla scheda di dettaglio (tabDettaglio) e successivamente ad un'ulteriore scheda di dettaglio del singolo campo (tabDettaglioCampo). La navigazione è gestita tramite gli event handler innescati dal click dell'utente e comporta l'aggiornamento dei dati visualizzati nelle varie schede, realizzata mediante i metodi AggiornaLista che utilizzano i dati prevenienti dal livello sottostante.

TWM012FElencoDipendentiMW

La classe TWM012FElencoDipendentiMW (sottoclasse di TWM000FDataSetMW) implementa la logica applicativa richiesta dalla funzionalità, incapsulando tramite il metodo pubblico AggiornaDipendentiSRV la chiamata al servizio B110. I dati acquisiti vengono poi conservati in un dataset il cui contenuto è accessibile e navigabile tramite i vari metodi pubblici disponibili. Anche in questo caso viene riutilizzata l'istanza della classe TB110FClientModule condivisa a livello di sessione web.

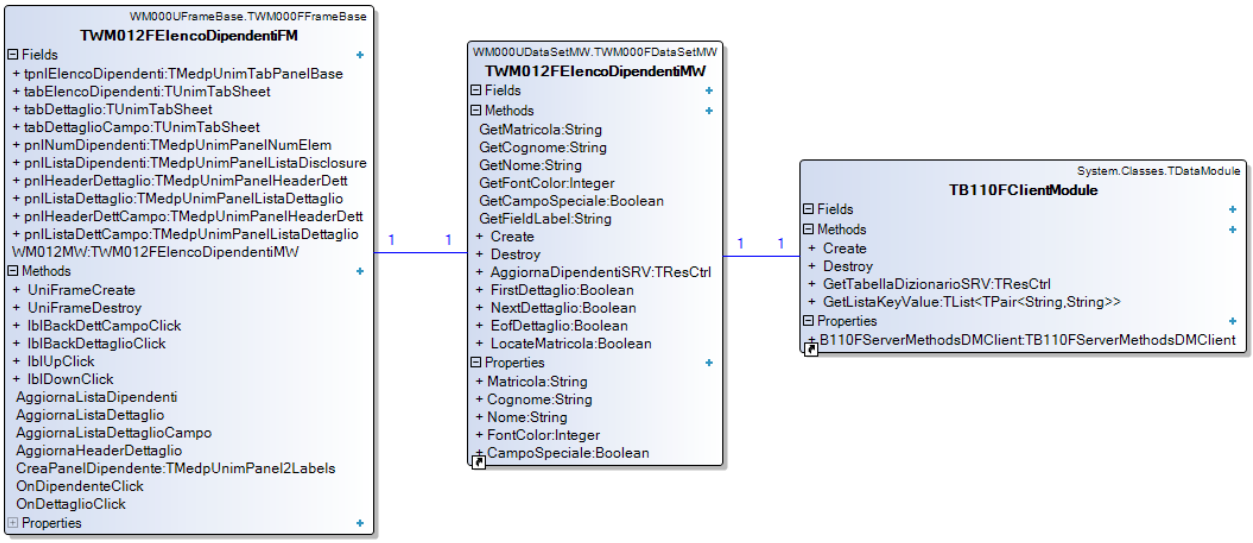


Figura 3.29 – Le classi utilizzate per l'implementazione della funzionalità Elenco dipendenti

Stampa cartellino

La funzionalità Stampa cartellino consente al dipendente il download del file in formato pdf del cartellino mensile. L'implementazione della funzionalità ha richiesto la creazione di due classi per la gestione dell'interfaccia grafica (presentation layer) e per incapsulare la logica applicativa (business logic layer). Anche per questa funzionalità l'accesso ai dati avviene tramite il web service B110, non è stato quindi necessario creare una classe dedicata per l'interazione con il database (data access layer).

TWM004FCartellinoFM

All'interno della classe TWM004FCartellinoFM (che eredita da TWM000FFrameBase) sono definiti tutti i componenti che compongono l'interfaccia grafica del frame e l'istanza della classe utilizzata per la gestione della logica applicativa. Utilizzando i dati selezionati dall'utente nell'interfaccia viene aggiornata la lista dei cartellini disponibili, mediante l'interazione con livello sottostante (funzione AggiornaListaCartellini). Tramite l'event handler OnDisclCartellinoClick viene avviata la procedura di stampa del cartellino, che richiede prima la chiamata al web service B110 per l'elaborazione del documento, successivamente la creazione del corrispondente file pdf in una cartella temporanea, per ultimo l'apertura del file nel browser tramite il link generato. Il file viene ricreato ad ogni richiesta con la parametrizzazione indicata dall'utente. Per evitare l'utilizzo della cache del browser che potrebbe contenere un file generato con layout diverso, il file viene ogni volta creato all'interno di una cartella con un nome casuale, in modo da ottenere ad ogni richiesta un link diverso.

TWM004FCartellinoMW

La classe TWM004FCartellinoMW implementa la logica applicativa richiesta dalla funzionalità, consentendo l'acquisizione della lista dei cartellini mensili disponibili e dei file in formato pdf dal servizio B110, utilizzando i metodi pubblici AggiornaListaCartellniSRV e StampaCartellinoPdfSRV. In particolare, la creazione del file richiede prima l'acquisizione del corrispondente array di byte e successivamente il suo salvataggio su file system, nel percorso passato come parametro alla funzione StampaCartellinoPdf. La lista dei cartellini viene conservata sotto forma di dataset di tipo TFDMemTable, ritornato dalla chiamata al web service.

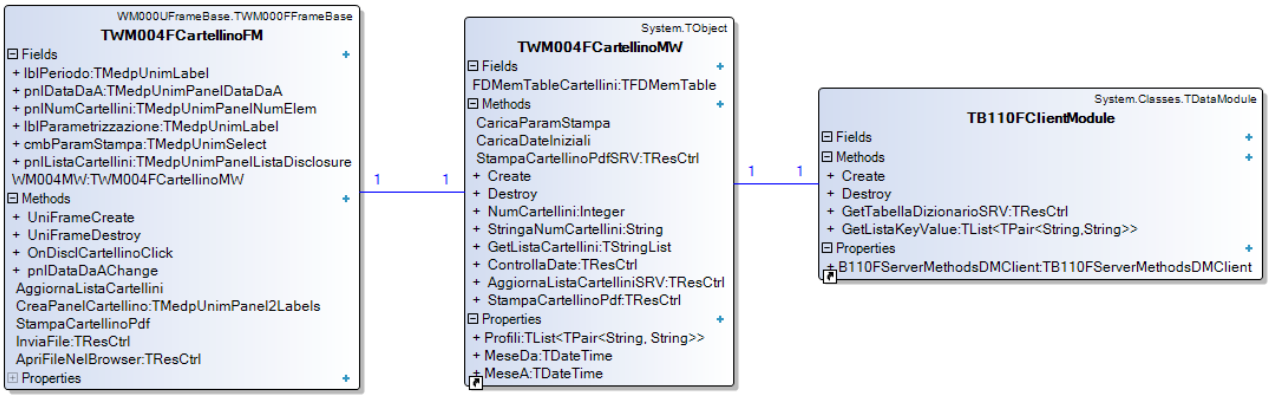


Figura 3.30 – Le classi utilizzate per l'implementazione della funzionalità Stampa cartellino

Stampa cedolino

La funzionalità Stampa cedolino consente al dipendente il download del file in formato pdf del cedolino paga mensile. L'implementazione della funzionalità ha richiesto la creazione di due classi per la gestione dell'interfaccia grafica (presentation layer) e per incapsulare la logica applicativa (business logic layer). Un'ulteriore classe TDatiCedolino viene utilizzata per incapsulare i dati che caratterizzano il singolo documento, condivisi tra le due classi principali mediante l'uso di liste generiche. Anche per questa funzionalità l'accesso ai dati avviene tramite il web service B110, non è stato quindi necessario creare una classe dedicata per l'interazione con il database (data access layer).

TWM005FCedolinoFM

All'interno della classe TWM005FCedolinoFM (che eredita da TWM000FFrameBase) sono definiti tutti i componenti che compongono l'interfaccia grafica del frame e l'istanza della classe utilizzata per la gestione della logica applicativa. La gestione della creazione del file e del download è identica a quella descritta nella funzionalità precedente, in aggiunta, dopo la visualizzazione del file può essere richiesta all'utente la conferma sull'inserimento della data di consegna del cedolino, aggiornata tramite la funzione AggiornaDataConsegnaCedolino.

TWM005FCedolinoMW

La classe TWM005FCedolinoMW implementa la logica applicativa richiesta dalla funzionalità, consentendo l'acquisizione della lista dei cedolini disponibili (AggiornaListaCedolinoSRV), il download dei file in formato pdf (StampaCedolinoPdfSRV) e l'aggiornamento della data di consegna (ImpostaDataConsegnaCedolinoSRV). La procedura di creazione del file è analoga a quella della funzione Stampa cartellino. Le proprietà in sola lettura MeseDa e MeseA rappresentano l'intervallo di mesi su cui è consentita la stampa del documento, vengono inizializzate nel costruttore a partire dalla data corrente e dai parametri aziendali WEBCedoliniDataMin e WEBCedoliniMMPrec.

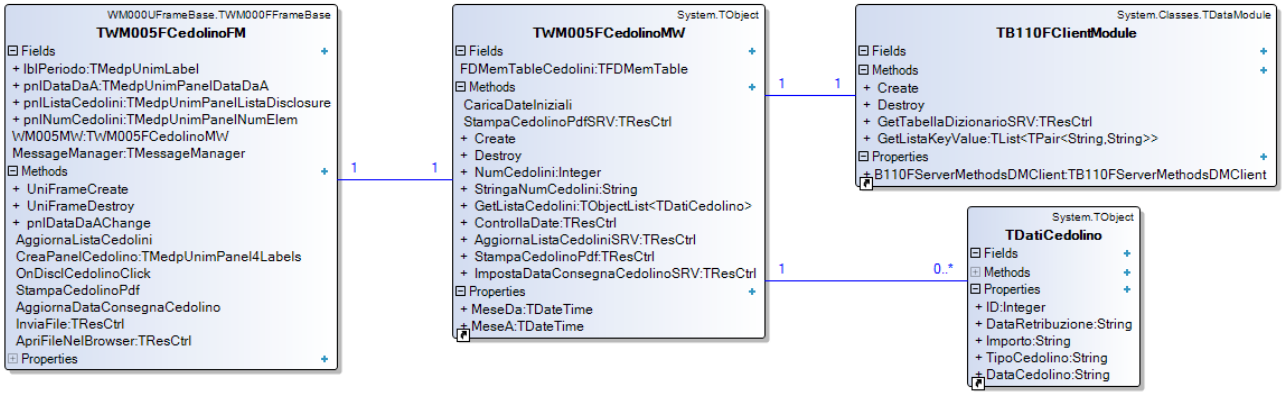


Figura 3.31 – Le classi utilizzate per l'implementazione della funzionalità Stampa cedolino

Timbratrice virtuale mobile

La funzionalità Timbratrice virtuale mobile consente al dipendente l'inserimento di una nuova timbratura, utilizzando opzionalmente la geolocalizzazione del dispositivo per agganciare uno dei rilevatori disponibili. L'implementazione della funzionalità ha richiesto la creazione di due classi per la gestione dell'interfaccia grafica (presentation layer) e per incapsulare la logica applicativa (business logic layer), a cui si aggiunge una form ulteriore per la visualizzazione della mappa. L'accesso ai dati avviene sia tramite il web service B110, sia tramite l'accesso diretto alla base dati, è stato quindi necessario creare un'ulteriore classe dedicata all'interazione con il database (data access layer).

TWM015FTimbraturaVirtualeFM
All'interno della classe TWM015FTimbraturaVirtualeFM (che eredita da TWM000FFrameBase) sono definiti tutti i componenti che compongono l'interfaccia grafica del frame e l'istanza della classe utilizzata per la gestione della logica applicativa. Utilizzando l'evento javascript "beforeInit" accessibile tramite la proprietà ClientEvent.UniEvent del frame sono state definite una serie di funzioni javascript utilizzate nel browser.

Molto importante è la gestione dell'aggiornamento dell'orologio mostrato nella pagina, che deve essere sincronizzato con l'ora del server per evitare discordanze tra l'interfaccia grafica e l'orario della timbratura inserita. L'orario mostrato nella label viene aggiornato durante l'accesso alla funzionalità ed incrementato ogni secondo lato client con la funzione javascript updateSeconds senza interazioni con il server dell'applicazione. Per evitare derive nell'orario mostrato e garantire la precisione richiesta, dopo l'avvio della funzionalità viene avviato lato server un timer rappresentato dall'oggetto tmrAggiornaOrario (di tipo TUnimTimer), configurato in modo da innescare l'evento OnTimer ogni 30 secondi. L'event handler associato chiama nel browser la funzione javascript updateTime, passando come parametro l'orario aggiornato campionato lato server, assicurando così l'allineamento del valore mostrato ogni 30 secondi.

Se abilitata, la funzionalità di geolocalizzazione richiede l'utilizzo della Geolocation Web API, in particolare la funzione getCurrentPosition che viene eseguita ad intervalli di 15 secondi a partire dall'apertura del frame, consentendo l'acquisizione delle coordinate geografiche del dispositivo ad intervalli regolari. L'operazione di localizzazione innesca poi una chiamata non bloccante verso il server, che utilizzando il metodo PositionRead verifica se le coordinate sono all'interno del raggio di validità di un rilevatore, abilitando o disabilitando di conseguenza la possibilità di timbratura da parte dell'utente.

TWM015FTimbraturaVirtualeMW
La classe TWM015FTimbraturaVirtualeMW implementa la logica applicativa richiesta dalla funzionalità, consentendo l'acquisizione della lista dei rilevatori e delle causali, la determinazione del rilevatore agganciato per le coordinate geografiche fornite e l'inserimento della timbratura. Per la gestione dei dati relativi al singolo rilevatore è stata creata la classe TRilevatore, con l'obiettivo di incapsulare i dati che lo caratterizzano e fornire i metodi pubblici necessari per la valutazione delle coordinate geografiche.

Per l'interazione con la base dati viene utilizzato il web service B110, ad eccezione dell'acquisizione dei rilevatori disponibili per l'utente, che richiede un'istanza della classe TWM015FTimbraturaVirtualeDM creata all'avvio della funzionalità.

TWM015FTimbraturaVirtualeDM

Come anticipato in precedenza, la classe TWM015FTimbraturaVirtualeDM, che eredita da TWM000FDataModuleBaseDM, consente l'acquisizione dei dati relativi ai rilevatori disponibili per l'utente che accede alla timbratura virtuale, senza interagire con il servizio B110. Utilizzando le query SQL contenute nei dataset selT361 e selT430 vengono creati gli oggetti di tipo TRilevatore richiesti dalla logica applicativa, resi disponibili al livello superiore sotto forma di lista generica.

TWM015FGoogleMaps

Sottoclasse di TUnimForm, rappresenta la form utilizzata per mostrare la mappa con la posizione del dispositivo e dei rilevatori disponibili. Mostrando a schermo intero un componente di tipo TUnimHTMLFrame, il cui contenuto è popolato con la funzione javascript DrawMap, permette di sfruttare la Google Maps API per disegnare tramite codice javascript la mappa ed i cerchi di raggio variabile che rappresentano i rilevatori. Essendo una form accessibile soltanto tramite la funzionalità di timbratura, le funzioni javascript utilizzate sono definite nello stesso evento già descritto per la classe TWM015FTimbraturaVirtualeFM.

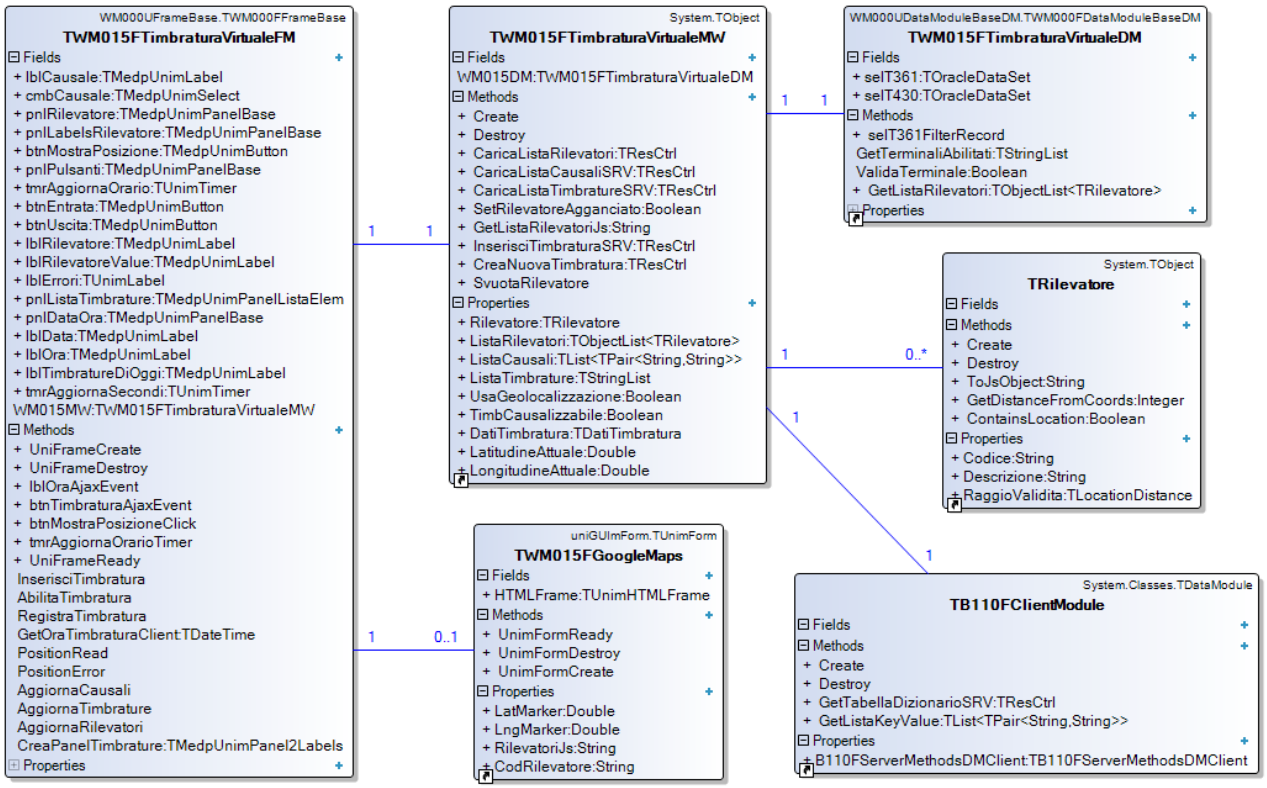


Figura 3.32 - Le classi utilizzate per l'implementazione della funzionalità Timbratura virtuale

Gestione deleghe

La funzionalità Gestione deleghe consente al responsabile l'inserimento, la modifica o la cancellazione di una delega esistente. L'implementazione della funzionalità ha richiesto la creazione di tre classi principali, per la gestione dell'interfaccia grafica (presentation layer), per incapsulare la logica applicativa (business logic layer) e per l'interazione con il database (data access layer). A queste si aggiungono tre classi di supporto che incapsulano i dati della delega, dell'utente delegante e dell'utente delegato, rispettivamente TDelega, TUtenteDelegante e TUtenteDelegato.

TWM006FGestioneDelegheFM
All'interno della classe TWM006FGestioneDelegheFM (sottoclasse di TWM000FFrameBase) sono definiti tutti i componenti che compongono l'interfaccia grafica del frame e l'istanza della classe utilizzata per la gestione della logica applicativa. La struttura della pagina prevede tre schede, rappresentate dai tre tab all'interno di un componente di tipo TMedpUnimTabPanelBase. La prima scheda riporta l'elenco delle deleghe esistenti, creato utilizzando il metodo AggiornaListaDeleghe che utilizza la funzione CreaPanelDelega per istanziare i singoli elementi della lista. La seconda scheda, a cui si accede tramite gli event handler InserimentoDelegaClick e ModificaDelegaClick, viene utilizzata per l'inserimento, la modifica o la cancellazione di una delega esistente. I due event handler, innescati dal click dell'utente sul pulsante di inserimento o sulla singola delega, modificano il contenuto della scheda adattandola all'uso per le due modalità. Le tre operazioni richiedono una conferma da parte dell'utente, gestita tramite la funzione MessageDlg, utilizzano il livello sottostante per validare i dati inseriti e per applicare le modifiche sul database, tramite gli event handler OkInserimentoDelegaClick, OkModificaDelegaClick e EliminaDelegaClick.
La terza scheda consente la selezione dell'utente e presenta una lista di possibili valori all'interno di un componente di tipo TMedpUnimPanelListaDisclosure, popolato a partire dal filtro impostato dal responsabile utilizzando la funzione AggiornaListaUtenti.

TWM006FGestioneDelegheMW
La classe TWM006FGestioneDelegheMW implementa la logica applicativa richiesta dalla funzionalità, mettendo a disposizione una serie di funzioni pubbliche per l'acquisizione delle liste di deleghe attive, la validazione dei dati inseriti, l'inserimento di una nuova delega e la modifica o cancellazione di una esistente. L'operazione di modifica richiede prima la selezione della delega da editare, indicata usando la funzione SelezionaDelegaInModifica che riceve come parametro l'id univoco della delega, e poi la chiamata della funzione ModificaDelega che applica le modifiche da apportare. La cancellazione è possibile solo sulla delega precedentemente selezionata per la modifica. I dati incapsulati sono accessibili al livello superiore tramite le proprietà pubbliche della classe, così come la delega in modifica selezionata.

TWM006FGestioneDelegheDM
La classe TWM006FGestioneDelegheDM (sottoclasse di TWM000FDataModuleBaseDM) consente al livello superiore l'accesso ai dati presenti nel database. I principali dataset utilizzati sono selI060Utenti, che estrae la lista dei possibili utenti delegati, e selI061DelegheUtente,

utilizzato per la lettura delle deleghe esistenti, l'inserimento, la modifica e la cancellazione della delega, identificata da un id univoco. La funzione VerificaProfiloAssegnato viene utilizzata per determinare se il profilo da creare sull'utente delegato non sia già presente, mentre la funzione VerificaDelegheEsistenti viene utilizzata per escludere la presenza di un'altra delega nel periodo selezionato.

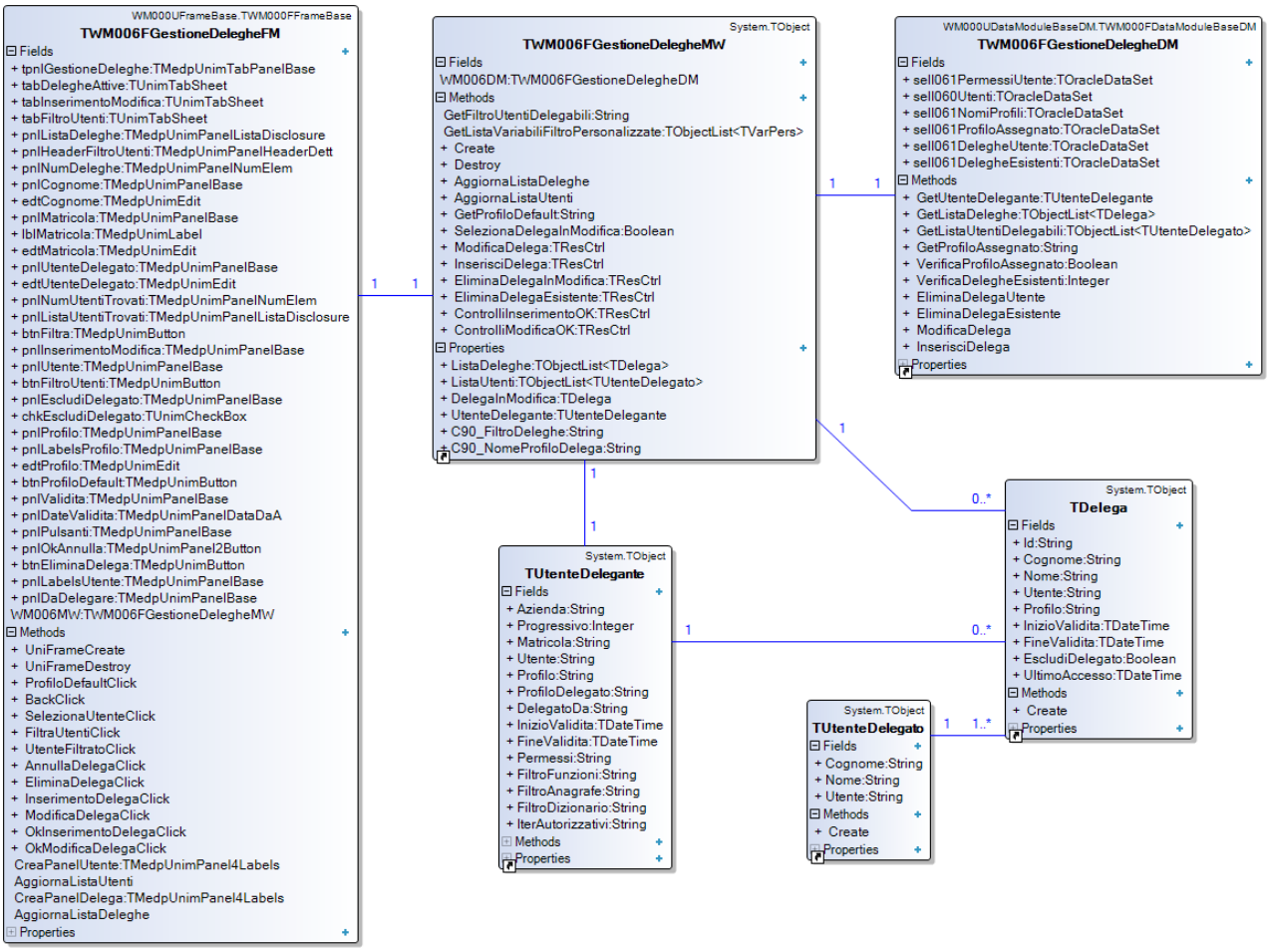


Figura 3.33 - Le classi utilizzate per l'implementazione della funzionalità Gestione Deleghe

Messaggistica

La funzionalità Messaggistica consente al dipendente la lettura dei messaggi ricevuti e inviati nel contesto del sistema informativo Iris. L'implementazione della funzionalità ha richiesto la creazione di tre classi principali, per la gestione dell'interfaccia grafica (presentation layer), per incapsulare la logica applicativa (business logic layer) e per l'interazione con il database (data access layer). A queste si aggiungono numerose classi di supporto che rappresentano il filtro applicato ai messaggi visualizzati (TFiltroMessaggi e sottoclassi), i messaggi inviati e ricevuti (TMessaggio e sottoclassi), gli allegati (TAllegatoMessaggio) e i destinatari (TDestinatario e sottoclassi).

TWM018FMessaggisticaFM

All'interno della classe TWM018FMessaggisticaFM (sottoclasse di TWM000FFrameBase) sono definiti tutti i componenti che compongono l'interfaccia grafica del frame e l'istanza della classe utilizzata per la gestione della logica applicativa. L'interfaccia grafica richiede la gestione di molti componenti visuali per la gestione dei filtri e del dettaglio dei messaggi. Anche in questa funzionalità la struttura della pagina è formata da un componente di tipo TMedpUnimTabPanelBase con all'interno quattro schede, utilizzate per visualizzare i messaggi ricevuti ed inviati, per il dettaglio del singolo messaggio e per l'elenco dei destinatari dei messaggi inviati. Il download degli allegati viene gestito con la stessa procedura descritta per la funzionalità Stampa cartellino. Un particolare della pagina è la visualizzazione dei filtri, differenti tra messaggi inviati e ricevuti, che richiedono alcuni campi sempre visibili ed altri opzionali. Gli oggetti utilizzati allo scopo sono pnlFiltroRicevuti e pnlFiltroInviati, entrambi di tipo TMedpUnimPanelBase, la cui dimensione in altezza è gestita dinamicamente in base al click dell'utente, la transizione tra i due stati possibili è arricchita da un'animazione impostata mediante le proprietà CSS.

TWM018FMessaggisticaMW

La classe TWM018FMessaggisticaMW implementa la logica applicativa richiesta dalla funzionalità, mettendo a disposizione del livello superiore una serie di funzioni pubbliche per l'acquisizione dei messaggi inviati e ricevuti in base ai filtri passati come parametro. Utilizzando la proprietà Modalita è possibile decidere di utilizzare l'istanza della classe per i messaggi ricevuti o inviati. Gli oggetti di tipo TFiltroMessaggi vengono creati a partire dai dati passati come parametro alle funzioni AggiornaListaMessaggi ed utilizzati dal livello inferiore per filtrare i dati acquisiti dal database. La classe consente l'aggiornamento dello stato dei messaggi (ricezione/lettura), permette di valutare se sono presenti messaggi obbligatori da leggere e gestisce il download degli allegati. I messaggi sono accessibili tramite le proprietà ListaMessaggi e MessaggioSelezionato, nel primo caso sotto forma di lista generica, nel secondo caso viene invece letto il singolo messaggio, selezionato con lo spostamento nella lista tramite le funzioni Next, Prior e Eof.

TWM018FMessaggisticaDM

La classe TWM018FMessaggisticaDM (sottoclasse di TWM000FDataModuleBaseDM) consente al livello superiore l'accesso ai dati presenti nel database. I principali dataset utilizzati sono

selT282Inviati, selT282Ricevuti e selT283Allegato, i primi due sono utilizzati per estrarre la lista dei messaggi inviati e ricevuti in base al filtro impostato, mentre il terzo è utilizzato per la gestione degli allegati. Il filtro ricevuto come parametro dal livello superiore nella funzione GetListaMessaggi viene trasformato in una condizione SQL, inserita nella query per l'estrazione dei messaggi. Le operazioni di aggiornamento dello stato sono gestite con gli oggetti updT284Ricezione e updT284Lettura, entrambi di tipo TOracleQuery.

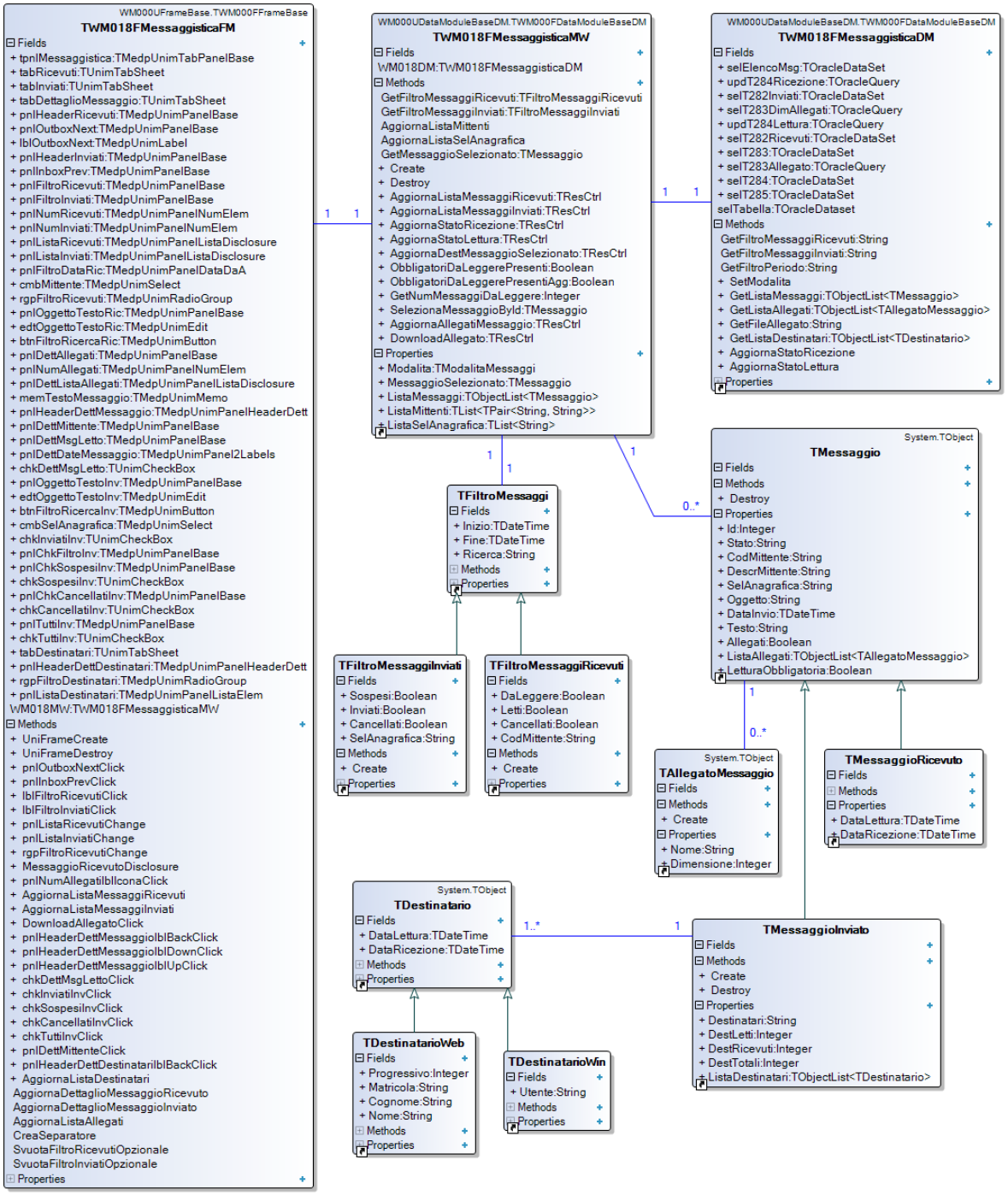


Figura 3.34 - Le classi utilizzate per l'implementazione della funzionalità Messaggistica

3.5.3 Iter autorizzativi

La fase di analisi ha consentito di individuare numerosi elementi in comune nelle funzionalità che rappresentano gli iter autorizzativi, non soltanto nei due elementi che compongono il singolo iter (richiesta/autorizzazione) ma anche tra gli iter stessi, dovendo gestire dati dalla struttura simile ed eseguendo su di essi operazioni comuni a più funzionalità. Per questo motivo, si è ritenuto opportuno sfruttare l'ereditarietà della programmazione ad oggetti per lo sviluppo di una serie di classi che costituiscono la base comune per l'implementazione di tutti gli iter autorizzativi.

TWR002FRichiesteFM
Rappresenta il frame di base che fornisce l'interfaccia grafica e le funzioni per la gestione delle richieste, incapsulando anche le istanze delle classi utilizzate per la gestione di note ed allegati. L'interfaccia presenta un componente di tipo TMedpUnimTabPanelBase, mediante il quale vengono gestite quattro schede;

- tabElenco: consente di visualizzate l'elenco delle richieste, gestendo un filtro predefinito per stato e per data, lasciando poi allo sviluppatore la possibilità di definire filtri personalizzati.
- tabDettaglio: consente di visualizzare il dettaglio della richiesta selezionata.
- tabNote: consente la gestione delle note della richiesta selezionata.
- tabAllegati: consente di gestire gli eventuali file allegati alla richiesta.

La classe contiene una serie di funzioni astratte la cui implementazione è lasciata alle sottoclassi dello specifico iter autorizzativo, definendo così un comportamento personalizzato nella rappresentazione grafica delle richieste nel tabElenco e dei dati di dettaglio nel tabDettaglio.

TWR003FGestRichiesteFM
Sottoclasse di TWR002FRichiesteFM, rappresenta il frame di base per l'implementazione delle funzionalità di inserimento della richiesta. Aggiunge alla superclasse una scheda per l'inserimento dei dati della nuova richiesta o la modifica di una esistente. Una serie di metodi astratti consente la gestione specifica per l'iter delle operazioni di validazione e di conferma delle operazioni eseguite dall'utente. L'interfaccia si arricchisce inoltre con i componenti visuali necessari per la gestione dell'eliminazione e della revoca delle richieste.

TWR004FAutRichiesteFM
Sottoclasse di TWR002FRichiesteFM, rappresenta il frame di base per l'implementazione delle funzionalità di autorizzazione delle richieste. Fornisce i metodi utilizzati per autorizzare o negare la richiesta, oltre ai componenti visuali necessari per la selezione dell'operazione da parte dell'utente. All'oggetto tabElenco della superclasse è inoltre aggiunto un componente di tipo TMedpUnimPanelSlideUp, utilizzato per contenere i pulsanti a disposizione del responsabile per autorizzare o negare tutte le richieste visualizzate. Il contenuto dell'elemento può essere visibile o nascosto a seconda delle necessità.

TWR002FRichiesteMW
Sottoclasse di TWM000FDataSetMW, rappresenta la classe principale del business logic layer per la gestione delle richieste. Incapsula il dataset principale utilizzato per contenere le richieste da visualizzare, sia in caso di acquisizione dei dati tramite il servizio B110, sia in caso di accesso diretto al database. Fornisce alle sottoclassi una serie di funzioni astratte da implementare per realizzare le operazioni sulle richieste, come inserimento, modifica, autorizzazione e revoca. Nel caso non venga utilizzato il web service B110, viene creata ed utilizzata un'istanza della classe TC018FIterAutDM, componente software già in uso in azienda che semplifica la gestione delle tabelle di base utilizzate per gli iter autorizzativi.

TWR003FNoteMW
Rappresenta la classe utilizzata per la gestione delle note della richiesta utilizzando il servizio B110. Mediante i metodi pubblici GetNoteRichiestaSRV e SetNoteRichiestaSRV è possibile acquisire e editare le note sui vari livelli disponibili per la richiesta selezionata. I dati associati alla singola nota sono accessibili tramite le proprietà pubbliche della classe, mentre ci si può muovere tra i vari livelli utilizzando le funzioni First, Last, Next e Prior.

TWR006FAllegatiMW
Rappresenta la classe utilizzata per la gestione degli allegati della richiesta. Fornisce i metodi pubblici necessari per il download, l'inserimento e l'eliminazione degli allegati, oltre a garantire la possibilità di modificare i dati associati agli allegati stessi. Per gestire le operazioni di download ed inserimento viene utilizzata un'istanza della classe TC021FDocumentiManagerDM, riferimento aziendale per la gestione dei documenti nel sistema informativo Iris.

TWR002FIterDM
Sottoclasse di TDataModule, rappresenta la superclasse per la gestione dell'interazione con il database per gli iter autorizzativi che non utilizzano il web service B110. Incapsula il dataset utilizzato per l'interazione con la tabella di base dell'iter autorizzativo, consentendo l'aggiornamento delle richieste da visualizzare in base ai filtri definiti ai livelli superiori. Permette inoltre di impostare un ordinamento specifico dei dati estratti un base all'iter gestito dalla sottoclasse.

TC018FIterAutDM
È la classe utilizzata da tutti gli applicativi del sistema informativo Iris per l'interazione con le varie tabelle che contengono i dati associati alle richieste. Attraverso numerose funzioni e attributi pubblici permette di acquisire i dati necessari ad eseguire le operazioni sulle richieste previste dai vari iter.

TC021FDocumentiManagerDM
È la classe utilizzata per la gestione dell'upload e del download dei file salvati nel sistema di gestione documentale del sistema informativo Iris. Incapsula il codice necessario per manipolare i documenti in base alla configurazione prevista dall'azienda cliente ed in modo indipendente dalla tipologia di storage utilizzata (database, file system o servizio esterno).

Figura 3.35 – Le classi utilizzate come base per lo sviluppo degli iter autorizzativi

Richiesta/Autorizzazione giustificativi

L'iter di gestione dei giustificativi consente al dipendente ed al responsabile l'inserimento e l'autorizzazione delle richieste di giustificativo (come ferie e permessi). Le classi implementate si basano sulla struttura di base descritta in precedenza, mantenendo l'organizzazione basata su tre livelli. Per questo iter è disponibile il web service B110 per l'accesso ai dati e per eseguire le operazioni sulle richieste.

TWM013FRicAssenzeFM
Sottoclasse di TWR003FGestRichiesteFM, rappresenta il frame messo a disposizione del dipendente per l'inserimento, l'eliminazione e la revoca delle richieste di giustificativo. Aggiunge alla struttura di base una scheda utilizzata per visualizzare le competenze del giustificativo selezionato (tabCompetenze). Completa la scheda di inserimento con i campi editabili necessari per la creazione della nuova richiesta. L'accesso ai dati richiesti per popolare l'interfaccia avviene tramite le istanze di due classi del business logic layer, utilizzate per gestire separatamente le richieste di giustificativo e le competenze.

TWM013FRicAssenzeMW
Sottoclasse di TWR002FRichiesteMW, fornisce al frame di inserimento delle richieste i dati necessari per visualizzare e per popolare i campi editabili della scheda di inserimento, sotto forma di liste generiche. Implementando i metodi astratti della superclasse, consente di eseguire le operazioni di inserimento, eliminazione e revoca della richiesta selezionata, utilizzando il servizio B110 tramite l'istanza della classe TB110FClientModule passata come parametro alle funzioni.

TWR004FCompetenzeMW
Sottoclasse di TObject, consente la gestione delle competenze calcolate per la tipologia di giustificativo selezionato. Dialoga con il web service B110 per l'acquisizione dei dati con la funzione GetCompentenzeSRV ed espone una serie di proprietà pubbliche per consentire al livello superiore l'accesso alle informazioni ricevute.

TWM003FAutAssenzeFM
Sottoclasse di TWR004FAutRichiesteFM, rappresenta il frame a disposizione del responsabile per l'autorizzazione singola o massiva delle richieste di giustificativo. Anche in questo caso è presente una scheda aggiuntiva per la visualizzazione delle competenze relative alla tipologia di giustificativo selezionata, i cui dati sono acquisiti utilizzando un'istanza della stessa classe TWR004FCompetenzeMW.

TWM003FAutAssenzeMW
Sottoclasse di TWR002FRichiesteMW, fornisce al frame di autorizzazione i dati necessari per rappresentare graficamente le richieste. Implementando i metodi astratti della superclasse, consente di eseguire le operazioni di autorizzazione della richiesta selezionata. Utilizza il servizio B110 tramite l'oggetto di tipo TB110FClientModule passato come parametro alla funzione AutorizzaRichiestaSRV, necessaria per autorizzare o negare la richiesta selezionata.

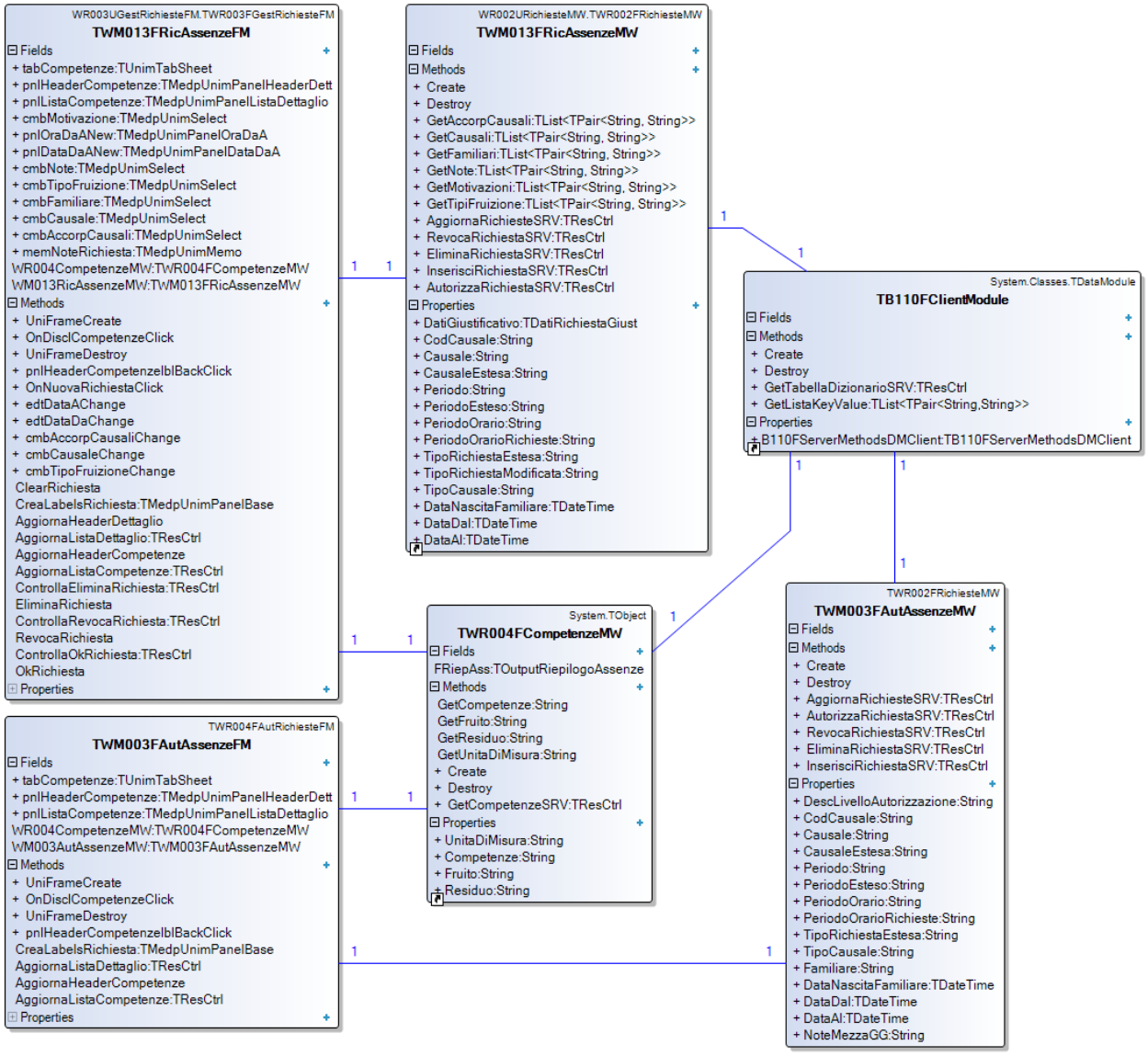


Figura 3.36 - Le classi utilizzate per l'implementazione delle funzionalità Richiesta e Autorizzazione giustificativi

Richiesta/Autorizzazione timbrature

L'iter di gestione delle timbrature consente l'inserimento e l'autorizzazione delle richieste di timbratura al dipendente ed al relativo responsabile. Le classi implementate si basano sulla struttura di base descritta in precedenza mantenendo l'organizzazione basata su tre livelli. Anche per questo iter è disponibile il web service B110 per l'accesso ai dati e per eseguire le operazioni fondamentali sulle richieste.

TWM014FRicTimbratureFM
Sottoclasse di TWR003FGestRichiesteFM, rappresenta il frame messo a disposizione del dipendente per la creazione di una nuova richiesta di inserimento, modifica o eliminazione di una timbratura. Aggiunge una scheda (tabTimbrature) alla struttura di base definita nella superclasse, consentendo di visualizzare le timbrature presenti sul giorno selezionato, acquisite tramite un'istanza della classe TWR005FTimbratureModificabiliMW. L'implementazione delle funzioni astratte consente di gestire le operazioni di inserimento, cancellazione e revoca nel contesto specifico dell'iter in esame, senza la necessità di definire ulteriori meccanismi per il funzionamento in quanto già presenti nella superclasse.

TWM014FRicTimbratureMW
Sottoclasse di TWR002FRichiesteMW, è utilizzata dalla classe TWM014FRicTimbratureFM per accedere ai dati necessari per rappresentare graficamente le richieste, applicando il filtro selezionato. Le operazioni richieste dalla logica applicativa vengono eseguite mediante l'interazione con il servizio B110, l'implementazione dei metodi InserisciRichiestaSRV e EliminaRichiestaSRV utilizza l'istanza della classe TB110FClientModule passata come parametro.

TWR005FTimbratureModificabiliMW
Sottoclasse di TWM000FDataSetMW, consente l'acquisizione dei dati relativi alle timbrature inserite nel giorno selezionato, mediante la funzione pubblica AggiornaTimbratureModifSRV che utilizza il web service B110. I dati ricevuti sono incapsulati nel dataset gestito dalla superclasse e accessibili tramite le proprietà pubbliche.

TWM010FAutTimbratureFM
Sottoclasse di TWR004FAutRichiesteFM, rappresenta il frame a disposizione del responsabile per l'autorizzazione singola o massiva delle richieste di timbratura. Utilizzando l'implementazione concreta delle funzioni astratte CreaLabelsRichiesta e AggiornaListDettaglio, consente di sfruttare i meccanismi già definiti nella superclasse, rispettivamente per la creazione della lista delle richieste e l'aggiornamento del dettaglio della richiesta selezionata.

TWM010FAutTimbratureMW
Sottoclasse di TWR002FRichiesteMW, è utilizzata come componente del business logic layer per la gestione della funzionalità di autorizzazione delle richieste di timbratura. Le proprietà pubbliche consentono di accedere ai dati delle richieste acquisite dal servizio B110, mentre le funzioni AggiornaRichiesteSRV e AutorizzaRichesteSRV consentono di gestire l'interazione con il servizio, incapsulando la gestione degli errori e la logica richiesta dall'applicazione.

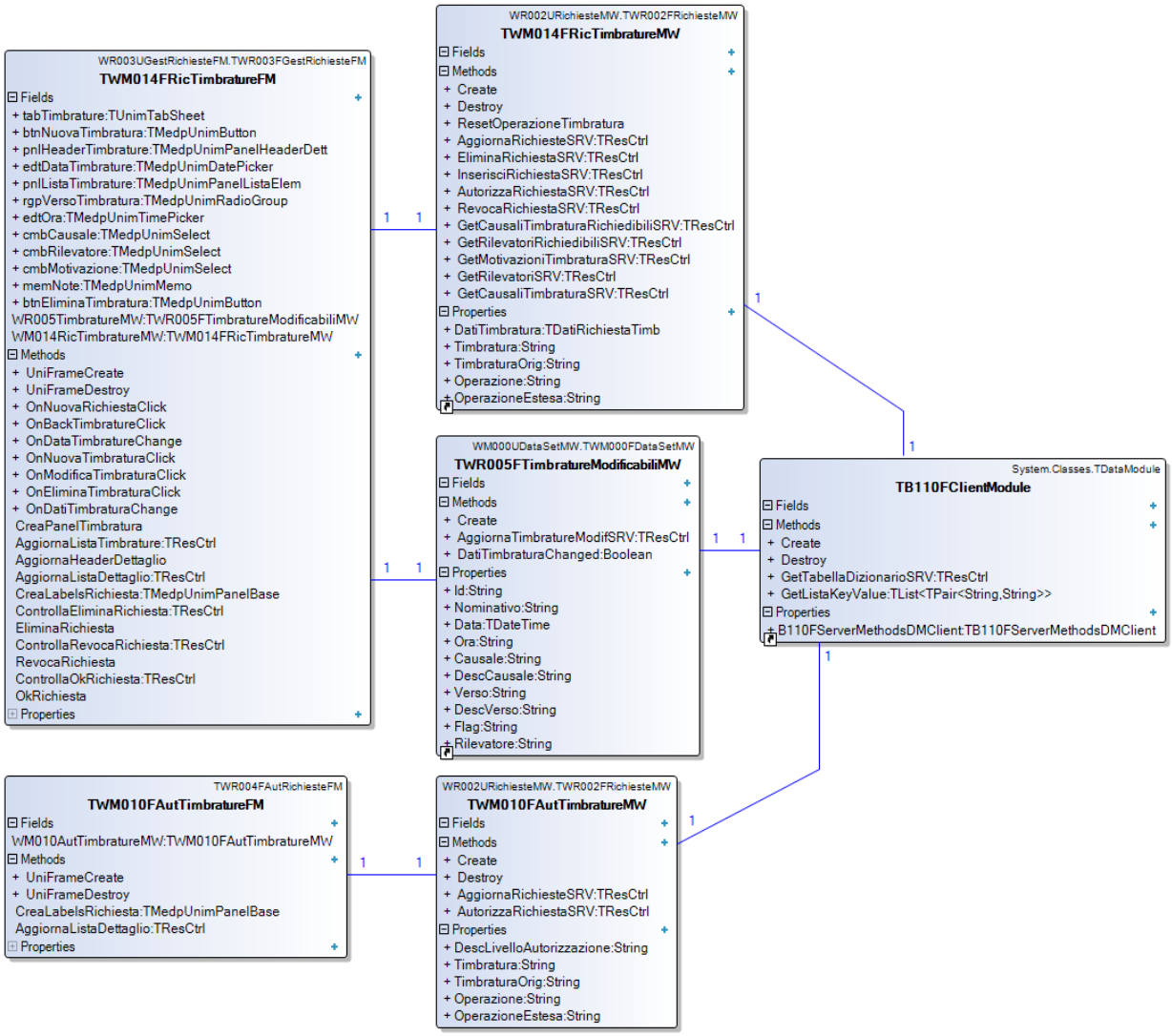


Figura 3.37 - Le classi utilizzate per l'implementazione delle funzionalità Richiesta e Autorizzazione timbrature

Richiesta/Autorizzazione cambio orario

Le funzionalità di richiesta e autorizzazione cambio orario consentono di implementare l'omonimo iter autorizzativo, permettendo l'inserimento e l'autorizzazione delle richieste da parte del dipendente e del responsabile. Anche in questo caso le classi si basano sulla struttura di base descritta in precedenza, mantenendo l'organizzazione basata su tre livelli sovrapposti. Non è tuttavia disponibile il web service B110, richiedendo quindi l'accesso diretto al database.

TWM016FRicCambioOrarioFM
Sottoclasse di TWR003FGestRichiesteFM, rappresenta il frame utilizzato dal dipendente per l'inserimento, l'eliminazione o la revoca di una richiesta di cambio orario. I componenti visuali mostrati nella scheda di inserimento (tabInserimento della superclasse) variano in base alla modalità selezionata dall'utente utilizzando i componenti di tipo TMedpUnimRadioButton, il cui evento OnChange innesca l'event handler rbtnModRichiestaChange.

TWM017FAutCambioOrarioFM
Sottoclasse di TWR004FAutRichiesteFM, rappresenta il frame visualizzato dal responsabile per autorizzare o negare le richieste inserite dai collaboratori. Utilizzando i metodi ed i componenti visuali della superclasse richiede solamente l'implementazione delle funzioni astratte AggiornaListaDettaglio e CreaLabelsRichiesta, quest'ultima necessaria per la creazione del panel utilizzato per visualizzare i dati della richiesta inserita dal dipendente.

TWM016FCambioOrarioMW
Sottoclasse di TWR002FRichiesteMW, è la classe utilizzata per implementare la logica applicativa di entrambe le funzionalità di richiesta ed autorizzazione. Consente al livello superiore di eseguire le operazioni necessarie incapsulando un'istanza della classe TC018FIterAutDM. Per compatibilità con la gestione che richiede l'utilizzo del servizio B110, l'implementazione delle funzioni astratte della superclasse mantiene la firma definita in precedenza ma non utilizza tutti i parametri passati dal chiamante. I nuovi metodi realizzati consentono di validare la nuova richiesta in fase di inserimento (istanza della classe TRichiestaCambioOrario) e di acquisire i giorni e gli orari disponibili necessari per la sua creazione.

TWM016FCambioOrarioDM
Sottoclasse di TWR002FIterDM, rappresenta la classe appartenente al data access layer dell'iter autorizzativo. I metodi pubblici consentono la gestione degli oggetti di tipo TOracleDataset e TOracleQuery utilizzati per l'interazione con il database. In particolare, sono utilizzati per l'inserimento della nuova richiesta, l'aggiornamento della pianificazione degli orari dei dipendenti e del calendario individuale del dipendente interessato dalla modifica.

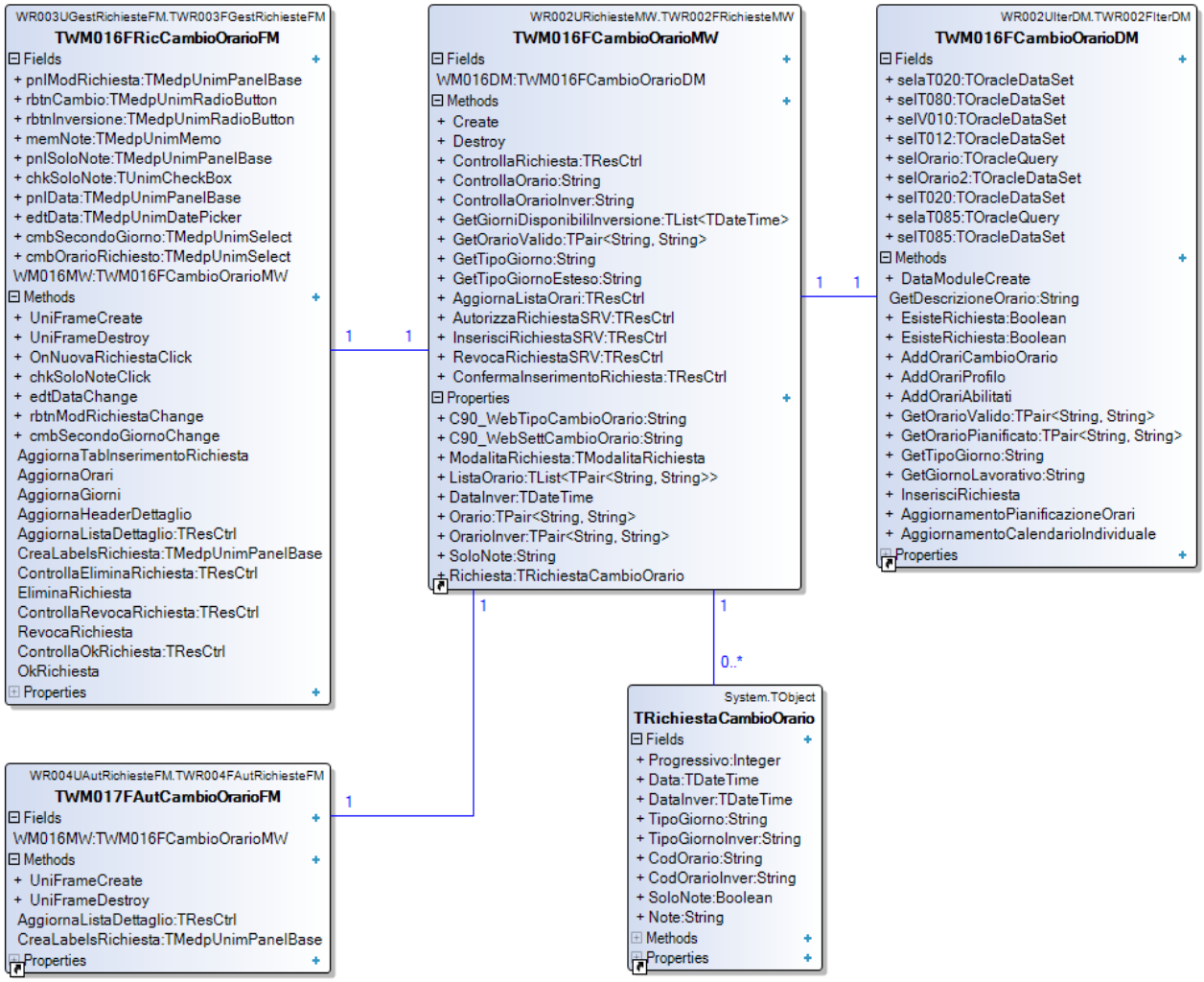


Figura 3.38 - Le classi utilizzate per l'implementazione delle funzionalità Richiesta e Autorizzazione cambio orario

Compilazione/Validazione scheda informativa

Le funzionalità di compilazione e validazione scheda informativa consentono di implementare l'omonimo iter autorizzativo, consentendo al dipendente la compilazione di modelli predefiniti di documenti ed al responsabile la validazione dei dati inseriti. Le classi implementate si basano sulla struttura degli iter autorizzativi descritta in precedenza, senza utilizzare il servizio B110. Molto importante è la gestione della rappresentazione grafica della scheda informativa, all'interno della quale sono presenti i campi di input utilizzati dal dipendente per la compilazione.

TWM019FRicCertificazioniFM
Sottoclasse di TWR003FGestRichiesteFM, rappresenta il frame utilizzato dal dipendente per la compilazione del modello di scheda informativa associato alla richiesta inserita. Aggiunge alla struttura definita nella superclasse una scheda (tabModello) utilizzata per visualizzare e compilare il modello, a cui si accede dalla scheda di inserimento o modifica della richiesta.

TWM020FAutCertificazioniFM
Sottoclasse di TWR004FAutRichiesteFM, rappresenta il frame utilizzato dal responsabile per validare le schede informative compilate dai collaboratori. Anche in questa funzionalità è presente la scheda aggiuntiva che consente all'utente di visualizzare il modello compilato associato alla richiesta in valutazione.

TWM019FCertificazioniMW
Sottoclasse di TWR002FRichiesteMW, è la classe utilizzata per implementare la logica applicativa di entrambe le funzionalità di compilazione e validazione. Utilizza l'istanza di una classe che implementa l'interfaccia IWM019FModelloCertificazioneMW per la gestione del modello. L'oggetto viene passato come parametro nel costruttore per evitare dipendenze dal framework UniGUI su questo livello. Per la gestione delle richieste utilizza la classe TC018FIterAutDM.

IWM019FModelloCertificazioneMW
È l'interfaccia definita per permette lo sviluppo di una classe dedicata alla creazione dell'interfaccia grafica del modello, mantenendo però l'indipendenza della classe del business logic layer dal framework UniGUI. I metodi definiti consentono la creazione del modello, la validazione dei dati inseriti e l'inserimento del modello associato alla richiesta.

TWM019FModelloCertificazioneMW
Sottoclasse di TWM000FDataModuleBaseDM, implementa l'interfaccia descritta in precedenza per gestire il modello di certificazione nel contesto del framework UniGUI. Non può essere inserita nell'architettura a tre livelli in quanto viene utilizzata per la creazione dell'interfaccia grafica, per la validazione dei dati inseriti e per l'interazione con il database nella gestione del modello. Incapsula un oggetto di tipo TMedpUnimPanelBase passato come parametro nel costruttore, all'interno del quale vengono creati dinamicamente i componenti visuali che compongono la scheda. Parallelamente alla creazione dell'interfaccia grafica viene mantenuta una lista di oggetti di tipo TDatoEditabile, utilizzati per rappresentare i dati da acquisire ed il puntatore al relativo componente di input. L'operazione di validazione è gestita nella funzione CtrlDatiModello che scorre la lista per acquisire il valore inserito nel componente, valutando poi l'obbligatorietà del dato ed applicando le regole di validazione definite dal creatore del modello di scheda informativa.

TWM019FCertificazioniDM

Sottoclasse di TWR002FIterDM, rappresenta la classe appartenente al data access layer dell'iter autorizzativo. I metodi pubblici consentono l'inserimento e la modifica dei dati associati alla richiesta, oltre all'acquisizione dei modelli disponibili e delle loro caratteristiche, ritornati al chiamante sotto forma di liste generiche.

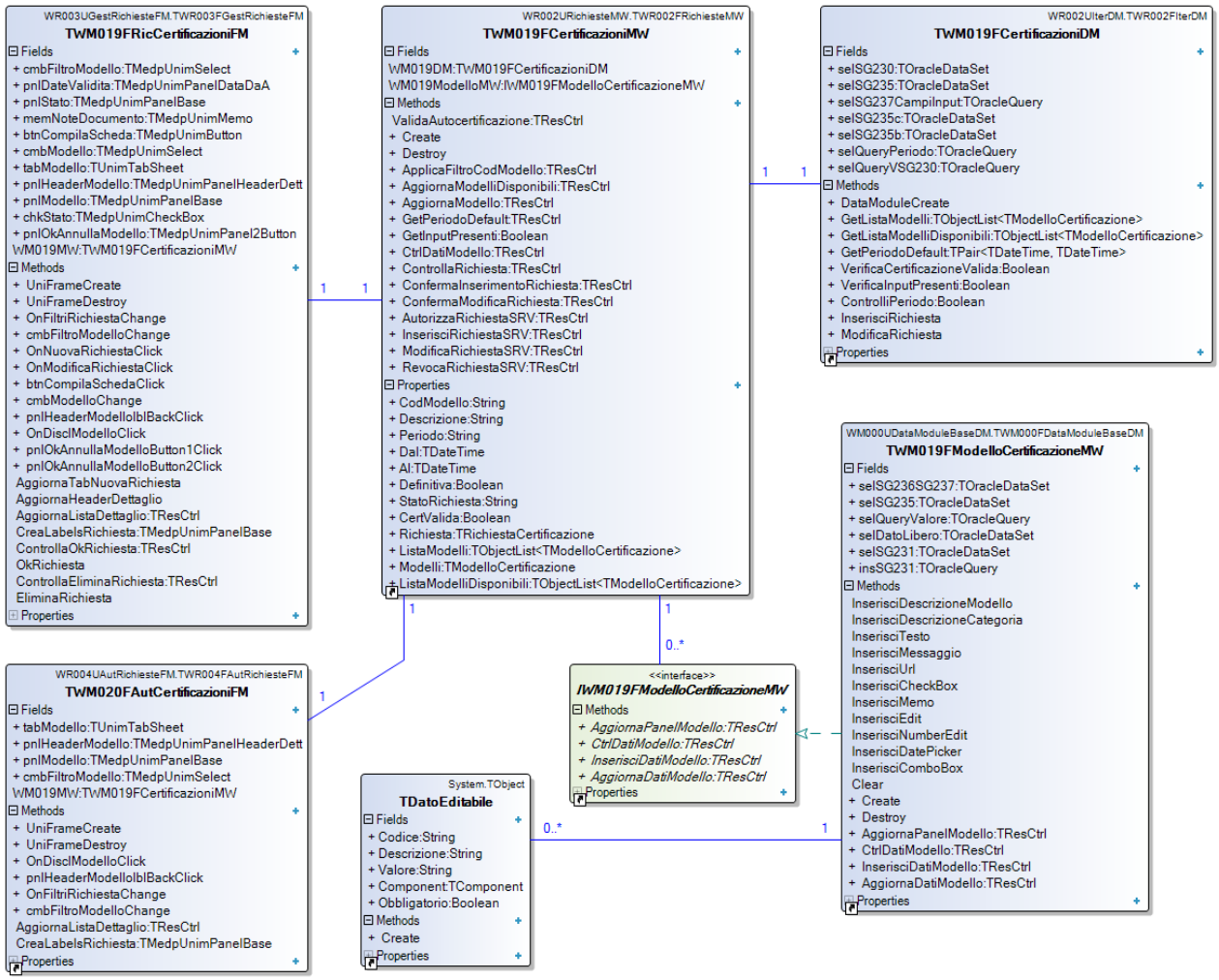


Figura 3.39 - Le classi utilizzate per l'implementazione delle funzionalità Compilazione e Validazione scheda informativa

4. Testing

I test eseguiti sull'applicazione hanno consentito di validare i risultati ottenuti ed evidenziare le correzioni necessarie prima del rilascio dell'applicazione alle aziende clienti. Possono essere suddivisi in tre tipologie:

1) Unit testing: i test di unità sono stati condotti parallelamente allo sviluppo con un approccio bottom-up, partendo dal basso con le classi del data access layer, proseguendo con le classi del business logic layer e terminando con il livello più alto, il presentation layer. I test eseguiti sulle classi di un livello includevano l'utilizzo del livello sottostante già testato, fornendo quindi un ulteriore riscontro sull'integrazione tra i componenti che compongono le funzionalità dell'applicazione. Particolare importanza è stata data alla gestione dei memory leak, tutti i test sono infatti stati condotti attivando il tool madExcept del Delphi IDE, che consente di evidenziare le aree di memoria non rilasciate dopo la chiusura dell'applicazione avviata in debug, fornendo inoltre informazioni utili per identificare e risolvere i problemi evidenziati.
2) Use case testing: al termine della fase di sviluppo sono stati condotti dei test basati su casi d'uso (approfonditi in seguito).
3) Alpha testing: l'ultima fase ha richiesto l'installazione dell'applicazione internamente all'azienda Mondo Edp, dove è stata utilizzata per un certo periodo dal personale dell'azienda per verificarne il corretto funzionamento, con particolare attenzione alla creazione e distruzione delle sessioni web.

4.1 Use case testing

La tecnica di use case testing si colloca nella categoria dei test funzionali di tipo black box e consente di definire dei casi di test incentrati sulle azioni svolte dall'utente nelle varie funzionalità dell'applicazione. A partire dai singoli casi d'uso sono state definite le precondizioni, i passi richiesti ed i risultati attesi per i vari casi di test.

Sono riportati in seguito alcuni esempi significativi dei casi di test utilizzati per verificare il corretto funzionamento dell'applicazione.

Test case ID	TC001	
Descrizione	L'obiettivo di questo caso di test è verificare il corretto funzionamento dell'autenticazione dell'utente, con le successive operazioni di modifica della password e dei contatti, seguite dal logout dall'applicazione.	
Prerequisiti	L'utente deve avere in corrispondente utente web registrato nel sistema informativo Iris.	
Passi	1	Inserimento di nome utente e password nella form di login
	2	Click sul pulsante accedi
	3	Click sul pulsante di gestione utente
	4	Click sul pulsante di modifica password
	5	Inserimento della password attuale e della nuova password
	6	Click sul pulsante di conferma

	7	Click sul pulsante di modifica contatti
	8	Inserimento dei nuovo valore da associare a cellulare, e-mail e PEC
	9	Click sul pulsante di conferma
	10	Click sul pulsante di chiusure della form
	11	Click sul pulsante di logout
Risultati attesi	L'utente deve riuscire ad accedere all'applicazione con le credenziali fornite. Le modifiche apportate dall'utente devono essere correttamente riportare sul database del sistema informativo Iris. Il logout dall'applicazione deve riportare l'utente sulla form di login.	

Test case ID	TC002	
Descrizione	L'obiettivo di questo caso di test è verificare il corretto funzionamento della creazione del menu principale, della funzionalità di cambio profilo e dell'opzione di login "Ricorda credenziali".	
Prerequisiti	L'utente deve avere in corrispondente utente web registrato nel sistema informativo Iris a cui sono associati due profili distinti, un profilo principale di tipo dipendente ed un secondo profilo di tipo responsabile.	
Passi	1	Inserimento di nome utente e password nella form di login
	2	Attivazione dell'opzione "Ricorda credenziali" nella form di login
	3	Click sul pulsante Accedi
	4	Visualizzazione del menu principale con le funzionalità del dipendente
	5	Click sul pulsante di gestione utente
	6	Selezione del secondo profilo dell'utente dalla combo box
	7	Click sul pulsante di cambio profilo
	8	Ritorno al menu principale con le funzionalità del responsabile
	9	Refresh della pagina con il pulsante del browser
	10	Click sul pulsante di logout
Risultati attesi	Dopo l'autenticazione l'utente deve visualizzare le funzionalità del dipendente, mentre dopo l'operazione di cambio profilo le funzionalità del responsabile. Il refresh della pagina deve riportare l'utente nel menu principale senza richiedere il login, il logout successivo deve portare l'utente a visualizzare una pagina bianca.	

Test case ID	TC003	
Descrizione	L'obiettivo di questo caso di test è verificare il corretto funzionamento della funzionalità Dati giornalieri.	
Prerequisiti	L'utente deve aver eseguito il login e deve avere dei dati significativi (timbrature, giustificativi, anomalie) su un giorno oltre al corrente.	
Passi	1	Apertura della funzionalità Dati giornalieri
	2	Scorrimento verticale dei dati visualizzati nel giorno corrente
	3	Selezione del secondo giorno da visualizzare
	4	Scorrimento verticale dei dati visualizzati nel secondo giorno
Risultati attesi	L'utente deve riuscire ad avviare la funzionalità ed a visualizzare correttamente tutti i dati del giorno corrente. Dopo aver selezionato una seconda data, il contenuto della pagina si deve aggiornare con i nuovi dati.	

Test case ID		TC004
Descrizione		L'obiettivo di questo caso di test è verificare il corretto funzionamento della funzionalità Elenco dipendenti.
Prerequisiti		L'utente di tipo responsabile deve aver eseguito il login e deve avere dei dati significativi (dipendenti assegnati nei vari stati possibili).
Passi	1	Apertura della funzionalità Elenco dipendenti
	2	Scorrimento verticale dei dipendenti associati
	3	Selezione di un dipendente della lista
	4	Selezione della timbratura in corso
Risultati attesi		L'utente deve riuscire ad avviare la funzionalità e visualizzare l'elenco completo dei dipendenti gestiti. Selezionando un dipendente deve visualizzare i dati associati e la timbratura del giorno.

Test case ID		TC005
Descrizione		L'obiettivo di questo caso di test è verificare il corretto funzionamento della funzionalità Timbratura virtuale mobile con geolocalizzazione.
Prerequisiti		L'utente di tipo responsabile deve aver eseguito il login e deve avere essere localizzato nei pressi di un rilevatore virtuale.
Passi	1	Apertura della funzionalità Timbratura virtuale mobile
	2	Autorizzazione all'utilizzo della posizione del dispositivo
	3	Spostamento del dipendente fuori dal raggio di validità del rilevatore
	4	Spostamento del dipendente all'interno del raggio di validità del rilevatore
	5	Click sul pulsante di visualizzazione della mappa
	6	Click sul pulsante di chiusura della mappa
	7	Click sul pulsante di inserimento timbratura
Risultati attesi		L'utente deve riuscire ad avviare la funzionalità. La timbratura deve essere consentita soltanto quando il dispositivo si trova nel raggio del rilevatore. La mappa con la posizione del dipendente e del rilevatore deve essere visualizzata correttamente.

Test case ID		TC006
Descrizione		L'obiettivo di questo caso di test è verificare il corretto funzionamento della funzionalità Stampa cartellino.
Prerequisiti		L'utente deve aver eseguito il login e avere dei cartellini da elaborare nel sistema informativo Iris.
Passi	1	Apertura della funzionalità Stampa cartellino
	2	Selezione di una parametrizzazione dalla combo box
	3	Click sul mese da scaricare
	4	Click sul pulsante di chiusura della nuova scheda del pdf
	5	Selezione di una diversa parametrizzazione dalla combo box
	6	Click sul mese da scaricare
	7	Click sul pulsante di chiusura della nuova scheda del pdf

Risultati attesi	L'utente deve riuscire ad avviare la funzionalità. La stampa del cartellino deve aprire il documento pdf in una nuova scheda del browser. Il cambio di parametrizzazione deve portare al download di un documento con layout diverso dal precedente.

Test case ID		TC007
Descrizione		L'obiettivo di questo caso di test è verificare il corretto funzionamento della funzionalità Stampa cedolino.
Prerequisiti		L'utente deve aver eseguito il login e avere dei cedolini elaborati e visibili nel sistema informativo Iris.
Passi	1	Apertura della funzionalità Stampa cedolino
	2	Selezione di un intervallo di mesi con le due combobox
	3	Click sul cedolino da scaricare
	4	Click sul pulsante di chiusura della nuova scheda del pdf
Risultati attesi		L'utente deve riuscire ad avviare la funzionalità. La selezione dell'intervallo temporale deve portare all'aggiornamento della lista dei cedolini disponibili. La stampa del cedolino deve aprire il documento pdf in una nuova scheda del browser.

Test case ID		TC008
Descrizione		L'obiettivo di questo caso di test è verificare il corretto funzionamento della funzionalità Messaggistica.
Prerequisiti		L'utente deve aver eseguito il login e avere dei messaggi ricevuti da leggere con allegati.
Passi	1	Apertura della funzionalità Messaggistica
	2	Click su un messaggio da leggere
	3	Click sulla checkbox di conferma della lettura
	4	Click sul pulsante di apertura del panel degli allegati
	5	Click su un allegato
Risultati attesi		L'utente deve riuscire ad avviare la funzionalità. La vista deve presentare i messaggi ricevuti per primi nell'elenco. La scheda di dettaglio deve consentire di visualizzare i dati e di scaricare gli allegati. L'inserimento della lettura del messaggio deve salvare correttamente la data nel database.

Test case ID		TC009
Descrizione		L'obiettivo di questo caso di test è verificare il corretto funzionamento della funzionalità Gestione deleghe.
Prerequisiti		L'utente deve aver eseguito il login come responsabile e avere delle deleghe attive per il profilo in uso.
Passi	1	Apertura della funzionalità Gestione deleghe
	2	Click su una delega visualizzata
	3	Click sul pulsante annulla
	4	Click sul pulsante di inserimento di una nuova delega
	5	Inserimento dei dati richiesti nei campi di input
	6	Click sul pulsante di selezione dell'utente

	7	Inserimento del filtro da applicare sugli utenti
	8	Click sull'utente selezionato
	9	Click sul pulsante Ok per confermare l'inserimento
Risultati attesi	L'utente deve riuscire ad avviare la funzionalità. La vista deve presentare la lista delle deleghe attive. Selezionando la singola delega l'utente deve portare alla scheda di modifica. L'inserimento della delega deve portare al corretto salvataggio dei dati sul database.	

Test case ID	TC010	
Descrizione	L'obiettivo di questo caso di test è verificare il corretto funzionamento dell'inserimento di una richiesta nella funzionalità Richiesta giustificativi	
Prerequisiti	L'utente deve aver eseguito il login come dipendente e avere la possibilità di inserire dei giustificativi nel sistema informativo Iris	
Passi	1	Apertura della funzionalità Richiesta Giustificativi
	2	Click sul pulsante di inserimento di una nuova richiesta
	3	Inserimento dei dati richiesti nei campi di input
	4	Click sul pulsante Ok per l'inserimento della richiesta
	5	Click sulla richiesta appena creata dall'elenco visualizzato
	6	Click sulla riga allegati
	7	Click sul pulsante di inserimento dell'allegato
	8	Click sul pulsante di selezione dell'allegato
	9	Selezione del file dal file system del dispositivo
	10	Click sulla checkbox di conformità
	11	Click sul pulsante Ok per confermare l'inserimento dell'allegato
Risultati attesi	L'utente deve riuscire ad avviare la funzionalità. L'inserimento del giustificativo deve proporre i campi richiesti dal tipo scelto. Dopo l'inserimento l'utente deve poter visualizzare i dati della richiesta e inserire correttamente l'allegato.	

Test case ID	TC011	
Descrizione	L'obiettivo di questo caso di test è verificare il corretto funzionamento dell'autorizzazione di una richiesta nella funzionalità Autorizzazione timbrature.	
Prerequisiti	L'utente deve aver eseguito il login come responsabile e avere delle richieste di timbratura con note del dipendente da autorizzare.	
Passi	1	Apertura della funzionalità Autorizzazione timbrature
	2	Click su una richiesta da autorizzare
	3	Click sulla riga Note
	4	Click sul pulsante di modifica della note
	5	Inserimento di un nota dell'autorizzatore
	6	Click sull'icona di conferma
	7	Click sul pulsante indietro
	8	Click sul pulsante Autorizza per autorizzare la richiesta

Risultati attesi	L'utente deve riuscire ad avviare la funzionalità, deve poter visualizzare correttamente i dati della richiesta e le note inserite dal dipendente. L'operazione di autorizzazione deve andare a buon fine con il salvataggio del dato sul database.

Test case ID	TC012	
Descrizione	L'obiettivo di questo caso di test è verificare il corretto funzionamento dell'inserimento di una richiesta di scheda informativa nella funzionalità Compilazione scheda informativa.	
Prerequisiti	L'utente deve aver eseguito il login come dipendente e deve avere a disposizione dei modelli di scheda informativa da compilare.	
Passi	1	Apertura della funzionalità Compilazione scheda informativa
	2	Click sul pulsante di inserimento di una nuova richiesta
	3	Selezione di un modello disponibile dalla combobox
	4	Inserimento dei dati associati alla richiesta
	5	Click sul pulsante compila scheda
	6	Compilazione del modello
	7	Click sul pulsante Ok per la conferma dei dati inseriti
	8	Click sul pulsante Ok per l'inserimento della richiesta
Risultati attesi	L'utente deve riuscire ad avviare la funzionalità e deve poter inserire la richiesta compilando correttamente i campi editabili del modello.	

Test case ID	TC013	
Descrizione	L'obiettivo di questo caso di test è verificare il corretto funzionamento dell'autorizzazione massiva di richieste nella funzionalità Autorizzazione cambio orario.	
Prerequisiti	L'utente deve aver eseguito il login come responsabile e avere delle richieste di cambio orario da autorizzare.	
Passi	1	Apertura della funzionalità Autorizzazione cambio orario
	2	Inserimento di un filtro temporale sulla richieste da autorizzare
	3	Click sull'icona che rende visibili i pulsanti di autorizzazione massiva
	4	Click sul pulsante autorizza tutti
	5	Click sul pulsante di conferma dell’operazione richiesta
Risultati attesi	L'utente deve riuscire ad avviare la funzionalità. Il filtro temporale deve aggiornare la lista delle richieste visualizzate di conseguenza. L'autorizzazione massiva deve registrare correttamente i dati dell'operazione sul database.	

5. Conclusioni

Il lavoro svolto ha consentito di realizzare l'applicazione con ottimi risultati ed in tempi relativamente brevi, fornendo anche un riscontro positivo sul nuovo framework UniGUI utilizzato.

La prima fase del progetto ha evidenziato delle lacune nella documentazione messa a disposizione dal fornitore del framework, comportando una fase di apprendimento abbastanza difficoltosa. Sono stato aiutato però dalla presenza di un forum dedicato ai clienti sul sito web di UniGUI, in cui il fornitore risponde con buone tempistiche alle richieste avanzate ed ai problemi evidenziati, fornendo workaround per la soluzione immediata, sostituiti poi dalle correzioni definitive nelle versioni rilasciate successivamente.

La scelta di sviluppare dei componenti visuali personalizzati si è rivelata corretta in quanto ha permesso di riutilizzare funzionalità comuni in più punti dell'applicazione, senza replicare il codice, e di velocizzare notevolmente lo sviluppo dell'interfaccia grafica richiesta, con benefici anche sulla creazione di nuove funzionalità in futuro. La gerarchia di classi utilizzata per l'implementazione degli iter autorizzativi, frutto di un compromesso tra l'interazione con il web service B110 e l'accesso diretto al database, a fronte di uno sforzo iniziale e di adattamenti successivi, ha consentito di creare una base utile per lo sviluppo di questo tipo di funzionalità, adattabile a tutti i contesti che richiedono la gestione di richieste e autorizzazioni. L'architettura a tre livelli adottata ha richiesto uno sforzo maggiore in fase di sviluppo per gestire la separazione dei compiti tra i vari componenti, ma ha permesso di produrre codice più facile da manutenere nel medio-lungo termine, consentendo anche di riutilizzare il business logic layer ed il data access layer nel caso l'azienda decidesse in futuro di convertire la web app usando un altro framework per lo sviluppo di applicazioni web.

IrisAPP è attualmente (novembre 2021) in uso presso una quindicina di clienti dell'azienda, tra aziende sanitarie, università ed enti locali, ed è stata utilizzata dai dipendenti della pubblica amministrazione anche durante il periodo di smart working generalizzato causato all'epidemia di COVID-19.

5.1 Possibili sviluppi futuri

Possibili miglioramenti da implementare in futuro sono:

- Aggiunta di nuove funzionalità, sia al profilo dipendente che al profilo responsabile, in modo da raggiungere l'effettiva equivalenza dal punto di vista funzionale con l'applicativo IrisWEB.
- Miglioramento della navigazione nelle funzionalità dell'applicazione riducendo il tempo di caricamento delle form. Realizzato mediante l'uso di richieste non bloccanti per i soli elementi che richiedono lunghe elaborazioni da parte del server (come il download delle richieste degli iter autorizzativi).
- Miglioramento dell'interazione con l'utente mediante l'utilizzo di azioni che non richiedono la comunicazione con il server. Usando maggiormente il codice javascript

eseguito lato client che consente di ridurre i tempi di risposta percepiti dall'utente durante l'interazione con la web app.

- Miglioramento dell'interfaccia grafica dell'applicazione, in modo da adattare il contenuto delle pagine in modo più funzionale all'utilizzo su tablet con risoluzione elevata.
- Utilizzo di componenti visuali personalizzati che implementano la paginazione, per ridurre il tempo di caricamento delle funzionalità e migliorare la navigazione da parte dell'utente.
- Integrazione delle notifiche web push, utili per segnalare all'utente eventi significativi nell'applicazione sui dispositivi mobili.
- Implementazione della funzionalità di routing, modificando l'url visualizzato nella barra degli indirizzi del browser durante la navigazione. Gestendo di conseguenza l'accesso alle funzionalità con un url diverso da quello di base della single page application.

Bibliografia

[1] Mondo Edp S.r.l. - https://www.mondoedp.com/

[2] NTML - https://it.wikipedia.org/wiki/NTLM

[3] ADSI - https://docs.microsoft.com/en-us/windows/win32/adsi/using-adsi

[4] SSPI - https://docs.microsoft.com/en-us/windows-server/security/windows-authentication/security-support-provider-interface-architecture

[5] OAuth 2.0 - https://datatracker.ietf.org/doc/html/rfc6749

[6] Oracle - https://docs.oracle.com/

[7] Microsoft IIS - https://www.iis.net/

[8] Delphi - https://www.embarcadero.com/products/delphi

[9] UniGUI - http://www.unigui.com/

[10] ExtJS - https://www.sencha.com/products/extjs/