C/O Polo Tecnologico
Via Solteri, 38 38121 Trento
TEL. +39 0461 1865865 FAX +39 0461 1865866

Adevico DSS
Documentazione software

Dicembre 2017

E-mail: info@edutech.it Web: www.edutech.it
Capitale Sociale 65.000,00 Euro - Iscrizione CCIAA TN
Codice fiscale/Partita Iva 02160210221

edutech

C/O Polo Tecnologico
Via Solteri, 38 38121 Trento
TEL. +39 0461 1865865 FAX +39 0461 1865866

SOMMARIO

C/O Polo Tecnologico
Via Solteri, 38 38121 Trento
TEL. +39 0461 1865865 FAX +39 0461 1865866

1 STRUTTURA BASE DELLA PIATTAFORMA ADEVICO

L'applicativo è suddiviso in verticale su più strati.

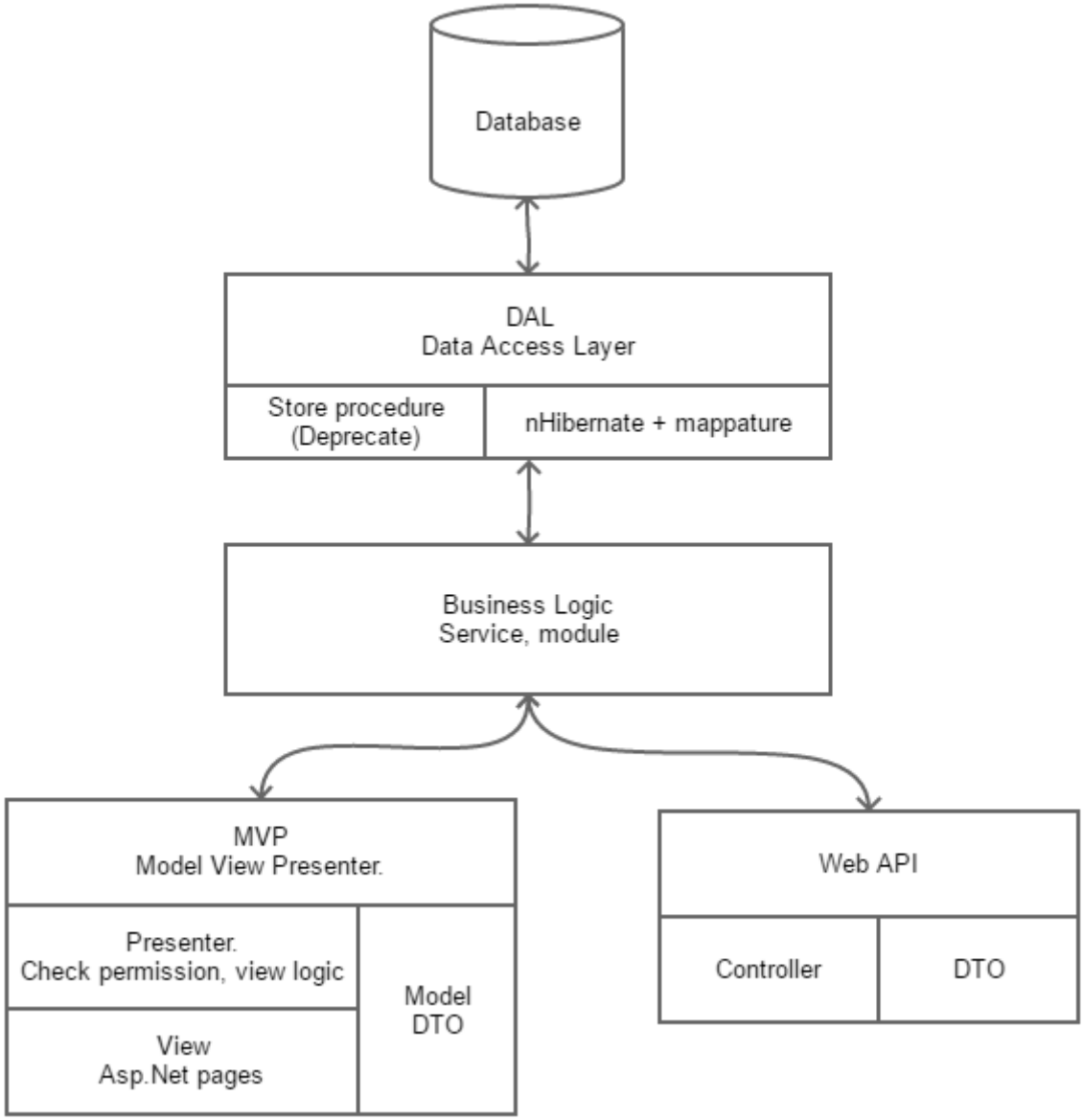

Figura 1:Schema generale applicativo

Partendo dal database si scende passando per il DAL che si occupa della comunicazione con il database. Lo strato di Business Logic si occupa della gestione delle principali logiche di business, garantendo che le regole applicative siano rispettate. Più sotto troviamo lo strato di Presentation, implementato attraverso il pattern MVP (Model View Presenter). La Presentation si occupa di ricevere le chiamate dalla pagina, fare alcuni controlli preliminari, principalmente permessi, prima di girare le richieste alla Business. Inoltre, nello strato di Presentation abbiamo anche la gestione di chiamate più generiche, come quelle necessarie alla tracciatura, l'invio di notifiche e l'interazione tra strati di business diversi. Infine le View, che fanno da interfaccia tra l'utente e gli strati precedentemente descritti, che si basano su Pagine Asp.Net.

C/O Polo Tecnologico
Via Solteri, 38 38121 Trento
TEL. +39 0461 1865865 FAX +39 0461 1865866

1.1 Struttura orizzontale

La struttura verticale descritta in precendeza, vale generalmente per tutti i moduli e serivzi presenti in piattaforma. Esiste però anche una suddivisione orissontale, ovvero ogni modulo o servizio presente in piattaforma ha la struttura descritta in precedenza, ma ognuno si affianca agli altri in senso orizzontale.
I moduli definiti “di core” sono i moduli principali che gestiscono gli oggetti applicativi utilizzati da tutti gli altri servizi. Principalmente si occupano di gestire gli utenti, i loro permessi e le comunità.
A fianco del core, abbiamo i primi moduli che sono spesso, ma non sempre, utilizzati dagli altri moduli. E’ il caso dell’invio delle mail, del repository, ma anche della gestione dei template. Sono moduli che offrono funzionalità ad altri servizi.
Infine tutti gli altri moduli, che forniscono funzionalità specifiche, come i questionari o i bandi, ma che talvolta permettono di aggregare più moduli, come il Percoso Formativo, che a seconda dei moduli disponibili, permette di organizzare questionari, materiale del repository, materiale multimediale, certificati attraverso i template, fornendo statistiche proprie date dall’utilizzo di oggetti prevenienti da altri servizi.

1.2 Regole ed oggetti di base

Di norma, per le classi principali descritte nel diagramma precedente, si prevede che gli oggetti implementino o estendano una serie di classi base di dominio, chiamate appunto “Base”. Tali classi permettono ad oggetti specifici all’interno della catena di avere sempre a disposizioni classi e funzioni generiche, come pure di mantenere una certa coerenza di sviluppo nel momento in cui si rende necessario implementare funzioni specifiche.
Nel capitolo successivo verrano descritti nel dettalgio gli strati visti inizialmente e per ognuno verranno evidenziate le classi base utilizzate.

2 DESCRIZIONE STRUTTURA

Di seguito viene dettagliata la struttura vista inizialmente. Quanto qui descritto è comune a tutti i servizi, pertanto nella descrizione degli stessi tali nozioni saranno date per scontate.

2.1 Database

L’applicazione nasce e cerca di essere il più agnostica possibile in relazione al database. Nonostante questo, gli sforzi per testare ed adattare tutto l’applicato a database diversi non è mai stato affrontanto per mancanza di una reale necessità. Per questo motivo si rimane legati ad SQL Server.

2.2 DAL – Data Access Layer

2.2.1 Store procedure, Trigger e programmazione database

L’applicativo nasce utilizzando store procedure per l’accesso ai dati. Successivamente si è passati all’utilizzo di nHibernate. Non tutto l’applicativo o le sue componenti sono però state aggiornate ed alcuni servizi utilizzano ancora vecchie strutture dati. Tale pratica è però considerata DEPRECATA.
L’utilizzo di Trigger, invece, è stato per lo più abbandonato in favore di una miglior gestione delle logiche di business. Sono quindi relegati a casi specifici, in cui si intende monitorare specifiche attività che esulano delle logiche applicative, senza toccare le attuali implementazioni. Tali implementazioni hanno un ambito estremamente limitato, in particolare ad esigenze molto specifiche e per singole istanze.

2.2.2 Altre funzionalità legate al database

Nei casi in cui sono stati tracciati degli storici a fini statistici della piattaforma, si è preferito agire non su trigger ma piuttosto estraendo dati specifici ad intervalli regolari, piuttosto che creando viste specifiche sulle specifiche esigenze.

C/O Polo Tecnologico
Via Solteri, 38 38121 Trento
TEL. +39 0461 1865865 FAX +39 0461 1865866

Al contrario, per la tracciatura regolare di normali attività di piattaforma, ci si rivolge a servizi specifici, come per le User Action.

2.2.3 Hibernate e mappature

Questo strato è quello attualmente in uso per lo sviluppo della piattaforma. Le mappature sono in realtà fisicamente nello strato di business.

2.3 Business Logic

Questo strato rappresenta lo strato più corposo dell'applicativo. Contiene tutti gli oggetti di dominio e relative mappature, i dto, le logiche e le funzioni di business.

2.3.1 Oggetti di dominio

Gli oggetti di dominio sono relativi al singolo servizio e sono completamente mappati su database. Questi derivano tutti da poche classi base (BaseObject) che definiscono alcuni campi comini a tutti gli oggetti, tra cui:

- Creatore (utente che ha creato l'oggetto, IP, IP Proxy e data di creazione)
- Modifica (utente che ha modificato l'oggetto, IP, IP Proxy e data di creazione)
- Cancellazione logica (basata su un enum che definisce se è stato cancellato dall'utente, in cascata in seguito alla cancellazione di un oggetto base, etc…)

Questi dati permettono un minimo di tracciabilità su tutti gli oggetti del sistema.
Per lo più, nel sistema, si preferisce mappare tutte le proprietà di un oggetto, in una mappatura per lo più 1-1 con i campi del database. Saranno poi le funzioni di recupero dei dati, creando appositi DTO a filtrare i dati provenienti dal database, facendo in modo che siano lette solo le proprietà di interesse.
Solo in casi specifici e di norma limitati, si ricorre all'utilizzo di oggetti "lite", ovvero oggetti che hanno e di cui sono mappate solo un numero limitato di proprietà. Questo viene utile nei casi in cui vengano letti un numero consistente di righe/oggetti.

2.3.2 Module

Per Module si intende una classe, presente in ogni servizio, che definisce gli aspetti principali del servizio setto. Nel module è presente il codice del servizio, i principali permessi di cui necessita, le funzioni base per tradurre i permessi generici del sistema nei permessi del servizio e le definizioni utilizzate per la tracciatura delle azioni utente, in particolare "ActionType" ed "ObjectType" che indicano rispettivamente le azioni che possono essere eseguite dagli utenti nel servizio e gli oggetti relativi a tali azioni.

2.3.3 Service

Le classi dei service contengono tutta le logiche di business del servizio stesso.
Al loro interno è presente quello che viene chiamato "Manager", che offre a tutti i servizi sia le funzioni generiche di accesso ai dati, che le funzioni principali per l'accesso e la modifica di oggetti comuni al sistema, come la comunità o la persona.
L'inizializzazione dei service è standard ed avviene tramite l'ApplicationContext, che a sua volta è composto dallo UserContext, che contiene le informazioni sull'utente corrente (id, comunità, tipo persona, etc…) che dal DataContext, che contiene le informazioni comuni all'applicativo per l'accesso ai dati.
Le altre funzioni del service, di norma, si occupano di manipolare gli oggetti di dominio del servizio, attraverso funzioni che si interfacciano per lo più attraverso parametri standard o dto specifici a seconda della complessità e dell'obiettivo della funzione stessa.

2.3.4 Manager

La classe Manager rappresenta il DAL.
È usata in tutti i servizi che implementa i metodi principali di accesso ai dati (crud) e fornisce funzioni per il caricamento di oggetti di "Core", come comunità o persona.

C/O Polo Tecnologico
Via Solteri, 38 38121 Trento
TEL. +39 0461 1865865 FAX +39 0461 1865866

2.3.5 Oggetti di dominio Vs Dto

Per oggetti di dominio si intendono gli oggetti propri di un singolo servizio, che come abbiamo visto ereditano specifiche caratteristiche da oggetti base. Questi servono per il corretto funzionamento del sistema (controlli sui dati e restrizioni di business), ma risultano talvola ridondanti. Inoltre la manipolazione da parte dello strato di presentation o delle webAPI comporta alcuni rischi. Per questi motivi si preferisce l'utilizzo di dto (data transfert object) per tali comunicazioni.
La principale differenza tra gli oggetti di dominio ed i dto è che i primi sono mappati su database e possono quindi essere persistiti, mentre i dto hanno il solo scopo di passare informazioni e non sono per questo persistiti.

2.4 Presentation

Lo strato di presentation utilizza il pattern MVP (model-view-presenter).
Il model è rappresentato dai dto, visti in precedenza.
La view è un interfaccia che descrive funzioni e proprietà che sono necessarie al corretto funzionamento e che viene implementata dal code-behind delle varie pagine aspx. Potenzialmente è quindi possibile avere pagine diverse che assolvono le stesse funzioni, anche se questo è destinato a casi particolari. Anche per le view esistono le relative classi base, che permettono in modo più agevole la gestione di comportamenti standard, come l'invio delle UserAction.
Infine il presenter è la classe che si occupa di far comunicare la view con lo strato di business. In questo livello vengono gestite tutte le logiche che riguardano la view in senso generico.

2.4.1 Pagine Web

Le pagine web rappresentano l'UI dell'applicazione. Vengono associate allo strato di presentation, in quanto sono fortemente interconnesse ed implementano le funzioni e le proprietà descritte nell'interfaccia della View.
Il code-behind si occupa di gestire tutti gli oggetti .net della pagina e mettere in relazione le varie azioni (dell'utente o su eventi specifici) con le funzioni del presenter.
Oltre alla view, le pagine ereditano dalla stessa classe base, definita "PageBase". Spesso ogni servizio estende il PageBase comune all'applicativo con un proprio PageBase, che permette un uso ottimizzato sul servizio delle funzioni del PageBase.
Il PageBase contiene alcune funzioni che vanno implementate nelle pagine (come il bind dei dati o la gestione della localizzazione), oltre ad un accesso omogeneo ad oggetti condivisi, come i dati utente (sessione), piuttosto che alle configurazioni in essere. Contiene anche alcune facility che permetto un accesso più agevole ed omogeneo, come il "BaseUrl".

2.5 Web Api

Le WebApi sono state introdotte per fornire all'esterno funzionalità specifiche della piattaforma, nate con la necessità di implementare un applicativo mobile. Oltre a questo le WebApi aprono la strada ad un nuovo paradigma di programmazione che permette di spostare il carico della presentation sul client.
Le logiche di Business in uso dalla WebAPI sono per lo più le stesse in uso dall'applicativo, garantendo l'integrità applicativa.
Alcuni nuovi servizi, inoltre, sono integrati direttamente in piattaforma, attraverso l'uso di pagine wrapper. Tali pagine hanno un code-behind comune, che segue le principali logiche dell'MVP di piattaforma e permettono la presenza di tutti gli elementi comuni (compresi menu, skin, gestione permessi, etc…), ma la cui componente HTML contiene chiamate dirette alle API stesse. In questo modo è possibile sviluppare pagine più complesse ed adatte a più dispositivi.
Una trattazione più approfondita delle WebApi è presente nel relativo documento.

C/O Polo Tecnologico
Via Solteri, 38 38121 Trento
TEL. +39 0461 1865865 FAX +39 0461 1865866

3 ELEMENTI COMUNI

Soprattutto a livello di interfaccia, emerge spesso la necessità di avere a disposizioni elementi comuni, con scopi diversi, ma con contenuti simili. Per manterene un'uniformità grafica e soprattutto riutilizzare lo stesso codice, sia a livello globale che per elementi di specifici servizi, vengono utilizzati elementi che possono essere condivisi.

3.1 User Control

All'interno dell'applicativo sono presenti sostanzialmente due tipologie di UserControl:

- Senza presentation: sono Usercontrol che hanno il solo scopo di standardizzare alcune interfacce, ma non contengono logiche di business proprie, che non siano legate ad eventuali configurazioni. Non prevedono quindi alcuna persistenza su database. Alcuni esempi sono gli UserControl relativi all'Editor, progress bar, etc…
- Con presentation: sono UserControl che hanno alle spalle la stessa struttura delle pagine web e consentono l'accesso e la condivisione di oggetti di sistema o di servizio. Alcuni esempi sono il selettore comunità o il selettore utenti

4 STRUTTURA CARTELLE

4.1 Web

La parte web è comune a tutti i servizi.
Vengono di seguito elencate le principali cartelle di uso comune:

- 1-WebApplication\AdevicoWeb\Risorse_XML***it-IT***\Modules***Servizio***\
  Tutti gli XML di localizzazione di un servizio
  it-IT la lingua relativa ai file contenuti
  Servizio il servizio a cui fanno riferimento le pagine
- 1-WebApplication\AdevicoWeb\Modules***Servizio***\
  Le pagine web e gli UserControl del servizio. Per servizi complessi, con molte aree diverse, vengono create più sotto-cartelle per i vari ambiti.
  Gli User control sono inseriti nella cartella /UC/
  Servizio il servizio a cui fanno riferimento le pagine

4.2 MVP

Tutta la parte dell'MVP è presente nel progetto del servizio stesso. I progetti sono strutturati in:

- \Presentation\iView\ Le iView implementate dalle pagine
- \Presentation\ i presenter
- \Domain\ contiene gli oggetti di dominio
- \Mapping\ contiene le mappature degli oggetti di dominio
- \dto\ contiene i dto del servizio
- \Business\ Classi Service, module, etc… con le logiche di business del servizio
- \Helper\ Se necessari, contiene le classi Helper

C/O Polo Tecnologico
Via Solteri, 38 38121 Trento
TEL. +39 0461 1865865 FAX +39 0461 1865866

Modulo Bandi (DSS)

1 INTRODUZIONE

Nel presente documento si fa riferimento in modo esclusivo agli sviluppi portati avanti per la parte DSS, escludendo in toto lo stato della piattaforma precedente agli interventi.
Il documento, inoltre, ha la sola finalità di mettere in luce le aree della Solution in cui sono stati eseguiti gli interventi, descrivendoli brevemente.
Per la descrizione funzionale si veda l'apposito documento, mentre per i dettagli sul codice si rimanda ai commenti del codice stesso.

1.1 Business

A livello di logiche di Business, la classe che gestisce tali logiche è il "ServiceCallOfPeaper.cs" (in \3-Business\3-Modules\Im.Comol.Modules.CallForPapers\Business).
Lo sviluppo di nuove funzionalità avviene estendendo tale classe con partial class che si occupano di aspetti specifici. Abbiamo quindi:

- ServiceCallOfPapers.Print.cs
  Per la gestione della stampa tramite Template (Step 1)
- ServiceCallOfPapers.adv.cs
  Che gestisce i nuovi oggetti (step e commissioni) e riscrive alcune funzioni per il loro utilizzo.
- ServiceCallOfPeapers.Economic.cs
  Per la gestione delle tabelle economiche in fase di valutazione economica, ovvero di occupa di tradurre quanto immesso dagli utenti nelle loro sottomissioni in oggetti specifici che permettono la valutazione economica. In questo modo i dati delle valutazioni economiche rimangono separati dalle sottomissioni utente e possono essere rimaneggiate secondo quando concordato.

2 STEP 1: TEMPLATE DI STAMPA, BOZZE, CONTROFIRME, TABELLE, TAG

Lo Step 1 ha lo scopo principale di allargare gli strumenti forniti agli amministratori dei Bandi per rispondere alle esigenze dei nuovi bandi, nella prospettiva di adattarli a sviluppi futuri.
In tal senso, lo step 1 prevede una forte integrazione con quanto già presente.

2.1 Template di stampa

I limiti del precedente sistema di stampa, impedivano qualsiasi personalizzazione da parte degli amministratori. Era quindi necessario un intervento correttivo lato software ove fosse necessario.
L'intervento avviene in due distinte aree dell'applicativo. La prima è all'interno dei template, aggiungendo i Bandi tra i servizi supportati. Questo comporta anche l'aggiunta di tag specifici del servizio e la relativa gestione in fase di modifica e stampa.
L'altra è nell'aggiungere ai bandi la possibilità di scegliere un template per la stampa, oltre che permettere alcune personalizzazioni specifiche per il servizio.

2.1.1 Template

All'interno della Solution, le componenti dei template sono così suddivise:

- Pagine web ed UserControl:
  \1-Webapplication\AdevicoWeb\Modules\DocTemplate\ Pagine dei docTemplate
  \1-Webapplication\AdevicoWeb\Modules\CallForPeaper\Edit\UC\ Impostazioni di stampa per I bandi
- Presentation e Busniess Logic:
  \3-Business\3-Modules\lm.Comol.Core.BaseModules\DocTemplate\

La struttura dati dei DocTemplate prevede già l'utilizzo di altri servizi e non sono quindi state apportate ulteriori modifiche alla struttura dati.

2.1.2 Call For Paper (bandi)

L'integrazione con i bandi prevede due livelli distinti. Il primo è la gestione dei nuovi tag relativi al servizio, ovvero quali dati essi debbano contenere, il secondo è quello descritto in precedenza, ovvero la personalizzazione di alcuni aspetti a livello di singolo bando.
La classi per la gestione dei tag nei bandi sono le seguenti:

- Integrazioni con i CallForPeaper:
  \3-Business\3-Modules\lm.comol.Modules.CallForPeapers\Business\
  - ExportHelper.cs
  - HelperExportToPdf.cs

2.1.3 Integrazioni con i bandi

A livello di Bandi, invece, è stata aggiunta la tabella "CP_PrintSettings", rappresentata e mappata sulla classe "CallPrintSettings.cs".
La classe non è inclusa nelle classi dei Call, ma viene gestista come oggetto a sé stante, per quanto comunque ogni CallPrintSettings sia associato ad un solo bando.

2.2 Bozza

La stampa di una bozza del bando è la conseguenza dell'introduzione nei template e permette la stampa di una bozza vuota o la stampa della compilazione corrente. Gli amministratori hanno sempre accesso a tale funzionalità, mentre per gli utenti è possibile impostarlo a livello di bando. Tale flag è inseriro all'interno di CallPrintSettings.

2.3 Controfirme

La gestione delle controfirme prevede per l'utente la stampa del bando compilato ed il successivo upload di una stampa controfirmata o con firma elettronica.
Per fare in modo che la controfirma non sia manipolabile, i bandi che hanno obbligo di controfirma, una volta sottomessi, vengono messi in uno stato particolare: "in attesa di controfirma". In questo stato non è possibile apportare alcuna modifica al bando, come se fosse sottomesso, ma per la data di sottomissione è valida quella relativa all'upload della controfirma stessa.
Le stampe caricate per le controfirme sono quelle ottenute dalle impostazioni a livello di bando, utilizzando il template ad esso associato.

2.4 Tabelle

Si è reso necessario dare la possibilità di creare della tabelle di immissione dati.
Le tabelle si dividono in tre tipologie:

- Tabelle semplici
  Tabelle che non prevedono alcuna logica, ma permettono l'immissione di dati in forma tabellare.
  Le intestazioni sono personalizzabili da parte degli amministratori in fase di creazione/modifica di un bando.

C/O Polo Tecnologico
Via Solteri, 38 38121 Trento
TEL. +39 0461 1865865 FAX +39 0461 1865866

- Tabelle Economiche
  Analoghe alle tabelle precedenti, aggiungono la possibilità per il sottomittore di inserire anche una colonna “Quantità” e “Prezzo unitario”, che vanno a definire il contenuto della colonna “Totale”.
  Per ogni tabella economica è possibile anche definire un massimale che non puo’ essere superato in fase di sottomissione del bando stesso.
- Tabelle di riepilogo
  Sono elementi che non sono modificabili, né compilabili, ma servono solamente a riassumere i contenuti delle tabelle economiche presenti fino a quel punto.

Queste sono state aggiunte tra i campi disponibili in fase di creazione della domanda, in particolare:

- \1-WebApplication\AdevicoWeb\Modules\CallForPapers\UC\UC_AddField.ascx
  UC per l’aggiunta di nuovi campi
- \1-WebApplication\AdevicoWeb\Modules\CallForPapers\UC\UC_EditField.ascx
  UC per la modifica dei campi inseriti
- \1-WebApplication\AdevicoWeb\Modules\CallForPapers\UC\UC_InputField.ascx
  UC per la compilazione del singolo campo

A livello di business è stata modificata la tabella “CP_FieldDefinition” con l’aggiunta della definizione delle colonne e degli eventuali massimali, mappata nella classe FieldDefinition.

2.5 Tag

In questa fase è stata solamente aggiunta la possibilità di associare a singoli campi dei tag, cno lo scopo futuro di poter filtrare i campi delle sottomissioni per agevolare il lavoro dei valutatori, evidenziando quali sono i campi relativi alle loro competenze.
I tag sono salvati in un campo nella tabella “CP_FieldDefinition” e presenti nella relativa classe “FieldDefinition”, sfruttando le precedenti funzioni.

3 STEP 2 – VALUTAZIONE BOOLEANA, PROCESSO DI VALUTAZIONE, INTEGRAZIONI

Lo step due ha lo scopo di ridefinire le procedure di valutazioni. Di base si è cercato di impattare il minimo possibile le precedenti funzionalità, pur sfruttando il più possibile tutto il codice e le pagine preesistenti.
Di base, quindi, abbiamo la parte relativa alla gestione delle nuove commissioni (“Processo di valutazione”), con la generazione e la creazione degli step e delle commissioni (commissioni avanzate), che rimane a sé stante, mentre i criteri applicabili e la parte di valutazione riutilizzano il vecchio codice e le vecchie pagine, estendendolo dove necessario.
Di base, per all’interno delle logiche precedenti, avviene un controllo sulla tipologia di valutazione in essere, ovvero dove si rende necessario utilizzare nuovi oggetti (Step ed AdvCommission), le richieste vengono deviate su nuove funzioni che riciclano il vecchio codice aggiornandolo con le nuove logiche ed oggetti.
Inoltre è stato aggiunto un nuovo tipo di valutazione, definita “Booleana”, che permette al valutatore di definire il superamento di una sottomissione in base ad un singolo flag.

3.1 Valutazione booleana

In questo caso è stato aggiunto all’enum che definisce le tipologie di valutazione (CriterionType in \3-Business\3-Modules\Im.Comol.Modules.CallForPapers\Domain\Base\Evaluation\BaseCriterion.cs) il relativo valore ed estendere la classe “BaseCriterion” con “BoolCriterion”.
Il nuovo tipo di valutazione va naturalmente gestito. Partendo dall’interfaccia utente, in 1-WebApplication\AdevicoWeb\Modules\CallForPapers\Evaluate\UC troviamo:

C/O Polo Tecnologico
Via Solteri, 38 38121 Trento
TEL. +39 0461 1865865 FAX +39 0461 1865866

- UC_AddCriterion.ascx
  User Control per l'inserimento di un nuovo criterio
- UC_EditCriterion.ascx
  User Control per la modifica di un criterio
- UC_InputCriterion.ascx
  User Control per l'assegnazione di un valore al criterio da parte di un valutatore
- UC_RenderCriterion.ascx
  User Control per la visualizzazione di un criterio dopo che gli è stato assegnato un valore da parte di un valutatore

In fase di creazione, modifica, valutazione e visualizzazione, non sono state fatte ulteriori modifiche al codice per la gestione di criteri stessi.
Per quanto riguarda l'utilizzo dei valori inseriti dai valutatori ai fini della classifica e del superamento di una data commissione, vengono analizzati successviamente.

3.2 Processo di valutazione

Il processo di valutazione è rappresentato da Step/Commissioni. Come regola per distinguere le tabelle specifiche di questa fase, alle tabelle viene aggiunto "_Adv_". Ad esempio, la tabella con le informazioni delle nuove commissioni diventa "CP_Adv_Commission".
In solution, tutti i nuovi oggetti e classi per la loro gestione sono contenuti in
Per comodità viene qui rappresentato il diagramma delle classi relative agli oggetti interessati, persenti in 3-Business\3-Modules\Im.Comol.Modules.CallForPapers\Advanced\Domain\ e gestiti tramite le funzioni in 3-Business\3-Modules\Im.Comol.Modules.CallForPapers\Business\ServiceCallOfPapers.adv.cs.

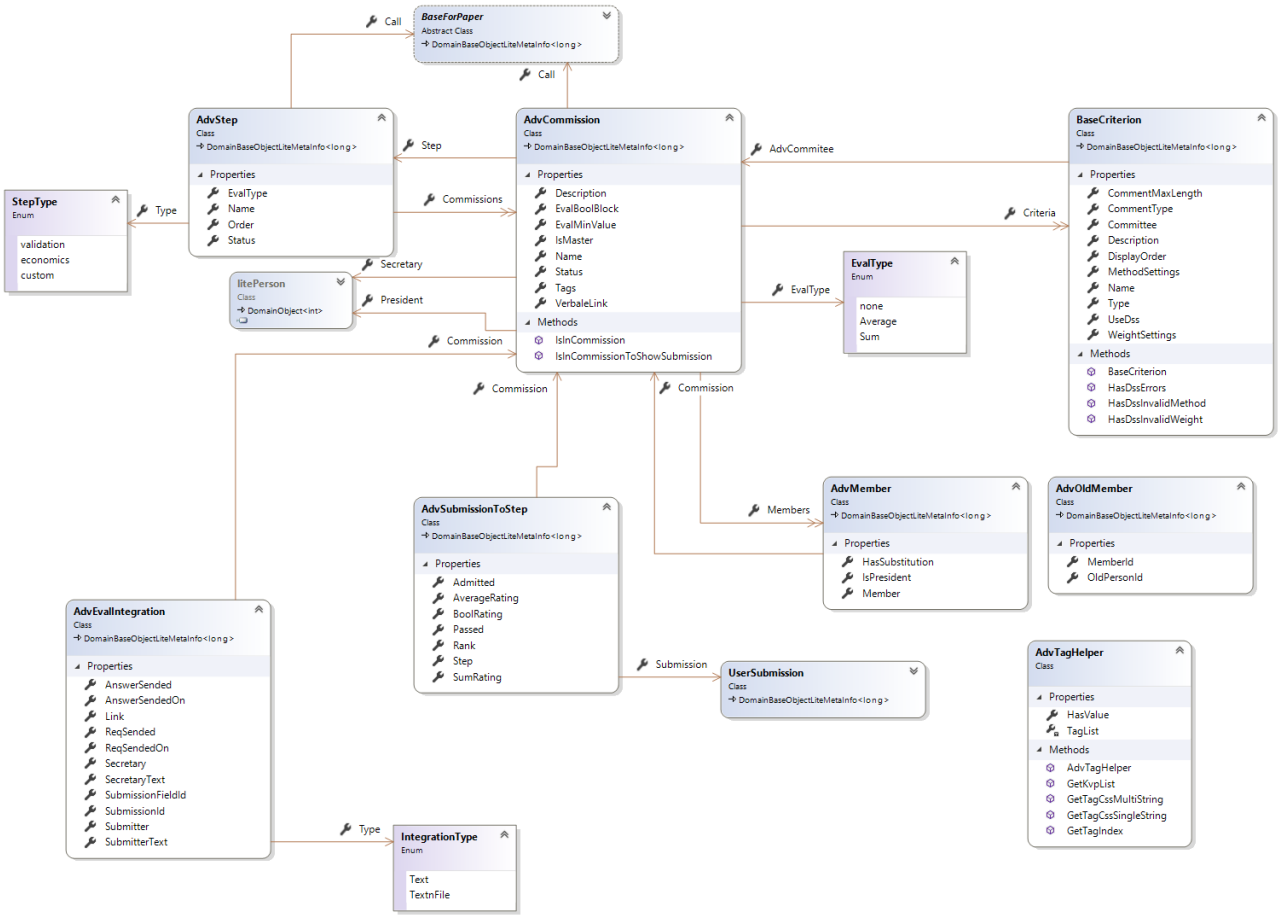

Figura 2: Diagramma classi Commissioni Avanzate

Le tabelle su database rispecchiano il diagramma delle classi. Per maggiori dettagli si rimanda ai commenti interni al codice.

3.2.1 Valutazione economica

L'ultimo step del processo valutativo prevede la valutazione economica dei progetti che hanno superato gli step precedenti.
Questo step prevede regole specifiche. Non sono più associati criteri di valutazione e non è prevista l'esclusione di partecipanti dalla gara, ma è previsto che vengano valutati gli aspetti economici, in particolare le voci inserite nelle tabelle economiche siano vagliate ed approvate e se approvate quale sia la cifra approvata.
In fase di apertura dello step, vengono analizzate tutte le sottomissioni e le tabelle economiche inserite, precedentemente salvate come tabelle html in formato testo, vengono analizzate ed inserite in apposite tabelle.
Viste le logiche diverse applicate a questa fase, tutti gli oggtti interessati sono stati inseriti in 3-Business\3-Modules\lm.Comol.Modules.CallForPapers\AdvEconomic, mentre la classe che gestisce le logiche di business diventa 3-Business\3-Modules\lm.Comol.Modules.CallForPapers\Business\ServiceCallOfPapers.Economic.cs.

Viene di seguito presentato il diagramma delle classi in uso per tale step.

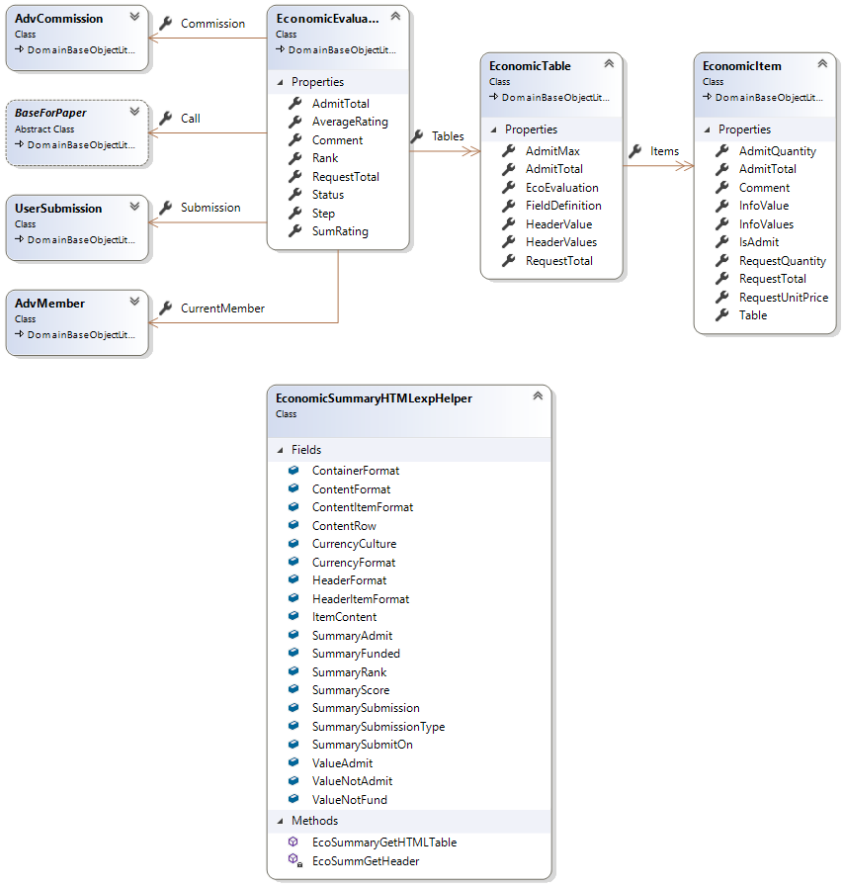

Figura 3: Diagramma classi dto Commissioni avanzate

Per ulteriori approfondimenti si rimanda ai commenti al codice.

C/O Polo Tecnologico
Via Solteri, 38 38121 Trento
TEL. +39 0461 1865865 FAX +39 0461 1865866